Troubleshooting Splunk Enterprise

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Do not distribute

Course Goals

- Identify the best practices for troubleshooting Splunk Enterprise
- List ways to gather useful Splunk diagnostic information
- Use Splunk diagnostic tools
- Identify common Splunk technical issues and solve them

Note

This course does not discuss the issues specific to Splunk Cloud, Splunk Clusters, or Splunk premium apps.

Course Outline

- Module 1: Splunk Troubleshooting Methods and Tools
- Module 2: Indexing Problems
- Module 3: Input Configuration Problems
- Module 4: Deployment and Forwarder Problems
- Module 5: Upgrading, Licensing, and User Management Problems
- Module 6: Search Management Problems
- Module 7: User Search Problems

Foundational Knowledge

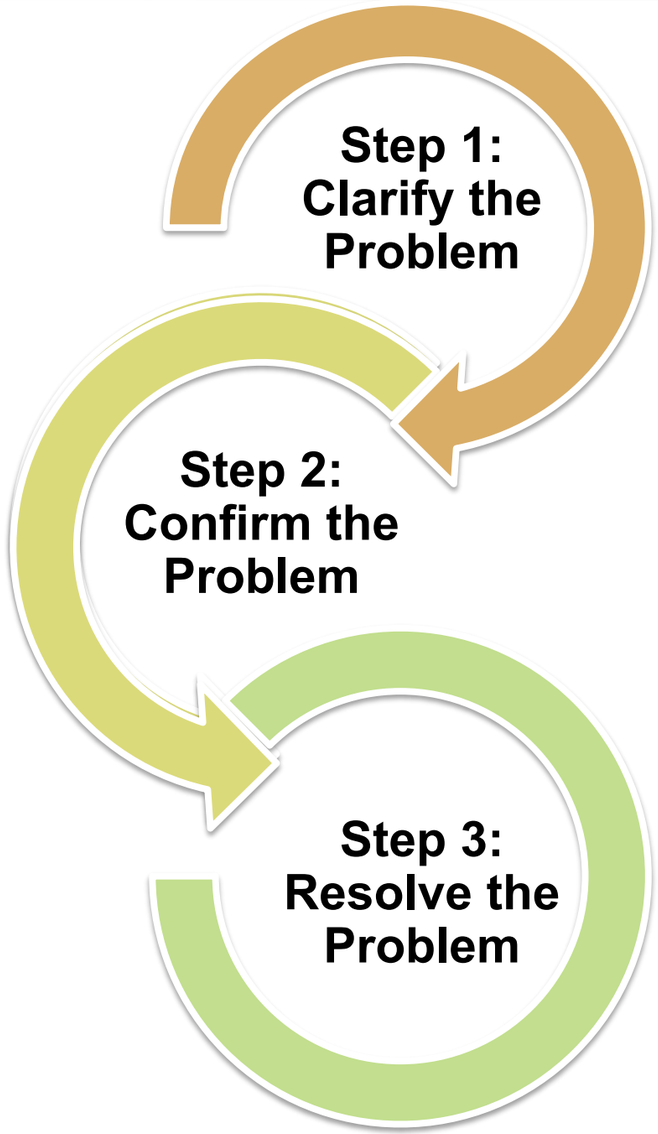
- To be successful, students should have completed these courses:
  - Splunk Fundamentals 1
  - Splunk Fundamentals 2
  - Splunk Enterprise System Administration
  - Splunk Enterprise Data Administration

Note

In order to receive credit for this course, you must complete all lab exercises.

Module 1:
Splunk Troubleshooting Methods and Tools

Splunk Troubleshooting Approach



- Define the problem in a single statement
- Gather the facts
- Investigate one issue at a time

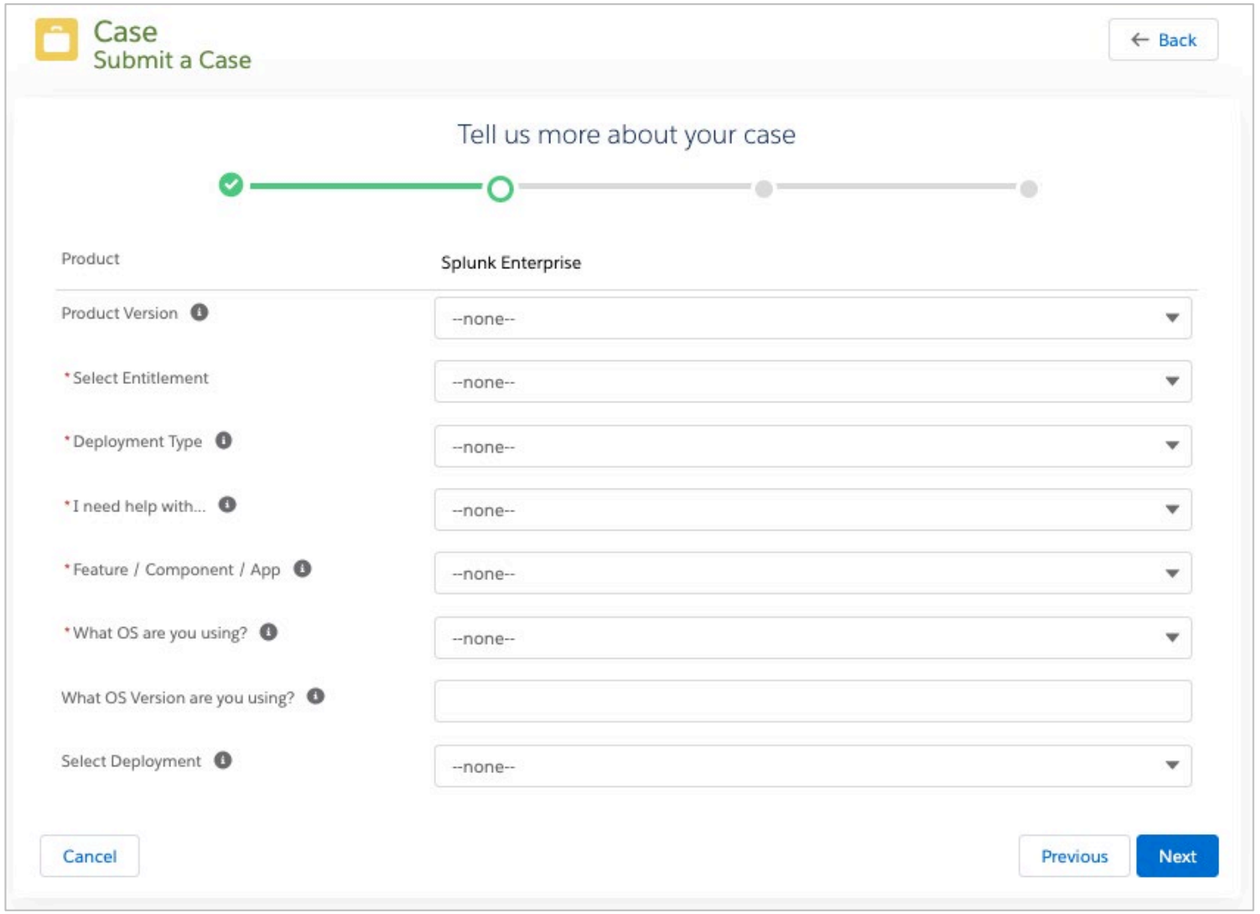
- What type of issue is occurring? What are the symptoms or common problems associated for this type of issue?
- Can you reproduce the issue? Is it intermittent?
- What log channels or Splunk tools can help you diagnose the problem?

- Work on resolving the problem once the issue is narrowed down
- Use the resources available to everyone to help resolve the issues

Submitting a Case

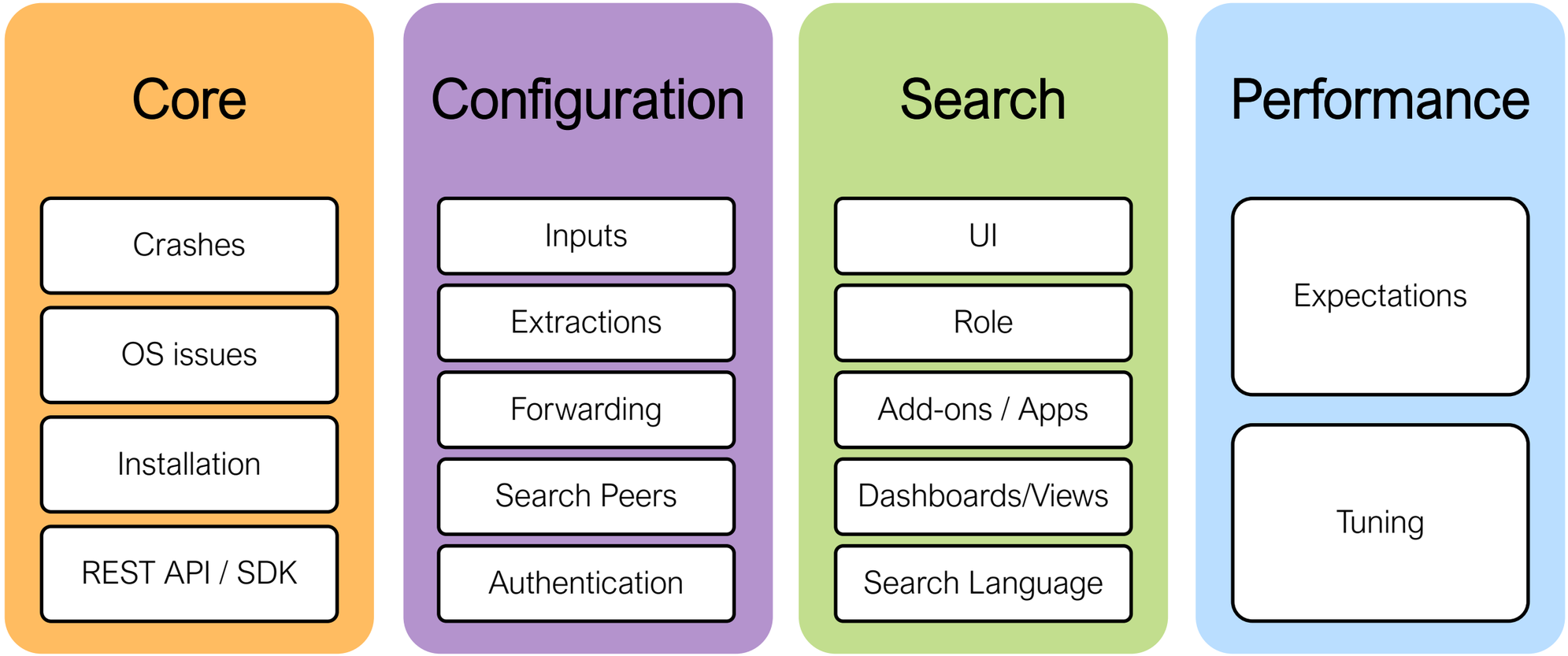
- Use the Splunk support portal if you need to submit a case
  - If it is reproducible, provide the detailed steps
  - Provide a diag
- After the case is submitted successfully, you can provide supporting files
  - Always attach your diag output
  - Helpful to include sample data, screenshots, network diagrams, Splunk architecture topology, etc.

http://login.splunk.com/page/sso_redirect?type=portal



http://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/HowtofileagreatSupportcase

Splunk Problem Classification



Splunk Diagnostic Tools

- Splunk btool
  - Displays merged configuration values
  - **`splunk btool <conf> list [options]`**
- Splunk REST API endpoints
- Splunk internal logs and indexes
- Configuration file change tracking
- Monitoring Console
- Proactive Splunk Component Monitoring
- Splunk Health Report
- Splunk instrumentation (telemetry)
- Splunk diag
- Splunk RapidDiag app

Note

More REST API topics are discussed in detail in the following courses:

- Building Splunk Apps
- Developing with Splunk REST API

Viewing Service Settings with REST Endpoints

https://<host>:<mPort>/services/data/inputs/tcp/raw

9514

_rcvbuf	1572864		
connection_host	ip		
disabled	0		
eai:acl	app	search	
	can_list	1	
	can_write	1	
	modifiable	0	
	owner	nobody	
	perms	read	1. admin 2. power 3. splunk-system-role 4. user
		write	1. admin 2. splunk-system-role
	removable	1	
	sharing	app	
group	listenerports		
host	so1		
host_resolved	so1		
index	web		
sourcetype	snort		

```
| rest /services/data/inputs/tcp/raw | fields - eai* | transpose
```

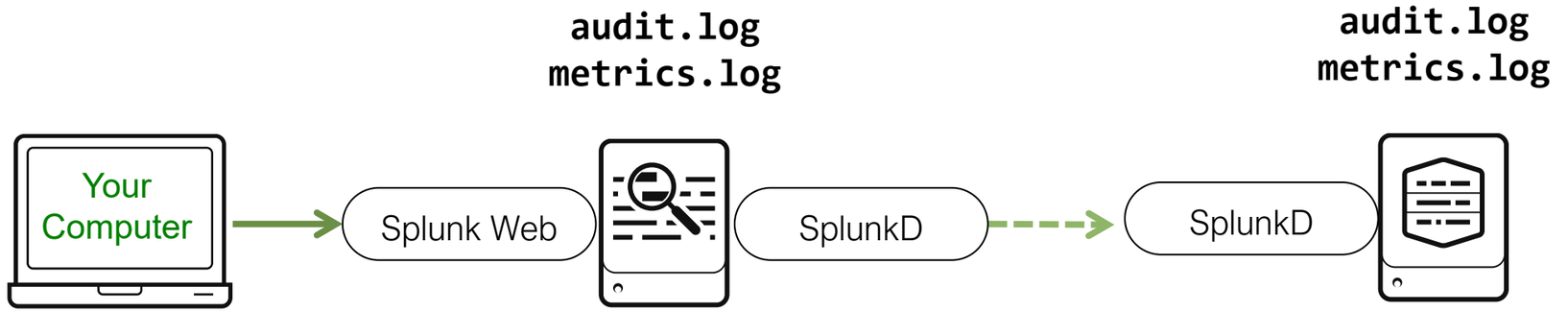
_rcvbuf	1572864
author	nobody
connection_host	ip
disabled	0
group	listenerports
host	so1
host_resolved	so1
id	https://127.0.0.1:8089/servicesNS/nobody/search/data/inputs/tcp/raw/9514
index	web
published	
sourcetype	snort
splunk_server	so1
title	9514

```
> ./splunk btool inputs list tcp://9514
[tcp://9514]
_rcvbuf = 1572864
connection_host = ip
host = so1
index = network
sourcetype = snort
```

http://docs.splunk.com/Documentation/Splunk/latest/RESTREF/RESTprolog

What Splunk Logs About Itself

1. A user log onto Splunk Web (**web_access.log + web_service.log**) and go to the search app (**splunkd_ui_access.log**).
2. The user submit a search (**splunkd_access.log**)
3. SH dispatches the search to IDX (**splunkd_access.log**)
4. IDX fulfills its portion of the search (**splunkd.log** + **remote_searches.log**)
5. SH tracks the search progress (**search.log**)



1 **web_access.log**
↳ **web_service.log**
splunkd_ui_access.log
2 ↳ **splunkd_access.log**

3 **splunkd.log** → 4 **splunkd_access.log**
scheduler.log ↳ **splunkd.log**
5 **search.log*** ↳ **remote_searches.log**

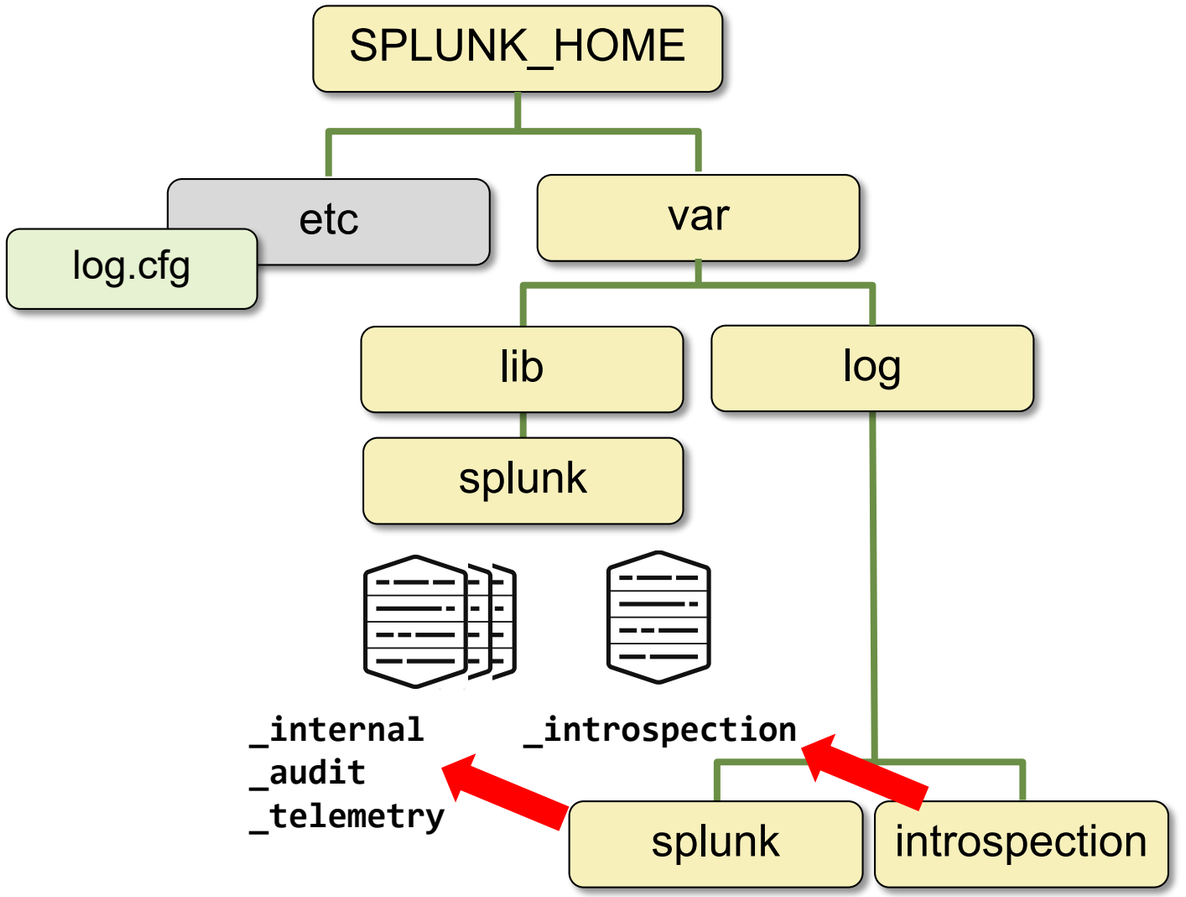
Note***

search.log is not indexed and is located in the following location per search ID in:
SPLUNK_HOME/var/run/splunk/dispatch/<sid>/

https://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/WhatSplunklogsaboutitself

Splunk Internal Log Files and Indexes

- Splunk internal logs are in **`SPLUNK_HOME/var/log`**
  - Logs are rolled based on size (25MB by default)
- Index retention:
  - **`_introspection`** = 14 days
  - **`_internal`** = 30 days
  - **`_audit`** = 6 years
  - **`_telemetry`** = 2 years (if opted in)



> **Note**
>
> Splunk logs can be an invaluable troubleshooting tool. Splunk your logs!

Splunk Log Channel Levels

- Splunk platform internal logging levels are:
  - **DEBUG** » **INFO** » **WARN** » **ERROR** » **CRITICAL** » **FATAL**
- To view and manage the log level of a particular channel, go to **Settings** > **Server settings** > **Server logging**
  - Change is immediate but not persistent
  - To make it permanent, edit **log-local.cfg** to override **log.cfg**
- To set the log level of a channel with CLI:

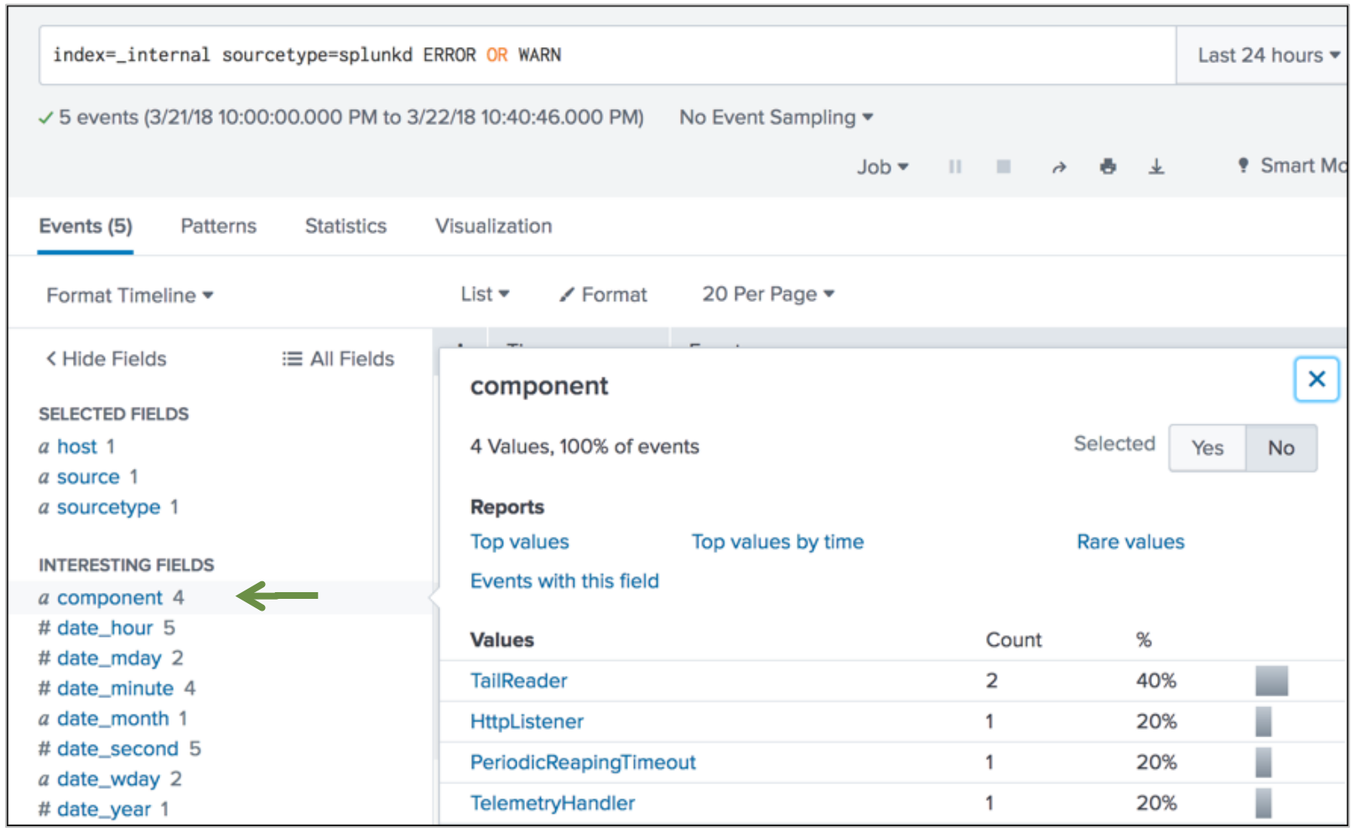
  splunk set log-level <channel> -level <level>
  - To make it permanent, edit **log-cmdline.cfg**

  https://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/Enabledebuglogging

index=_internal sourcetype=splunkd

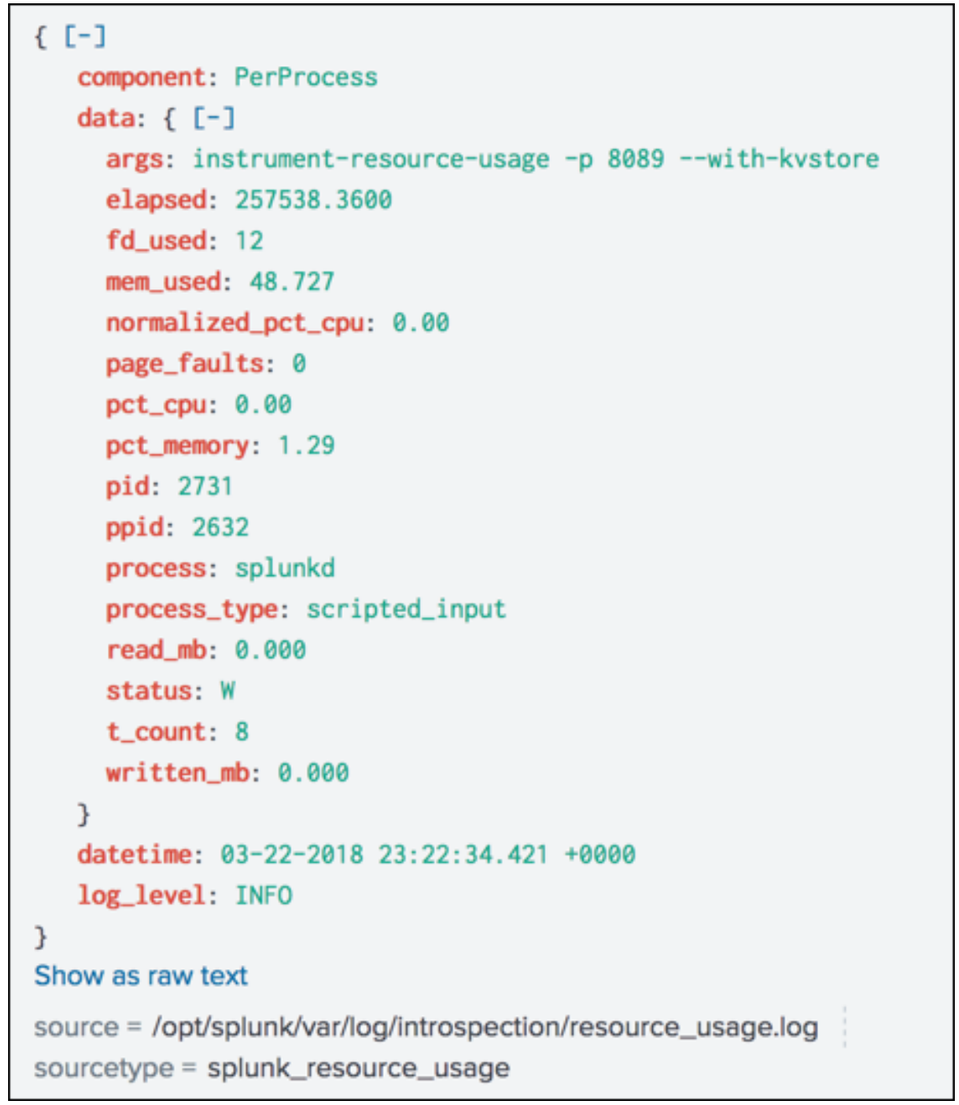
- **splunkd.log** records each *splunkd child processor* event
  - Processor log channels are assigned to the **component** field
  - To diagnose problems, filter on the **component** in question
- You may need to correlate the set of logs from multiple instances or components
  - For example:

```
index=_internal host=idx*
sourcetype=splunkd
component=tcp* (ERROR OR
WARN)
```



Introspection Data

- Splunk resource collection framework logs resource usage
  - **introspection_generator_addon** writes to **`SPLUNK_HOME/var/log/introspection`** and it is indexed to **`_introspection`**
- REST endpoints provide snapshot data
  - **`/services/server/info`** provides system configurations
    - OS info, core count, server roles, etc.
  - **`/services/server/status/resource-usage`** provides CPU, memory, and disk stats per host and per Splunk-process

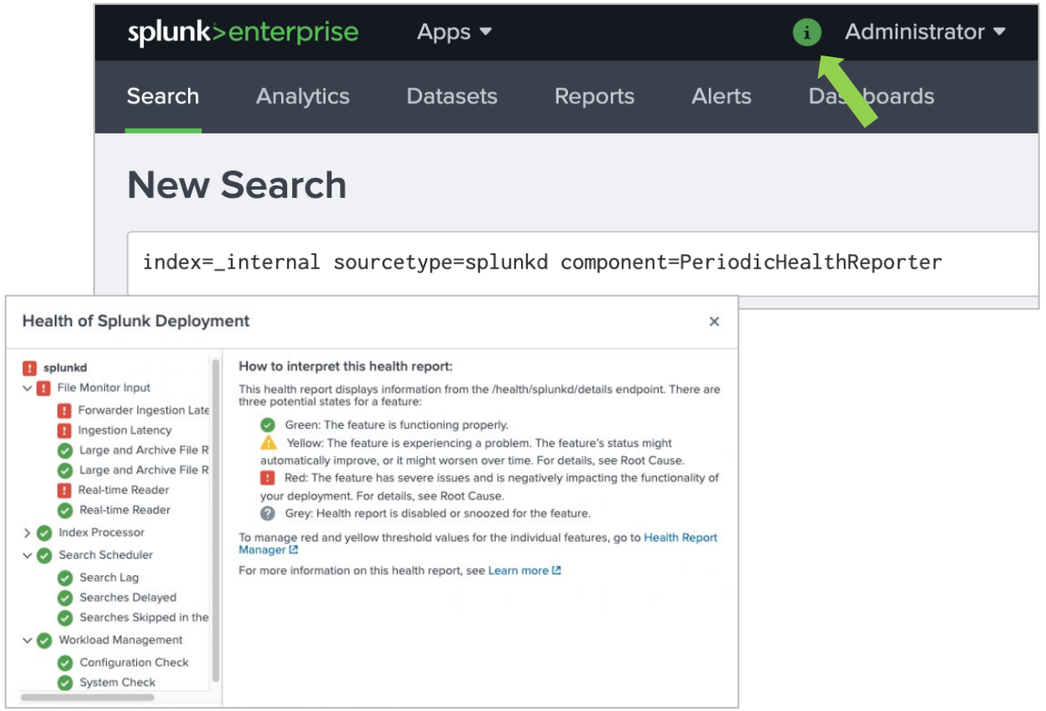


Tracking Configuration Changes

- Performed by default by the Configuration Change Tracker
  - Uses active *monitoring* if the Linux **`inotify`** API is available
  - Otherwise uses *polling* which checks **`.conf`** files every 30 seconds
  - Stores changes in the **`_configtracker`** index
- Monitors valid **`.conf`** files in **`default`** and **`local`** folders for:
  - **`SPLUNK_HOME/etc/system`**
  - **`SPLUNK_HOME/etc/apps`**
  - **`SPLUNK_HOME/etc/users`**
  - **`SPLUNK_HOME/etc/slave-apps`**
  - **`SPLUNK_HOME/etc/instance.cfg`**
- Ignores *values* for fields with sensitive data (such as hashed passwords)

Proactive Splunk Component Monitoring

- Checks the **`/services/server/health/splunkd`** REST endpoint
  - Records the status output to **`SPLUNK_HOME/var/log/splunk/health.log`**
- The status depends on the value of associated indicators
- Feature health status is reported in:
  - Green: the feature is functioning
  - Yellow: experiencing a problem
  - Red: negatively impacting the feature
  - Grey: disabled or snoozed feature



Note

- You can configure a set of alert actions (Only supports a single action per type)
- Need **`list_health`** and **`edit_health`** capability
- Configure alerts by editing **`health.conf`**. or **Settings > Health report manager**
- Each node can overwrite the settings

Splunk Instrumentation

- You can opt in to share telemetry data with Splunk
  - Improves customer support
  - Helps Splunk make future development decisions
- The ability to enable/disable instrumentation is controlled by the **`edit_telemetry_settings`** capability
  - The instrumentation page can only be accessed from each non-clustered search head
- Collected data is anonymized
  - The opt-in modal controls sharing for anonymized and support data
- License usage data is collected and sent to Splunk by default

https://docs.splunk.com/Documentation/Splunk/latest/Admin/Shareperformancedata

What is a Splunk Diag?

- Diag provides insight into your instance
  - How is the instance configured?
  - What was the condition up to the point that **`diag`** ran
- Gathers data based on Splunk components
  - OS settings, internal logs, configuration files, etc.
  - Produces a **`tar.gz`** file and **`diag.log`**
- No customer data is retrieved
  - Splunk anonymizer combs through files and replaces identified terms in a dictionary
  - http://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/AnonymizedatasamplestosendtoSupport
- Can serve as a quick backup of config files
- Run it before and after to track update/upgrade differences
  - You can index your own diag file

Inside a Diag

- **composite.xml** – the unified component list that **splunkd** uses at runtime to control its component sub-system
  - Resides in **SPLUNK_HOME/var/run/splunk**
- **etc** subdirectory – contains the diag system's etc directory
- **log** subdirectory – contains the diag system's Splunk logs
- **var** subdirectory – contains information about the indexes and index structure
- **dispatch** subdirectory – directory of search dispatches
- **systeminfo.txt** – contains OS and hardware info as well as Splunk version and build
  - **Msinfo-sum.txt** (only on Windows) – contains Windows system information: hardware, running process, services, hardware specification
- **introspection** – from **SPLUNK_HOME/var/log/introspection**
- And more...

A **diag** Run in Practice

```
[me@sh1 bin]$ ./splunk diag
Collecting components: conf_replication_summary,
consensus, dispatch, etc, file_validate, index_files,
index_listing, kvstore, log, pool, searchpeers,
suppression_listing
Skipping components: rest
Selected diag name of: diag-sh1-2019-03-23_19-55-56
Starting splunk diag...
Logged search filtering is enabled.
Skippping REST endpoint gathering...
Determining diag-launching user...
Getting version info...
Getting system version info...
Getting file integrity info...
Getting network interface config info...
Getting splunk processes info...
Getting netstat output...
(Not all processes could be identified, non-owned process
info will not be shown, you would have to be root to see
it all.)
Getting info about memory, ulimits, cpu (on windows this
takes a while)...
```

```
Getting etc/auth filenames...
Getting Sinkhole filenames...
Getting search peer bundles listings...
Getting conf replication summary listings...
Getting suppression files listings...
Getting KV Store listings...
Getting index listings...
Copying Splunk configuration files...
filtered file ...
The following certificates were excluded ...
If you have any certs that were not auto-detected, add
to an EXCLUDE rule in the [diag] stanza of server.conf.
Copying Splunk log files...
Copying Search Pool files...
Copying bucket info files...
Copying Splunk dispatch files...
Copying Splunk consensus files...
Adding manifest files...
dding cachemanager_upload.json...
Cleaning up...
Diag created: ../diag-sh1-2020-10-23_19-55-56.tar.gz
```


Diag Run Options – Component On/Off

- Diag can run for a long time, produce a gigantic file, or fail to complete

```
splunk diag
 Copying Splunk configuration files...
Exception occurred while generating diag, we are deeply sorry.
Traceback (most recent call last):
File "./site-packages/splunk/clilib/info_gather.py", line 1626, in copy_pool raise Exception("OMG!")
```

- Work around by collecting specific components:

```
splunk diag --collect=dispatch,etc
```

```
splunk diag --disable=pool --disable=etc
```

server.conf

```
[diag]
components = dispatch,etc
```

- Available components
  - index_files, index_listing, dispatch, etc, log, pool, and more

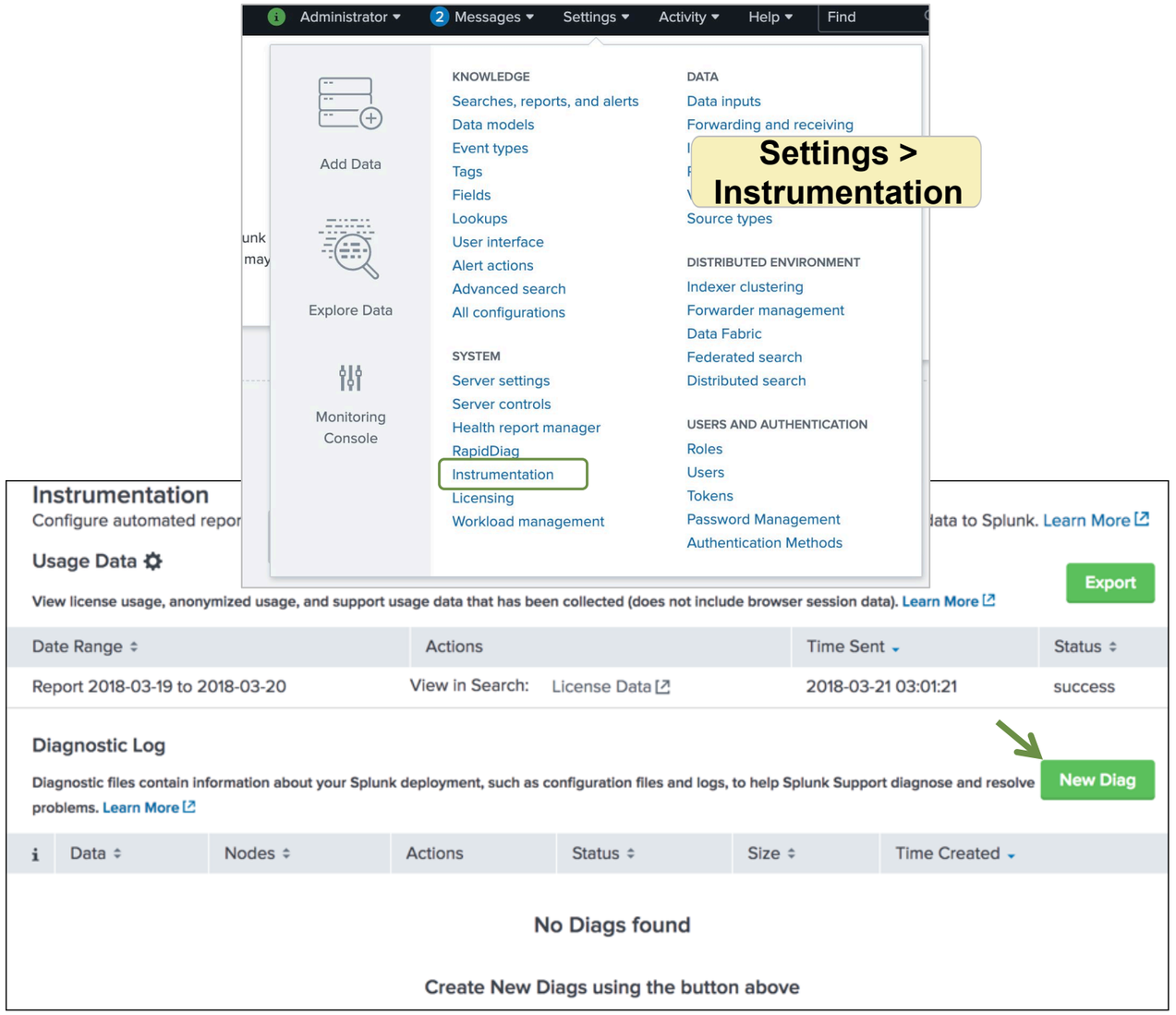
http://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/Generateadiag

Diag Run Options – Collection Filters

- Some components have levels of collection filters
  - **`--exclude "*/dispatch/*"`**
    - Files ruled out are logged in **`excluded_filelist.txt`**
  - **`--include-lookups <what>`**
  - **`--all-dumps=<bool>`**
  - **`--log-age=<days>`**
  - **`--index-files=[manifests|full]`**
  - **`--index-listing=[light|full]`**
  - **`--etc-filesize-limit=<size>`**
- Diag blocks **`splunk.secret`**, all files in **`etc/auth`**, **`etc/passwd`**
- Attempts to detect and exclude SSL certs anywhere in **`etc`**

Diag UI

- Provides a diag collection service from a search head
  - To use, the **`get_diag`** capability is required
  - Equivalent of running CLI: **`splunk diag -uri https://<host>:<mgmtPort>`**
- With non-clustered instances, the diag collection service authenticates with remote nodes via distributed search auth tokens
  - In clustered environment, it uses its respective cluster secrets
- Previously created diags expire after 30 days
  - Saved in **`SPLUNK_HOME/var/run/diags/`**
- Available actions: Recreate, Download, and Delete



RapidDiag – What is It?

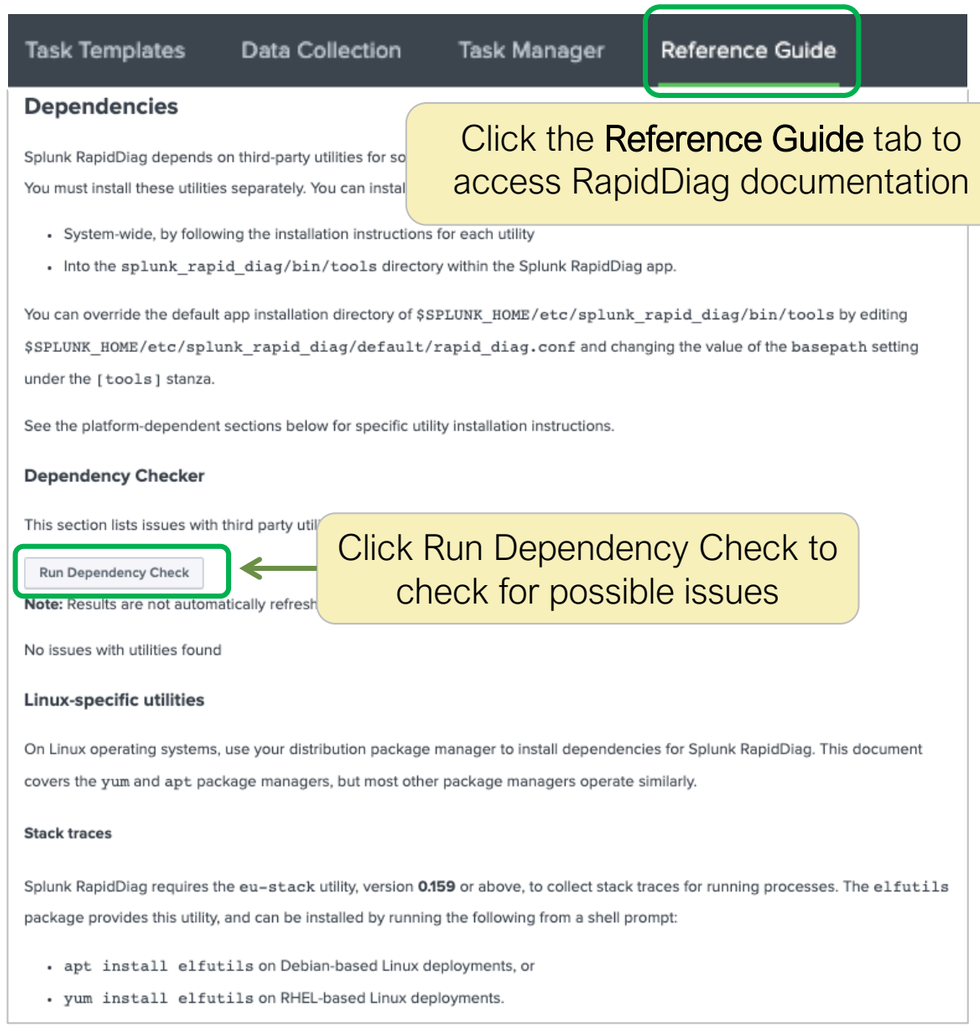
- RapidDiag is a Splunk supported app that enables “point-in-time” data collection to assist with diagnosing issues
  - Right data, Right place, Right time
- Compliments Splunk Diag by capturing additional diagnostics using pre-defined or custom templates
- Uses Json format to describe what, when, and how to collect
  - Type of collector
  - Triggers on resources (CPU/Memory)
    - For example, collects on high resource usage or long running searches
  - Triggers on logs
  - Supports periodic and one-off data collections

RapidDiag – What Does it Collect?

- Collects the following:
  - Stack dumps (pstack, eu-stacks, procdump)
  - System call traces (strace, procman)
  - Splunk diag
  - Rest endpoint outputs (`|rest search` exports)
  - I/O operations (logman, iostat)
  - Network stats and connections (netsh)
  - Process information (ps, lsof, or handle64)
- Common pre-defined templates:
  - Performance: for investigating issues related to excessive time or CPU load used by a process
    - Collector tools used: pstack (eu-stack), splunkdiag
  - Slow search performance: for investigating slow performing searches
    - Collector tools used: iops, netstat, pstack (eu-stack), splunkdiag

RapidDiag Requirements

- Currently only supports Linux OS
- Read/write privileges assigned to **admin** role
  - Also requires the **`get_diag`** capability to use app
- Rapid Diag depends on various third-party utilities to collect some of its data
  - Refer to RapidDiag documentation in the app
- Splunk RapidDiag requires the eu-stack utility, version 0.159 or above, to collect stack traces for running processes



RapidDiag CLI Upload

- Users can upload RapidDiag output file(s) to an existing support case:
  - Requires a valid support username/password

```
./splunk cmd rapidDiag upload <diag_file.tar.gz>
--<username> –auth <password> --upload_description
```

- View help/parameters with

```
./splunk cmd rapidDiag upload -h
```

- Troubleshooting:
  - Check the RapidDiag logs located at:

    `$SPLUNK_HOME/var/log/splunk/splunk_rapiddiag.log`
    - The logs show errors, number of chunks generated, and status of each chunk
  - Run netstat to check if **api.splunk.com** is accessible from the machine
- If you had a TB size file and only the 500th chunk failed, you can re-try the upload by using `-- firstchunk 500`

Splunk Troubleshooting Resources

Online Troubleshooting Manuals

https://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/Whatsinhere

http://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/IntrototroubleshootingSplunk

Common Symptoms and Solutions

https://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/Whatsinhere#Some_common_scenarios

Splunk Answers

https://community.splunk.com/t5/Splunk-Answers/ct-p/en-us-splunk-answers

Proactive Splunk Component Monitoring

https://docs.splunk.com/Documentation/Splunk/latest/DMC/Aboutfeaturemonitoring

Release Notes/Known Issues

http://docs.splunk.com/Documentation/Splunk/latest/ReleaseNotes/Knownissues

Lab Exercise 1 – Know Your Environment

Time: 20 minutes

Tasks:

- Access your Splunk CLI terminal and check the Splunk status
- Access Splunk Web and change the server name
- Install the **`tse_lab01.spl`** app
- Generate a diag for your Splunk instance and index
  - Monitor with the **index once** option
  - Index it into **`index=diag`**
- Search the diag for system information:
  - Splunk version, OS, CPU, RAM, and installed apps

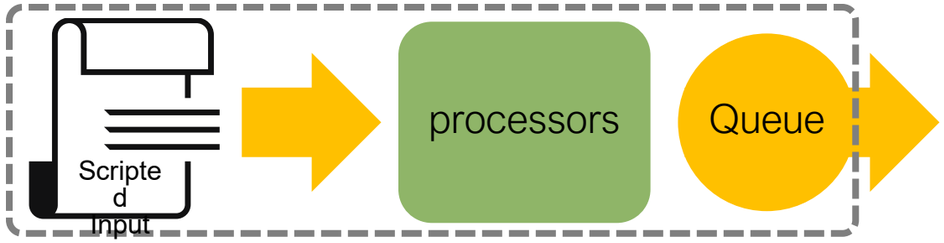
Module 2: Indexing Problems

Module Objectives

- Identify **`splunkd`** Indexing nomenclature
- Identify Indexing pipeline set and each phase
- Use Splunk Web to identify indexing performance and error messages
- Use the **`metrics.log`** to clarify the index-time problem

splunkd Indexing Nomenclature

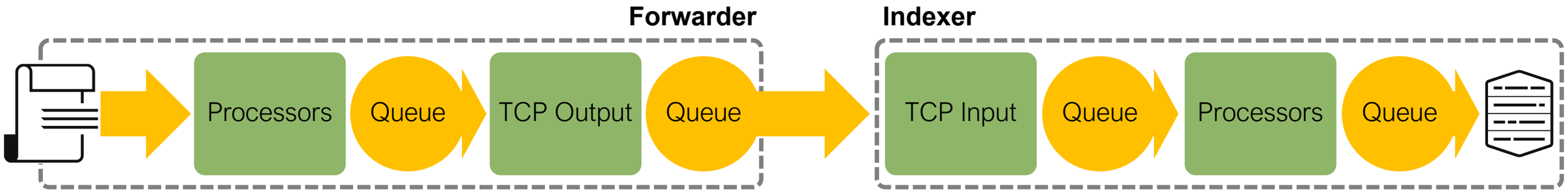
- Data enters Splunk and flows through a series of pipelines linearly
- **Module** is a container for pipelines
- **Pipeline** is a single thread grouped by type
- **Pipeline Set** is a series of pipelines that are strung together to process data from input to indexing
- **Processor** performs small but discrete tasks within a pipeline
- **Queue** is memory space to store between pipelines



SPLUNK_HOME/etc/modules/input/exec/config.xml

```
<module>
  <pipeline name="exec" type="startup">
    <processor name="exec"
               plugin="execprocessor">
      <config>
      </config>

    </processor>
    <processor name="sendOut"
               plugin="queueoutputprocessor">
      <config>
        <queueName>parsingQueue</queueName>
      </config>
    </processor>
  </pipeline>
</module>
```

Indexing Pipeline Set

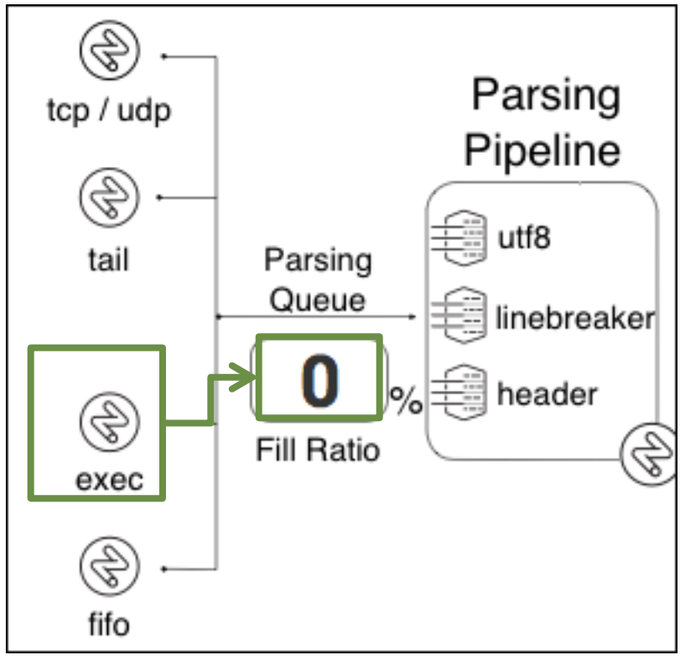


- A pipeline set spans across the index-time deployment topology
- Raw data from a source is pushed through the pipeline set using processors and queues
  - Processors manipulate events flowing in and out of queues
  - Queues are bounded by memory and can get blocked if full
  - Pipeline activities are captured in **metrics.log**
- With extra resources, Splunk can run multiple pipeline sets in parallel
- Detail diagram
  - http://wiki.splunk.com/Community:HowIndexingWorks

Input Pipeline

- How do we get from the **exec** pipeline to the **parsing** pipeline?
  - **readin** / **sendout** processors work in between queues

```
index=_internal source=*metrics.log group=pipeline name=exec processor=*
index=_internal source=*metrics.log group=queue name=parsingqueue
```



Configuration files of interest

- inputs.conf
- wmi.conf
- regmon-filters.conf
- props.conf

Attributes of interest

- CHARSET
- NO_BINARY_CHECK
- CHECK_METHOD
- initCrcLength

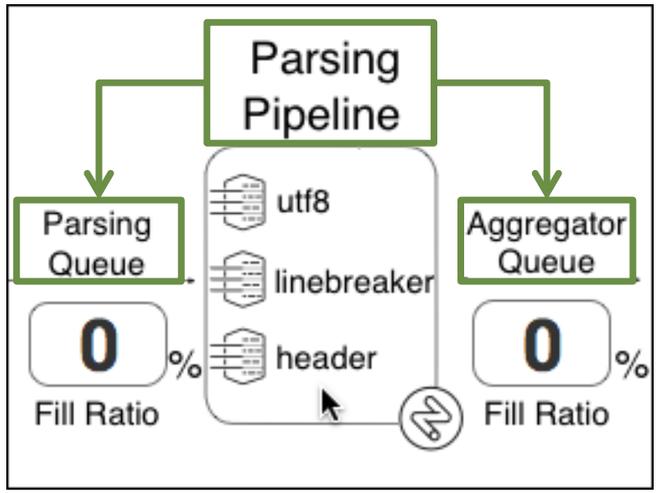
Note

Sendout will be at the end of all pipelines and it is responsible by taking the data and pushing to the next queue. Lots of **sendout** activity is not indicative of a problem and can be safely ignored when investigating a pipeline.

Parsing Pipeline

- What happens after parsing queue?
  - **`readerin`** processor takes in data from **`parsingqueue`**
    - Runs through **`utf8`**, **`linebreaker`**, and **`header`**
  - **`sendout`** processor sends out to **`aggqueue`**

```
index=_internal source=*metrics.log group=pipeline name=parsing processor=*
index=_internal source=*metrics.log group=queue name=aggqueue
```



Configuration files of interest

- `props.conf`

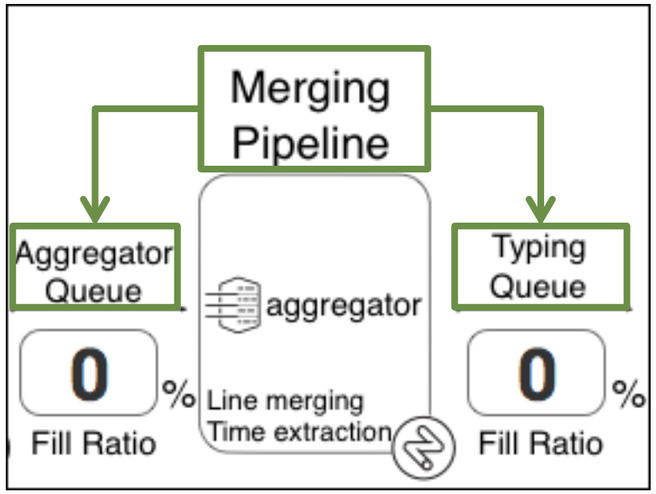
Attributes of interest

- `CHARSET` (is checked)
- `LINE_BREAKER` (single line)
- `TRUNCATE`
- `HEADER_MODE`

Merging Pipeline

- **aggregator** identifies the event boundary and performs timestamp extraction
  - Sends out to **typingqueue** queue
- **winparsingqueue** has its own **aggregator**
  - Feeds directly into **typingqueue**

```
index=_internal source=*metrics.log group=pipeline name=merging processor=*
index=_internal source=*metrics.log group=queue name=typingqueue
```



Configuration files of interest

- `props.conf`

Attributes of interest (events)

- `SHOULD_LINEMERGE`
- `BREAK_ONLY_BEFORE`
- `MUST_BREAK_AFTER`
- `MAX_EVENTS`

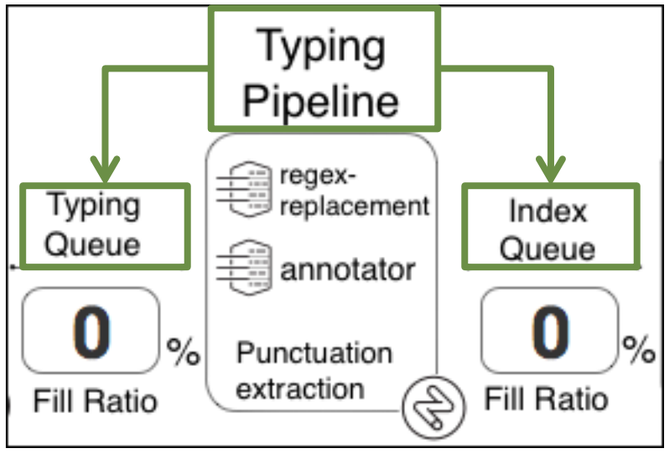
Attributes of interest (timestamps)

- `TIME_PREFIX`
- `TIME_FORMAT`
- `MAX_TIMESTAMP_LOOKAHEAD`
- `DATETIME_CONFIG`
- `MAX_DAYS_AGO`
- `MAX_DAYS_HENCE`
- `TZ`

Typing Pipeline

- Performs the transformation tasks
  - Runs through **`regexreplacement`**, **`annotator`**, and **`previewout`**
  - Sends out to **`indexqueue`**

```
index=_internal source=*metrics.log group=pipeline name=typing processor=*
index=_internal source=*metrics.log group=queue name=indexqueue
```



Configuration files of interest

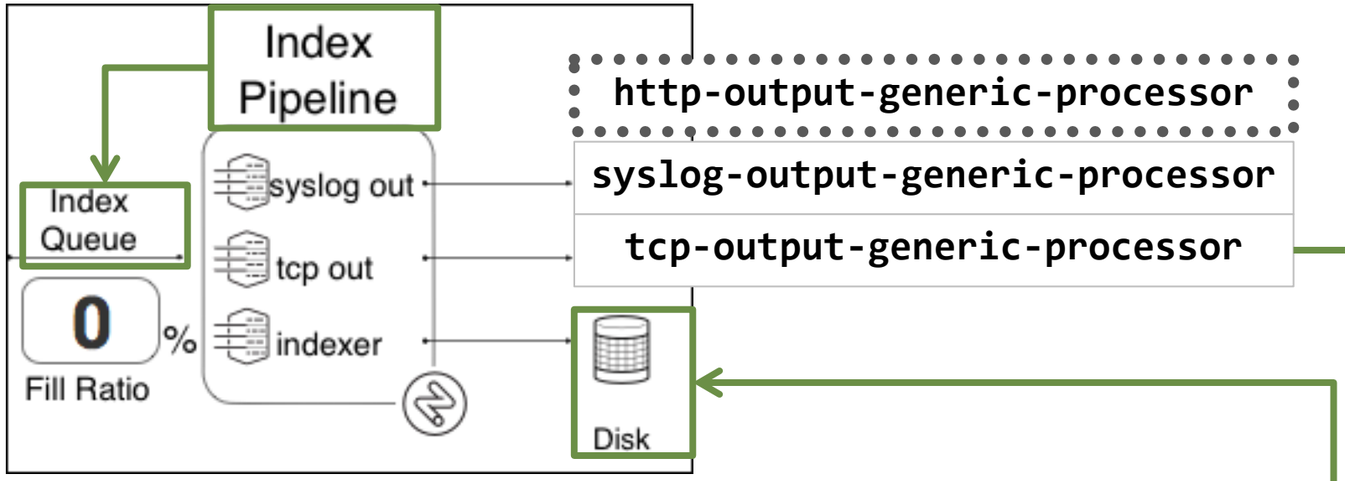
- `props.conf`
- `transforms.conf`

Attributes of interest

- `TRANSFORMS-xxxx`
- `SEDCMD`
- `SOURCE_KEY`
- `DEST_KEY`
- `REGEX`
- `FORMAT`

Index Pipeline

```
index=_internal source=*metrics.log group=pipeline name=indexerpipe processor=*
```



syslogout sends selected events to a configured 3rd party server using syslog format

tcpout sends selected events to a configured tcpout group in raw or cooked format

splunktcp cooked indicates the data will be output to another Splunk using Splunk S2S protocol

tcpout_* (per group)

↓

Splunktcp parsed (per event)

Value of processor field in **_internal** events

- **indexandforward**
- **signing**
- **indexer**
- **indexin**
- **Index_thruput**

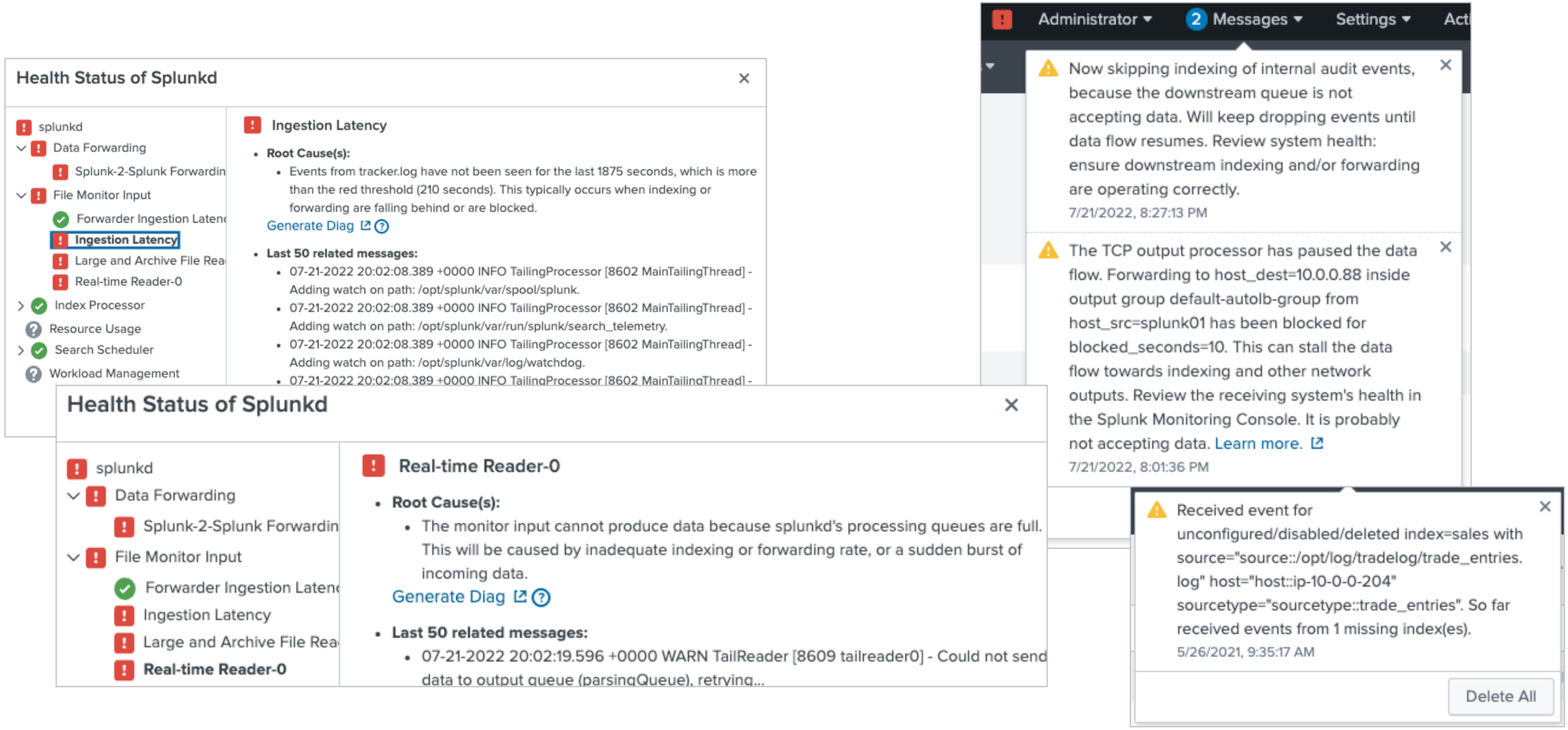
indexer meters Non-Splunk internal indexing

indexin is for internal logs and does not meter license usage

index_thruput emits indexing thruput measurements to **metrics.log**

 | 11 November 2022

Indexing Error Messages in Splunk Web

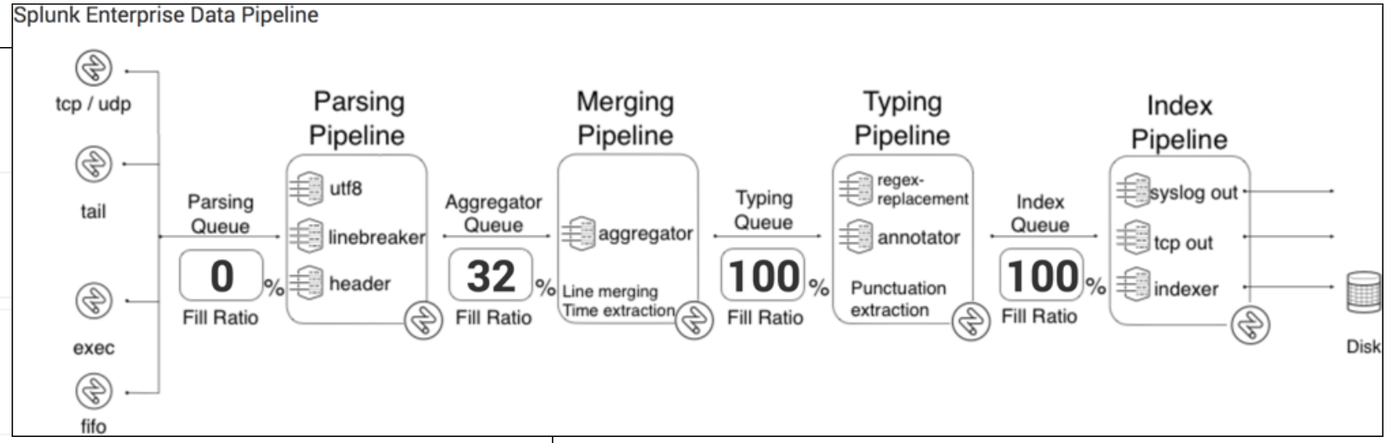


Indexer Health Report

- Proactive Splunk Component Monitoring reports the current health of an indexer, on a per instance basis
  - **`feature:disk_space`**
    - Tracks whether Splunk index filesystems contain sufficient free space based on the **`minFreeSpace`** setting
  - **`feature:buckets`** (reports per index)
    - Monitors a number of buckets created during last 60 minutes
    - Tracks the percentage of small buckets created over the last 24 hours
    - Buckets rolled during a restart or due to **`maxHotSpanSecs`** are not counted
  - **`feature:splunkoptimize_processes`**
    - Tracks whether the bucket optimization process is falling behind

Viewing MC Indexing Performance

Historical



Snapshot

Note

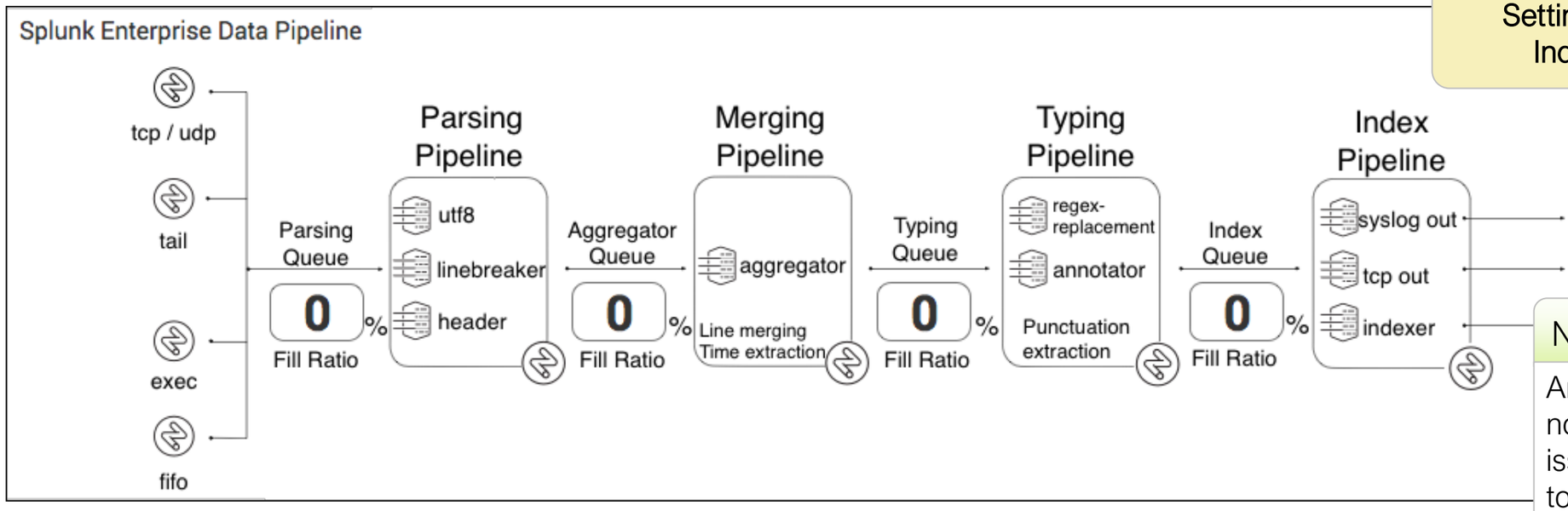
Snapshot may not tell the whole picture. Check the **Historical** option to detect blockages and interruptions.

Understanding metrics.log

- Contains a variety of introspection data including Splunk pipeline statistics
- Samples every 30 seconds and writes metrics of top 10 results in each category
  - Queue sizes
  - CPU usage per processor
  - Forwarding and indexing thruput
- **`group`** indicates the data type, such as pipeline, queue, thruput, etc.
  - **`group=pipeline`** plots the frequency (**`executes`**) and the duration of the pipeline process machinery (**`cpu_seconds`**)
  - **`group=queue`** displays the data to be processed
    - **`current_size`** can identify which queues are the bottlenecks
    - **`blocked=true`** indicates a busy pipeline
    - Cleared as soon as the pipeline processor catches up with its task
    - A long sequence of blocked queue messages indicates an indexing issue

Troubleshooting with metrics.log

- Checking **metrics.log** across the topology reveals the whole picture
  - Downstream slowdown results in upstream blockage



Settings > Indexing > Performance > Indexing Performance: Instance

Note

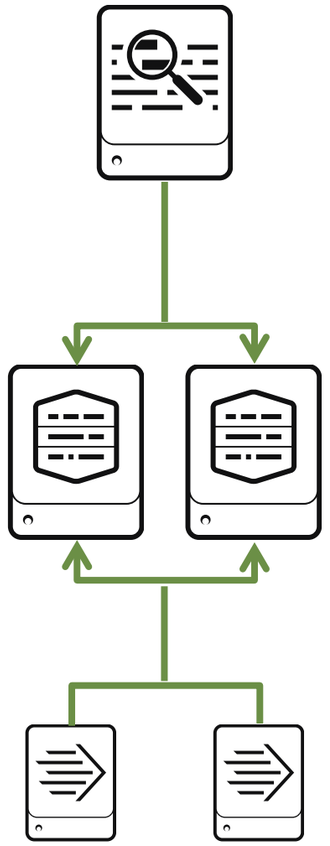
An occasional queue filling up does not indicate an issue. It becomes an issue when it remains full and starts to block other queues.

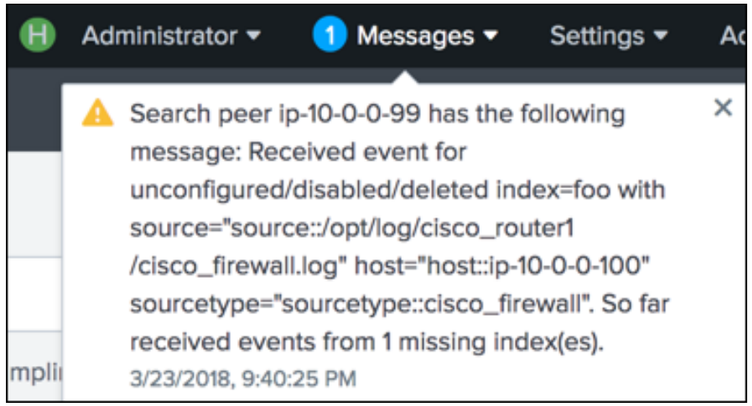
```
index=_internal source=*metrics.log host=xyz group IN(pipeline, queue)
```

```
 02-23-2021 01:08:43.802 +0000 INFO  Metrics - group=queue, name=indexqueue, blocked=true,
    max_size_kb=500, current_size_kb=499, current_size=968, largest_size=968,
smallest_size=968
 02-23-2019 01:10:39.802 +0000 INFO  Metrics - group=pipeline, name=typing, processor=sendout,
    cpu_seconds=0.05710199999999998, executes=134716, cumulative_hits=1180897
```

metrics.log Perspective (Gotchas)

- Metrics.log does not provide error messages or diagnostics
- For example, mistake in **`inputs.conf`** or **`transforms.conf`** drops events destined to non-existent indexes
  - **`metrics.log`** thruput data indicates normal operations
  - A warning registers with the **`IndexerService`** component
  - **`lastChanceIndex`** in **`indexes.conf`** provides ability to capture events destined to non-existent indexes
    - Assign to an index that only admins can access and set an alert






```
index=_internal group=per_index_thruput
series=foo metrics
03-23-2019 21:39:55.655 +0000 INFO
Metrics - group=per_index_thruput,
series="foo", kbps=4.775483123820209,
eps=0.16128387122579613,
kb=148.0458984375, ev=5, avg_age=202929,
max_age=338215
```

Useful Pipeline Searches with metrics.log

- How much time is Splunk spending within each pipeline?

```
index=_internal source=*metrics.log* group=pipeline
| timechart sum(cpu_seconds) by name
```

- How much time is Splunk spending within each processor?

```
index=_internal source=*metrics.log* group=pipeline
| timechart sum(cpu_seconds) by processor
```

- What is the 95th percentile of measured queue size?

```
index=_internal source=*metrics.log* group=queue
| timechart perc95(current_size) by name
```

- What is the maximum number of entries used in each queue? (1000 is max queue size, except for forwarding)

```
index=_internal source=*metrics.log* group=queue
| timechart max(current_size) by name
```


Lab Exercise 2 – Troubleshoot Indexing issue

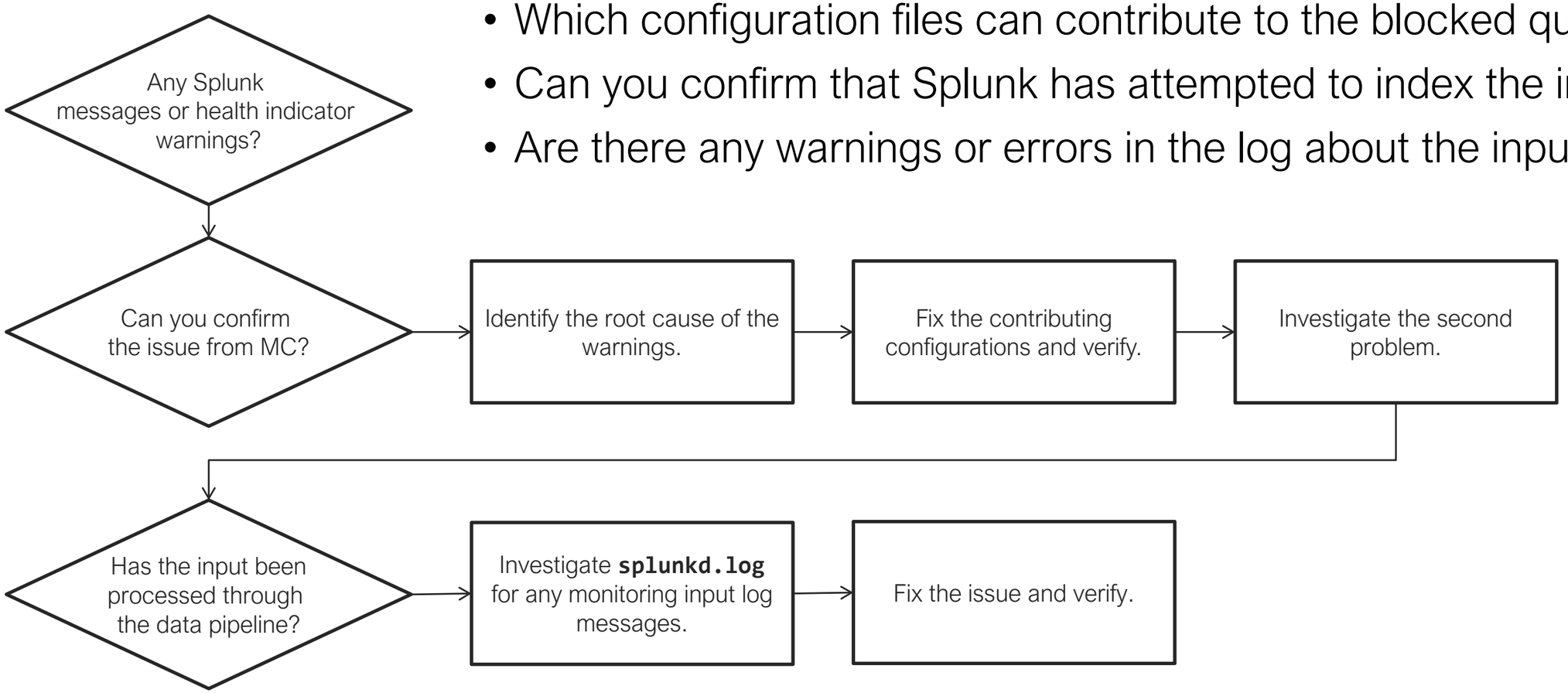
Time: 25 minutes

Tasks:

- Configure Monitoring Console in Standalone mode
- Install the **`tse_lab02.spl`** app
- Enable the **`trade_entries.log`** monitor input
- Search **`index=* sourcetype=trade_entries`** for events
- Investigate pipeline fill-ratio with Monitoring Console
  - Fix the pipeline issue
- Resolve the missing data problem
  - Analyze the **`splunkd.log`** to resolve the missing data problem

Lab Exercise 2 – Troubleshooting Suggestions

- In which processor has the blocked queue first occurred?
- Which configuration files can contribute to the blocked queue?
- Can you confirm that Splunk has attempted to index the input?
- Are there any warnings or errors in the log about the input?



Module 3:
Input Configuration Problems

Module Objectives

- Diagnose and clarify data input problems
- Use Monitoring Console to identify indexing issues

Review: Index Time Best Practices

- To maximize the performance and resource utilization, provide specific settings
  - The defaults are designed for flexibility, but they can be expensive
  - Remember the settings can be specified in an app
- For inputs, explicitly specify host, source type, source and index
  - Separate high-velocity sources from low-velocity sources
  - Combine things frequently searched together in the same index
- For parsing-phase pipelines, always specify:
  - Event boundary
  - Timestamp
- Distribute data uniformly to all indexers

splunk> Best Practice

The "Great 8" Per Sourcetype (props.conf)

HF/Indexer

```
MAX_TIMESTAMP_LOOKAHEAD
TIME_PREFIX
TIME_FORMAT
LINE_BREAKER
SHOULD_LINEMERGE = false
TRUNCATE
```

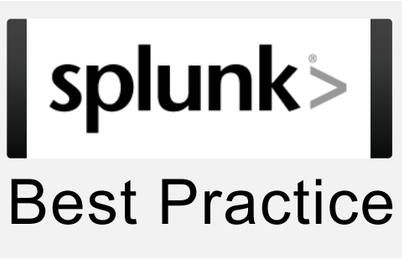
UF

```
EVENT_BREAKER
EVENT_BREAKER_ENABLE = true
```


Clarifying the Input Problems

- Start on the system that is reading the sources
- Establish expected vs. observed behavior
  - Is Splunk reading files it shouldn't read or failing to read the files it should be reading?
  - Is Splunk reading the same files more than once?
    - Are all files re-read, a subset of files, or only parts of a file?
- Enable Splunk debugging on the source system

  `./splunk start --debug`
  - Tells how often Splunk is actually checking the problem files
  - **BEST PRACTICE:** Roll the log files before enabling debugging
  - Do not leave debug enabled!
    - When you notice the problem, stop the instance and restart without the **`--debug`** flag



Notable splunkd Input Log Channels

Monitor	Network	Scripted
TailingProcessor	NetUtils	ExecProcessor
TailReader	TcpInputConfig	ModularInputs
WatchedFile	TcpInputProc	
	UDPInputProcessor	WinEventLogInputProcessor
ChunkedLBProcessor		WinEventLogChannel
ArchiveProcessor	TcpOutputProc	WinEventLog

Reminder

Changing log levels via Splunk Web is immediate and persists until the next reboot, but editing the **SPLUNK_HOME/etc/log-local.cfg** file is permanent and requires Splunk restart.

- Monitor input
  - Try to point to specific files instead of directories or using wildcards
  - When wildcards are used, Splunk creates an implicit whitelist
- Network input
  - The TCP/UDP data must be able to be parsed by the receiving Splunk ports
- Scripted input
  - Anything the script prints to **stdout** is indexed
  - Windows inputs are scripted (modular) inputs and have their own designated stanza

Common Monitor Input Errors

- Permission issue
  - Splunk is running as a user who doesn't have access permissions to the files specified in the input settings
- Network problem
  - If Splunk is monitoring the files over NFS or SMB, there could be something blocking that network access
- Status of monitored files
  - CRC problems – similar file confusion
  - Missing data or duplicate indexing
- Lagging input

Status of Monitored Files

- By default, Splunk performs a cyclical redundancy check (CRC) against the first and last 256 bytes of a file to prevent the same from re-indexing
  - Runs a CRC on every file it monitors during a startup
  - The CRC helps Splunk to determine if it has read the file before or not
  - **`fishbucket`** maintains the CRC state for monitored files
- Check the current file monitoring status
  - CLI: **`splunk list inputstatus [-input | -type]`**
  - REST: **`/services/admin/inputstatus/TailingProcessor:FileStatus`**

inputs./opt/log/tradelog.parent	/opt/log/tradelog/trade_entries.log
inputs./opt/log/tradelog.type	directory
inputs./opt/log/tradelog/trade_entries.log.file position	28393903
inputs./opt/log/tradelog/trade_entries.log.file size	28393903
inputs./opt/log/tradelog/trade_entries.log.parent	/opt/log/tradelog/trade_entries.log
inputs./opt/log/tradelog/trade_entries.log.percent	100.00
inputs./opt/log/tradelog/trade_entries.log.type	open file

Note

A fast growing file can show >100%. If tailing is not reading data, it could mean a queue is blocked or the fishbucket entry is corrupted.

CRC Problems

- A long standard header or small log files can cause CRC confusion
  - If the first half changes, Splunk re-reads the whole file
  - If the second half changes, Splunk reads from the last stored seek pointer
  - Check the internal index for possible clues:

```
index=_internal sourcetype=splunkd component=watchedfile
```

- Check the fishbucket and change/reset the tailing records if needed:

```
splunk cmd btprobe -h
splunk cmd btprobe -d <fishbucket_path> -k <file> --validate
splunk cmd btprobe -d <fishbucket_path> --file <file> --reset
```

- Possible remedy:
  - **initCrcLength** and/or **crcSalt=<SOURCE>** settings in **inputs.conf**
  - Do not use in a scenario in which log files get renamed or moved

Using `initCrcLength`

- By default, Splunk checks the first 256 characters of a file, and creates a CRC fingerprint for the file
  - Have I read this before?

```
[monitor://<path>]
...
initCrcLength = 256 <range 256 - 1048576>
```

- If a file won't read, it often has too long a header (same CRC)
  - "I've seen this already. Maybe it's a rolled file. I'm not going to re-read it"

Important

Changing *`initCrcLength`* and restarting Splunk re-indexes the data.

- Increase `initCrcLength` in `inputs.conf` to tell Splunk to read beyond the first 256 bytes, and create a unique CRC

Using `crcSalt`

- `crcSalt` forces the input to ingest files that have matching CRCs, by creating a unique CRC:
  - `crcSalt = <string>`: the string is added to the CRC
  - `crcSalt = <SOURCE>`: the full directory path to the source file is added to the CRC
- Useful if you need to
  - Forward to a new environment
  - Transition from test index to production

```
[monitor://<path>]
...
crcSalt = <string>|<SOURCE>
```

Important

Do not use `crcSalt = <SOURCE>` with rolling log files.

Duplicate Indexing

- If Splunk processes a file more than once, it results in duplicate data and can possibly lead to license violations
- Look for **`offset`** in the **`WatchedFile`** component:

  `index=_internal sourcetype=splunkd component=watchedfile`

  - *`Checksum for seekptr didn't match, will re-read entire file`*
  - *`File too small to check seekcrc`*
  - *`Will begin reading at offset=0`* means a file is new (or rolled)
    - Seeing this twice in other conditions means it is not good

```
index=abc sourcetype=xyz
| convert ctime(_indextime) AS idxtime
| stats count dc(idxtime) as numIndexed, values(source), values(idxtime) by _raw
| where count > 1
```


File Monitor Input Health Report

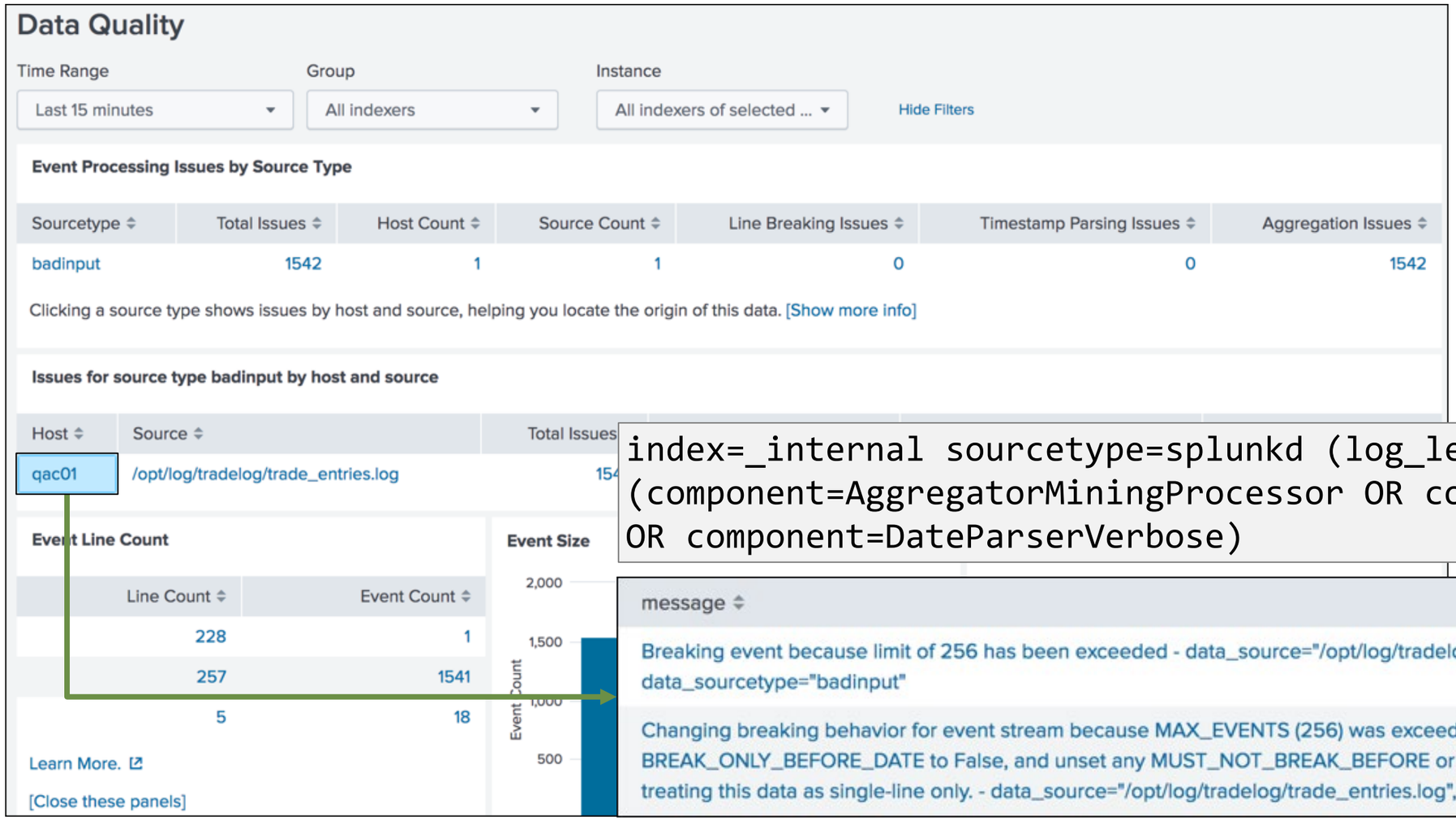
- Proactive Splunk Component Monitoring reports the current health of file monitor inputs
  - **`feature:batchreader`**
    - Reflects the number of consecutive times the Batch File Reader was unable to insert data into Splunk queues
  - **`feature:tailreader`**
    - Reflects the number of consecutive times the Tail File Reader was unable to insert data into Splunk queues

Network Input Errors and Fixes

- Network problems – the source is not reaching the indexer
  - Unblock what is blocking (most likely a firewall or a wrong port)
- Routing problems – UDP traffic may not be routable to the indexer
  - Use TCP when you can or use an intermediate Splunk forwarder
- Reliability problems – a flaky network can cause data loss or corruption
  - Set up an intermediate forwarder which can cache the stream
  - Stand up a syslog server to collect the stream and have Splunk monitor
- Host extraction problems – Splunk uses reverse DNS lookup
  - Look to the data itself for a host extraction
  - Use separate ports for each host and explicitly set the host name
  - Consider using tags to help clarify hosts

Monitoring Input Data Quality

MC > Indexing > Inputs > Data Quality



Reveals issues with line breaking, event breaking, and timestamp extraction

```
index=_internal sourcetype=splunkd (log_level=WARN OR log_level=ERROR)
(component=AggregatorMiningProcessor OR component=LineBreakingProcessor
OR component=DateParserVerbose)
```

message
Breaking event because limit of 256 has been exceeded - data_source="/opt/log/tradelog/trade_entries.log", data_host="qac01", data_sourcetype="badinput"
Changing breaking behavior for event stream because MAX_EVENTS (256) was exceeded without a single event break. Will set BREAK_ONLY_BEFORE_DATE to False, and unset any MUST_NOT_BREAK_BEFORE or MUST_NOT_BREAK_AFTER rules. Typically this will amount to treating this data as single-line only. - data_source="/opt/log/tradelog/trade_entries.log", data_host="qac01", data_sourcetype="badinput"

Troubleshooting Indexing Latency

- A large delta can lead to missing results
  - Search results are events with timestamps only within the window at the time of the search
  - Inputs using **`ArchiveProcessor`** are handled serially
    - The second file in the zip package always gets indexed after the first one
- How long does it take to make your indexed data searchable?

```
index=a source=b | eval latency = round((_indextime - _time),2)
| timechart min(latency) avg(latency) max(latency) by host
```

- Compare the delay between your log and the Splunk internal logs
  - If the delay is about the same, then it is likely a forwarding issue
    - Check the **`thruput`** setting: **`splunk btool limits list thruput`**
    - Check **`metrics.log`**

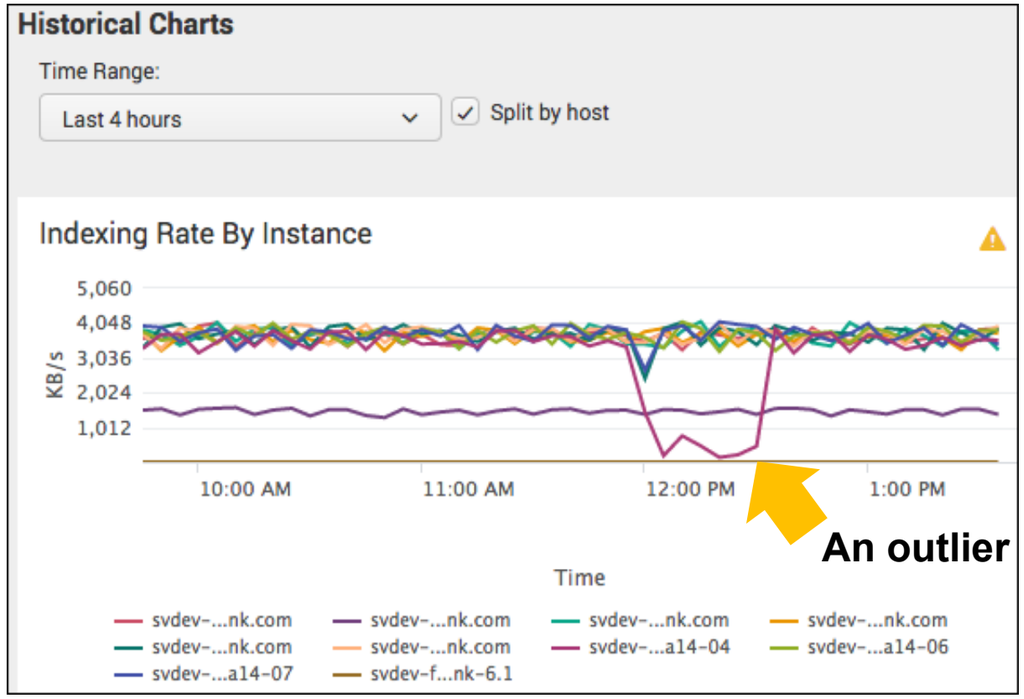
```
index=_internal metrics host=<uf> group=per_sourcetype_thruput
| timechart avg(kbps) by series
```

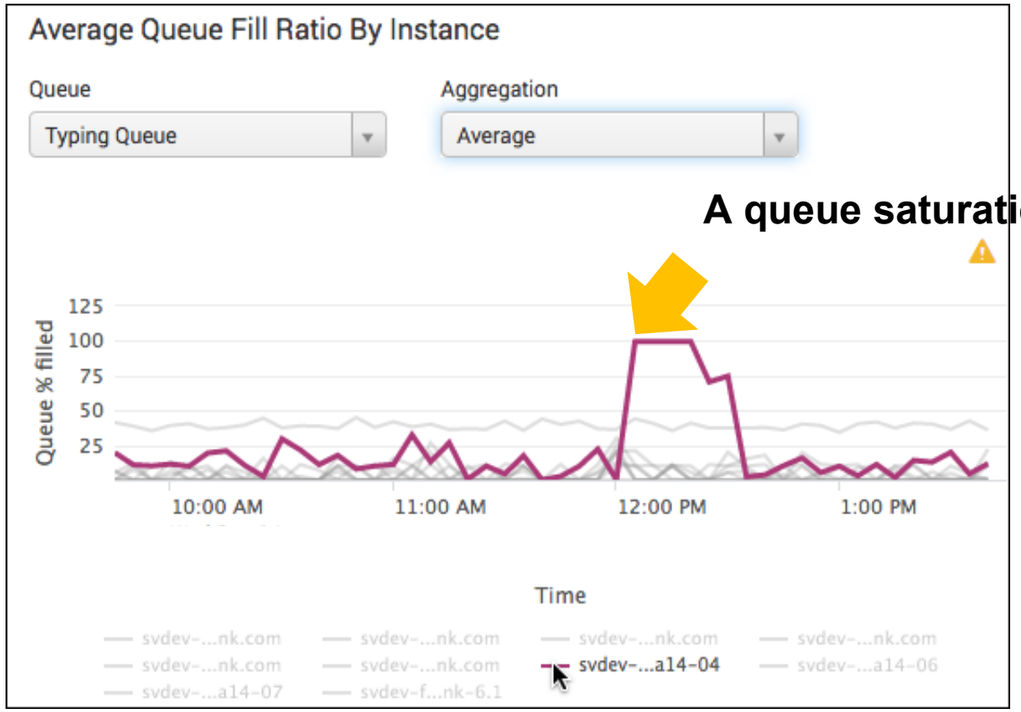
  - If the delay is significant only on your log, then it indicates an input issue

Troubleshooting Indexing Latency (cont.)

Monitoring Console's indexing performance dashboards display increasingly granular details about indexing issues

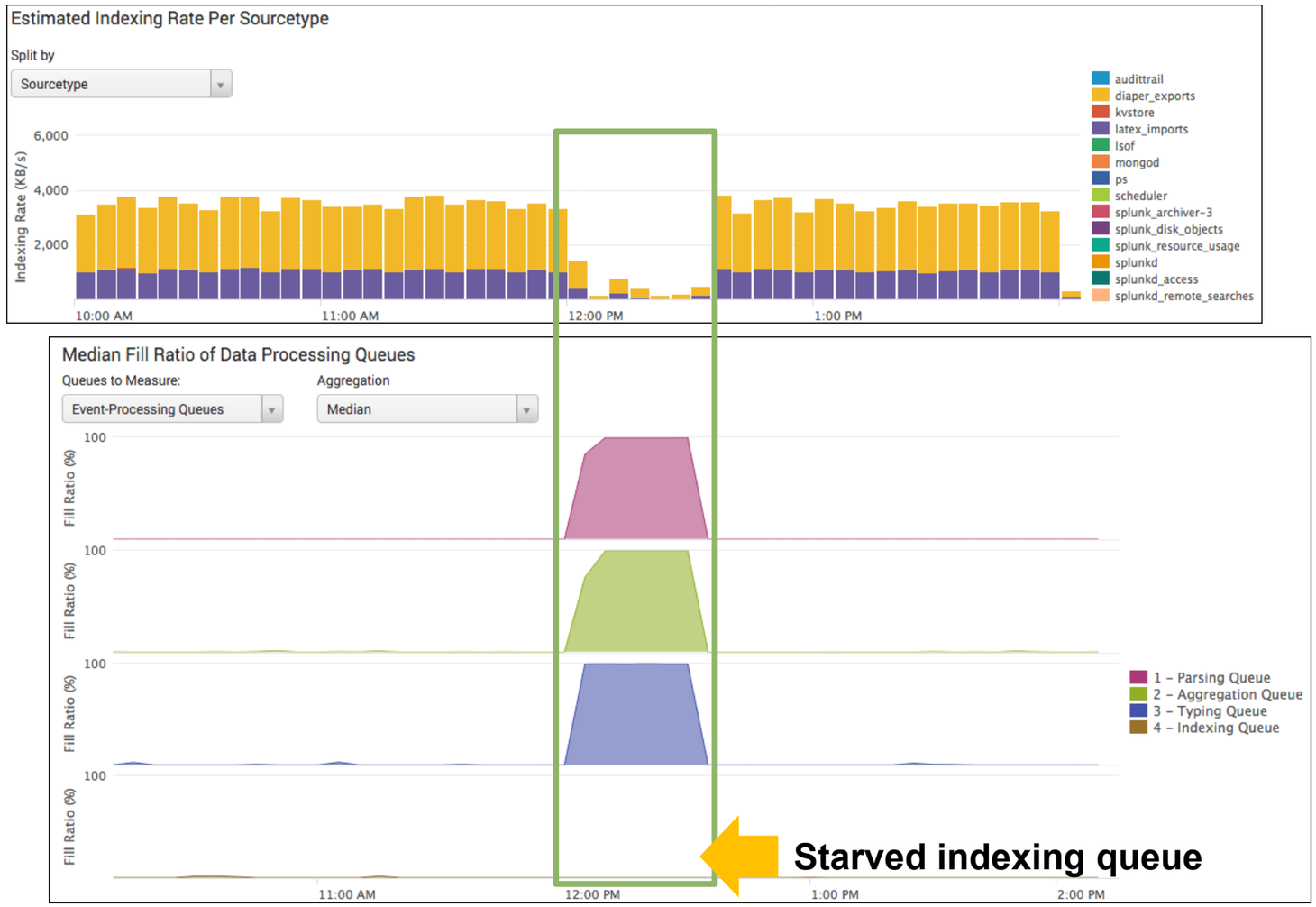
MC > Indexing > Performance > Indexing Performance: Deployment





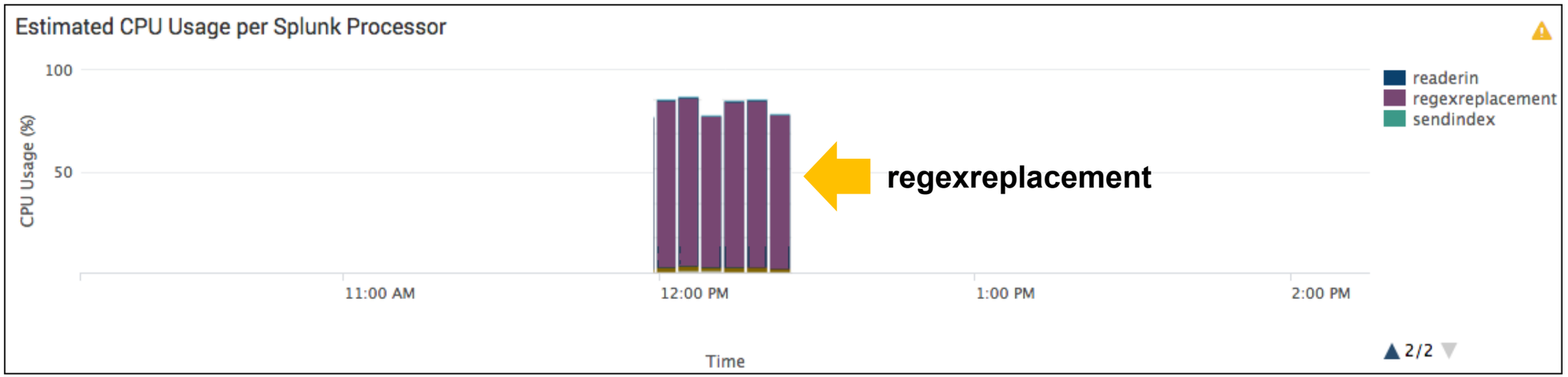
Troubleshooting Indexing Latency (cont.)

MC > Indexing > Performance > Indexing Performance: Instance



Troubleshooting Indexing Latency (cont.)

MC > Indexing > Performance > Indexing Performance: Instance



- Search for specific hosts / sources / sourcetypes to find duplicate stanzas for possible anomalies and precedence issues
- Enable CPU time metrics for **`RegexProcessor`** for debugging purposes

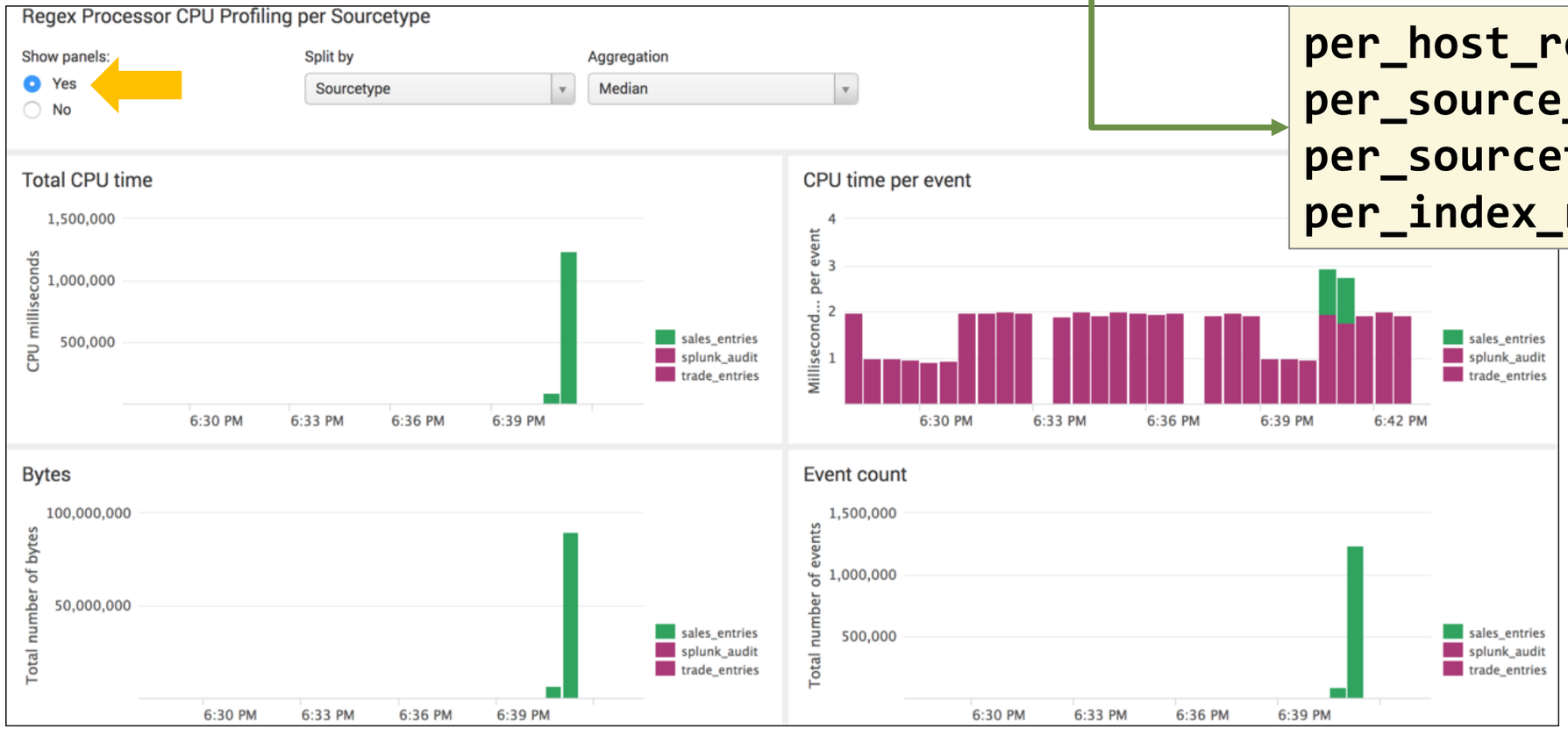
limits.conf

```
[default]
regex_cpu_profiling = true

[metrics]
maxseries = 100
```


 | 11 November 2022

Troubleshooting Indexing Latency (cont.)

`regex_cpu_profiling` outputs **`regex_cpu`** groups to **metrics.log**




```
per_host_regex_cpu
per_source_regex_cpu
per_sourcetype_regex_cpu
per_index_regex_cpu
```

Lab Exercise 3 – Troubleshoot Input Issues

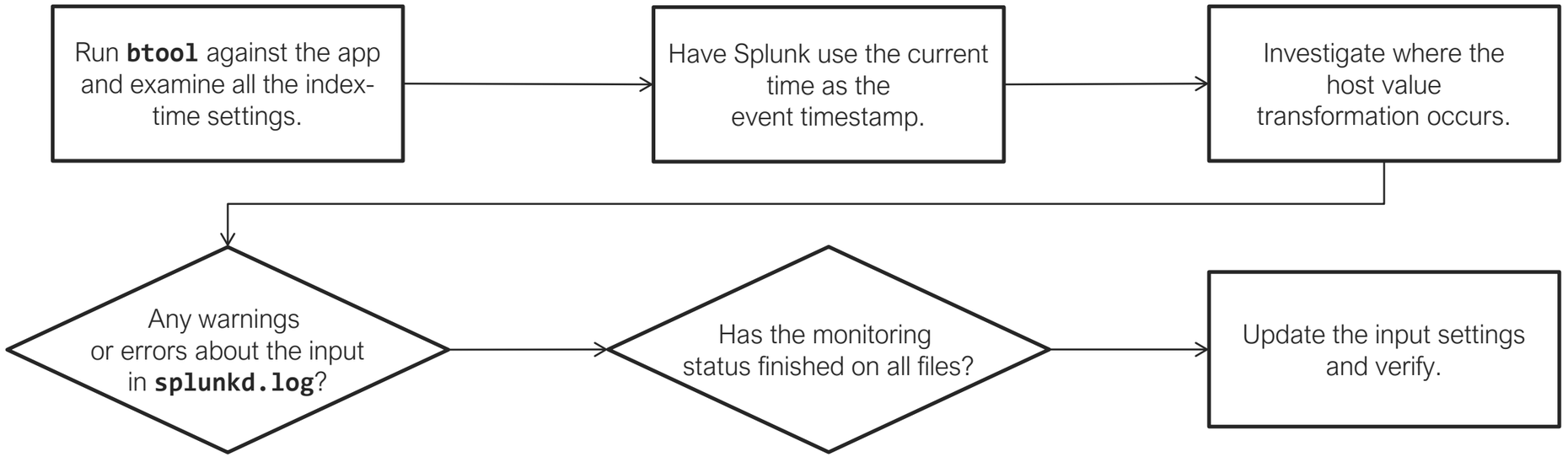
Time: 25 minutes

Tasks:

- Install the **`tse_lab03.spl`** app
- Enable the input **`tse_lab03/data/cidr`**
- Search **`index=cidr`** for events
  - Only shows 1 host and 1 source
  - Auto timestamp extracted a wrong event time
- Fix the input configurations
  - There should be 20 hosts and 20 sources
  - Events should be set to system time

Lab Exercise 3 – Troubleshooting Suggestions

- What index-time configurations has the app introduced?
- What Splunk time extraction options can adjust the event-time?
- Where does the host transformation get invoked?
- Are there any warnings or errors in the log about the input? How about the status of tailing?



Module 4: Deployment and Forwarder Problems

Module Objectives

- Troubleshoot deployment server problems
- Review forwarding and receiving problems

Deployment Problems

- Deployment server (DS) performance depends on the DS specs, the phone home interval, the number of clients, and the total size of the apps

http://docs.splunk.com/Documentation/Splunk/latest/Updating/Calculatedeploymentserverperformance

- Forwarder management UI supports a subset of the deployment options
  - For advanced configurations, edit **`serverclass.conf`**
    - **CAUTION**: Once edited, you may not be able to configure it via the UI anymore

http://docs.splunk.com/Documentation/Splunk/latest/Updating/Forwardermanagementcompatibility

- My apps are not appearing on my client instances
  - Is the client contacting the correct deployment server?
  - Is the deployment server correctly matching the client in **`serverclass.conf`**?
  - Are Include and Exclude in use? Are they correct?

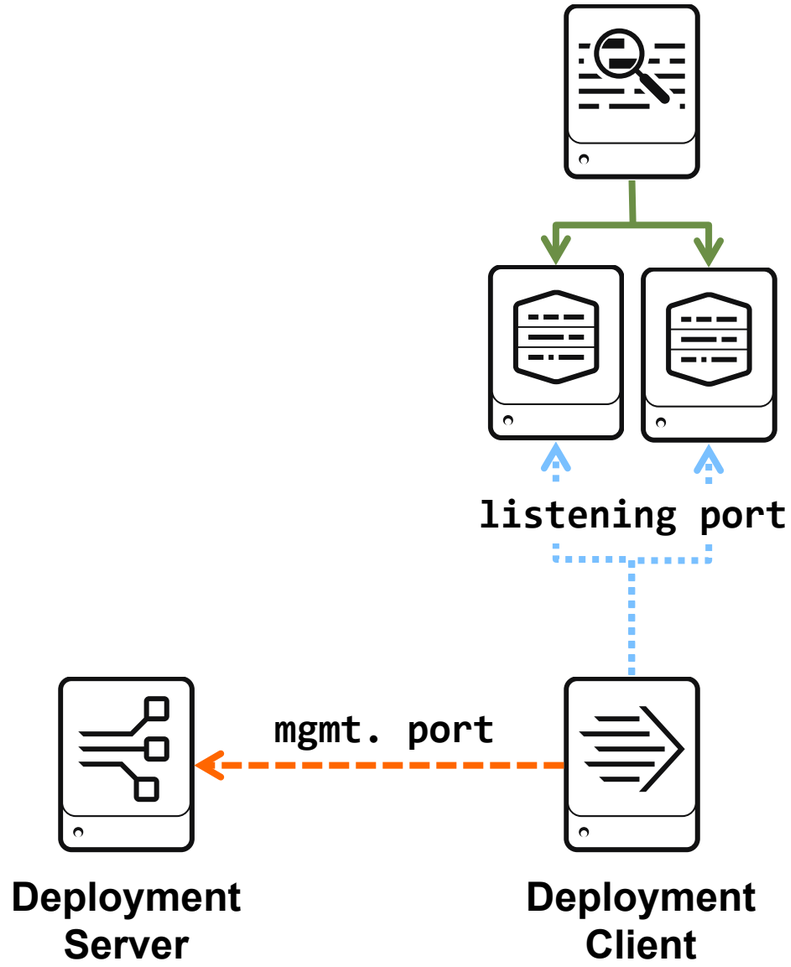
Notable Deployment Server/Client Channels

Deployment Server	Both	Deployment Client
DSManager DeploymentServer	DS_DC_Common	DC:DeploymentClient DC:HandshakeReplyHandler DC:PhonehomeThread DCManager DeployedApplication

- Show me all deployment messages on the deployment server:
  index=_internal component=DS* host=<ds> | stats count by event_message
- Show me all deployment client messages from the client:
  index=_internal component=DC* host=<uf> | stats count by event_message
- Show me all deployment app-related activities:
  index=_internal component=Deploy* host=<uf> | stats count by event_message

Review: Forwarding Problem

- Is the listening port on the receiver enabled?
- Is the forwarder sending the data to the correct address and the port?
- Is a firewall blocking?
  - Traffic from the forwarder address/port
  - Traffic to the receiver address/port
- Is the server not running as root but using a root-only port?
  - Only the root user can access ports lower than 1024 on a Linux server
  - Have Splunk listen on a port higher than 1024
  - Run Splunk as the **`root`** user



http://docs.splunk.com/Documentation/Splunk/latest/Forwarding/Receiverconnection

TCP Output Health Report

- Proactive Splunk Component Monitoring reports the current health of auto load-balanced TCP output
  - **`feature:s2s_autolb`**
    - Tracks whether this forwarder can successfully connect to all indexers configured in **outputs.conf**
    - Yellow when 20% of indexers are unreachable
    - Red when 70% of indexers are unreachable

Checking the Network Connection

- Verify that the network to the destination is working
  - Stop Splunk on the receiver
  - Use a network monitoring tool to check the receiving port

```
> tcpdump src port 9997 and tcp
> nc -z host 9997
```

    - If you see data, then connectivity is established
    - Check the parsing settings on your indexer to see where data is going
- If data is not being received, it is typically a TCP connection issue
  - Check if any firewalls between the servers are blocking the traffic
    - Include the correct addresses and ports on the Include/Exclude list

Checking the Connection Using splunkd.log

- Are the servers connected?

Indexer

```
index=_internal sourcetype=splunkd Metrics group=tcpin_connections
03-15-2021 01:13:59.135 +0000 INFO – Metrics - group=tcpin_connections,
10.0.0.50:60022:9997, connectionType=cooked, sourcePort=60022, sourceHost=10.0.0.50,
sourceIp=10.0.0.50, destPort=9997
```

Forwarder

```
index=_internal host=uf* component=TcpOutputProc
03-15-2021 23:44:59.647 +0000 INFO  TcpOutputProc - Connected to idx=10.0.0.200:9997
using ACK.
```

- Is the forwarder working?

```
index=_internal host=uf* component=Metrics group=queue name=tcp*
03-15-2021 01:28:10.416 +0000 INFO Metrics - group=queue, name=tcpin_queue,
max_size_kb=500, current_size_kb=0, current_size=0, largest_size=0, smallest_size=0

03-15-2019 01:28:37.415 +0000 INFO Metrics - group=queue, name=tcpout_default-
autolb-group, max_size=7340032, current_size=0, largest_size=6756, smallest_size=0
```


Helpful Searches

- What are the current server roles of the Splunk instances?

```
| rest /services/server/info | fields server_roles
```

- Which forwarders are sending data to Splunk and how much?

```
index=_internal sourcetype=splunkd host=<indexer> metrics
tcpin_connections | timechart span=5m max(tcp_KBps) by
sourceIp
```

- Where is the forwarder trying to send data to?

```
index=_internal host=<uf> sourcetype=splunkd destIp
```

- What output queues are set up?

```
index=_internal host=<uf> source=*metrics.log group=queue
tcpout | stats count by name
```


Example: Deployment Error Messages

- Possible error messages in **splunkd.log**
  - `07-13-2022 17:48:30.183 +0000 INFO DC:DeploymentClient [12522 PhonehomeThread] - channel=tenantService/handshake Will retry sending handshake message to DS; err=not_connected`
  - `07-12-2022 19:50:56.681 +0000 INFO DC:DeploymentClient [2204 PhonehomeThread] - channel=deploymentServer/phoneHome/default Will retry sending phonehome to DS; err=not_connected`
  - `07-12-2022 15:18:07.106 WARN  DeployedApplication - Installing app: windows to location: C:\Program Files\Splunk\etc\apps\windows`
  - `02-12-2022 15:18:07.106 ERROR DeployedApplication - There was a problem unarchiving file to: C:\Program Files\Splunk\etc\apps\windows\local\service?WSDL due to The filename, directory name, or volume label syntax is incorrect`
- There are other possibilities; DEBUG logging can highlight any errors

Lab Exercise 4 – Troubleshoot Deployment Client Issues

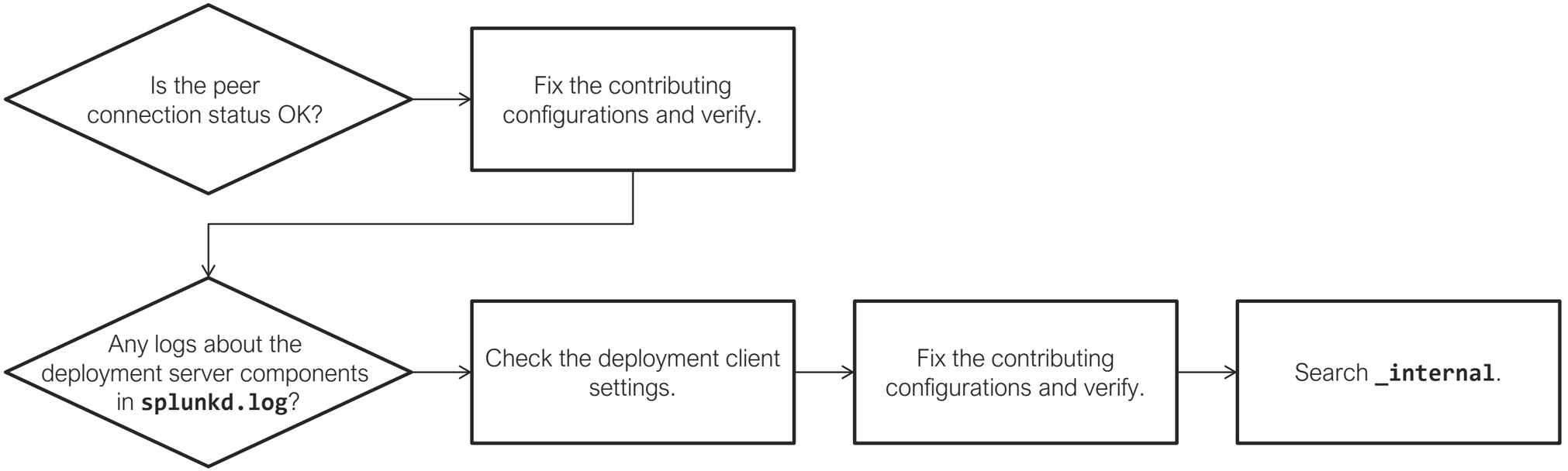
Time: 25 minutes

Tasks:

- Install the **`tse_lab04.spl`** app
- Check the internal log for any issues and resolve them
  - Distributed peer issue
  - Deployment client issue
- Piece together a complete picture of your lab environment topology
  - Who is forwarding?
  - How many universal forwarders are there?
    - Are they deployment clients?
  - Where is the deployment server?

Lab Exercise 4 – Troubleshooting Suggestions

- What is the status of search peer connections?
- What is the status of deployment client connection?
- What configuration files can affect the deployment connection?



Module 5: Upgrading, Licensing, and User Management Problems

Module Objectives

- Diagnose installation problems
- Understand and troubleshoot Splunk licensing
- Describe how to troubleshoot users and roles

Diagnosing Installation Issues

- Confirm the correct package was installed for your installation/upgrade path
  - Example: Supported OS, 64-bit vs 32-bit
- Confirm user credentials
  - Example: The Windows domain account must have permission to run as a service
- See the docs for individual OS patch and hardware requirements
- Upgrades and updates require similar considerations

Your current version	First upgrade to latest available	Then upgrade to latest available	README link	Rel. Notes link
7.0.x, 7.1.x, 7.2.x, 7.3.x	8.0.x or 8.1.x	9.0.x	8.0 README, 8.1 README	8.0 Rel. Notes, 8.1 Rel. Notes
8.0.x	8.1.x or 8.2.x	9.0.x	8.1 README, 8.2 README	8.1 Rel. Notes, 8.2 Rel. Notes
8.1.x, 8.2.x	9.0.x	N/A	9.0 README	9.0 Rel. Notes

https://docs.splunk.com/Documentation/Splunk/latest/Installation/HowtoupgradeSplunk#Upgrade_paths_to_version_9.0

Upgrading Splunk Enterprise

1. Back up **`etc`** directory
2. Stop the running Splunk instance
3. Install the new version onto the existing Splunk directory
4. Start Splunk
   - *`Perform migration and upgrade without previewing configuration changes?`*
     - To review the proposed migration changes, respond with **`n`**
     - **`SPLUNK_HOME/var/log/splunk/migration.log.`**`<timestamp>`

http://docs.splunk.com/Documentation/Splunk/latest/Installation/HowtoupgradeSplunk

Considerations for Upgrading to 9.0

- Splunk Enterprise 9.0 has security enhancements that can be enabled
  - Splunk may enable these enhancements in a later release
  - Introduces a fix for a critical deployment server vulnerability
    - Clients must be running 7.0.0 or higher
- Splunk Enterprise 9.0 is Python 3 only
  - The option to select Python 2 runtime has been removed
- Migrate your KV store storage engine from the Memory Mapped (MMAP) storage engine to the WiredTiger storage engine
  - Update MongoDB version from 3.6 to 4.2
- Splunk automatically opts you in to sharing telemetry data, one can still opt out
- Data Fabric Search (DFS) and associated server-side components are removed in 9.0

https://docs.splunk.com/Documentation/Splunk/latest/Installation/AboutupgradingREADTHISFIRST

Splunk Security 9.0 Improvements

New Security Features

Splunk Assist
Upgrade Readiness App
Native Smart Card Auth

No-action Needed Fixes

Secure by default.
Automatically implemented:
enforcement mode

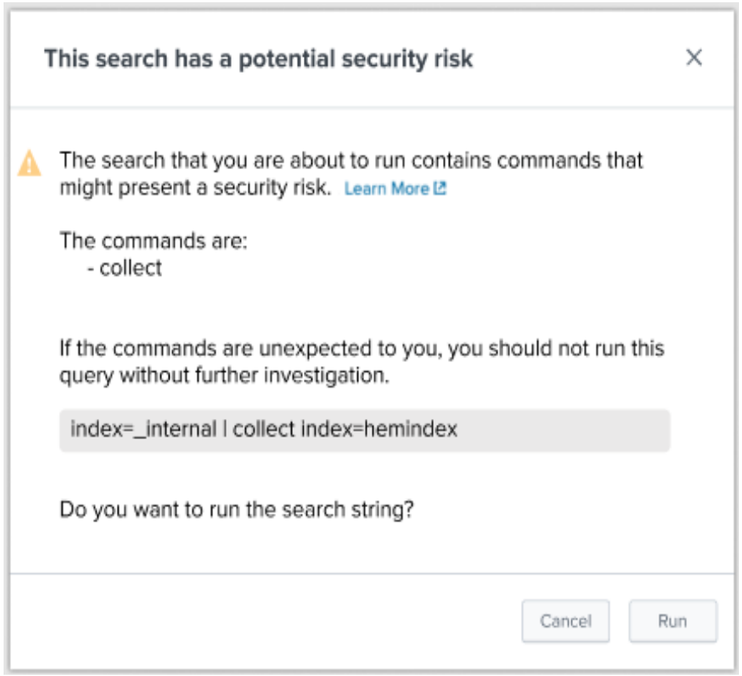
Action-required Advisories*

Require proactive steps to work. Mostly in **warning mode**

** For June 2022*

After Splunk 9.0 Upgrade – No Action Needed

- Improved dashboard security with sanitization on input fields
- Increased admin control with greater Splunk roles and capabilities with restriction options
- Updates to third party packages
  - Node.js
  - OpenSSL
- Easier risk management with user-friendly SPL safeguards
- Role-based filtering



After Splunk 9.0 Upgrade – Action Required

- Configure any action-required procedures:

  docs.splunk.com/Documentation/Splunk/latest/Security/Updates

Summary of changes

The following table lists a summary of the changes, the Splunk platforms on which the changes ship, the enforcement mode in which they currently operate, and links to procedures on how to configure Splunk software to enforce the changes.

Change	Description	Introduced in	Addresses	Current mode	Learn more
TLS certificate host name validation	Splunk platform instances verify the hostname in the TLS certificate they receive when they connect to other Splunk platform instances.	Splunk Cloud Platform (SCP) 8.2.2202, Splunk Enterprise (SE) 9.0	SVD-2022-0602, SVD-2022-0603	Warning	Learn more
Python module TLS connection hardening	Python modules on Splunk platform instances always validate TLS connections.	SCP 8.2.2202, SE 9.0	SVD-2022-0601	Warning	Learn more
TLS certificate host name validation using Splunk CLI	The Splunk Command Line Interface (CLI) validates TLS certificates for any connections it makes to other Splunk platform instances.	SCP 8.2.2203, SE 9.0	SVD-2022-0606	Warning	Learn more
Universal forwarder security	Universal forwarders always validate connections from	SE 9.0	SVD-2022-0605	Enforcement	Learn more

 | 11 November 2022

Splunk Security Upgrade Resources

- Splunk Docs: Product Security page

  www.splunk.com/en_us/product-security.html

- Splunk Docs: Security Update Documentation

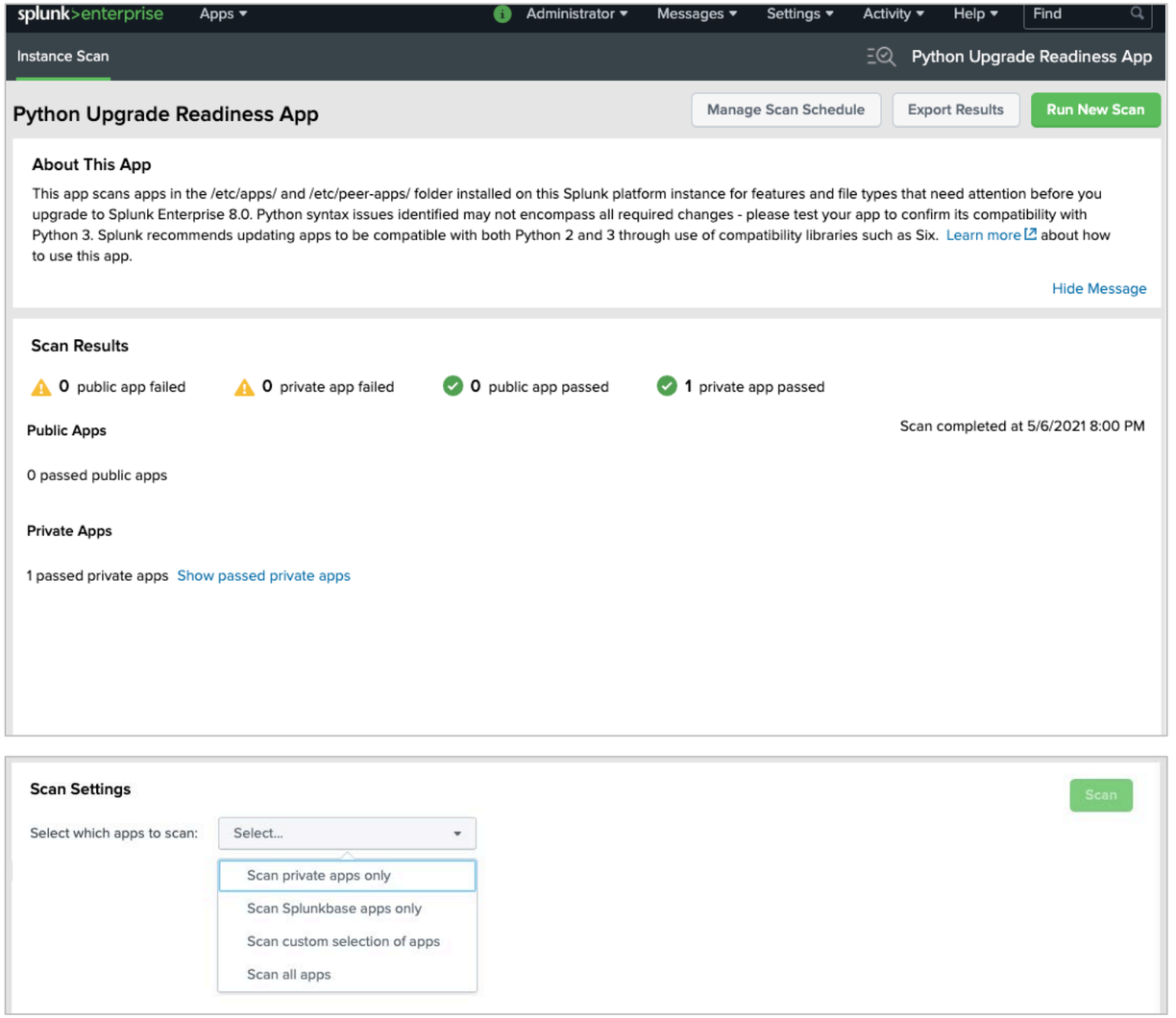
  docs.splunk.com/Documentation/Splunk/latest/Security/Updates

- Splunk Lantern: Upgrading Splunk Enterprise

  lantern.splunk.com/Splunk_Platform/Product_Tips/Enterprise/Upgrading_Splunk_Enterprise

Python Upgrade Readiness App

- Admin tool used to identify at-risk apps ahead of upgrading
  - On-prem only app
- Scans apps present in **`$SPLUNK_HOME/etc/apps`**
- Identifies file paths and line number of offending items and proposes changes
- Identifies old versions of SDK



Review: License Issues Overview

- Splunk operates with *no-enforcement license* (since 6.5) with a license volume of 100GB/day
  - http://docs.splunk.com/Documentation/Splunk/latest/Admin/TypesofSplunklicenses
  - Splunk licensing allows occasional data “bursts”
  - Exceeding the daily license quota raises a warning
- 5 warnings in a rolling 30-day period is a violation
  - The license manager still tracks violation
  - Indexing and searching continues
- Starting with 8.1, search is disabled for licenses with less than 100GB/day with 45 warning in a rolling 60-day period
  - Free licenses that generate 3 or more warnings in a rolling 30-day period will continue to index and search is disabled

Clarifying a License Problem

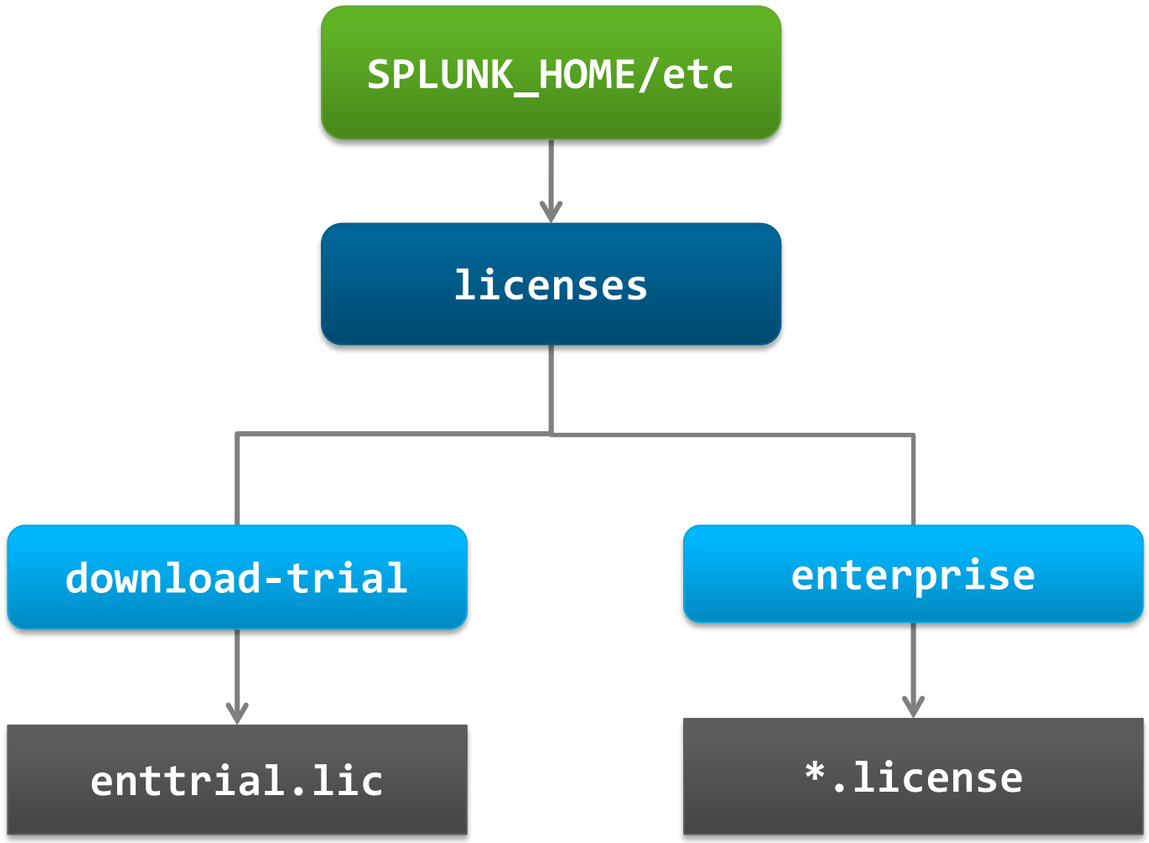
- Are you unable to search?
  - Has your license expired?
- Do you need to reset a locked license?
  - Escalate to Splunk Support
- Got a warning?
  - What's taking up most of the quota?
- Are you trying to do something that is not allowed with the installed license?
  - Contact your sales account manager

Confirming with splunkd.log

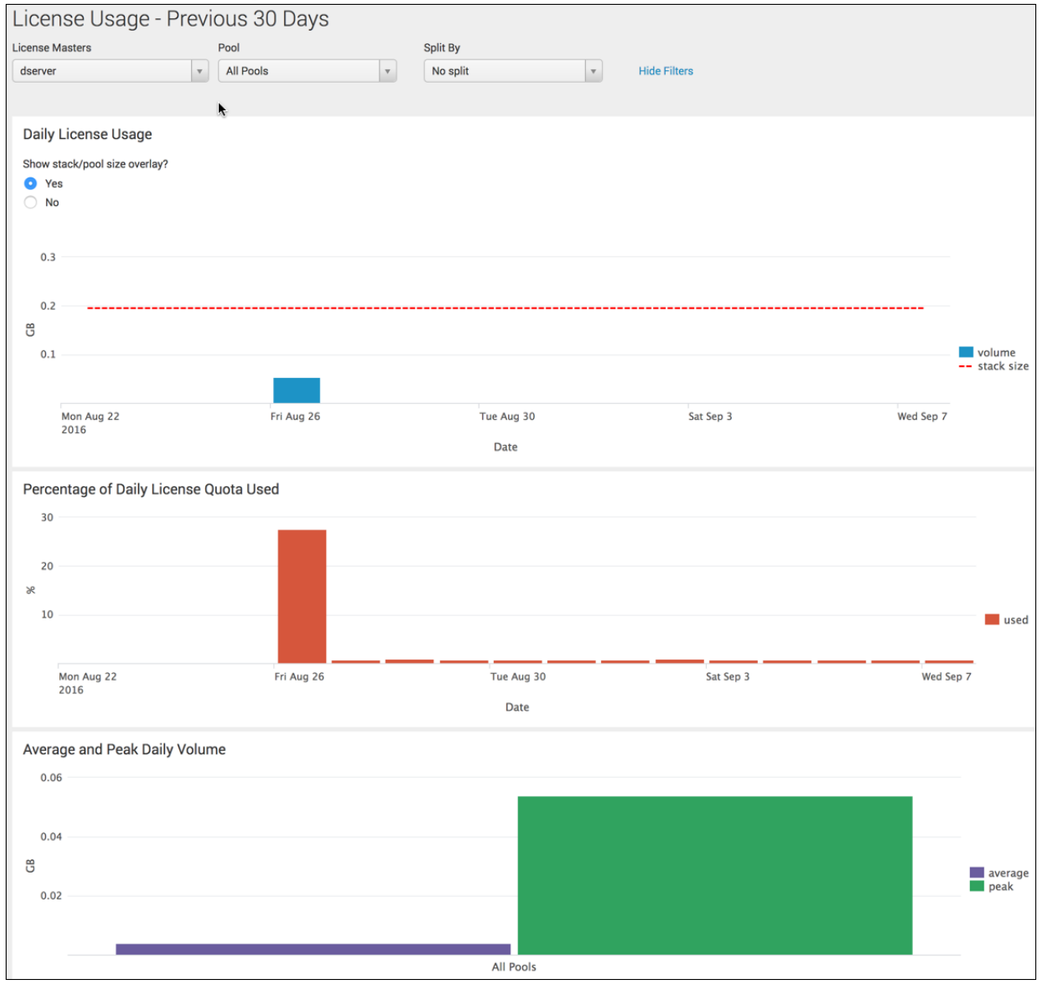
index=_internal sourcetype=splunkd component=LM*

```
03-09-2019 00:00:00.352 -0700 WARN  LMStackMgr - Indexing quota exceeded for this pool,
quota=524288000 bytes
03-15-2019 07:00:54.561 +0000 WARN  LMTracker - license expired, revoking all session keys
03-20-2019 00:00:00.967 +0000 INFO  LMStackMgr - should rollover=true because
_lastRolloverTime=1552953600 lastRolloverDay=1552953600 snappedNow=1553040000
03-20-2019 00:00:00.969 +0000 WARN  LMStackMgr - This pool has exceeded its configured
poolsize=209715200 bytes. A warning has been recorded for all members
03-20-2019 00:00:00.969 +0000 INFO  LMStackMgr - finished rollover, new
lastRolloverTime=1553040000
03-20-2019 15:08:55.720 +0000 ERROR LMAdminHandlerLicenses - splunk.license.big.license: failed
to add because: stack already has this license, cannot add again
03-20-2019 15:53:39.495 +0000 ERROR LMAdminHandlerPools - failed to create pool: 'A catch all
pool already exists(auto_generated_pool_enterprise). Cannot create two catch all pools'
03-20-2019 15:53:49.165 +0000 ERROR LMAdminHandlerPools - failed to create pool: 'Stack is fully
allocated. No more quota to create the new pool newpool'
03-22-2019 00:00:04.370 +0000 INFO  LMSlaveInfo - Detected that masterTimeFromSlave(Thur Mar 21
23:59:03 2019) < lastRolloverTime(Fri Mar 22 00:00:00 2019), meaning that the master has already
rolled over. Ignore slave persisted usage.
```

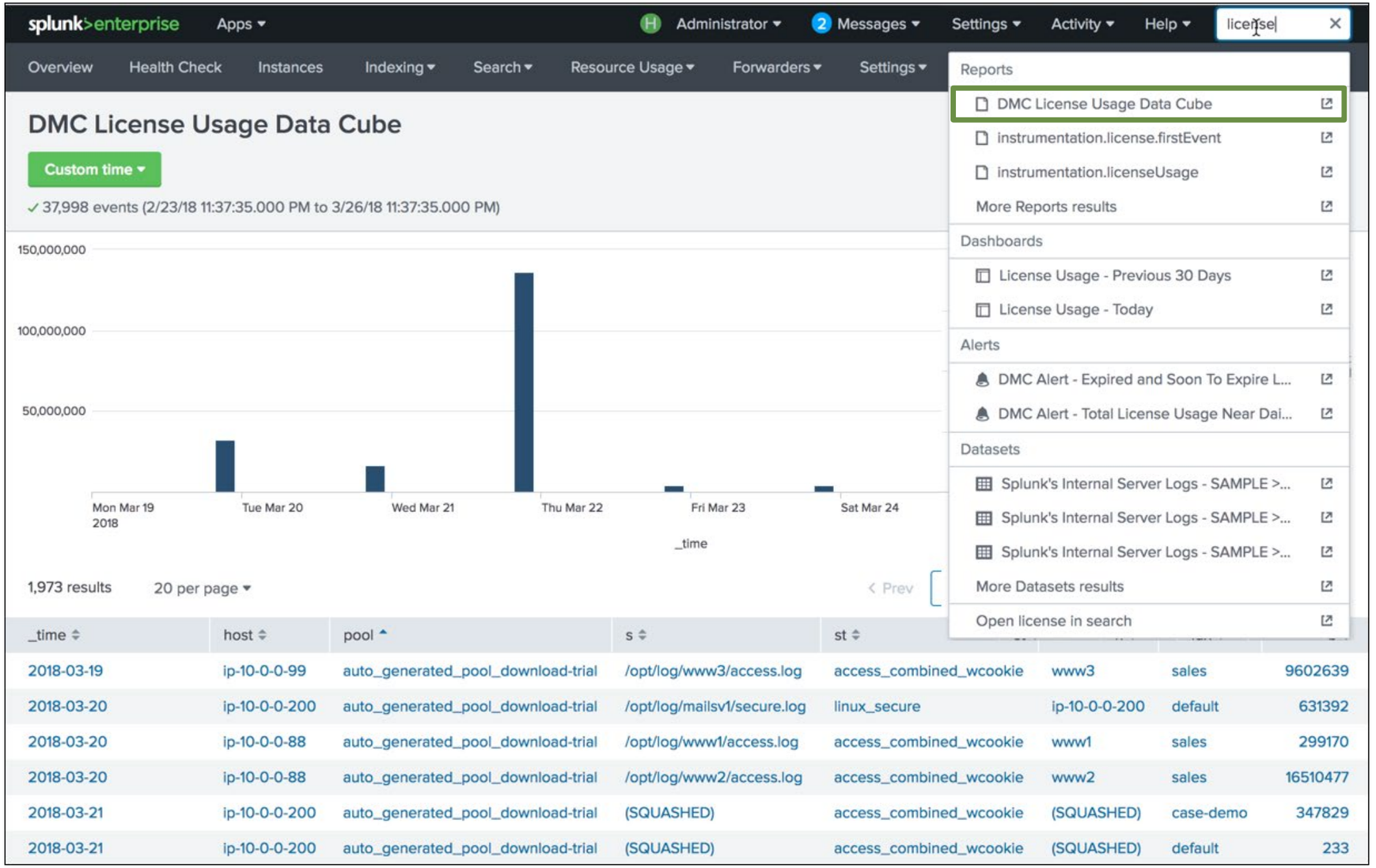

License File Locations and Usage Check



MC > Indexing > License Usage

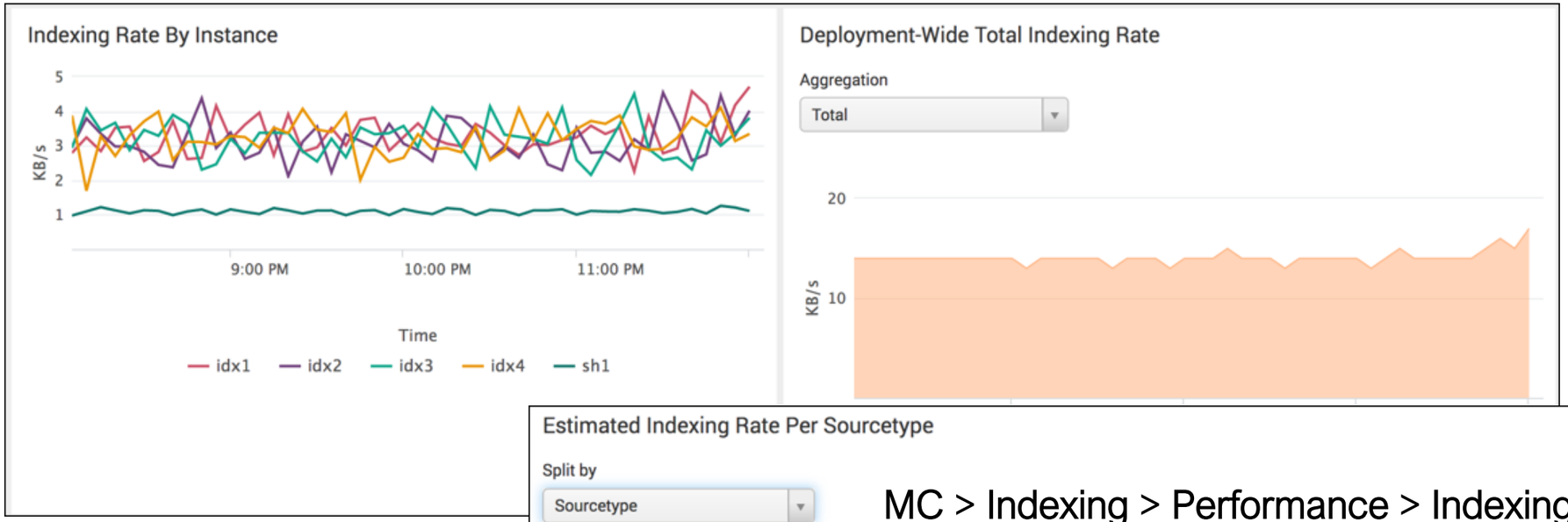


Where is all that Data Coming From?



Where is all that Data Coming From? (cont.)

MC > Indexing > Performance > Indexing Performance: Advanced



Notable Logs for License Problem

- **`license_usage.log`**
  - Contains indexed volume in bytes per pool, index, host, sourcetype, and source
  - Only available on the license master
  - Generates once a day (generally just after midnight spanning - 24h)
    - Or, when splunkd is restarted
- **`metrics.log`**
  - Contains a periodic report taken every 30 seconds
  - Reports top 10 results per each type
  - Useful for approximation only

License Problem – Log Channels of Interest

- Notable license_usage.log fields

Field	Description
s	Source of the license usage
st	Source type of the license usage
h	Host of the license usage
idx	Index of the license usage
type	Message type

- Log channels of interest
  - **LicenseUsage**
  - **LicenseUsageSummary**
  - **LicenseMgr**
  - **Metrics**

```
index=_internal component=LicenseUsage*
| top type
```

- Metrics license-related fields

Field	Description
group	What type of data is being reported on
kb	The number of kilobytes indexed per thirty seconds
series	The specific data being reported on

- Metrics group
  - **thruput**
  - **per_index_thruput**
  - **per_host_thruput**
  - **per_source_thruput**
  - **per_sourcetype_thruput**

```
index=_internal component=Metrics per_index_thruput
| eval mb=(kb/1024) | timechart span=1h sum(mb) by
series | addtotals
```

Metrics Group Values and Examples

```
03-09-2019 18:43:21.416 +0000 INFO  Metrics - group=per_host_thruput,
series="www3", kbps=0.014144, eps=0.354835, kb=0.438477, ev=11, avg_age=0.000000,
max_age=0

03-09-2019 18:43:21.416 +0000 INFO  Metrics - group=per_index_thruput,
series="web", kbps=0.021547, eps=0.193546, kb=0.667969, ev=6, avg_age=0.000000,
max_age=0

03-09-2019 18:43:21.417 +0000 INFO  Metrics - group=per_source_thruput,
series="/opt/log/www3/secure.log", kbps=0.014144, eps=0.354835, kb=0.438477,
ev=11, avg_age=0.000000, max_age=0

03-09-2019 18:43:21.417 +0000 INFO  Metrics - group=per_sourcetype_thruput,
series="linux_secure", kbps=0.007088, eps=0.225808, kb=0.219727, ev=7,
avg_age=0.000000, max_age=0

03-09-2019 18:43:21.417 +0000 INFO  Metrics - group=thruput, name=cooked_output,
instantaneous_kbps=0.303109, instantaneous_eps=0.967732, average_kbps=0.251002,
total_k_processed=304294.000000, kb=9.396484, ev=30.000000

03-09-2019 18:43:21.417 +0000 INFO  Metrics - group=thruput, name=thruput,
instantaneous_kbps=0.303109, instantaneous_eps=0.935475, average_kbps=0.251022,
total_k_processed=304319.000000, kb=9.396484, ev=29.000000, load_average=0.030000
```

Note

Thruput metrics measure the indexing pipeline. No data in these events means your data is not reaching the pipeline for some reason.

Troubleshooting Users and Roles

- Users have no meaning in Splunk outside of their role(s)
  - Without a role assignment, a user cannot log in
- Roles created with Splunk Web are saved in **`authorize.conf`**
- Roles define the capabilities
  - Which parts of the system can a user access?
  - What is a user ALLOWED to do in the system?
- For example, can a user search? If yes:
  - What is a user allowed to search?
    - Restrictions on search terms, time range, and allowed indexes
  - Has a user reached his/her allowed quota?
    - Restrictions on concurrent job limits and storage usage

http://docs.splunk.com/Documentation/Splunk/latest/Security/Rolesandcapabilities

Troubleshooting Users and Roles (cont.)

- Users with multiple roles inherit broadest permissions
  - Exception: the user assigned to a role without search restrictions and another with search restrictions inherits the search restrictions
- Validate the user capabilities
  - CLI: **`splunk list user`** displays the roles assigned to a given user
  - Search:

```
| rest /services/authentication/users | table title capabilities
```

- Check the user's runtime quota restrictions
  - Search:

```
| rest /services/admin/quota-usage | search title=<user> |
fields srchDiskQuota, diskUsage, srchJobsQuota,
historicalSearchCount, rtSrchJobsQuota, realTimeSearchCount,
srchMaxTime
```

Working with LDAP Problems

- LDAP is a series of key value pairs organized in a tree
  - The configuration tells Splunk what to look for and where
- Notable splunkd channels:
  - **`UiAuth, UserManagerPro, AdminHandler:AuthenticationHandler`**
  - **`AuthenticationManagerLDAP`**
  - **`ScopedLDAPConnection`**
- All the information that you need to make the configuration work is contained in the user and group LDIF entry
  - UNIX: **`ldapsearch -x -h <host> -p <port> -b "<(user|group)BaseDN>" -D "<bindDN>" -w <realNameAttribute>`**
  - Windows: **`ldifde -f output.ldif`**
  - Or, Active Directory Explorer app from Microsoft: https://technet.microsoft.com/en-us/adexplorer

Things to Consider with LDAP

- DO NOT grab everything in a LDAP tree
  - Choose the point in the tree that is narrow enough in scope to return the users that you want in Splunk
  - The fewer records Splunk fetches, the more efficient it is
- Use the base filters to get additional fidelity with the configuration
- NOTES:
  - Entries in **`authentication.conf`** are case sensitive
  - Supports multi-domain, but does not support anonymous LDAP login
  - The DN value of **`userBaseDN`** cannot be set to the same value as **`groupBaseDN`**
    - Workaround is to remove one level from the **`groupBaseDN`** (or vice versa)

Optional LDAP Default Role

- Without a role assignment, a user cannot log into Splunk
- If you want to give access to Splunk without needing an explicit group-to-role mapping, a default role can be specified
  - Valid only if LDAP authentication is a successful request
    - The LDAP server does not return any groups or groups cannot be mapped to Splunk roles
  - Not valid for failed request or connectivity issues

authentication.conf

```
[authentication]
authSettings = AD_splunkers
authType = LDAP
defaultRoleIfMissing = sales

[roleMap_AD_splunkers]
admin = splunkAdmins
...
```

Working with SAML Problems

- Requires the **`change_authentication`** capability on the search head
- Splunk must be enabled to query the identity provider (IdP)
  - IdP needs SPMetadata.xml (Service Provider Metadata)
    - Contains Splunk endpoint URI, required attributes and format, and Splunk cert
  - Splunk needs IdPMetadata.xml from the provider
    - Contains IdP endpoint URI, entity ID, Name ID format, and IdP cert
- SAML passes tokens (not credentials)
  - IdP SSO must respond (SAML Assertion) with the **`realName`**, **`role`**, and **`mail`** attributes
- Notable Splunk log channels:
  - **`SAMLConfig, UiSAML, UserManagerPro`**
  - **`AuthenticationManagerSAML`**
  - **`Saml, SamlIdpCertDatabase`**

Admin Tip

For troubleshooting purposes, create a non-SAML admin user to bypass SAML SSO:
https://<SplunkWeb>/en-US/account/login?loginType=Splunk

http://docs.splunk.com/Documentation/Splunk/latest/Security/TroubleshootSAMLSSO

Managing Deactivated Users

- When a Splunk account is removed or deactivated from an external directory, Splunk does not automatically remove the corresponding user's knowledge objects (KOs)
- Do not allow KOs to operate on behalf of owners who are no longer in the system
- Orphaned KOs can cause problems in certain conditions
  - Splunk scheduler cannot run a scheduled report on behalf of a non-existent owner
  - May generate search errors if the KOs have global permissions
- As a result, dashboards would look to be broken, summary indexes may have gaps, etc.

Ways To Detect Orphaned KOs

- Review orphaned scheduled search notifications in **Messages**
- Use the **Orphaned scheduled searches** report and the **Orphaned Scheduled Searches, Reports, and Alerts** dashboard
- Run the Monitoring Console health check
- **NOTE:** These options have no way of knowing when an account has deactivated from an external directory
- To detect deactivated external user, check **splunkd.log** on the search head for any authentication errors

Removing Orphaned User Objects

- To remove the user directory from Splunk:
  1. Back up the **SPLUNK_HOME/etc/users/<user>** folder
  2. Search any reports, dashboards, and other KOs with global permissions owned by the user under **SPLUNK_HOME/etc/apps**
     - See ways to detect orphaned KOs mentioned in the previous slide
  3. Redirect all global-context objects to a new owner
     - Go to **Settings** > **All configurations** and click **Reassign Knowledge Objects**
     - Private objects cannot be reassigned through the UI
  4. Remove the **SPLUNK_HOME/etc/users/<user>** folder
- To reassign private objects:
  1. Temporarily recreate the Splunk account with the exact name used
  2. Log in as the user and reassign or delete the objects
  3. Delete the temporal account and the directory

Warning

Changing KO ownership may inadvertently grant access to previously inaccessible data or vice versa. Always review the permissions before reassigning.

 |

Lab Exercise 5 – Troubleshoot LDAP Issues

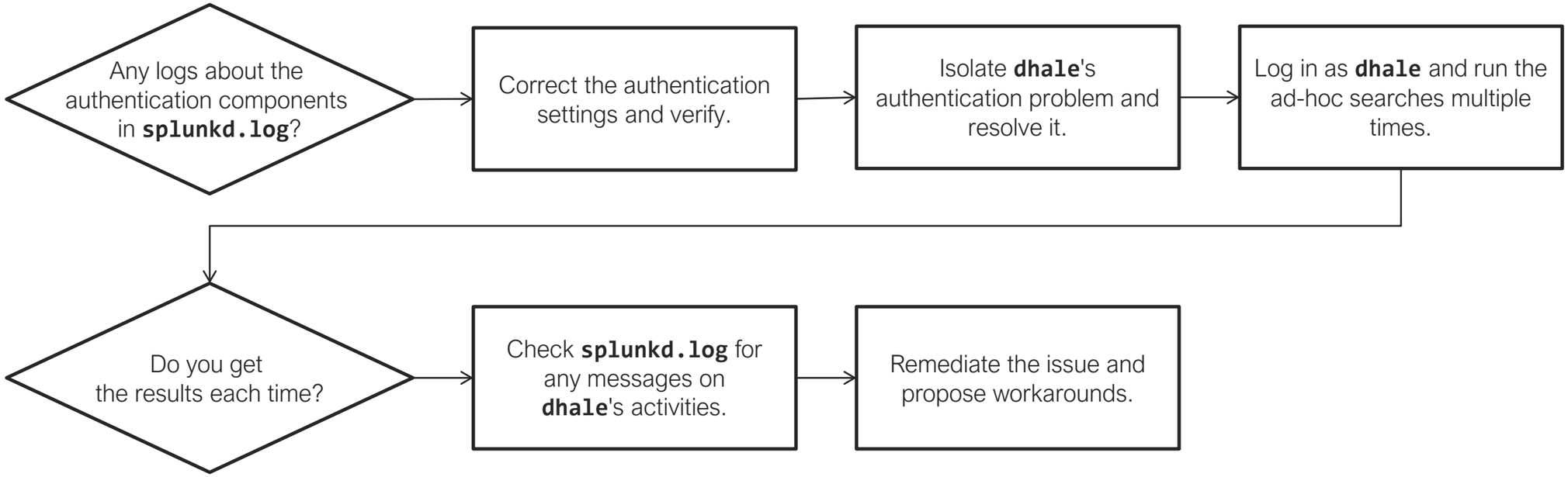
Time: 25 minutes

LIVE Tasks:

- Install the **`tse_lab05`** app
- Confirm the user **`dhale`**'s authentication issue
- Restore the Splunk LDAP authentication service
- Troubleshoot the user **`dhale`**'s search problem

Lab Exercise 5 – Troubleshooting Suggestions

- Are there any warnings or errors in the log about the authentication process?
- Are there any warnings or errors in the log about **dhale**'s Splunk activities?
- Are there any warnings or errors in the log regarding **dhale**'s search jobs?



Module 6: Search Management Problems

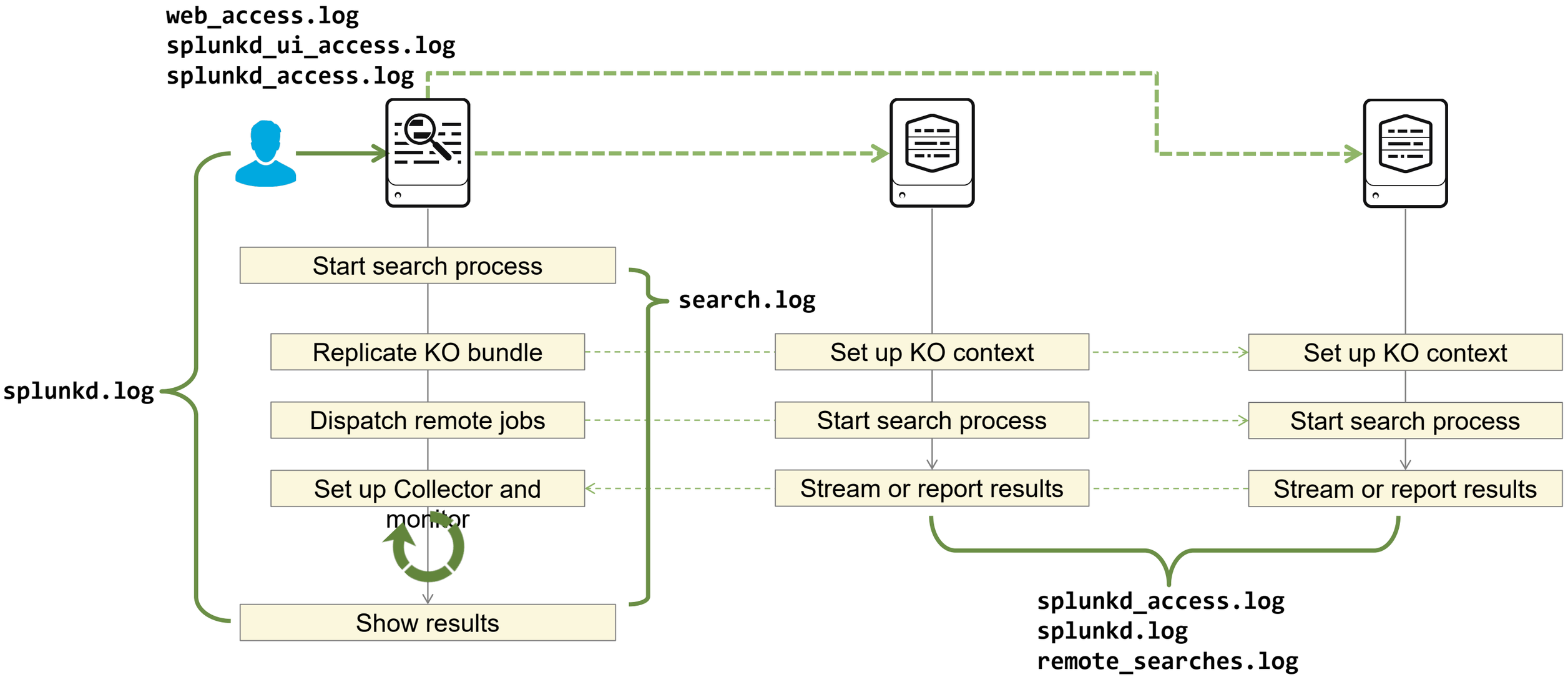
Module Objectives

- Troubleshoot distributed search problems
- Identify job scheduling problems
- Troubleshoot skipped search problems
- Learn to diagnose crashing problems
- Describe how to prioritize resources for critical Splunk processes

What is a Splunk Job?

- A job is the individual search process kicked off by a search, alert, etc.
- Manifests as Splunk processes in the OS
  - Has its own child processes on the search head and on every peer that participates in the search
  - Used by the **`splunkd`** process to do further work as needed
- In Linux, each search job starts two processes
  - **`search-launcher`** and **`process-runner`**
  - Can isolate all the *splunk-search* processes with: **`ps -ef | grep search`**
  - The main job is the one using system resources
    - Contains the **`search --id`** in its name
- In Windows, each job spins up a new **`splunkd service`**

Distributed Search Job Overview



Other Notable Logs in Distributed Searches

- **splunkd_access.log** on search peers
  - Shows splunkd requests via the UI and the REST calls
  - Includes the time taken to respond to the requests and the artifact size
- **remote_searches.log** on search peers
  - Contains messages regarding searches executed on the search peers in response to a search head request
- **scheduler.log** on search head
  - Shows scheduled search activity performed by the splunkd and alert scheduler

Knowledge Bundle Replication

- Distributed searches utilize a replicated knowledge bundle on the peers
  - SH periodically replicates its knowledge objects (KO) – fields, tags, lookups, and other entities used in a search – to its search peers
  - A bundle contains nearly everything under **`etc/system`**, **`etc/apps`**, **`etc/users}`** by default
  - By default, a bundle larger than 2GB is not replicated to the search peers
- SH generates and maintains the source bundle in **`SPLUNK_HOME/var/run`**
  - Generates a bundle in the form of **`<serverName>-<time>.bundle`**
  - Or, a delta bundle in the form of **`<serverName>-<oldTime>-<newTime>.delta`**
    - A delta bundle represents a difference between the prior and current bundles
- Splunk tries to replicate **`.delta`** first
  - If **`.delta`** replication fails, do the full **`.bundle`** replication
- The search peers ignore their local KOs and use only this bundle in their part of the search on SH's behalf

Knowledge Bundle Management

- On each search peer, replicated bundles are stored in **`SPLUNK_HOME/var/run/searchpeers`**
  - Keeps *N* number of bundle generation per search head (5 by default)
  - Currently-running searches use the bundle from the time they were started
  - The no-longer-needed generation gets deleted periodically
- Check the high-level replication status in:

  Settings > **Distributed search** > **Search peers**
- Notable Splunk log channels on SH:
  - **`Archiver`**
  - **`ArchiverFilters`**
  - **`BundleDeltaHandler`**
  - **`BundleArchiver`**
  - **`DistributedBundleReplicationManager`**

Bundle Management Options

- Even minimal KO changes can result in a new bundle replication
  - Frequently updated files can trigger retransmission
  - Asynchronous and in some cases partial (**`.delta`**)
- Limit the size of the knowledge bundle
- Migrate largest csv lookups to KV Store

Notable Logs on Search Peers

- splunkd_access.log

  index=_internal sourcetype=splunkd_access host=<peer> uri_path="*bundle*/<sh>"

```
..."GET /services/admin/bundles/sh?count=-1 HTTP/1.1" 200 9671 - - - 1ms
..."POST /services/receivers/bundle/sh HTTP/1.0" 200 19 - - - 700ms
..."GET /services/receivers/bundle/sh HTTP/1.0" 200 212 - - - 0ms

..."GET /services/admin/bundles/sh?count=-1 HTTP/1.1" 200 3339 - - - 0ms
..."POST /services/receivers/bundle-delta/sh HTTP/1.0" 200 2027 - - - 11ms
```

- splunkd.log channels
  - **DistributedBundleReplicationManager**
    - Change the log level to **INFO** for debugging purposes (**WARN** by default)
  - **DistBundleRestHandler**
  - **BundleDeltaHandler**
  - **NewDistBundleHandler**
  - **SearchPeerBundlesSetup**

Monitoring Bundle Replication Activity

- Monitoring Console in distributed mode
  - **Search > Distributed Search > Distributed Search: Deployment**
- Knowledge Bundle Activity panel shows the replication of both full bundle and delta bundles
- Check both upload and download activities

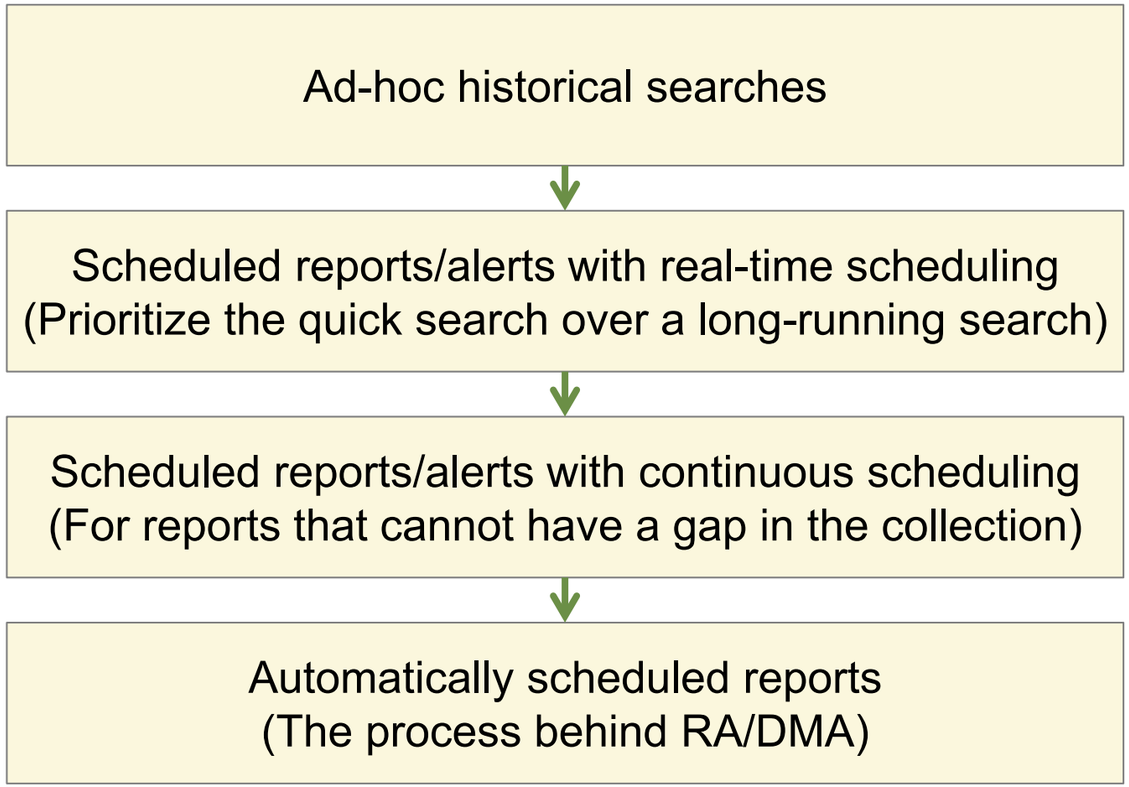
Search Scheduler

- Scheduler manages if, when, and how searches/alerts can be triggered
  - Tracks priority based on search settings and prior-run scoring rules
- When the scheduler is oversubscribed, each job's scheduling mode determines if it is ***skipped*** or ***deferred***
- There are two types of scheduling modes
  - **Real-time scheduling** can skip a job (default)
  - **Continuous scheduling** defers a job
- The scheduler prioritizes reports with real-time scheduling over reports with continuous scheduling
- **`realtime_schedule`** in **`savedsearches.conf`** indicates the scheduling mode of a search
- If a search is skipped or deferred, check the reported reason in **`scheduler.log`**

Splunk Scheduling Modes and Priority

- Real-time scheduling
  - **`realtime_schedule = 1`**
  - Not same as the real-time search
  - Use when you want the latest reports to return up-to-date data
  - A job is skipped if more reports are set to run at the same time than the available capacity
- Continuous scheduling
  - **`realtime_schedule = 0`**
  - Use when you value data availability over recency
  - A job runs eventually when there is available capacity

- Scheduling Priority

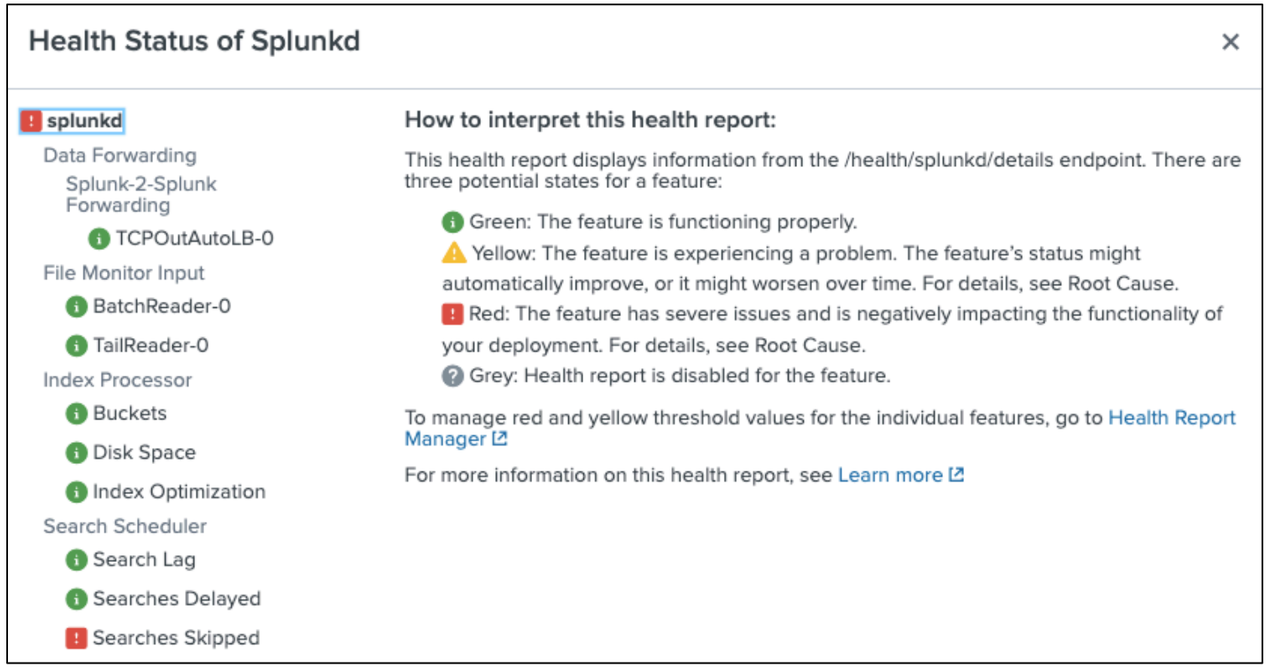


Note

Reports that have been skipped get improved priority scores until they execute.

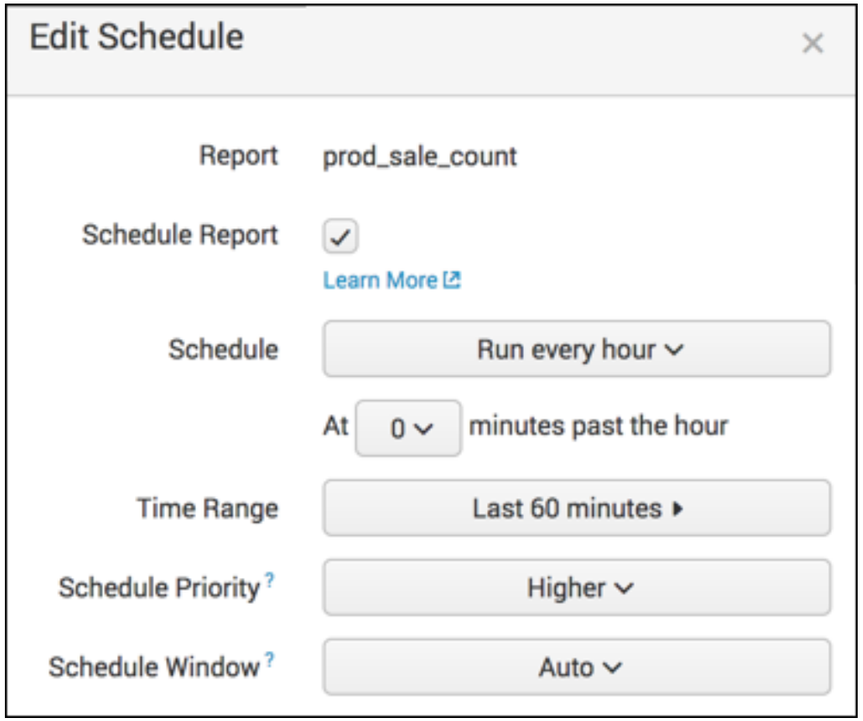
Scheduler Oversubscription – Symptoms

- Health Check indicator provides a single pane of glass of current health status
- Blank dashboard panels
- Alerts are not triggered on time or at all
- Summary indexes have gaps
- DMAs are not 100% up to date
- High number of skipped or deferred searches
  - Check **Search** > **Scheduler Activity: Instance** in Monitoring Console



Scheduler Oversubscription – Workarounds

- Convert inline searches to scheduled searches
- Utilize base searches in dashboards
- Use Splunk guardrails to prevent users from overloading SHs
  http://docs.splunk.com/Documentation/Splunk/latest/Report/PrioritizescheduledreportsinSplunkWeb
  - **Schedule Priority**
    - Require the **`edit_search_scheduler_priority`** capability
  - **Schedule Window**
  - **`allow_skew`** in **`savedsearches.conf`**
  - User/role-based quotas in **`authorize.conf`**
  - Workload management
- Check **Activity** > **Jobs** and cancel any rogue jobs
- Monitor MC for search patterns and adjust the schedules
- Add more physical CPU cores
- Use Search Head Clustering

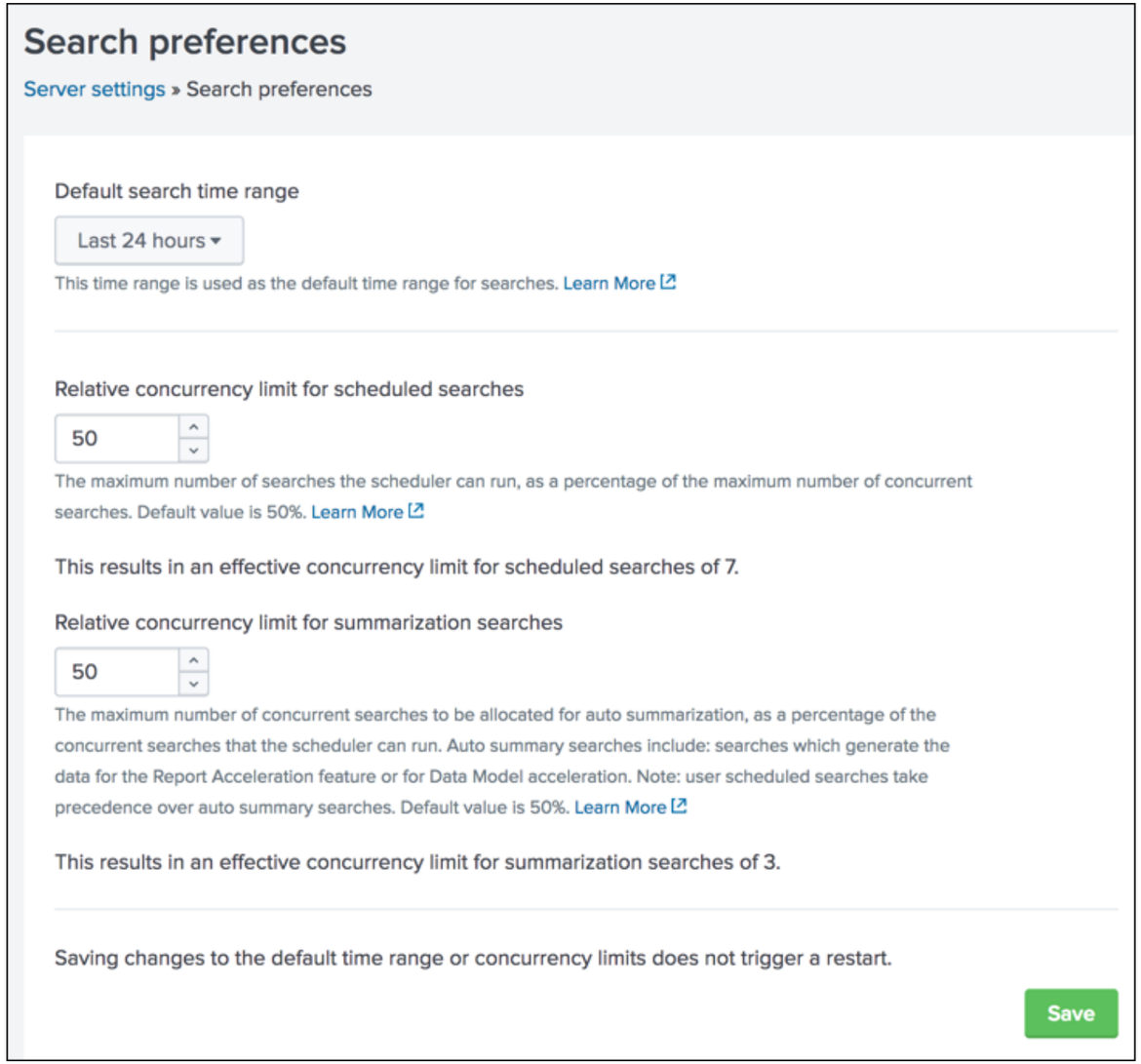


What Is Using Cores?

	Summary Index	Report Acceleration	Data Model Acceleration
Notes	• Extract the precise results and save to a designated index • Can keep longer than the index retention	• Speed up qualified transforming reports and dashboards • Not useful for reports that generate very high cardinality (use DMA)	• Accelerate many fields as dataset from different searches
Requirements	• Access to indexes	• **`accelerate_search`** and **`schedule_search`** capabilities • Have write permissions for the report	• **`accelerate_datamodel`** capability
Configuration	• Scheduled reports or SPL	• UI	• UI
Schedule	• User-defined interval • 1 historical scheduler slot per schedule	• After the initial build, runs scheduled searches on a 10 minute interval • Each acceleration uses 1 summary scheduler slot	• After the initial build, runs scheduled searches on a 5 minute interval • Each acceleration uses 3 summary scheduler slots
Scheduler Mode	• Continuous scheduling	• Real-time scheduling	• Real-time scheduling
Status in MC	• complete / deferred	• complete / skipped	• complete / skipped

Adjusting Scheduler Guardrails

- A large number of concurrent searches can overload network resources
  - This is not a Splunk issue
  - Use the **`allow_skew`** setting in **`savedsearches.conf`**
  - A valid timescale is **`0`**, **`%`**, **`m`**, **`h`**, or **`d`**
- After evaluating the search patterns in MC, if you need to adjust the concurrency allocation, you can adjust them via:
  - **`[scheduler]`** settings in **`limits.conf`**
  - Settings > Server settings > Search preferences



Scheduled Search Best Practice

- Verify when events occurred and actually indexed

```
<base_search> | head | convert ctime(_time) as evtTime
ctime(_indextime) as idxTime | eval lag=_indextime-_time |
table evtTime idxTime lag
```

- Offset search range to compensate for the lag in indexing

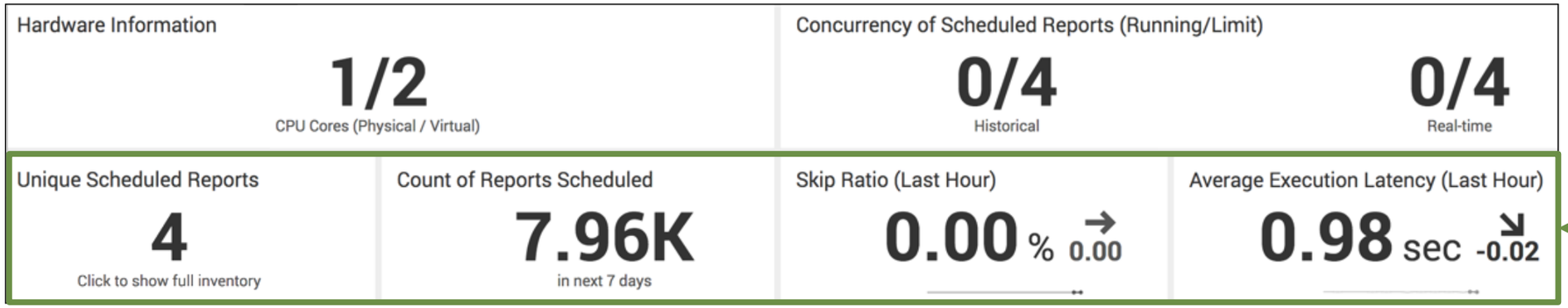
  Example for a less than a minute lag:

  - Before: `<base_search> earliest=-1m@m latest=now`
  - After: `<base_search> earliest=-2m@m latest=-1m@m`
    - Can accommodate indexing lag and searches over a same span

- Classify and reserve resources for critical services with Workload Management

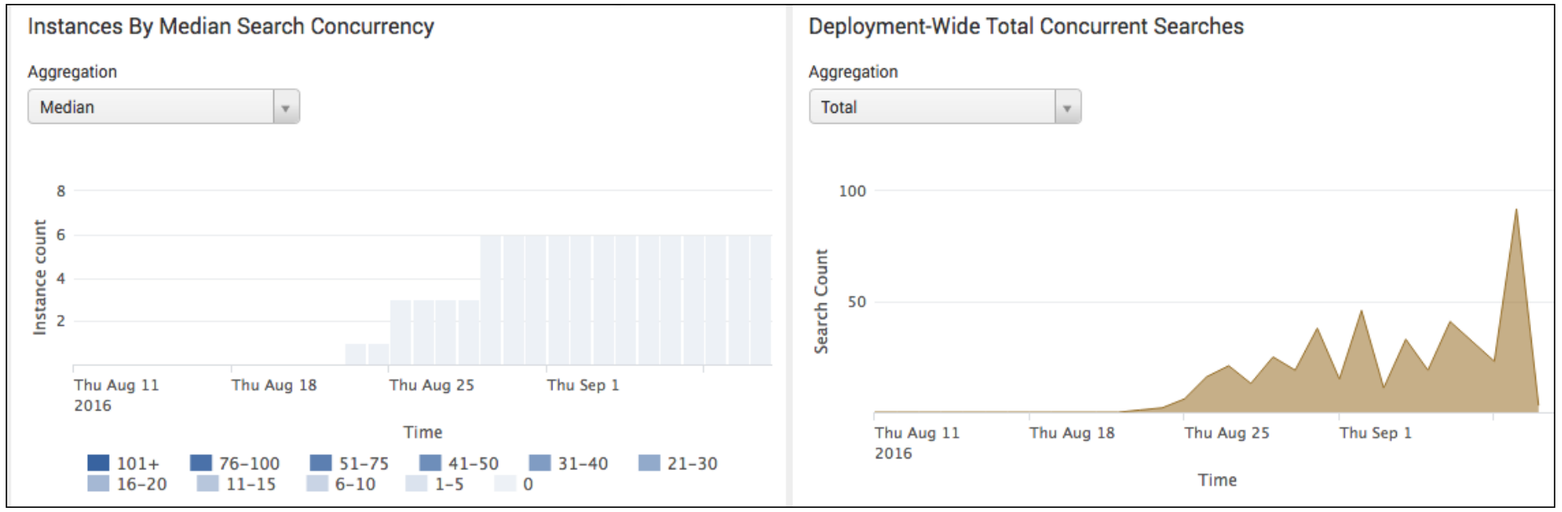
Troubleshooting Skipped Searches

MC > Search > Scheduler Activity: Instance



Scheduler load forecasting

MC > Search > Search Activity: Deployment



Notable Logs

- scheduler.log

```
index=_internal
sourcetype=scheduler
status=skipped | stats
count values(alert_actions)
by savedsearch_name, reason
```

- audit.log

```
index=_audit action=search
| stats count by info
```

Troubleshooting a Crash

- Splunk can crash if there is not enough memory
  - On Linux, number of open file descriptors and max user process limit can also be contributing factors
    - http://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/ulimitErrors
    - https://www.splunk.com/en_us/blog/tips-and-tricks/whats-your-ulimit.html
- Most frequent cause – the search ran out of memory
  - Too many events from **`join`**, **`append`**, **`transaction`**, or SPLs with **`BY`** clause
    - Check the search to see if it can be refined
    - Usually the parent process should still be up
    - Make use of summarization methods
- Not enough disk space (too many job artifacts)
  - Indexing may continue but new searches may not be serviced

Note

Crash problems are VERY difficult to track down and you often need to work with Splunk Support.

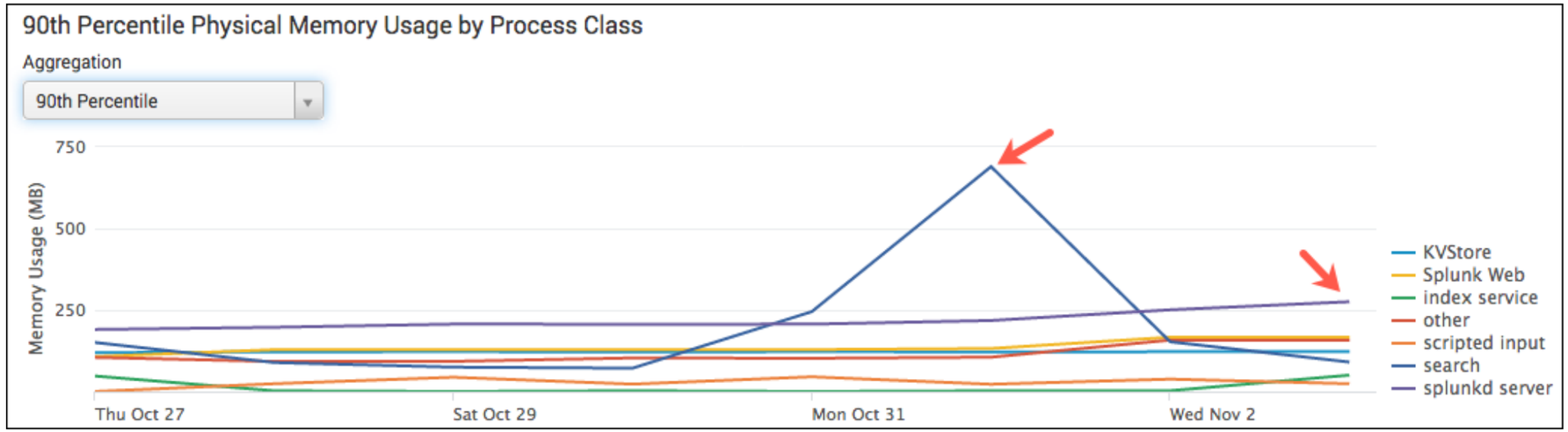
Not Enough Memory

- Find System Memory section in the diag:

  `index=mydiag source="*systeminfo.txt" "System Memory"`

  – Is the memory full? If so, you need to free up more RAM

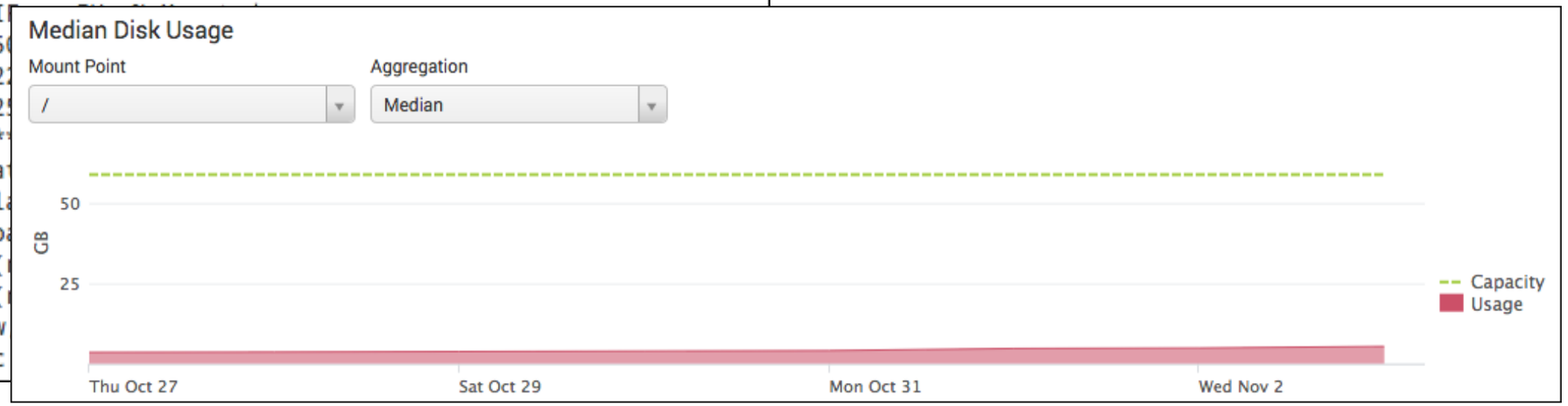
- Monitoring Console > Resource Usage
  – Per instance or machine
  – The historical data comes from **`_introspection`** index



Not Enough Disk Space

- Monitoring Console > Resource Usage
- Find DF output section in the diag:
  - **`index=mydiag source="*systeminfo.txt" "DF output"`**
- Is the Splunk partition full?
- Indexes, search artifacts, old search bundles, and rogue lookup files contribute to disk usage

```
********** DF output **********
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/xvda1      61794280 7186836  54507196  12% /
devtmpfs         7693400      60   7693340   1% /dev
tmpfs            7701972       0   7701972   0% /dev/shm
also df -i
Filesystem      Inodes IUsed   I
/dev/xvda1     3932160 81703 385
devtmpfs       1923350   477 192
tmpfs          1925493     1 192
********** mount output *********
proc on /proc type proc (rw,rela
sysfs on /sys type sysfs (rw,rel
/dev/xvda1 on / type ext4 (rw,no
devtmpfs on /dev type devtmpfs (
devpts on /dev/pts type devpts (
tmpfs on /dev/shm type tmpfs (rw
none on /proc/sys/fs/binfmt_misc
```



Notable Logs for Crashes

- Check if **`crash*log`** exists
  - For example: **`crash-2019-04-27-11:54:26.log`**
    - The timestamp occurs when the crash is reported
  - **`splunkd_stderr.log`**
  - For *nix, this contains times of healthy start and stop events, exceptions, assertions, and other errors generated by libraries and the OS
- On Windows OS, the dump files are created (**`*.dmp`**)
  - Contains what was in memory at the time of crash
- Check other logs to determine what Splunk was doing at the time
  - **`audit.log`**
  - **`splunkd.log`**
  - **`metrics.log`**
  - **`web*.log`**

Helpful Searches

- Show me all Splunk restarts based on loader?

```
index=_internal sourcetype=splunkd loader event_message=*xml
```

  - Check `splunkd.xml` and `composite.xml` for enabled/disabled processors

- When did Splunk last crash?

```
index=_internal sourcetype=splunkd_crash_log
```

- If no crash log exists:

```
index=_internal sourcetype=splunkd ("pipelines finished" OR "My GUID") | transaction startswith="My GUID" endswith="pipelines finished" keepevicted=true keeporphans=true | search closed_txn=0 | head 1
```

Splunk Workload Management (WLM)

- Policy-based Splunk system resource manager
  - **Workload pools** allocate dedicated compute and memory resources to Splunk processes
    - Assign individual scheduled or ad-hoc searches to designated workload pools, based on policies that you define in *workload rules*
  - **Workload categories** determine the resources available to *workload pools* by process types
    - Three types: search, ingest, and misc.
  - **Workload rules** control access to *workload pools*
    - Protects high-priority search workloads
    - Minimizes indexing latency due to heavy search load
- Create search rules to reserve and assign system resources based on apps and roles
  - Allow admins to take action on runaway search offenders

  http://docs.splunk.com/Documentation/Splunk/latest/Workloads/Keyconcepts

Lab Exercise 6 – Troubleshoot Job Problems

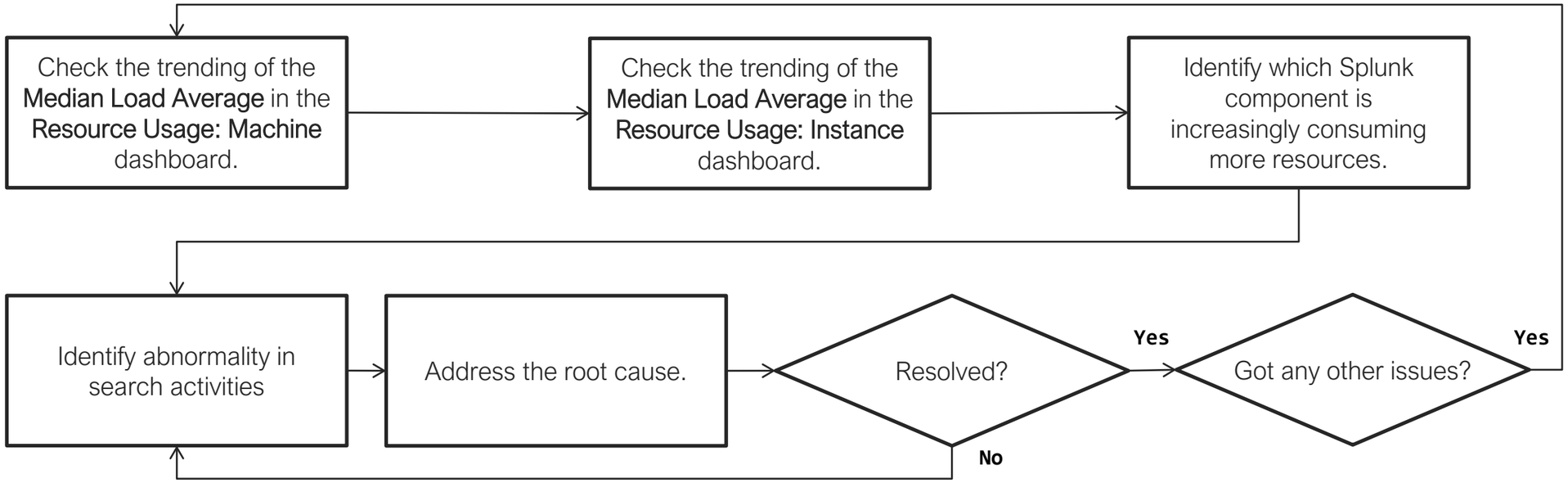
Time: 25 minutes

Tasks:

- Review the **Scheduler Activity: Instance** dashboard
- Confirm the search problems
- Investigate the root cause
- Restore the search functionality

Lab Exercise 6 – Troubleshooting Suggestions

- What is the limit of job types your search head can handle concurrently?
- Can you identify any search job patterns?
  - Who's running the searches, what type of searches, and from which apps?
- How does Splunk handle the search jobs when it is overloaded?
- What Splunk search-time options can help distribute the load?



Module 7:
User Search Problems

Module Objectives

- Identify the types of search problems
- Isolate and troubleshoot search problems

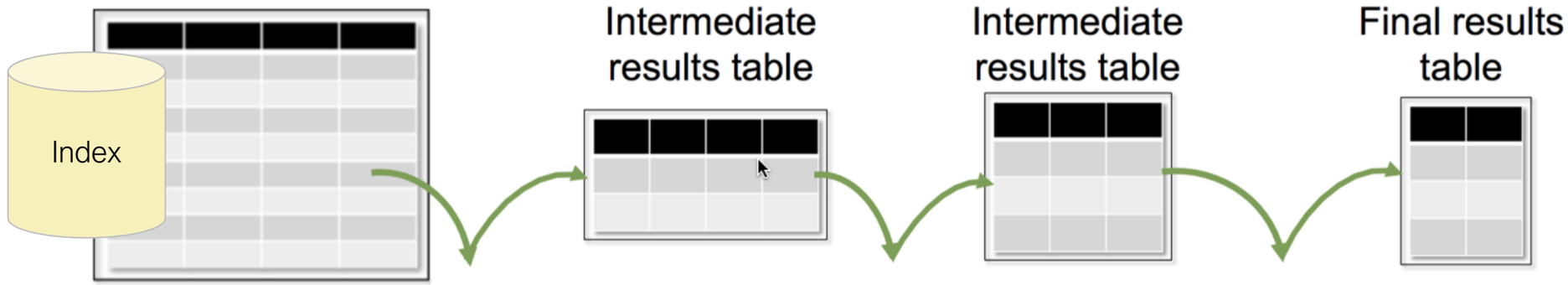
Identifying the Search Problem

- What type of search is this?
  - Ad-hoc search
  - Real-time search
  - Saved search
  - Scheduled search
  - Alerts
  - Load jobs
- What is the problem?
  - No results
  - Too many results
    - Duplicated events?
  - Partial results
    - Access permission?
  - Slow results
  - Error generated

Note

All search heads and search peers in the environment must have a unique name.

Search Pipeline and Processors



- Splunk Search Processing Language (SPL) encompasses all the search commands and their functions, arguments, and clauses
- Search command is a search processor
  - Each command after a | in a query is a search processor
- Search pipeline is an ordered group of search processors in SPL
  - Translates the commands into processor categories and optimizes them
- Each processor receives events in batch, processes them, and outputs the results based on the processor category

Determine Where the Search Breaks

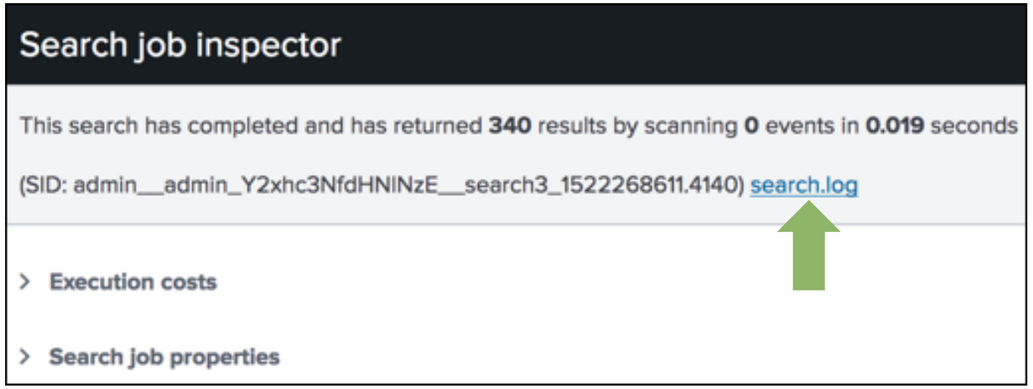
- As part of your fact finding, you should "run" the problem search
  - Run the search yourself (on example data) with same permissions
  - Break the search query down into logical pipe segments and run each one incrementally
- Iterate through each subset of the search and check the job inspector
  - Review the search properties and **`search.log`**
    - Pay attention to how Splunk optimizes the base *lispy* search https://en.wikipedia.org/wiki/Lisp_(programming_language)
  - Check the execution cost chart for any anomalies
  - Check the **`status.csv`** in the job directory

Search Job Inspector

- Job inspector provides details about a search in progress or finished
  - **Execution cost** can help you determine search processor performance
  - **Search job properties** is like a "query plan" but more
- Investigate search errors with **search.log**
  - Control the search logging level by:
    - Editing **limits.conf**

```
[search_info]
infocsv_log_level = DEBUG
```

    - Insert **| noop log_debug=*** after the base search
  - The default logging can use up to 30 MB (10 MB x 3)
  - Can adjust the size in **SPLUNK_HOME/etc/log-searchprocess.cfg**



http://docs.splunk.com/Documentation/Splunk/latest/Search/ViewsearchjobpropertieswiththeJobInspector

More on Splunk Jobs

- The job result is aggregated to SH and saved in: **`SPLUNK_HOME/var/run/splunk/dispatch/<search_id>/`**
  - Scheduled jobs include the search name as part of the directory name
- Job limits are defined as role restrictions in **`authorize.conf`**:
  - **`srchJobsQuota`** and **`cumulativeSrchJobsQuota`**
    - Maximum number of concurrently running searches a member of this role can have
    - Can be scoped at the role-level and the user-level
  - **`srchDiskQuota`**
    - Maximum disk space (MB) that can be taken by search jobs belonging to this role
- A list of all current jobs is available under **Activity** > **Jobs**
  - Or, via REST: **`| rest /services/search/jobs`**
- Job TTL determines how long the artifact remains in the **`dispatch`** directory

Default TTL for Job Artifacts

Ad-hoc search

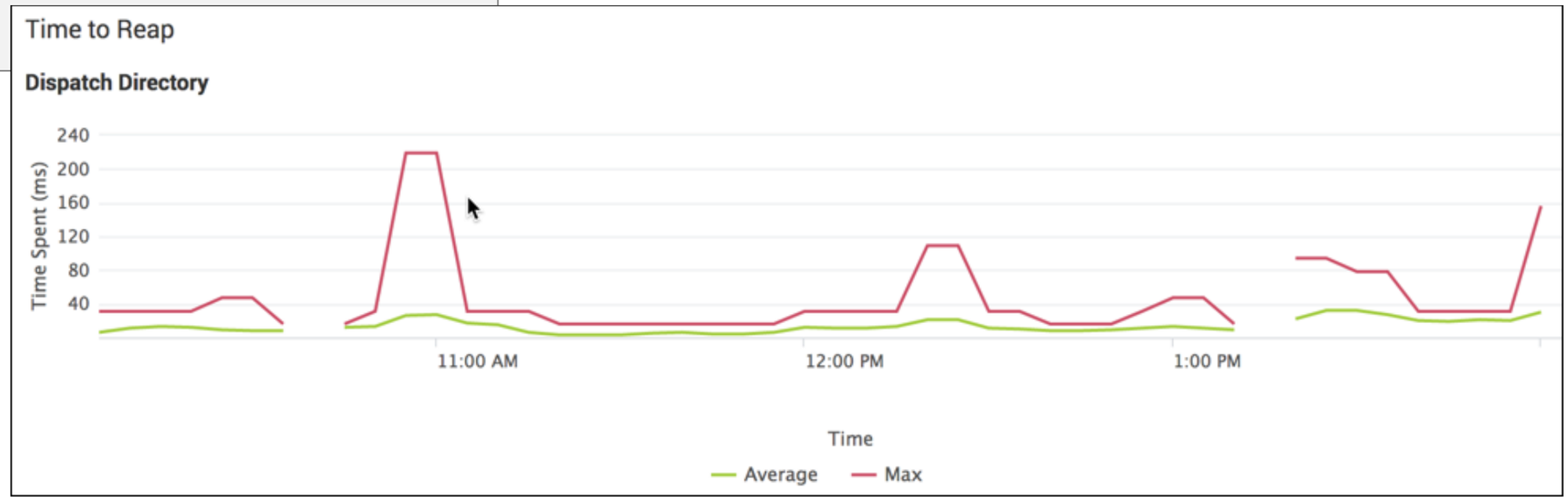
```
[search]
ttl = 600 (10 min)
default_save_ttl = 604800 (1 week)
remote_ttl = 600 (10 minutes)
failed_job_ttl = 86400 (24 hours)

[subsearch]
ttl = 300 (5 min)
```

limits.conf

Scheduled search

```
[search_name]
dispatch.ttl = 2p (2x the scheduled period)
```

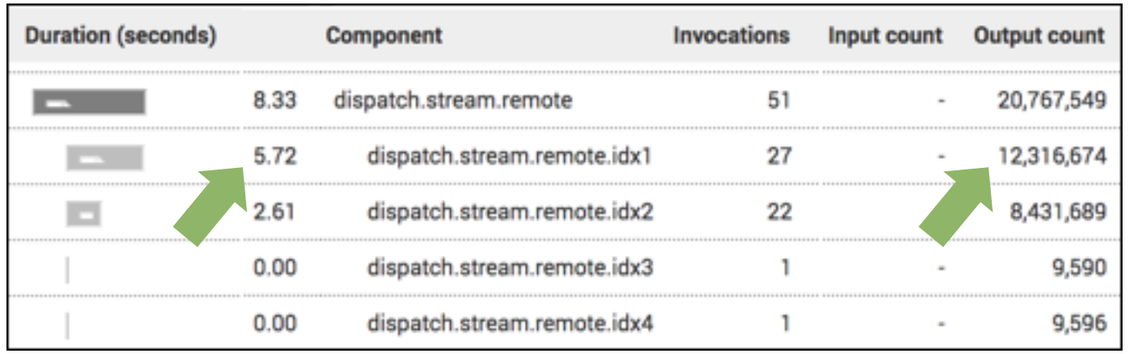
savedsearches.conf



MC > Search > Distributed Search > Distributed Search: Instance

Troubleshooting Slow Search Problems

- Check resource usage and search processes in Monitoring Console
- Verify and optimize the search query
  - Check the execution cost
  - Strip out the data you don't need
  - Avoid reporting commands early in the search



Duration (seconds)	Component	Invocations	Input count	Output count
8.33	dispatch.stream.remote	51	-	20,767,549
5.72	dispatch.stream.remote.idx1	27	-	12,316,674
2.61	dispatch.stream.remote.idx2	22		8,431,689
0.00	dispatch.stream.remote.idx3	1	-	9,590
0.00	dispatch.stream.remote.idx4	1	-	9,596

- Use the **splunk_server** field to determine if correct search peers are participating
- Restrict searches to certain peers using **splunk_server_group**

```
[distributedSearch]
servers = https://idx1:8089,https://idx2:8089,https://idx3:8089,https://sh1:8089
[distributedSearch:peers4myapp]
servers = localhost:localhost,https://idx1:8089,https://idx2:8089
```

distsearch.conf

Useful Searches

- Long-running search?

```
index="_audit" action="search" (id=* OR search_id=*) | eval
user=if(user=="n/a",null(),user) | stats max(total_run_time) as
total_run_time first(user) as user by search_id | stats count
perc95(total_run_time) median(total_run_time) by user
```

- How much time are the indexers spending in response to queries from SH?

```
index=_internal source=*remote_searches.log server=<sh> | stats
max(elapsedTime) by search_id host
```

- Identify all splunkd responses taking more than 100ms

```
index=_internal sourcetype=splunkd_access host=<sh> user=<user> | rex
"(?<spent>\d+)ms" | search spent > 100
```

- What is the size of the artifacts?

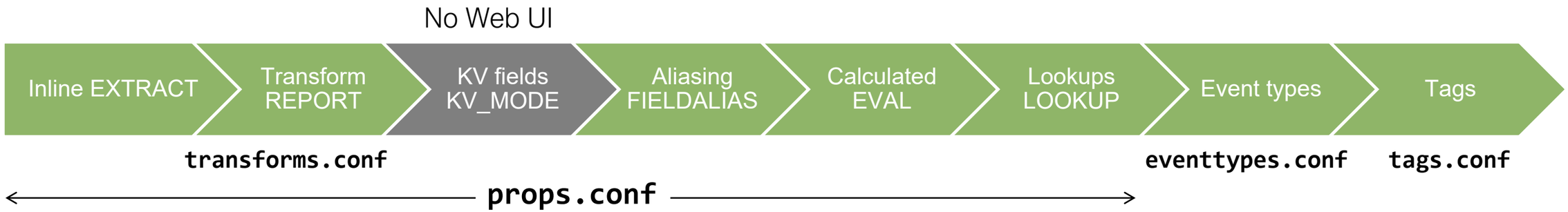
```
index=_internal sourcetype=splunkd_access method=GET jobs | stats
sum(bytes) by uri
```


Troubleshooting No Results Problems

- Modify time span
  - Do not just use **All Time**
  - First try **`_index_earliest=-10m`**
- Double check the logic
  - For example, is the user trying to average a non-numeric field?
- Use explicit **`index`**, **`host`**, **`sourcetype`**, **`source`**, and **`splunk_server`**
  - **`index=* host=<x> sourcetype=<y> splunk_server=<indexer>`**
- Has the data actually been indexed? (segmenters and base LISPY)
- Do you have permissions to see the data?
- Is the quota OK? (free space, role, and user)
- Is the peer you are searching in automatic detention?

Search-Time Operation Sequence

- Splunk search derives knowledge objects into a flattened search following a specific sequence
  - Example: Fields from lookups are unavailable when calculated fields reference them in an **`eval`** expression



http://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Searchtimeoperationssequence

Lab Exercise 7 – Troubleshoot Search Issues

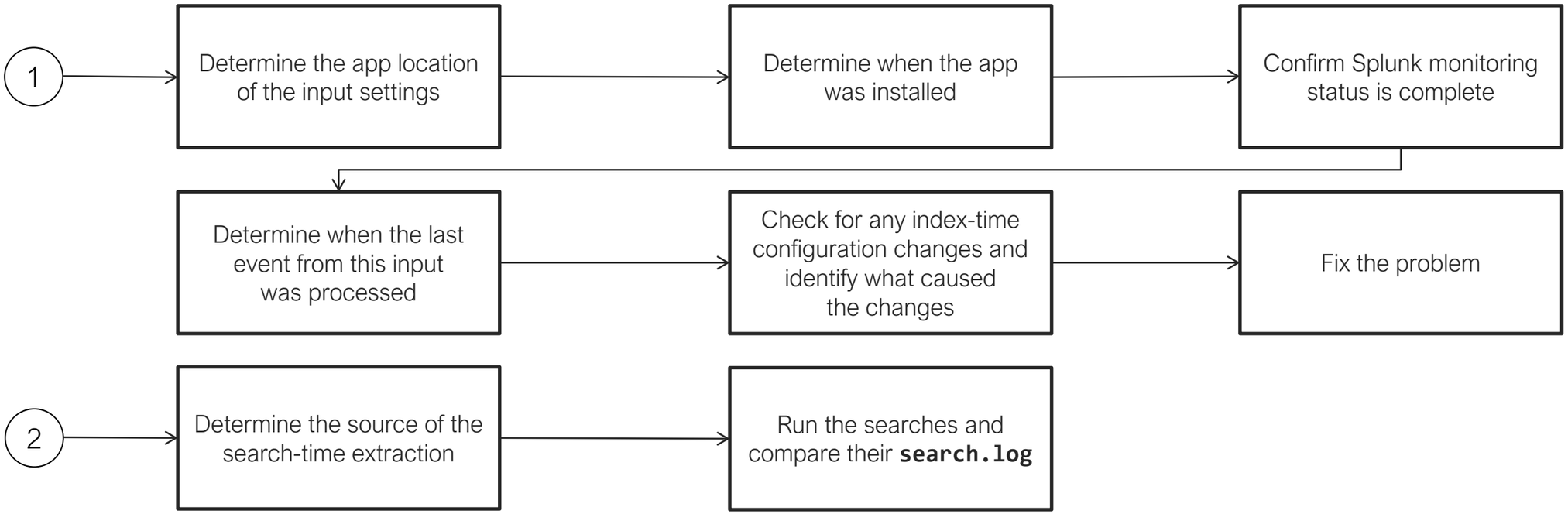
Time: 25 minutes

Tasks:

- Install the **`tse_lab07`** app
- Two search problems:
  - **`index=itops sourcetype=ghostwww | timechart count by host`** shows data is missing from a host
  - **`index=itops sourcetype=wocheese country=BA OR country=*ZB`** shows results but **`index=itops sourcetype=wocheese country=ZB`** does not
- Investigate the root causes and resolve the issues

Lab Exercise 7 – Troubleshooting Suggestions

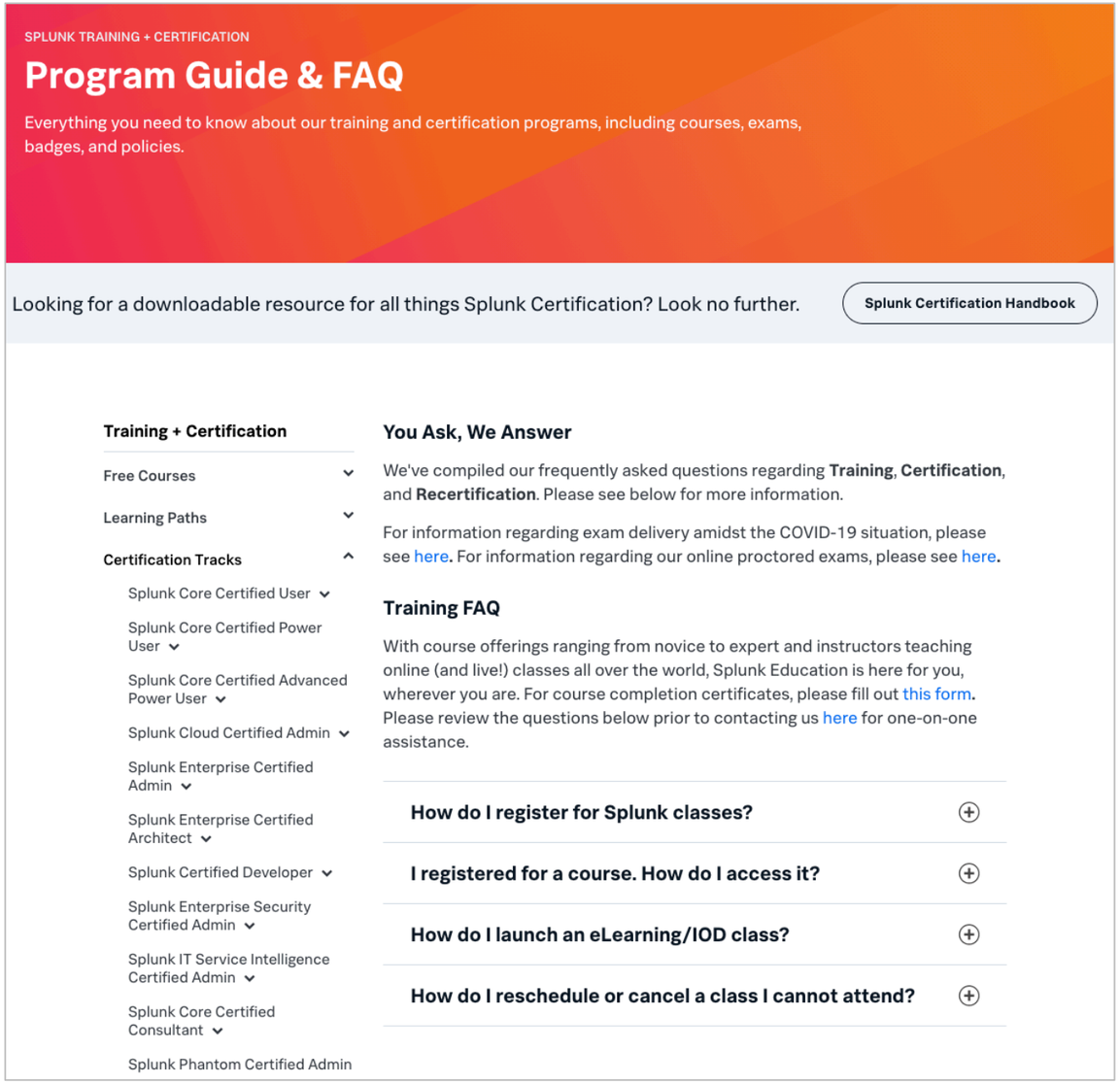
1. Can you confirm the input was working before?
   - Since when did the events go missing?
   - Any changes to the system prior to this time?
2. Can you confirm if the searched keywords are in the index?



Wrap-up Slides

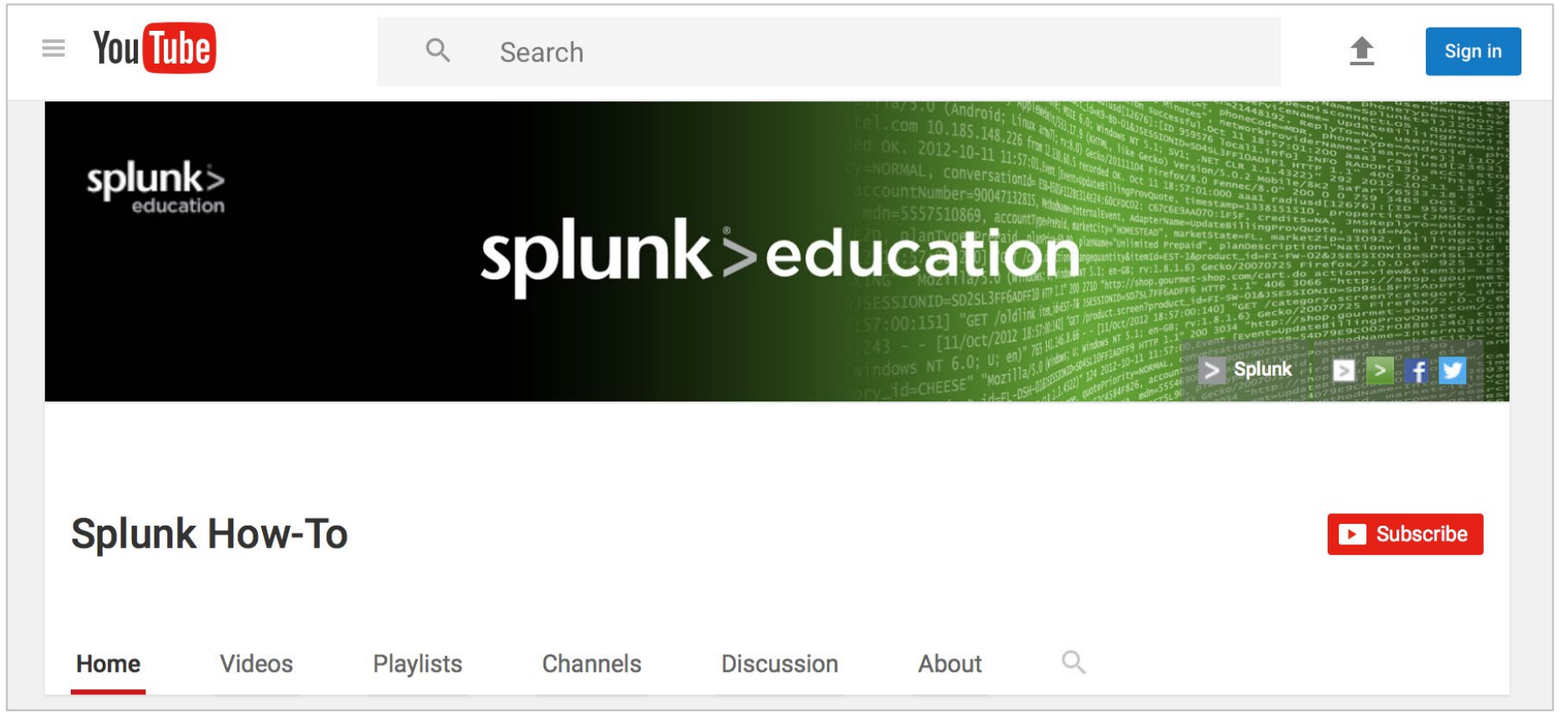
What's Next?

- Splunk Certification program
  https://www.splunk.com/en_us/training/faq-training.html
- Program information
  - https://www.splunk.com/pdfs/training/Splunk-Certification-Candidate-Handbook.pdf
- Exam registration
  - https://www.splunk.com/pdfs/training/Exam-Registration-Tutorial.pdf
- If you have further questions, send an email to: certification@splunk.com



YouTube: The Splunk How-To Channel

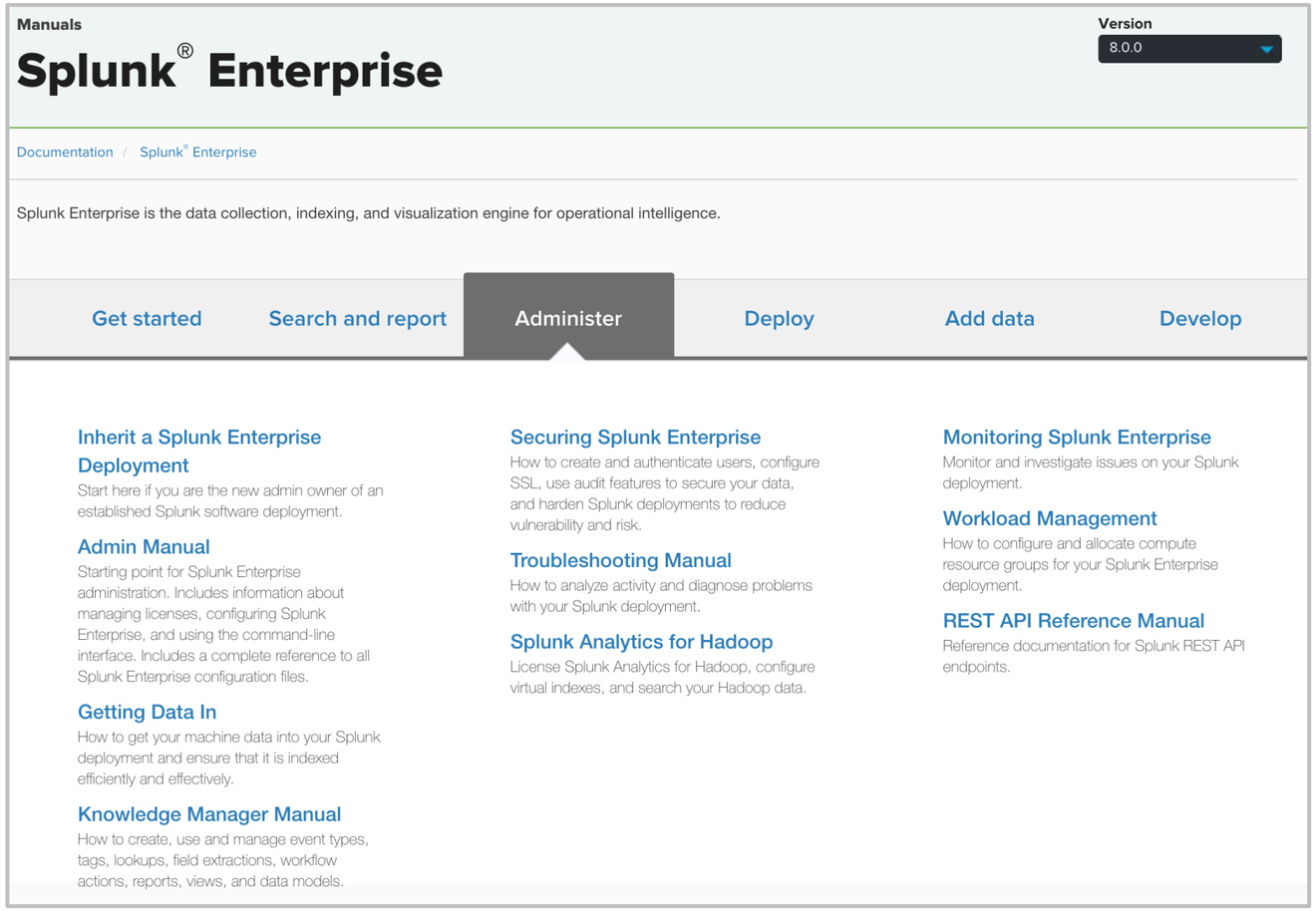
- In addition to our roster of training courses, check out the Splunk Education How-To channel: http://www.youtube.com/c/SplunkHowTo
- This site provides useful, short videos on a variety of Splunk topics



Appendix A: Splunk Self-Service and Support Programs

Splunk Self-Service Support Resources

- Splunk online documentation
  - http://docs.splunk.com
  - Make sure you select the correct version
- Splunk community portal
  - http://community.splunk.com



Splunk Community Portal

- Ask an expert: https://community.splunk.com
  - Get answers to your questions from Splunk experts
- Splunk blogs: https://www.splunk.com/en_us/blog
- Splunk for developers: http://dev.splunk.com
- Hot wiki topics: http://wiki.splunk.com
  - Includes best practices and how-tos
- Splunk user groups: https://usergroups.splunk.com
  - Connect with like-minded Splunk professionals near you

Splunk Support Programs

	Community	Standard	Premium
	(Free)	(Enterprise, Cloud, Premium Apps)	
Access to Splunk Documentation	✓	✓	✓
Access to Splunk Answers	✓	✓	✓
Live Product Roadmap Input	✓	✓	✓
Online Case Submission		✓	✓
Online Case Status		✓	✓
Response Time Targets		✓	✓
Phone Support		✓	✓
Direct Access to Advanced Support Team			✓
24/7 Availability		P1	P1 & P2

Splunk support versions: https://www.splunk.com/en_us/support-and-services/support-programs.html

Splunk Support Case Priority Levels

Priority Level	Description
P1	• A production environment is completely inaccessible • Most functionality is unusable
P2	• One or more important features of a production environment is unusable
P3	• Any feature of purchased Splunk software is not operating as documented
P4	• All enhancement requests and general questions

Check targeted response times:
https://www.splunk.com/en_us/support-and-services/support-programs.html

Note

Targeted response times are based on support program and priority level.

Escalating to Splunk Support

- Splunk Support is your communication channel to all things Splunk
  - Splunk Support Portal
    - http://login.splunk.com/page/sso_redirect?type=portal
  - Phone: (855) SPLUNKS or (855) 775-8657
    - Regional business hours, 24x7 for critical issues
  - More in-country toll-free hotlines:
    - http://www.splunk.com/en_us/about-us/contact.html#tabs/customer-support

Splunk Success Plans

- An additional tiered, Splunk support entitlement

  https://www.splunk.com/en_us/support-and-services/support-programs.html#success

- Includes Support, Admin on Demand (AoD), and Customer Success Managers (CSM)
  - AoD is a credit-based, technical assistance service to provide pre-defined PS tasks

    https://www.splunk.com/pdfs/legal/Splunk-Admin-On-Demand-CSM-Datasheet-101.pdf

    - Click **Admin on Demand Submit Request** from the Splunk Customer Support Portal
  - A CSM is your Splunk advocate and will help you with use cases, data lifecycle management, program management, etc.

Appendix B: Windows Input Error

Scripted Input Errors and Fixes

- Fix the permissions
  - Splunk is running as a user who doesn't have the permission to run the script
    - Or the script is trying to do something that the Splunk user can't do
  - Make sure that the script actually works by running it manually

    ./splunk cmd ../etc/apps/<app>/bin/<script>
- Wrong directory
  - Splunk runs scripted inputs from designated **bin** directories
    - **SPLUNK_HOME/etc/system/bin**
    - **SPLUNK_HOME/etc/apps/<app_name>/bin**
  - Move the script to the proper location
- Duplicate data
  - A custom script may lack the intelligence that the monitor has
  - Make sure the script "knows" what to output on each run
  - Modular input provides a checkpoint facility

Windows Inputs Errors and Fixes

- Use local Windows collection whenever possible
  - Local Windows inputs behave differently than remote collection
  - Local Windows inputs only involve a forwarder setup
- Too much data from AD monitor (**admon**)
  - Splunk AD monitor recursively traverses the AD tree from a given start point
    - When set without any restrictions, Splunk may read too much data
  - Limit with **startingNode** and **monitorSubtree**
  - http://docs.splunk.com/Documentation/Splunk/latest/Data/MonitorActiveDirectory
- For WMI inputs, the user must have the proper domain privileges to gather remote data
  - Can only support a small number of remote clients
  - Remote collection on desktop clients is turned off

Windows Inputs Errors and Fixes (cont.)

- Windows servers can generate a lot of events
- A busy Windows UF can encounter collection limit, especially when the UF has been restarted
  - Might still be collecting the older events first
  - If you can, upgrade the forwarder to the latest version
  - If upgrade is not possible, collect only recent logs or monitor the recent events first
    - **`current_only=1`**
    - **`start_from=newest`**
- Reduce the volume of events to collect
  - Use **`whitelist/blacklist`** to filter specific **`EventCodes`** or ***`key=regex`***
- Sysinternals: https://docs.microsoft.com/en-us/sysinternals/

Considerations for Windows Installer Inputs

- When you select inputs from the list while using the Windows universal forwarder installer, the resulting **`inputs.conf`** is:
  - Put in **etc/system/local**
  - Owned by msi-installer
  - Cannot be managed via deployment server
  - Will not allow Windows to install TA on top