Forward-Looking Statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

splunk> .conf21

From Noob to Ninja: Growing and Managing Splunk Enterprise as a Team of One

PLA1410C

Dan Burras

Distinguished Engineer | Verizon

splunk> .conf21

Dan Burras

Distinguished Engineer | Verizon

splunk> .conf21

Agenda

1) Background
   Why we're here - both myself and you in the audience

2) Automation
   How to take your hands off the wheel, at least for a bit

3) Visualizations
   How not to get overwhelmed with too much to look at

4) Service Health Score
   How to let the tool watch the tool so that you can take a break from watching things

5) Takeaways
   Wrap up and some nuggets of wisdom!

Disclaimer

Any information or opinions expressed during this presentation are solely based on my own expertise and experience and are not presented on behalf of my employer.

Background

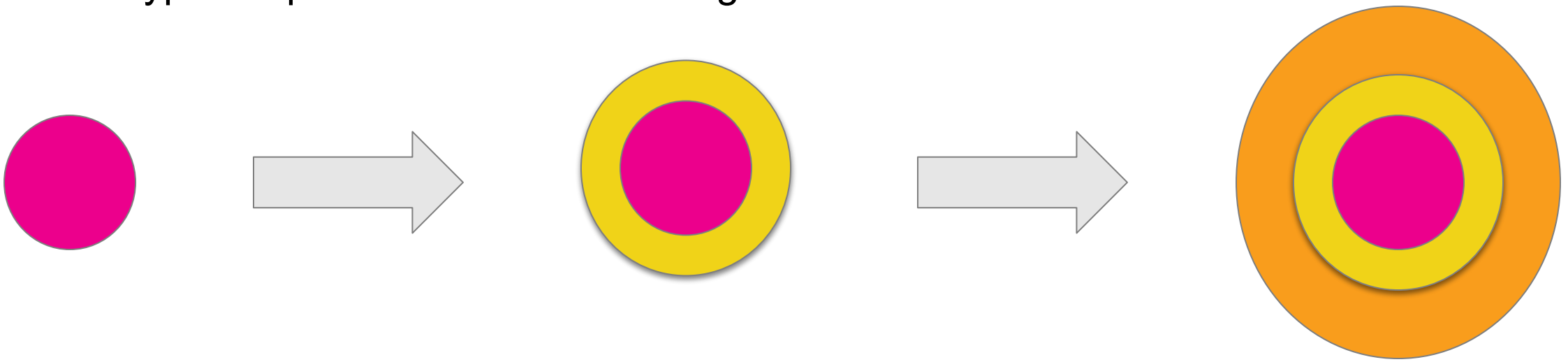
Started in 2013 as a small single case Splunk system - less than 10 servers overall

Having extra capacity = adding more use cases

Then people start using the platform and everyone wants more….

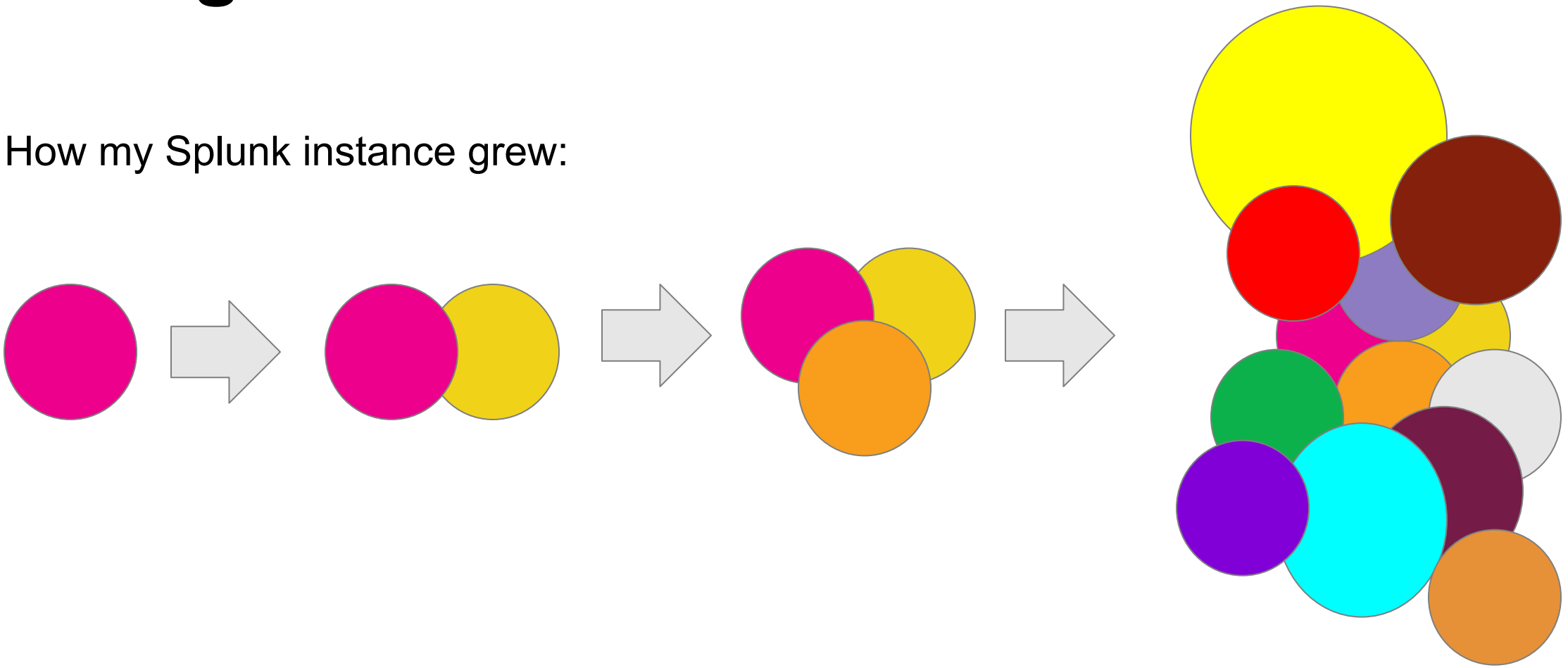
Background

How a typical Splunk instance tends to grow:



Background

How my Splunk instance grew:



Background

Non-traditional growth pattern means:

- Hundreds of indexes
- Thousands of sourcetypes
- Dozens of platforms
- Dozens (hundreds?) of different ways of collecting data
- Dozens of different executives and team

All with different requirements and needs!

Background

Platform growth numbers (over 500 servers in total)

- Indexers - from 1 to 75
- Search Heads - from 0 to 64
- Search Head Clusters - from 0 to 8
- Integration Servers (HF) - from 1 to 250
- DB Connect Servers (HF) - from 1 to 60

Growth Adds

- ITSI
- DSP
- Standalone training environment
- Dev environment
- Lab environment

Background

From 2013 to present:

- Number of Splunk admins = 1
- Number of Splunk architects = 1
- Number of Splunk visualization support folks = 1

In 2019 added a group responsible for Visualization Support and Dashboard Building

In 2021 added another person responsible for day-to-day Splunk admin work

- Account creation
- Daily health checks
- On-call and tier 1 support calls

Automation

What is automation?

- Any technology that minimizes human input or interaction with a process

What can we automate?

- Server builds
- Splunk installs and upgrades
- Backups
- User administration
- Any Splunk CLI interaction
- Any API (Splunk or otherwise)

Automation

How do we automate?

- Use any of the boundless numbers of automation tools out there:
  - Chef
  - Puppet
  - Ansible
  - Jenkins
  - Docker
  - HP Operations Orchestration
  - ActiveBatch

What's the right tool to use?

- Any of them! Whichever tool or tools work within the scope and structure of your organization or business

There is no single right answer!

Automation

My personal choice? PDSH

Automation Examples:

- Upgrade Splunk to a new version
  - pdsh -w ^server.txt -l splunk “/opt/splunk/bin/splunk stop”
  - pdsh -w ^server.txt -l splunk “tar xvzf splunk-8.2.1-ddff1c41e5cf-Linux-x86_64.tgz -C /opt”
  - pdsh -w ^server.txt -l splunk “/opt/splunk/bin/splunk start --accept-license --answer-yes”
- Update pass4SymmKey
  - pdsh -w ^server.txt “sed -i ‘3d’ /opt/splunk/etc/system/local/server.conf”
  - pdsh -w ^server.txt “sed -i ‘3ipass4SymmKey = newPass4SymmKey’ /opt/splunk/etc/system/local/server/conf”
  - pdsh -w ^server.txt -l splunk “/opt/splunk/bin/splunk restart”
- Add new admin user:
  - pdsh -w ^server.txt -l splunk “/opt/splunk/bin/splunk add user newAdmin -role admin -password PaSsWoRd”

Visualizations

Don’t we already have the DMC?

- Yes, but…

What about health reports?

- Yes, but…

What about any of the dozens of other awesome Splunk health apps out there?

- Yes, but...

Visualizations

Scale issues

- DMC less effective at large scale, especially in a single system
- API calls against hundreds of servers don’t always return in time

Scope Issues

- API calls are point-in-time references - how do I track over time?
- DMC is designed to be all-encompassing
- Health reports are only easily visible per system (or at best per cluster)
- External health apps - too all-encompassing AND too narrowly focused

As a stressed out admin I only want to see what I really need to see when and where I need to see it!

Visualizations

So how do we fix it?

- Figure out which visualizations you really need to see
- Figure out how often you really need to see those visualizations
- Build your own - that's what we tend to do anyway
- Summarize pertinent API calls

How many visualizations should I have?

- As many as you need to effectively do your job
- Make your views as focused as possible - separate by system or how often you'll view them
- Be purpose-driven - if it doesn't help don't have it!
- Use the DMC and other health apps for inspiration and then make them your own

How do I figure out what's purposeful for my environment? That's in the next section…

Visualizations

What do I have for my environment?

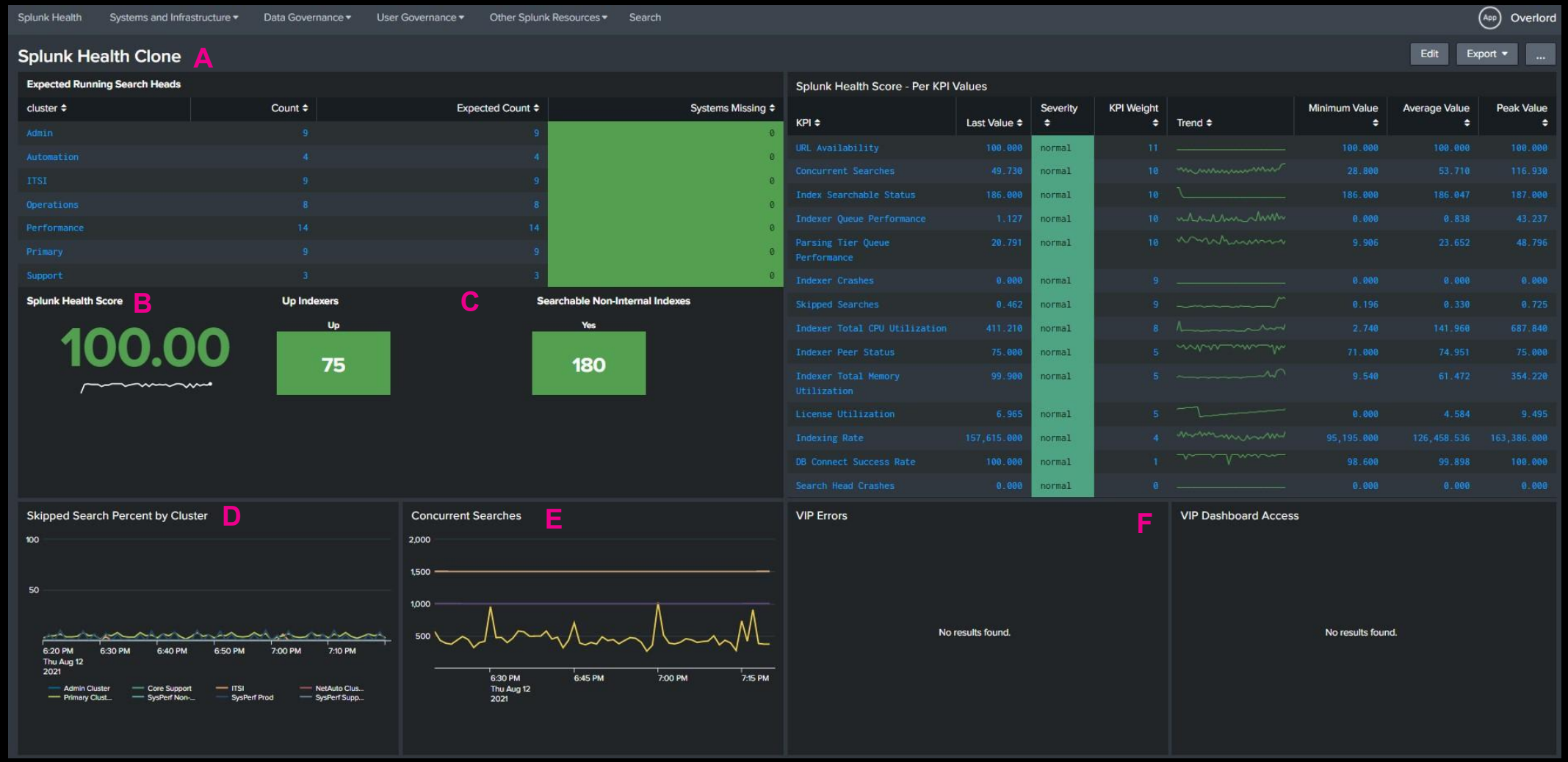
- One “overlord” view kept constantly running at all times
- A few daily/weekly health check dashboards
- A few dashboards for troubleshooting specific issue

What should you have in your setup?

– **Whatever makes sense for you!**

How should I make things look? Should I just copy your visuals?

– **Heck no! Do what makes sense for you!**



splunk> .conf21

Visualizations

This and the next visualization are actually a single display that is kept constantly running with auto-refreshing panels. It's our primary single pane of glass health view.

- A - Expected Running Search Heads
  - Counts systems in each cluster to ensure all are present
  - Turns red if any are missing
  - Search: | tstats count where index=_internal AND host IN (splsearch*) by host | eval cluster = case(match(host,"splsearchitsi\d+.*?"),"ITSI", match(host,"splsearch(06|07|08|09|29|30|31|32|33)\.*?"),"Admin",match(host,"splsearch(01|02|03|04|05|41|42|43|44)\.*?"),"Primary",match(host,"splsearch(37|38|39|40)\.*?"),"Automation",match(host,"splsearch(10|11|12|13|45|46|47|48)\.*?"),"Operations",match(host,"splsearch(15|16|17|18|19|20|21|22|23|24|25|26|27|28)\.*?"),"Performance",1==1,"Support") | eval expected_count = case(match(cluster,"Admin"),"9",match(cluster,"ITSI"),"9",match(cluster,"Primary"),"9",match(cluster,"Automation"),"4",match(cluster,"Operations"),"8",match(cluster,"Performance"),"14",1==1,"3") | stats dc(host) as running values(expected_count) as expected by cluster | eval missing = expected-running | rename missing AS "Systems Missing" running AS "Count" expected as "Expected Count"
- B - Splunk Health Score
  - A running tracker of the overall Splunk Health Score from ITSI
  - Search: index=itsi_summary serviceid=a51e5df0-5b64-4b83-9a13-433896149cd9 kpi=ServiceHealthScore | timechart avg(health_score) AS "Health Score"

Visualizations

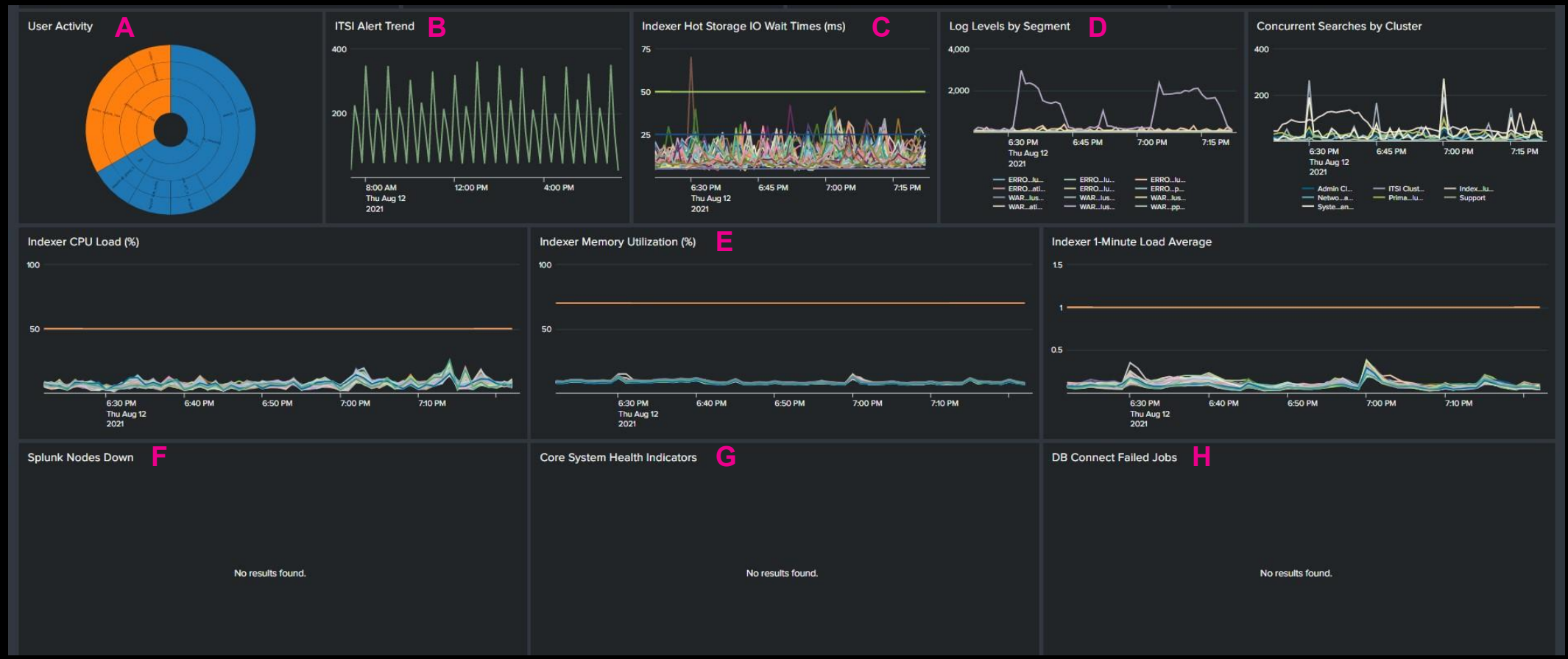
Notes:

- C - Up Indexers and Searchable non-Internal Indexes
  - Look familiar? It should - it's taken from the Indexer Cluster Master view
  - Count of how many indexers in an up state and how many non-internal indexes are currently searchable
  - Example of where we've taken APIs and summarized the data for long-term tracking
  - Search 1: index=splunk_metrics sourcetype=indexer_cluster_peers | fields label, status | stats latest(status) as status by label | stats dc(label) as count by status | where status="Up"
  - Search 2: index=splunk_metrics sourcetype=cluster_master_indexes title!="_*" | eval is_searchable = if((is_searchable == 1) or (is_searchable == "1"), "Yes", "No") | stats latest(is_searchable) as is_searchable by title | stats dc(title) as count by is_searchable | where is_searchable="Yes"
- D - Skipped Searches by Cluster
  - Shows the percent of skipped searches happening in any given cluster over _time
  - Search: index=_internal host=splsearch* sourcetype=scheduler | eval cluster = case(match(host,"splsearchitsi\d+.*?"),"ITSI", match(host,"splsearch(06|07|08|09|29|30|31|32|33)\.*?"),"Admin",match(host,"splsearch(01|02|03|04|05|41|42|43|44)\.*?"),"Primary",match(host,"splsearch(37|38|39|40)\.*?"),"Automation",match(host,"splsearch(10|11|12|13|45|46|47|48)\.*?"),"Operations",match(host,"splsearch(15|16|17|18|19|20|21|22|23|24|25|26|27|28)\.*?"),"Performance",1==1,"Support") | bucket _time span=1min | stats count as Total_Schedules, count(eval(status="skipped")) as skipped by cluster _time | eval pct_skipped=round(skipped/Total_Schedules * 100, 4) | timechart span=1m values(pct_skipped) as pct_skipped by cluster

Visualizations

Notes:

- E - Concurrent Searches
  - Shows the overall number of concurrent searches happening in the platform
  - Search: index=_internal source=*metrics.log group="search_concurrency" active_hist_searches=* active_hist_searches!=0 "system total" | timechart span=1m sum(active_hist_searches) as concurrent_searches | eval "start paying attention"=1000 | eval "start freaking out"=1500
- F - VIP Errors/Dashboards
  - Shows what “VIPs” are in using the system at any time and what errors they are experiencing
  - Search 1: index=_internal sourcetype=splunkd index=_internal host=splsearch* log_level!=INFO username IN (user1,user2,user3,user4) | stats count by username reason | lookup user_manager_data title as username OUTPUT realname | table realname reason count | rename realname AS "User Name" reason AS "Error Message" count AS "Number of times"
  - Search 2: index=_internal sourcetype=splunkd_ui_access "en-US/app" host=splsearch* user IN (user1,user2,user3,user4) | rex "GET /[^/]+/app/(?<app>[^/ ?]+)/(?<dashboard>[^/ ?]+)" | stats count by user dashboard | lookup user_manager_data title AS user OUTPUT realname | table realname dashboard| rename realname as User dashboard as "Page Accessed"



splunk> .conf21

Visualizations

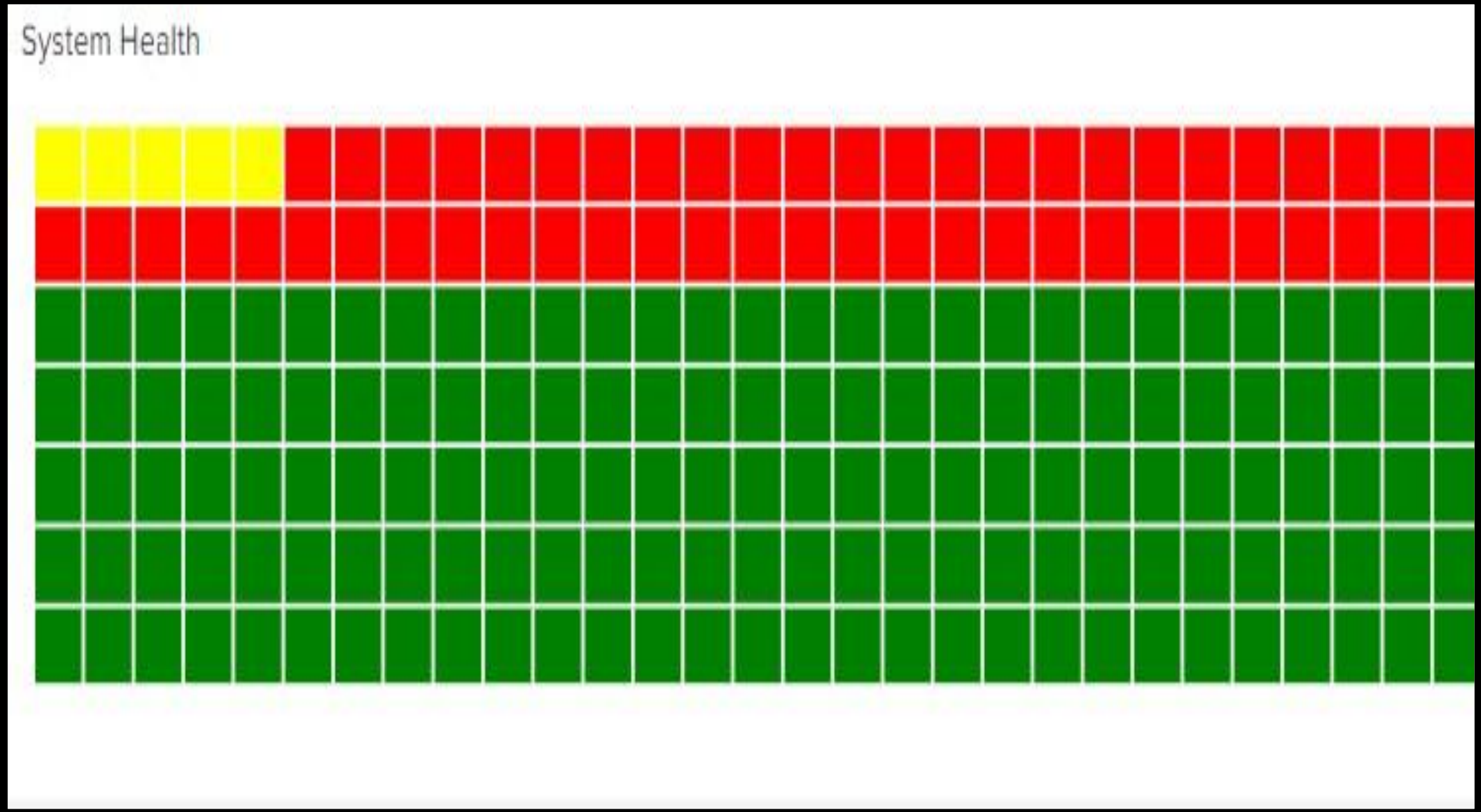
Continuation of the previous dashboard display...

- A - User Activity
  - Breakdown of search activity by cluster, app, user
  - Search: index=_internal sourcetype=splunkd_ui_access "en-US/app" user!="-" host=splsearch* | rex field=uri "en-US/app/(?<app>[^/]+)/(?<dashboard>[^?/\s]+)" | eval cluster = case(match(host,"splsearchitsi\d+.*?"),"ITSI Cluster",match(host,"splsearch(06|07|08|09|29|30|31|32|33)\.*?"),"Admin Cluster",match(host,"splsearch(01|02|03|04|05|41|42|43|44)\.*?"),"Primary Cluster",1==1,"Other") | stats count by cluster app dashboard user
- B - ITSI Episode Trend
  - A running track of the count of episodes created by ITSI
  - Let's us know when ITSI may not be functioning as intended
  - Search: (index=itsi_grouped_alerts NOT source=itsi@internal@group_closing_event sourcetype=itsi_notable:group) | fields + _time | timechart count by events | rename NULL as "Total Alerts"
- C - Indexer IO Wait Time
  - Timechart of wait time for hot storage with warning thresholds
  - Search: index=_introspection host=splindex* component=IOStats data.mount_point=*splunkdata_na* data.avg_total_ms!="-*" | rename data.mount_point AS mount_point | eval host-mount = host."-".mount_point | timechart limit=0 useother=f span=1m avg(data.avg_total_ms) AS iowait_ms by host-mount | eval low_threshold=5 | eval moderate_threshold=25 | eval high threshold=50

Visualizations

Notes:

- D - Log Levels by Segment
  - Breakdown of WARN/ERROR/CRITICAL/FATAL messages in Splunk logs by cluster over _time
  - Visual ability to look for abnormal patterns
- E - Indexer CPU/Memory Utilization and Load Average
  - Standard Indexer metrics from _introspection presented over time
- F - Splunk Nodes Down
  - Report of any Splunk nodes not talking in the last 5 minutes
- G - Core System Health Indicators
  - Breakdown of data from Splunk Health Reports looking for unexpected results
  - Search: index=_internal host=spl* source=*health.log color IN (red,yellow) | stats latest(color) AS Color values(reason) AS Reason by feature host | rename feature as Feature host AS Host
- H - DB Connect Failed Jobs
  - Breakdown of any DB Connect jobs that have failed in the last 15 minutes
  - Search: index=_internal host=*dbconn* sourcetype=dbx_server | rex field=_raw "Job \'(?<job_name>.*?)\' finished with status:\s(?<job_status>\w+)" | search job_status=FAILED | stats latest(job_status) by job_name



Visualizations

This dashboard is used daily as a quick health check

- Overall System Health
  - Presentation of Splunk Health report results by system
  - Filterable by location and function of the system
  - Saved Report: index=_internal sourcetype=splunkd source=*health.log host=spl*| stats latest(color) as color by host | foreach * [eval icon=case('color'=="yellow","times-circle",'color'=="red","exclamation-circle",1==1,"check-circle"), color=case('color'=="yellow","yellow",'color'=="red","red",1==1,"green")] | stats last(host) as value last(icon) as icon last(color) as color by host | sort - icon
  - Dashboard Panel Search: | loadjob savedsearch="user:app:infrastructure_health_all" | search host=* host=* color=* | stats last(host) as value last(icon) as icon last(color) as color by host | sort - color

Data Onboarding

Edit | Export | ...

Date Parsing Issues

70

affected sourcetypes

Truncation Issues

2

affected sourcetypes

Line Merging Issues

2

affected sourcetypes

Date Parsing Issue Specifics

sourcetype	count	Affected Source Count	Affected Host Count	Last Message
access_log	18678	8	8	Failed to parse timestamp in first MAX_TIMESTAMP_LOOKAHEAD (128) characters of event
dsstats	1547	1	4	Failed to parse timestamp in first MAX_TIMESTAMP_LOOKAHEAD (128) characters of event
splunk_vsphere_vim25.log	567	5	6	Failed to parse timestamp in first MAX_TIMESTAMP_LOOKAHEAD (44) characters of event
regional_schedules	168	56	1	A possible timestamp match (Wed Sep 26 23:29:42 2001) is outside of the acceptable time window
stb_schedules	51	22	1	A possible timestamp match (Wed Sep 26 23:29:41 2001) is outside of the acceptable time window

« Prev 1 2 3 Next »

Data Truncation Issue Specifics

sourcetype	Allowed Lines	count	Peak Lines	Affected Source Count	Affected Host Count
itsi_internal_log	50000	26	65536	3	9
debugmessages	10000	2	23784	1	1

Line Merging Issue Details

sourcetype	count	Error Message	Affected Source Count	Affected Host Count
regional_verifications	75	Breaking event because limit of 256 has been exceeded	75	1
regional_schedules	47	Breaking event because limit of 256 has been exceeded	45	1

Visualizations

This dashboard is used daily to verify data health

- Date Parsing Issues
  - Count of the number of sourcetypes affected by date parsing issues
  - Search can be used to extrapolate specifics about the issues in later panels
  - Search: index=_internal host=spl* sourcetype=splunkd component=DateParserVerbose log_level=WARN | rex "Context:\s+source=(?<data_source>[^\|]+)\|host=(?<data_host>[^\|]+)\|(?<data_sourcetype>[^\|]+)" | stats count values(data_source) values(data_host) dc(data_source) dc(data_host) BY data_sourcetype | sort - count | stats dc(data_sourcetype)
- Truncation Issues
  - Count of the number of sourcetypes affected by data truncation issues
  - Search can be used to extrapolate specifics about the issues in later panels
  - Search: index=_internal host=spl* sourcetype=splunkd component=LineBreakingProcessor | extract | rex "because\slimit\sof\s(?<limit>\S+).*>=\s(?<actual>\S+)" | stats count avg(actual) max(actual) values(data_source) values(data_host) dc(data_source) dc(data_host) BY data_sourcetype, limit | eval avg(actual)=round('avg(actual)') | sort - count | stats dc(data_sourcetype)
- Line Merging Issues
  - Count of the number of sourcetypes affected by line merging issues
  - Search can be used to extrapolate specifics about the issues in later panels
  - Search: index=_internal host=spl* sourcetype=splunkd component=Aggregator* NOT "Too many events * with the same timestamp" | rex "\s-\s(?<message_content>.*?)\s-\sdata" | extract | stats count values(message_content) values(data_source) values(data_host) dc(data_source), dc(data_host) BY data_sourcetype | sort - count | stats dc(data_sourcetype)

Visualizations

This dashboard is used every few days to check on indexer-specific health issues

- Indexing rate/searches/volume are pulled from APIs from the DMC
- Event Distribution
  - Trends the distribution of events across the indexer cluster
  - Abnormal bulges indicate a problem
  - Search: `| tstats prestats=t count WHERE index=* BY splunk_server, _time span=1d | timechart limit=100 span=1d count by splunk_server`

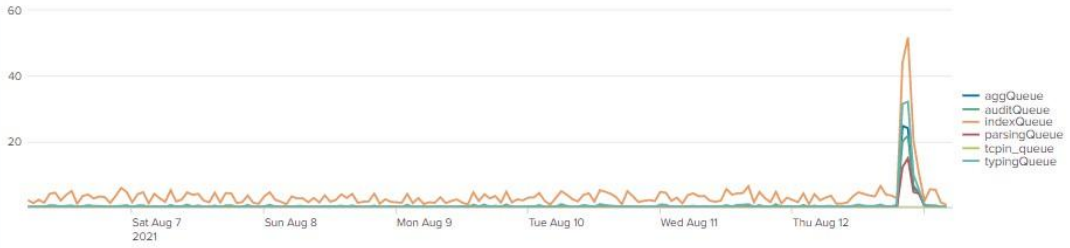
Queue Performance

Edit | Export | ...

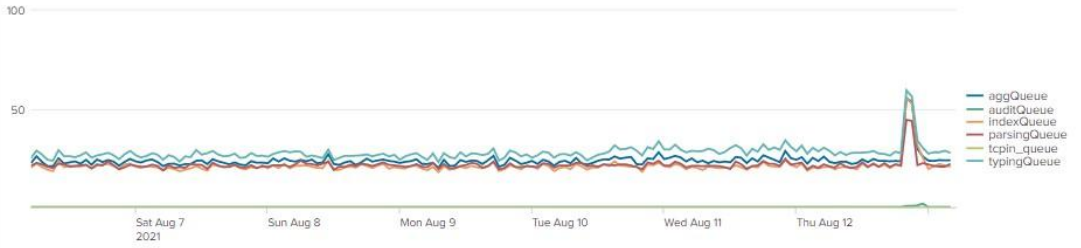
Avg Parsing Queue Utilization (Indexer Tier)	Avg Aggregation Queue Utilization (Indexer Tier)	Avg Typing Queue Utilization (Indexer Tier)	Avg Indexing Queue Utilization (Indexing Tier)
22.87 %	0.00 %	0.27 %	2.50 %

Avg Parsing Queue (Parsing Tier)	Avg Aggregation Queue Utilization (Parsing Tier)	Avg Typing Queue Utilization (Parsing Tier)	Avg Indexing Queue Utilization (Parsing Tier)
22.87 %	25.51 %	30.16 %	23.60 %

Indexer Queue Utilization Trends



Parsing Tier Queue Utilization Trends



splunk> .conf21

Visualizations

This dashboard is viewed weekly to look for unexpected utilization but also used for troubleshooting.

- Queue Utilization API
  - These operate on a summarized API call: | rest splunk_server=splindex* /services/server/introspection/queues | rex field=title "(?<queue_name>^\w+)(?:\.(?<pipeline_number>\d+))?" | join outer splunk_server [| rest splunk_server=rvaparsplindex* /services/server/introspection/indexer]
- Utilization of various queues in the indexer tier
  - Each queue gets its own breakdown for each identification of issues
  - Search: index=splunk_metrics sourcetype=indexer_queues queue_name=aggQueue | eval fill_perc=round(current_size_bytes / max_size_bytes * 100,2) | stats avg(fill_perc)
- Queue Utilization Trends
  - Breakdown of queue utilization by queue over _time
  - Search: index=splunk_metrics sourcetype=indexer_queues queue_name IN (aggQueue,auditQueue,indexQueue,parsingQueue,tcpin_queue,typingQueue) | eval fill_perc=round(current_size_bytes / max_size_bytes * 100,2) | timechart span=1h avg(fill_perc) by queue_name

Search Insights

Edit | Export | ...

Total Reports	Shared Reports	Private Reports	Scheduled Reports
4,604	1,941	1,555	994

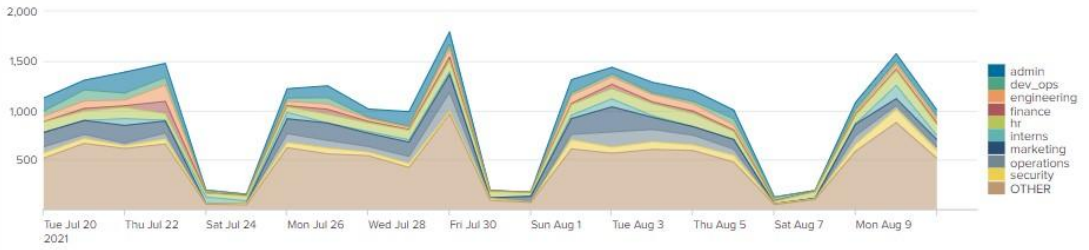
Alerts Configured for Alarming	Alerts Configured for Email	Alerts Configured for ITSI Notable Events
358	305	284

10h ago

Unique Dashboards

2,047

Dashboard Views by App



splunk> .conf21

Visualizations

This dashboard is used weekly to check for anomalies or changes in usage patterns

- Report/Alert/Dashboard Breakdowns
  - Generated through summarized API calls
  - Allows trended insight into usage of the platform
  - Report API: | rest splunk_server=* /servicesNS/-/-/saved/searches | rename "eai:acl.sharing" AS acl_sharing "eai:acl.owner" AS user
  - Dashboard API: | rest splunk_server=* /servicesNS/-/-/data/ui/views
- Dashboard Usage
  - Breaks down the top utilized apps in the system based on dashboards
  - Give great insight into what pre-built content users are using
  - Search: index="_internal" host=splsearch* sourcetype=splunk_web_access user!=”-” GET app | rex "GET /[^/]+/app/(?<app>[^/ ?]+)/" | timechart count by app where max in top10

Infrastructure Inventory

Edit | Export | ...

Select a site: All

Select a server type: Indexer

Hide Filters

System Health

Host	Splunk Version	Operating System	CPU Architecture	Core Count	Virtual Cores	% CPU Used	System Load Avg	Available Memory (GB)	% Memory Used	Last System Restart
splindex01	8.2.1	Linux	x86_64	84	84	9.15%	0.07	188.73	6.25%	08/04/21 02:26:29
splindex02	8.2.1	Linux	x86_64	84	84	7.66%	0.07	188.73	5.64%	08/04/21 02:34:33
splindex03	8.2.1	Linux	x86_64	84	84	5.91%	0.07	188.73	6.03%	08/04/21 02:29:26
splindex04	8.2.1	Linux	x86_64	84	84	6.02%	0.07	188.73	5.82%	08/04/21 02:16:18
splindex05	8.2.1	Linux	x86_64	84	84	8.26%	0.07	188.73	6.16%	08/04/21 02:38:06
splindex06	8.2.1	Linux	x86_64	84	84	4.99%	0.07	188.73	5.96%	08/04/21 02:16:54

Visualizations

This dashboard is used at need for troubleshooting

- Overall System Inventory
  - Presentation of Splunk Inventory and capacity by system
  - Filterable by location and function of the system
  - Summarized API data so that we can evaluate changes to inventory/capacity over time
  - Saved Report: index=splunk_metrics sourcetype=server_info orig_host=spl* | dedup orig_host | fields orig_host version os_name cpu_arch numberOfCores numberOfVirtualCores physicalMemoryMB startup_time | eval last_restart = strftime(startup_time,"%m/%d/%y %H:%M:%S") | fields - startup_time | eval physicalMemoryGB=round(physicalMemoryMB/1024,2) | join orig_host [search index=splunk_metrics sourcetype=resource_usage earliest=-90m@m latest=now | eval "% CPU Used"=(100-cpu_idle_pct)."%" | eval "% Memory Used"=round((mem_used/mem)*100,2)."%" | fields orig_splunk_server "% Memory Used" normalized_load_avg_1min "% CPU Used" | rename orig_splunk_server AS orig_host normalized_load_avg_1min AS "System Load Avg"] | table orig_host version os_name cpu_arch numberOfCores numberOfVirtualCores "% CPU Used" "System Load Avg" physicalMemoryGB "% Memory Used" "System Load Avg" last_restart | rename orig_host AS Host version AS "Splunk Version" os_name AS "Operating System" cpu_arch AS "CPU Architecture" numberOfCores AS "Core Count" numberOfVirtualCores AS "Virtual Cores" physicalMemoryGB AS "Available Memory (GB)" last_restart AS "Last System Restart" | sort Host

Service Health Score in ITSI

Service Decompositions - two primary questions to answer

- What's going to keep you from sleeping at night?
- What is your boss going to ask you about tomorrow?

How to do a decomposition

- Doesn't need to necessarily be as formal as a full business service decomposition
- Approach at a high level
- Be specific to your environment - your list will be unique!
- Keep it short and relevant
- Make it a living document/process
- Ensure each item can be broken down into a specific measurable KPI

Service Health Score in ITSI

Okay, now what?

- Turn your KPIs into searches
  - Make sure they return a numeric value!
- Identify thresholds
- Prioritize and assign weights to each item
- Familiarize yourself with the ITSI algorithm for health scores:

$$\text{Service Health Score} = \sum_{X=1}^{N} K_X * \frac{G_X}{\sum_{Y=1}^{N} G_Y}$$

Where:

- N = count of KPIs
- G = importance value of one KPI
- K = the score contribution of the KPI (Normal=100, Low=70, Medium=50, High=30, Critical=0)

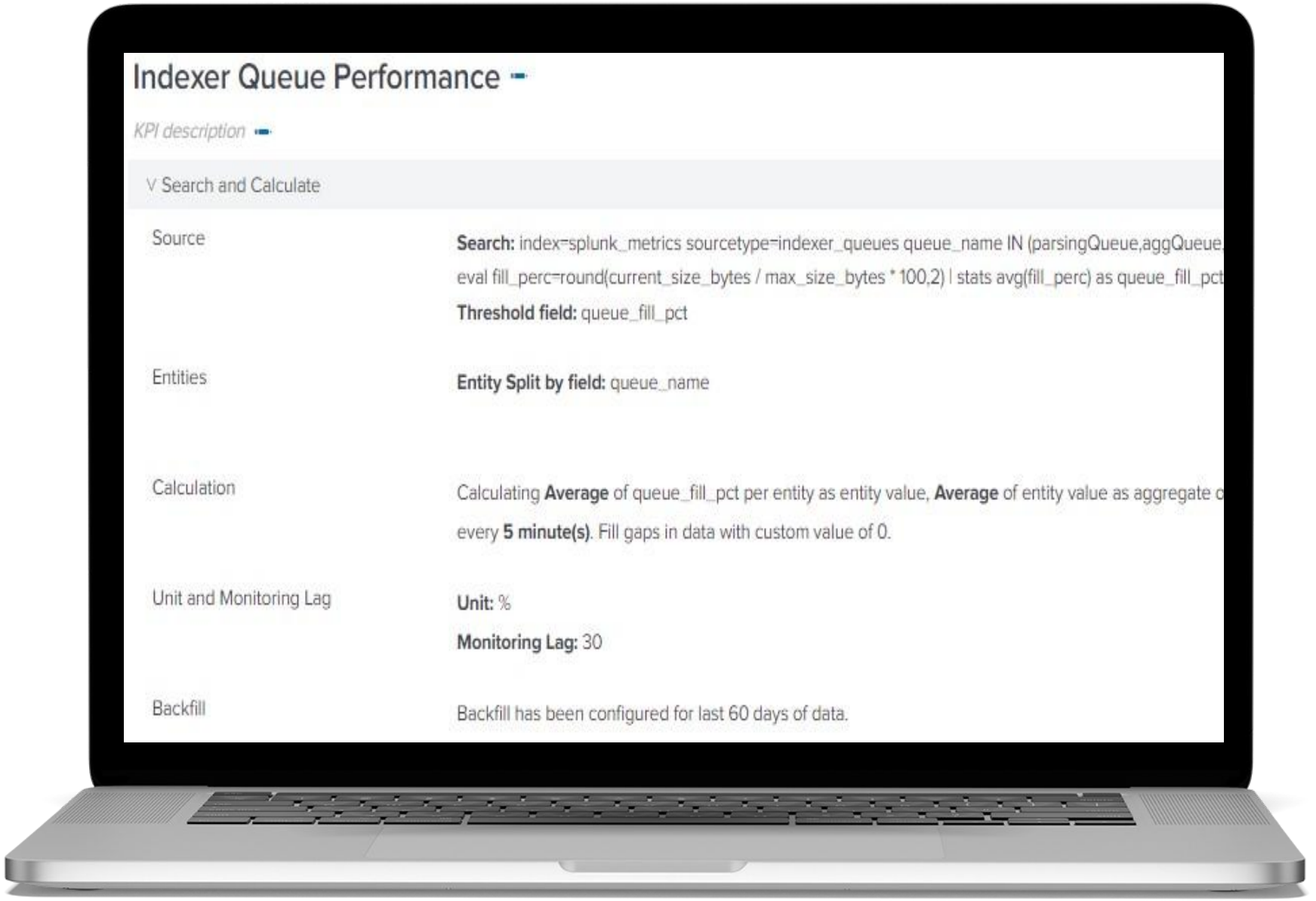
Service Health Score in ITSI

Build the service

- Build the service
- Define entities
- Define service dependencies
- Define teams

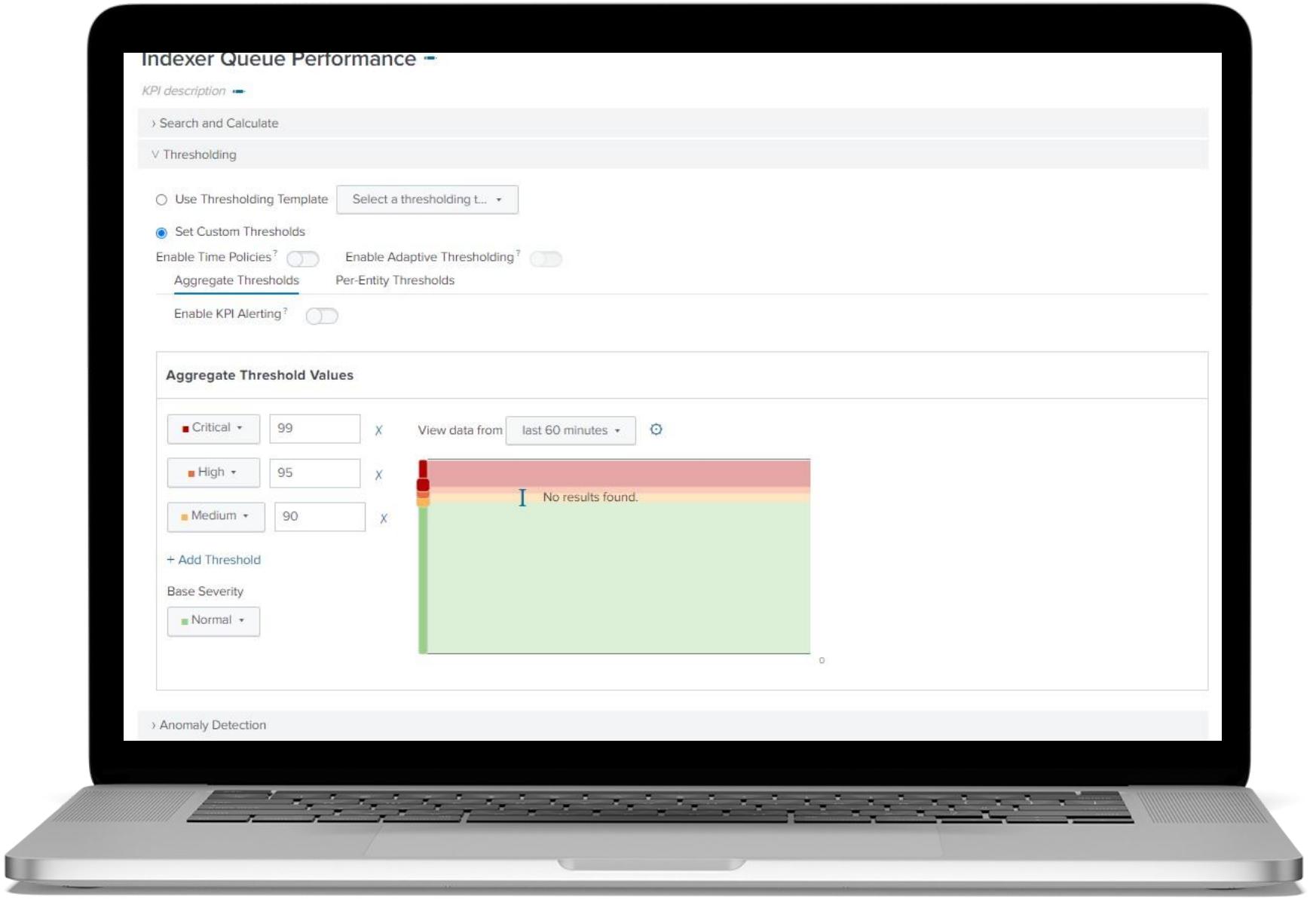
Build KPIs

- Define the search
- Define the calculation
- Define the backfill



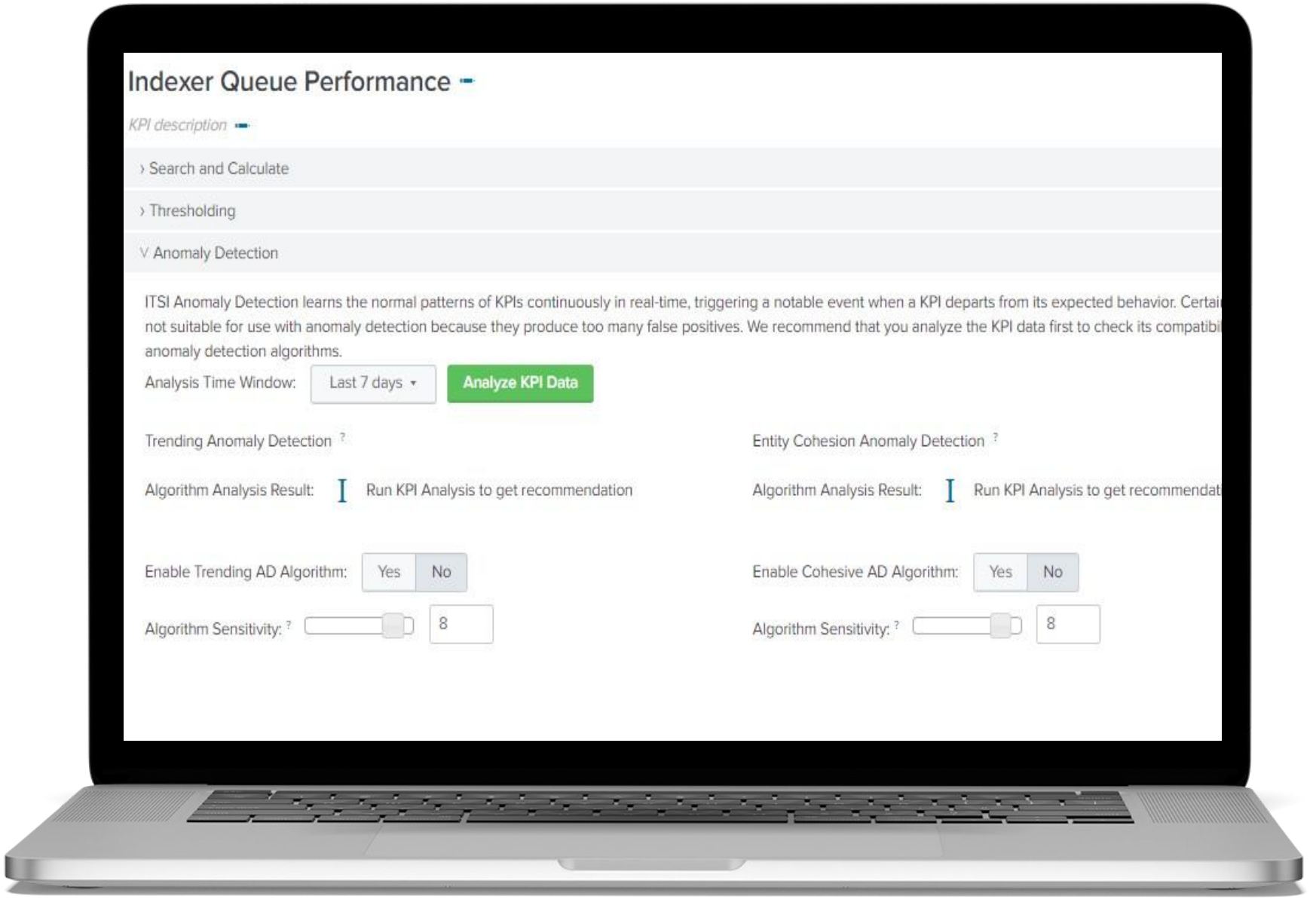
Build KPIs

- Define thresholds



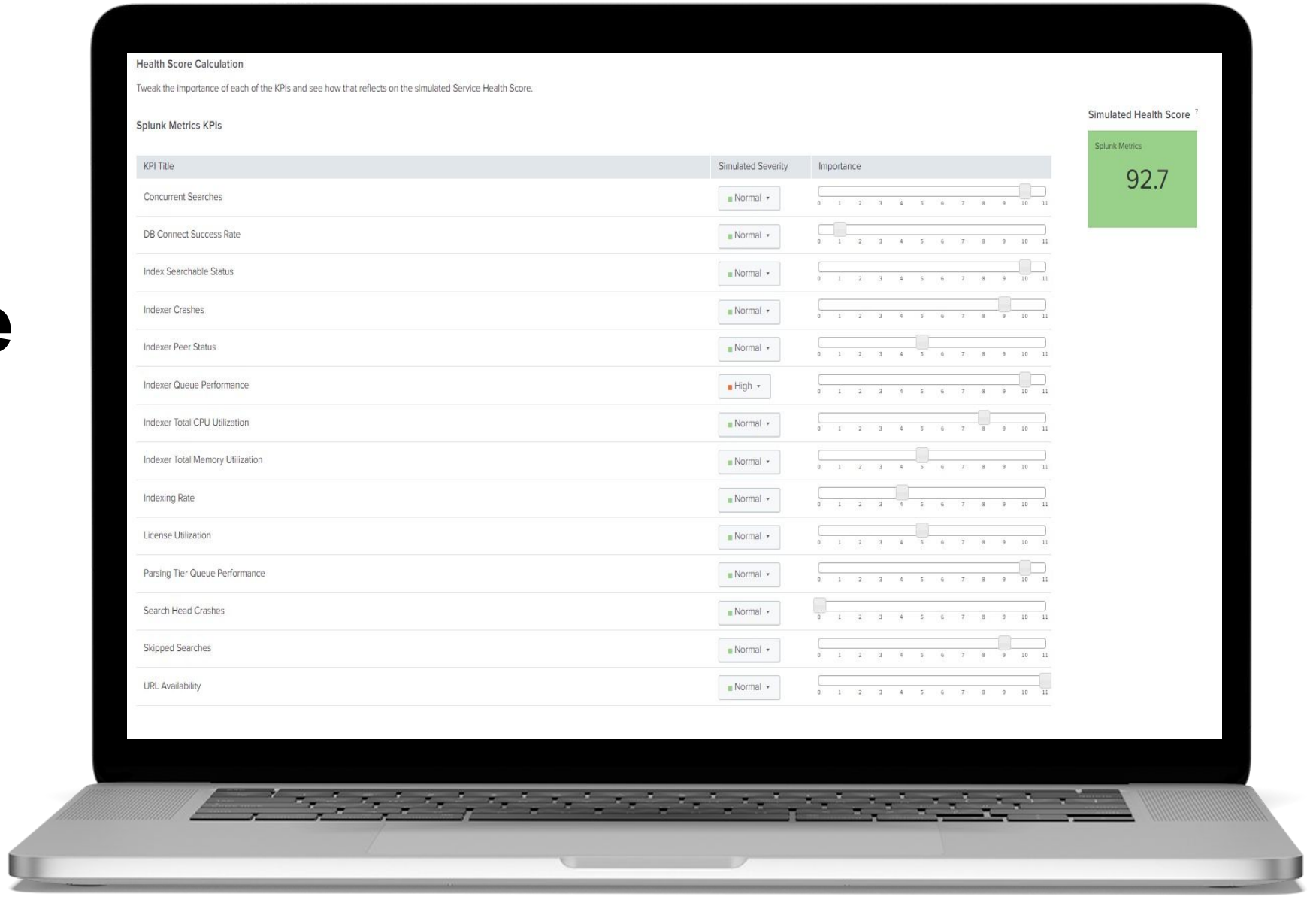
Build KPIs

- Anomaly detection

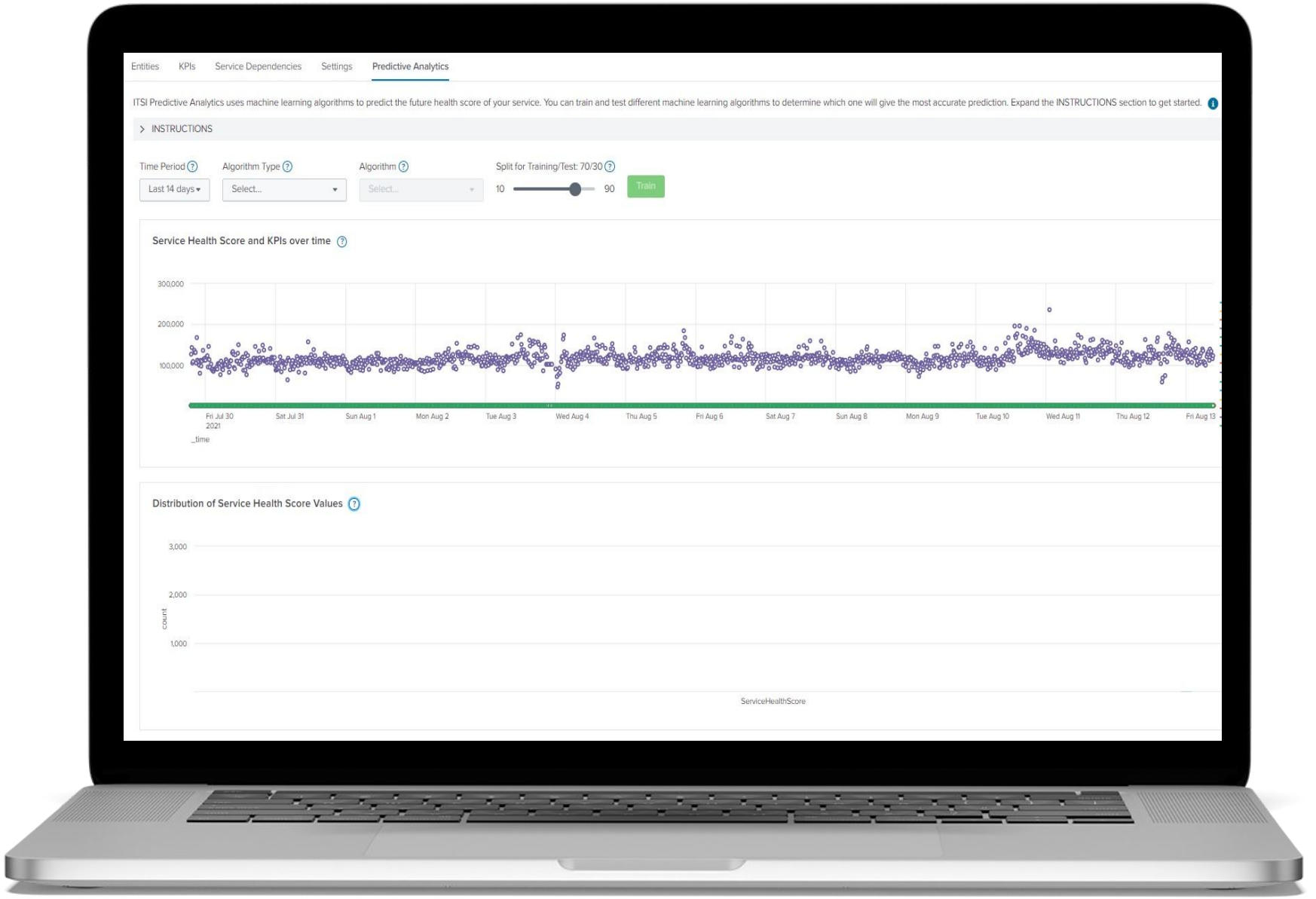


Calculate Health Score

- Remember that formula from earlier?



AI/ML Anyone?



splunk> .conf21

Service Health Score in ITSI

So, what's all in my health score?

- URL Availability
- Concurrent Searches
- Index Searchable Status
- Indexer Queue Performance
- Parsing Tier Queue Performance
- Indexer Crashes
- Skipped Searches
- Indexer Total CPU Utilization
- Indexer Total Memory Utilization
- Indexer Peer Status
- License Utilization
- Indexing Rate
- Search Head Crashes
- DB Connect Success Rate

What does that end up looking like?

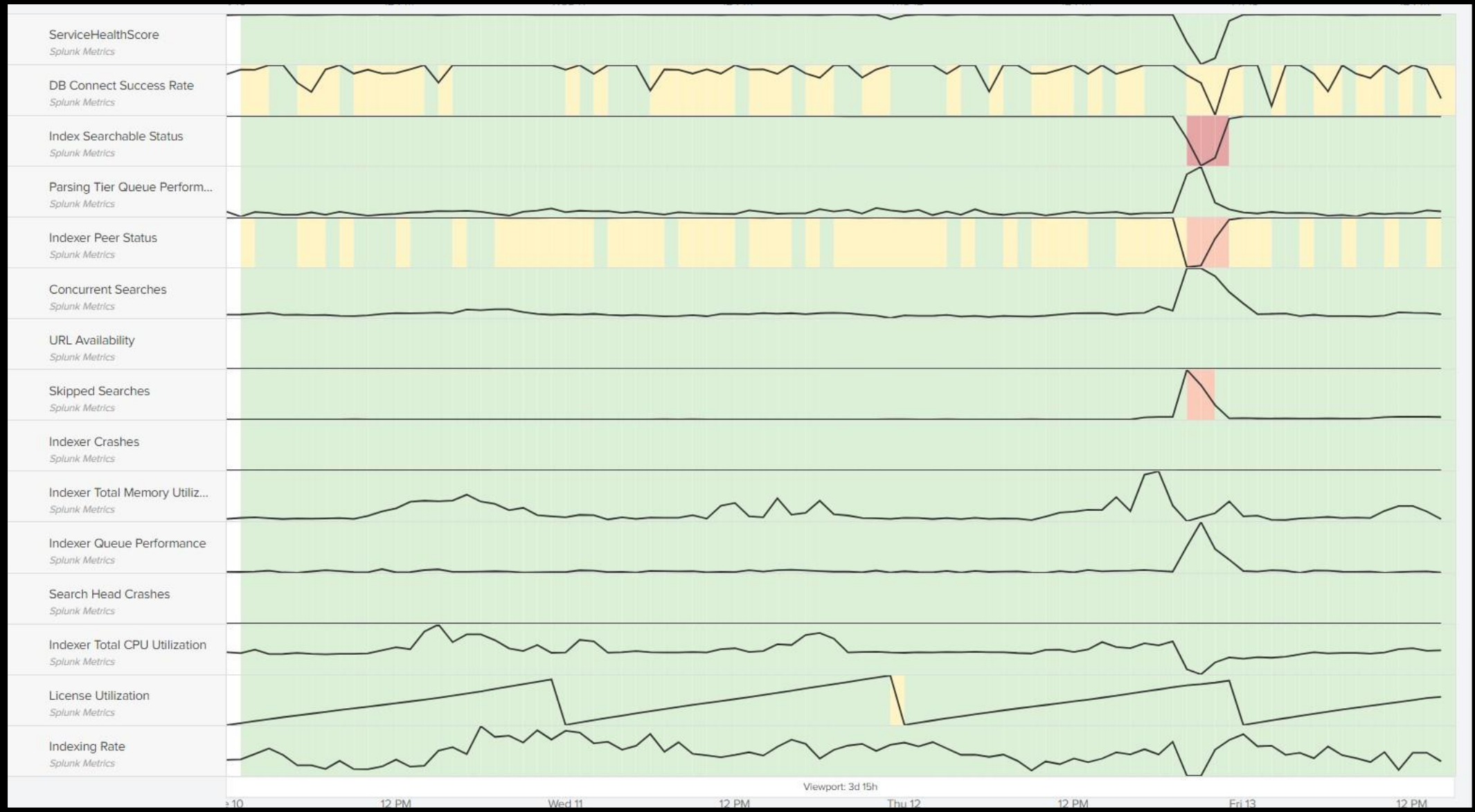
Splunk Health Score - Per KPI Values

KPI	Last Value	Severity	KPI Weight	Trend	Minimum Value	Average Value	Peak Value
URL Availability	100.000	normal	11		100.000	100.000	100.000
Concurrent Searches	51.200	normal	10		27.330	58.287	217.600
Index Searchable Status	186.000	normal	10		4.000	179.998	186.000
Indexer Queue Performance	0.000	normal	10		0.000	2.795	65.258
Parsing Tier Queue Performance	34.181	normal	10		8.788	25.328	70.954
Indexer Crashes	0.000	normal	9		0.000	0.000	0.000
Skipped Searches	0.372	normal	9		0.208	0.803	13.071
Indexer Total CPU Utilization	121.710	normal	8		0.000	122.008	411.210
Indexer Peer Status	75.000	normal	5		28.000	74.114	75.000
Indexer Total Memory Utilization	92.380	normal	5		0.000	67.330	354.220
License Utilization	6.077	normal	5		0.000	4.495	8.464
Indexing Rate	129,381.000	normal	4		58,040.000	126,142.032	177,192.000
DB Connect Success Rate	100.000	normal	1		95.260	99.764	100.000
Search Head Crashes	0.000	normal	0		0.000	0.000	0.000

Service Health Score in ITSI

This is part of our constant health view

- ITSI Health Scores
  - Presentation of each KPI with its name, weight, trend, last value and then mix/max/avg value
  - Search: index=itsi_summary a51e5df0-5b64-4b83-9a13-433896149cd9 kpi!=ServiceHealthScore | stats latest(alert_value) AS kpilastvalue latest(alert_severity) as kpiseverity latest(urgency) as weight sparkline(avg(alert_value)) AS Trend min(alert_value) AS kpiminvalue avg(alert_value) AS kpiavgvalue max(alert_value) AS kpimaxvalue by kpi | sort -weight +kpi | eval kpiavgvalue=round(kpiavgvalue,3) | eval kpilastvalue=round(kpilastvalue,3) | eval kpimaxvalue=round(kpimaxvalue,3) | eval kpiminvalue=round(kpiminvalue,3) | rename kpi AS "KPI" kpilastvalue AS "Last Value" kpiseverity AS "Severity" weight AS "KPI Weight" kpiavgvalue AS "Average Value" kpimaxvalue AS "Peak Value" kpiminvalue AS "Minimum Value"



splunk> .conf21

Service Health Score in ITSI

Great, now what do we do with it?

- Feed it into your alerting system so a NOC can let you know when scores drop
- Track it as a metric to predict future scores
- Use it to determine when platform growth is needed
- Great candidate for AI/ML

Make it a living process:

- Did you have an outage that didn’t reflect in your health score?
- Did you have an outage that too highly impacted your health score? Or vice versa?
- Regularly review everything and modify as needed

Key Takeaways

Automation

- Any tool is the right tool if you use it the right way - no right or wrong tool to use
- Automating tasks frees you up for more important work (or sleep!)
- Automated tasks can be more easily scripted or handed off to others

Visualizations

- Eliminate scope and scale issues with purpose-driven visualizations
- Build views based on what you actually need and when you need them

Service Health Scores

- Use ITSI capabilities to track platform health in metric form
- Have a review of KPIs on a scheduled basis and also as part of all post-outage work
- Stay one step ahead of problems that will bite you later

Words of Wisdom?

- Work smarter, not harder!
- Take advantage of training (even if it just means reading a lot of docs)!
- Set realistic expectations for yourself
- Set realistic expectations for others
- Use your resources!
  - Employee Resource Groups
  - Splunk Answers
  - Splunk Slack
  - Splunk Area Groups

splunk> .conf21

Thank You

Please provide feedback via the

SESSION SURVEY

splunk> .conf21