



Dashboards & Visualizations: What's New

Nicholas Filippi | Product Management, Splunk

Patrick Ogdin | Product Management, Splunk

September 2017 | Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

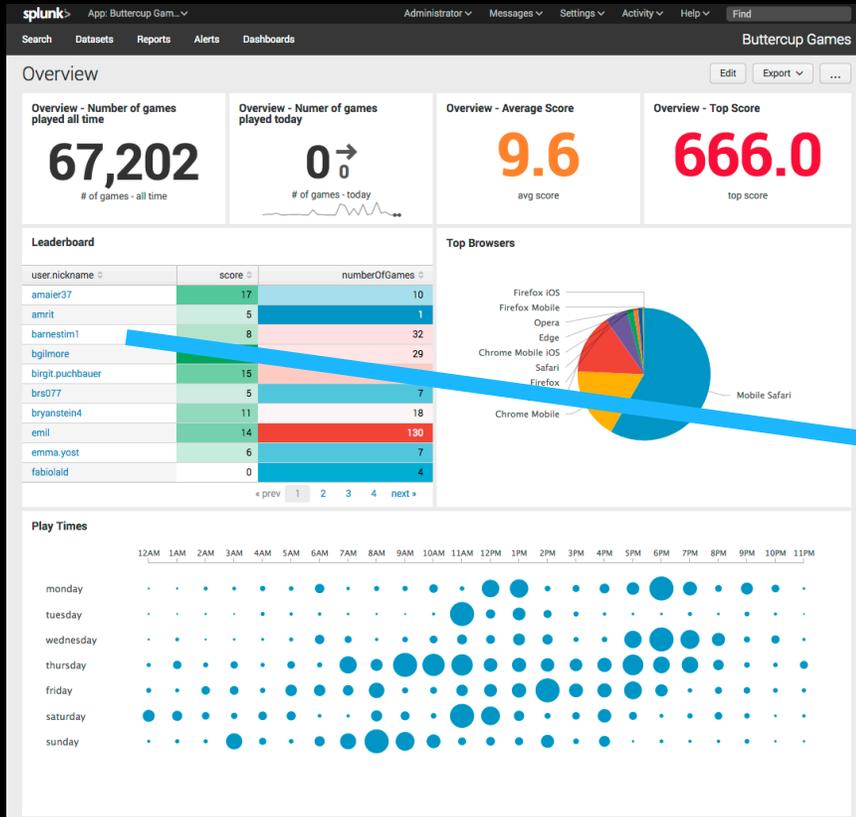
The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.

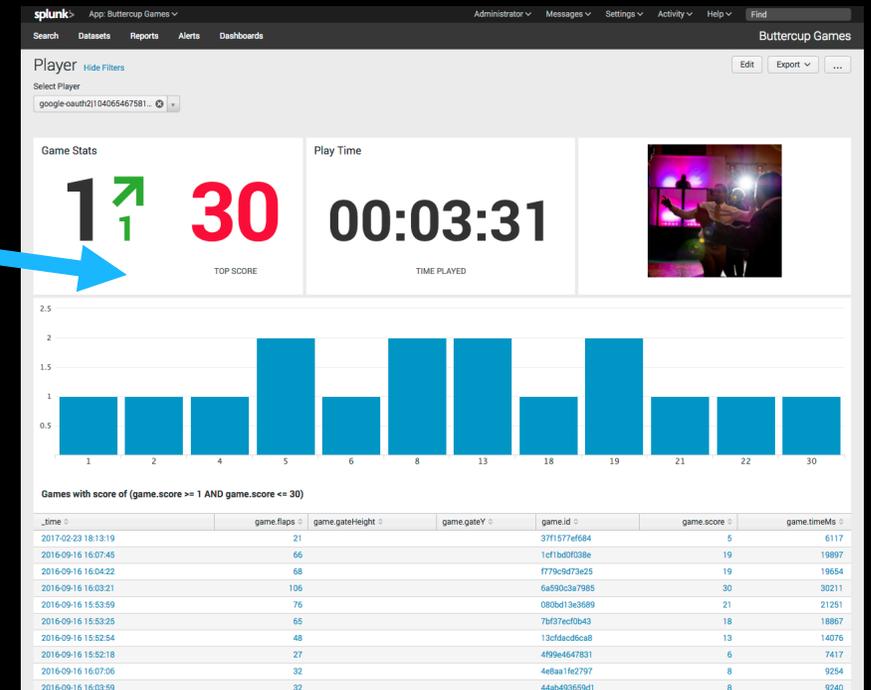
Dashboards

Drilldown Editor
Event Annotations
SearchBar Integration

Dashboard Drilldown (xml config)



Optionally configure drilldown event to direct users to another dashboard, passing context in the form of token variables (XML only)



```
<drilldown>
  <link target="_blank">/app/buttercup_games/player?form.player=$row.user.user_id$</link>
</drilldown>
```

Drilldown Editor

build interactivity in your dashboard without learning XML

► Objective:

- Promote more users to customize the drilldown experience
- Remove requirement to learn XML
- Default “Drill to Search” is often not the preferred behavior

► Key Details

- Introduce new “Edit Drilldown” configuration dialog
- Supports common use cases
 - Link to search
 - Link to dashboard
 - Link to report
 - Link to custom URL
 - Manage tokens
- No change to Simple XML syntax
- Disable drilldown by default
 - *only affects newly created content

Drilldown Editor
✕

On Click Link to dashboard ▾

App No action

Dashboard ✓ Link to dashboard

Open Link to report

Link to custom URL

▼ Advanced Manage tokens on this dashboard

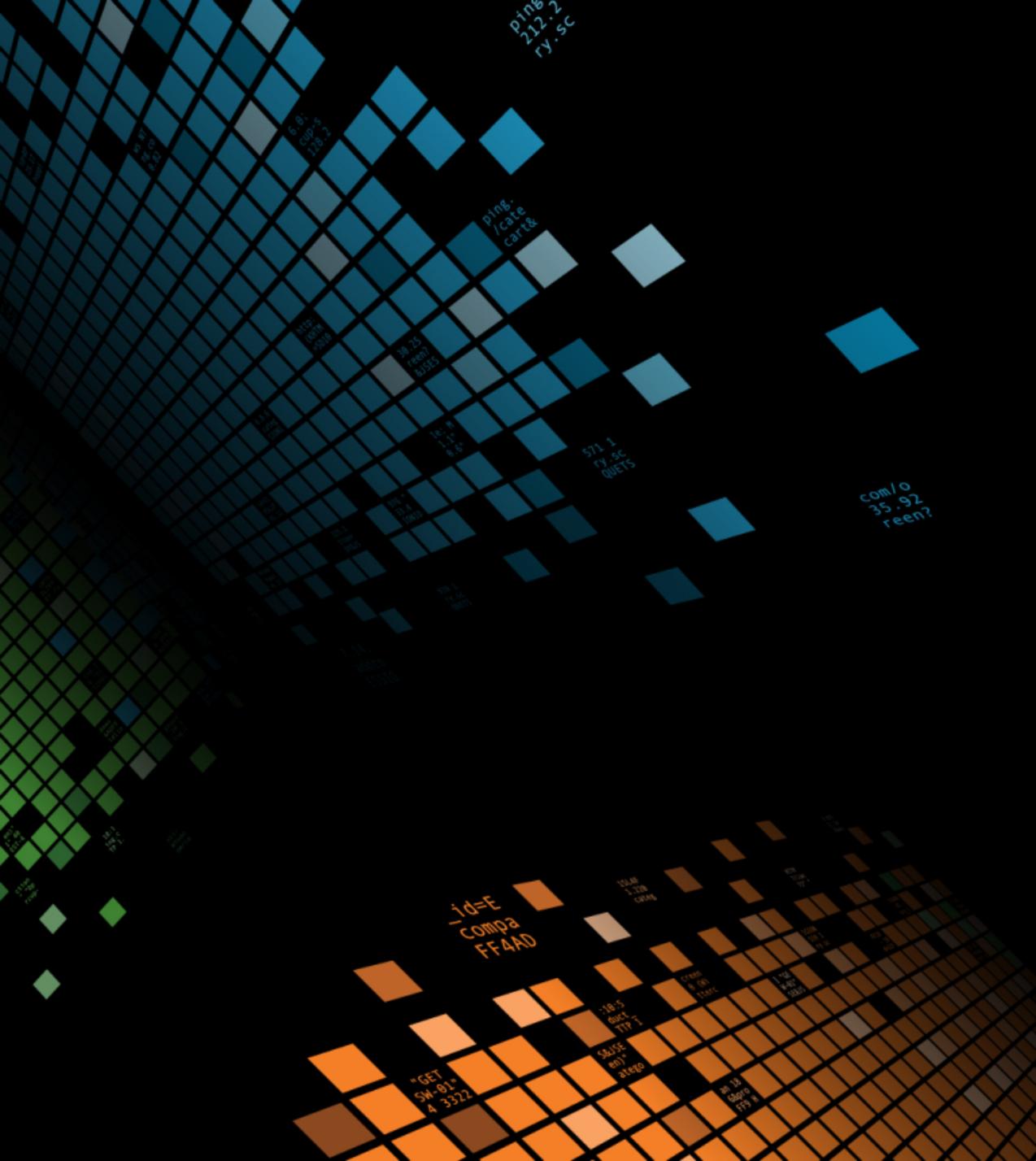
Parameters Enable in-page drilldown actions

[+ Add new](#)

Use parameters to set token values in the target dashboard.
For example, form.host = \$click.value2\$ or host = \$click.value2\$ [Learn more](#)

Preview URL /app/search/cities

Cancel
Apply



Demo

Drilldown Editor

Drilldown UI Editor – Surface Area

Supported via UI Editor

	Use Case
1	No action
2	Link to search <ul style="list-style-type: none"> Both default (uses intentions parser) and custom search string
3	Link to dashboard <ul style="list-style-type: none"> Same/different app context; pass tokens to target dashboard
4	Link to report <ul style="list-style-type: none"> Same/different app context
5	Link to custom URL <ul style="list-style-type: none"> Pass tokens to target URL
6	In-page interactivity (via token management) <ul style="list-style-type: none"> Set/Unset/Eval tokens on the page
7	Conditional field drilldown
8	Multiple Actions

Dashboard Search Bar

Improved search editing experience on dashboards

▶ Integrated SearchBar Component within Dashboards

- Add Panel & Edit Search Workflows

▶ Improved Productivity & Consistency

▶ Leverage Functionality

- Syntax Highlighting
- Keyboard Shortcuts
- Compact Search Assistant

Edit Search

Title

Search String

```
((index="main") (sourcetype="products_download"))
| fields "activity", "Platform", "salesforce_id", "version", "raw", "_time"
| fields - "_raw"
| eval Date=strftime('_time', "%m/%e/%Y")
| rename "activity" AS "Products"
| replace "Download" with "Enterprise" in "Products"
| lookup "sfdc" "salesforce_id" OUTPUT "AccountName"
| fields - "salesforce_id"
| dedup "AccountName" "Platform" "version"
| stats
```

Time Range

Auto Refresh Delay?

Refresh Indicator

Run Search

stats count	Command History
stats count by action, host	Command History
stats count by host	Command History
stats count by src_ip, dest_ip, dest_port	Command History
stats first(resident_on) as resident_on	Command History

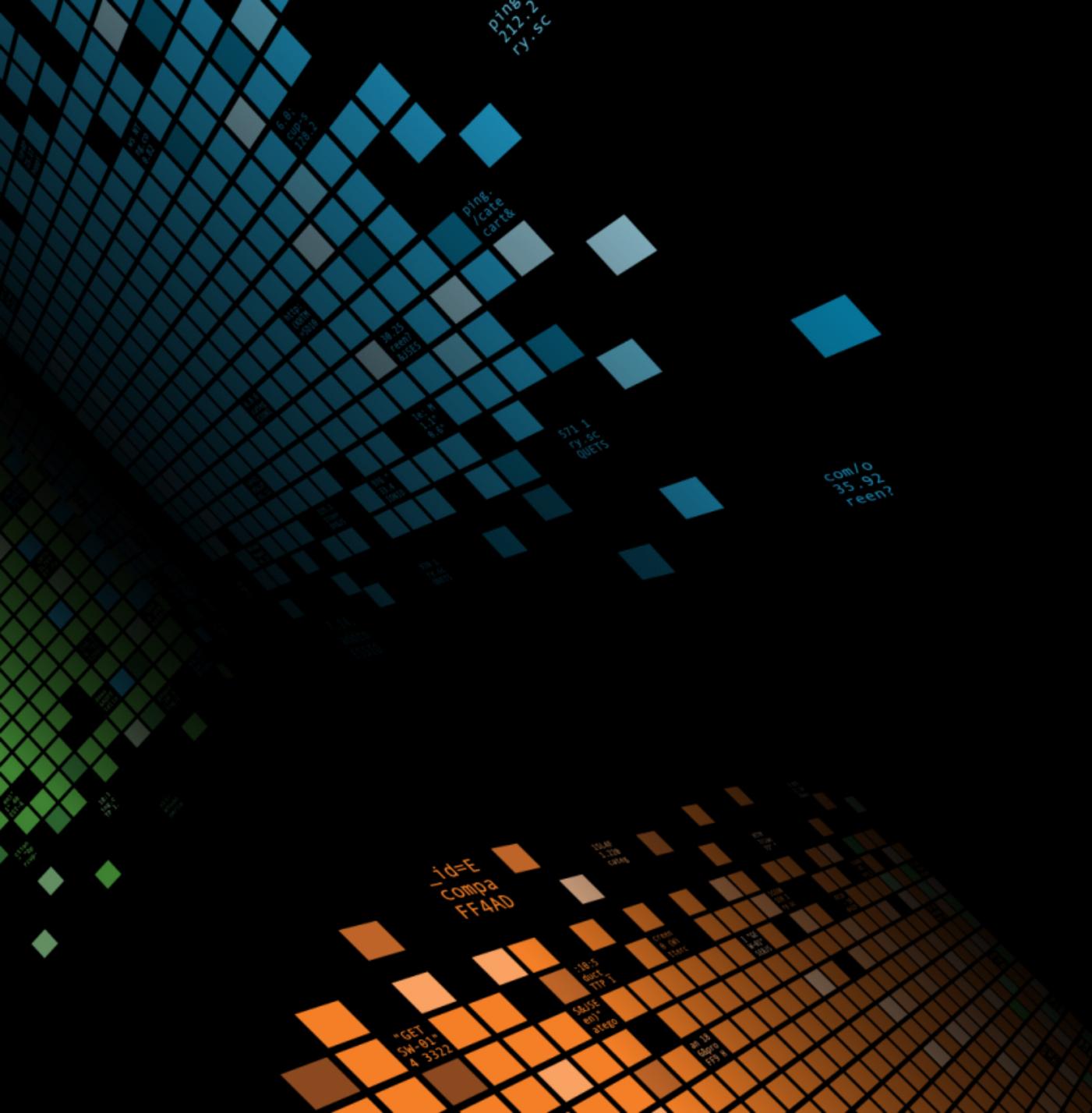
stats

[Learn More](#)

Provides statistics, grouped optionally by field.

Example:
sourcetype=access_combined | top limit=100 referer_domain | stats sum(count)

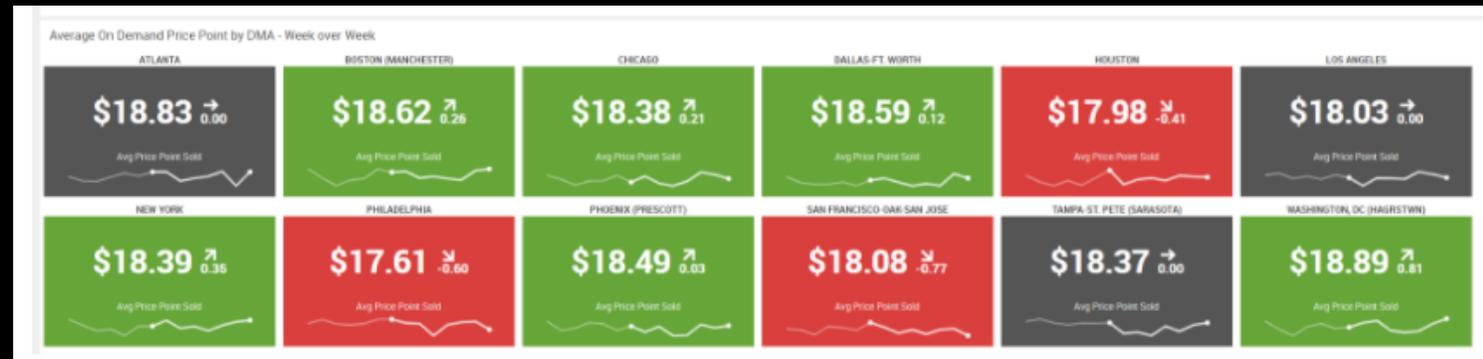
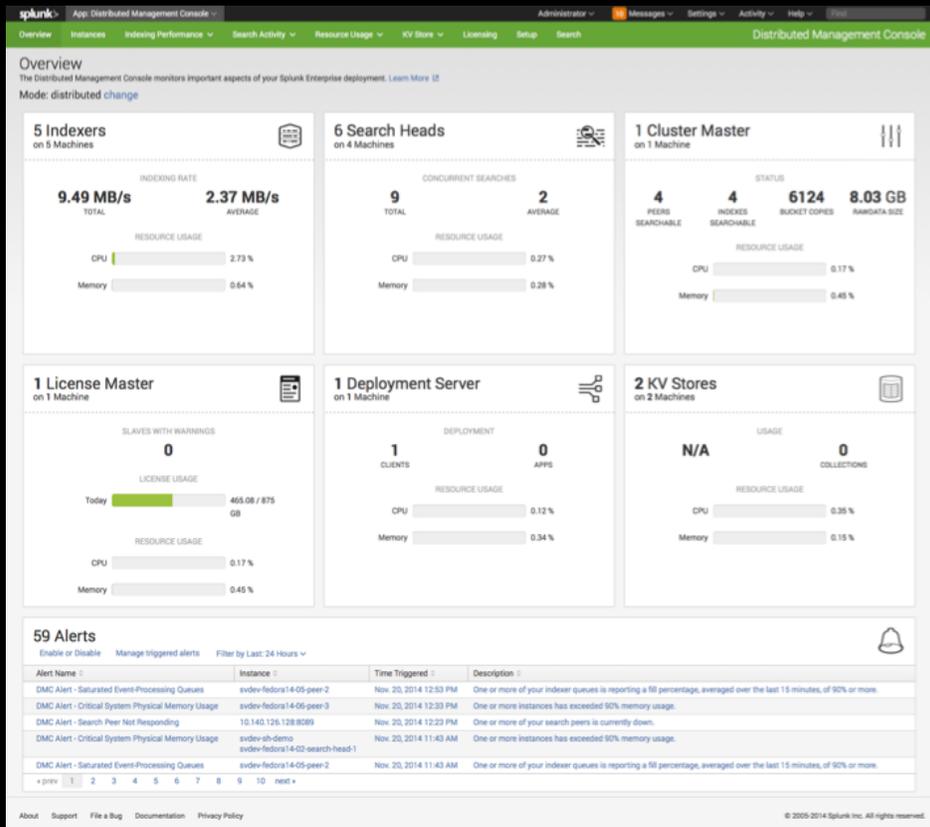
Cancel Apply



Visualizations

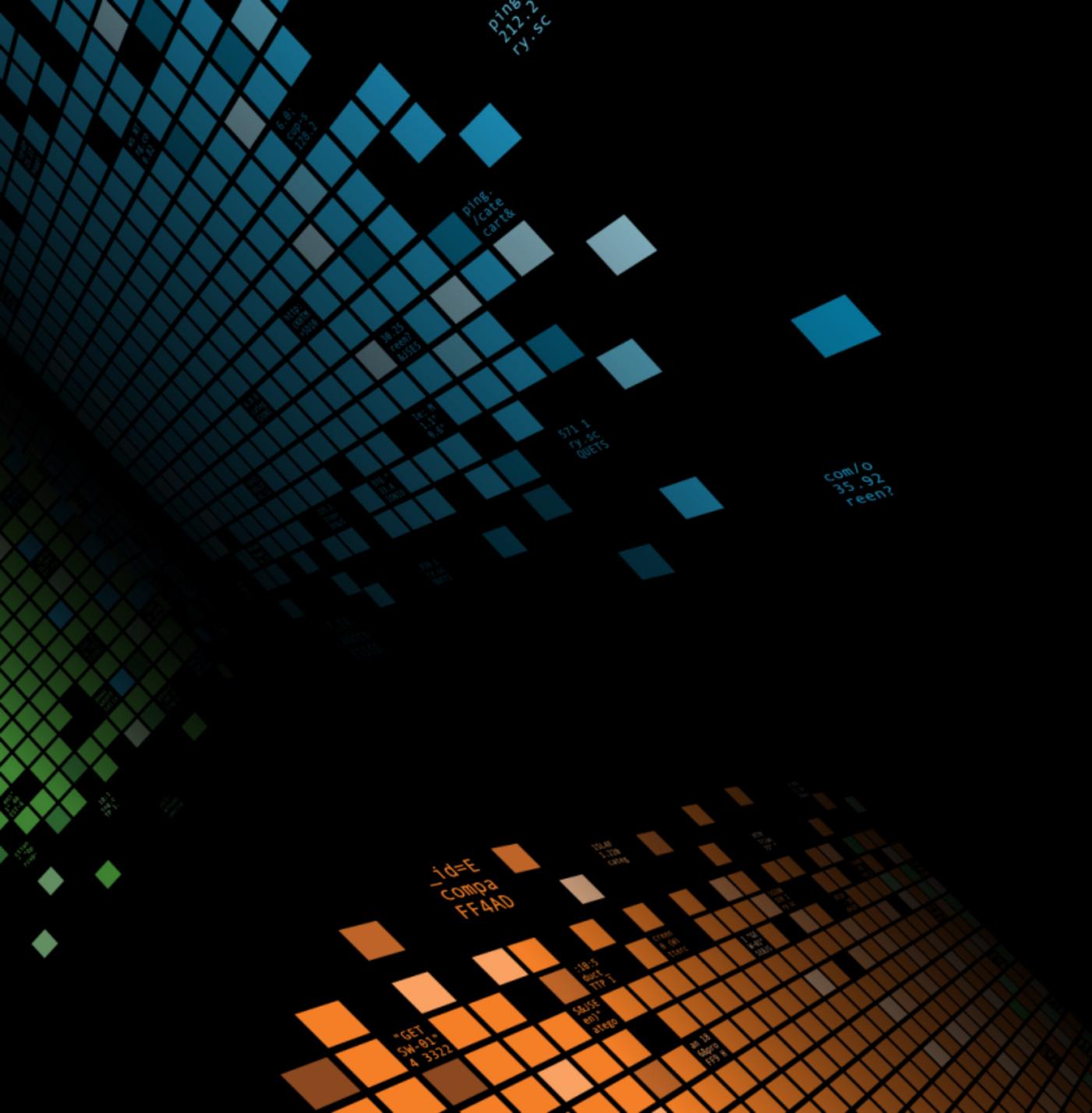
Trellis Layout
Result Truncation
Actions for Reports

Before & After



► Benefits

- Alternative is to use series of similar queries, which causes unnecessary load on the system
- Single visualization might hide relevant outliers by over-aggregating values
- Since values often change over time, Trellis Layout can dynamically show all values that are present in the selected time range



Demo

Trellis Layout

Thank You

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

splunk> .conf2017

Appendix A: Drilldown Editor

Drilldown UI Editor – Surface Area

Supported via UI
Editor

	Use Case
1	No action
2	Link to search <ul style="list-style-type: none"> Both default (uses intentions parser) and custom search string
3	Link to dashboard <ul style="list-style-type: none"> Same/different app context; pass tokens to target dashboard
4	Link to report <ul style="list-style-type: none"> Same/different app context
5	Link to custom URL <ul style="list-style-type: none"> Pass tokens to target URL
6	In-page interactivity (via token management) <ul style="list-style-type: none"> Set/Unset/Eval tokens on the page
7	Conditional field drilldown
8	Multiple Actions

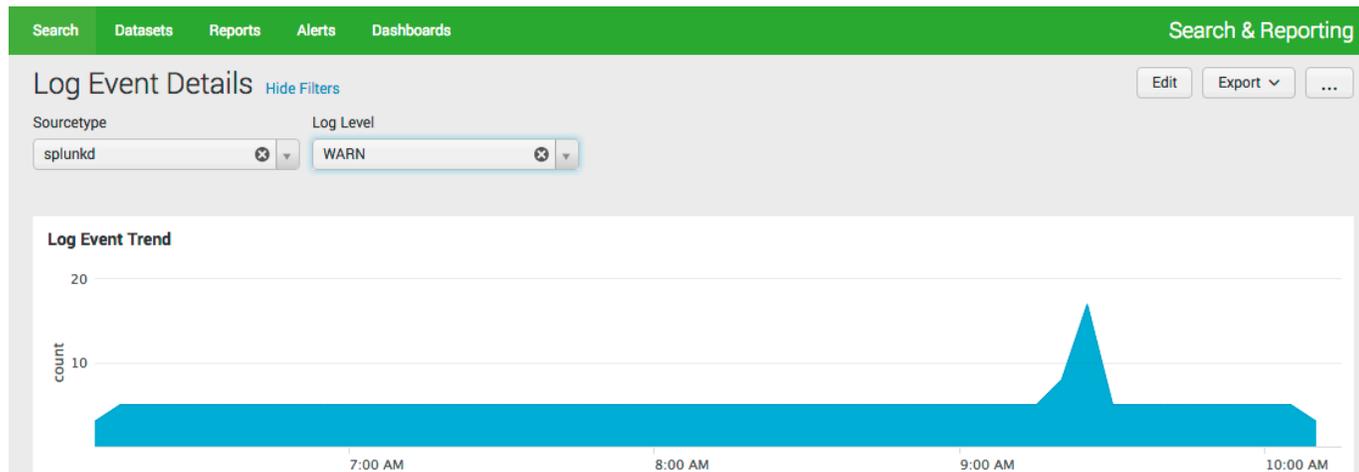
Conditional Field Drilldown (XML Only)

Search string

```
index=_internal (log_level=* log_level!="WARNING")
| chart count over sourcetype by log_level
| addtotals
| sort -ERROR
```

Log Events by Sourcetype

	sourcetype	ERROR	INFO	WARN	Total
1	splunkd	18	420955	1520	422493
2	splunk_web_service	1	510	0	511
3	scheduler	0	8341	0	8341
4	splunkd_conf	0	2	0	2



Conditional Field Drilldown (XML Only)

Happy
Case!

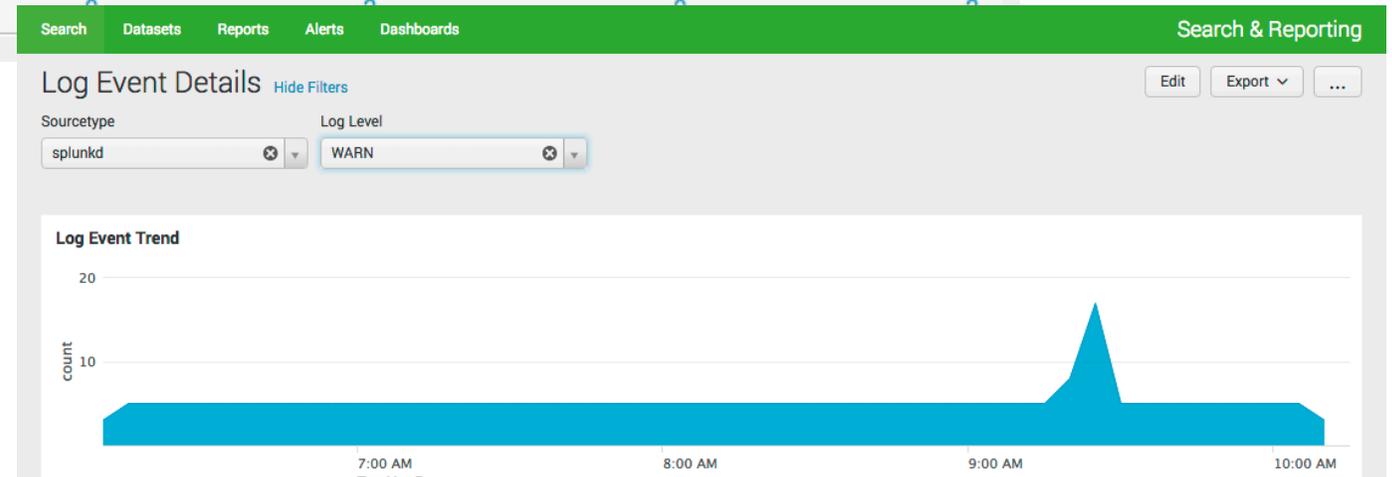
Search string

```
index=_internal (log_level=* log_level!="WARNING")
| chart count over sourcetype by log_level
| addtotals
| sort -ERROR
```

Log Events by Sourcetype

	sourcetype	ERROR	INFO	WARN	Total
1	splunkd	18	420955	1520	422493
2	splunk_web_service	0	510	0	511
3	scheduler	0	8341	0	8341
4	splunkd_conf	0	0	0	0

If user clicks on "18",
then direct user to "Log Events Details"
dashboard and pass sourcetype='splunkd' &
log_level="ERROR"



```
<drilldown>
  <link target="_blank">/app/search/log_event_details?form.sourcetype=$row.sourcetype&form.log_level=$click.name2$</link>
</drilldown>
```

Conditional Field Drilldown (XML Only)

Edge Case!

Search string

```
index=_internal (log_level=* log_level!="WARNING")
| chart count over sourcetype by log_level
| addtotals
| sort -ERROR
```

Log Events by Sourcetype

	sourcetype	ERROR	INFO	WARN	Total
1	splunkd	18	420955	1520	422493
2	splunk_web_service	1	510	0	
3	scheduler	0	8341	0	8341
4	splunkd_conf	0	2	0	2

Current Behavior

If user clicks on “splunkd”,

then direct to “Log Event Details” and pass sourcetype='splunkd' &

log_level='sourcetype'

If user clicks on “422493”,

then direct to “Log Event Details” and pass sourcetype='splunkd' & **log_level='Total'**

Conditional Field Drilldown (XML Only)

Edge Case!

Search string

```
index=_internal (log_level=* log_level!="WARNING")
| chart count over sourcetype by log_level
| addtotals
| sort -ERROR
```

Log Events by Sourcetype

	sourcetype	ERROR	INFO	WARN	Total
1	splunkd	18	420955	1520	422493
2	splunk_web_service	1	510	0	
3	scheduler	0	8341	0	8341
4	splunkd_conf	0	2	0	2

Desired Behavior

If user clicks on “splunkd”,

then direct to “Log Event Details” and pass sourcetype='splunkd' & log_level='*'

If user clicks on “422493”,

then direct to “Log Event Details” and pass sourcetype='splunkd' & log_level='*'

Conditional Field Drilldown (XML Only)

Solution!

Search string

```
index=_internal (log_level=* log_level!="WARNING")
| chart count over sourcetype by log_level
| addtotals
| sort -ERROR
```

Log Events by Sourcetype

	sourcetype	ERROR	INFO	WARN	Total
1	splunkd	18	420955	1520	422493
2	splunk_web_service	1	510	0	
3	scheduler	0	8341	0	8341
4	splunkd_conf	0	2	0	2

```
<drilldown>
  <condition field="sourcetype">
    <link target="_blank">/app/search/test_bug?form.sourcetype=$row.sourcetype&form.log_level=*</link>
  </condition>
  <condition field="Total">
    <link target="_blank">/app/search/test_bug?form.sourcetype=$row.sourcetype&form.log_level=*</link>
  </condition>
  <condition>
    <link target="_blank">/app/search/test_bug?form.sourcetype=$row.sourcetype&form.log_level=$click.name2$</link>
  </condition>
</drilldown>
```

Multiple Actions (XML Only)

Purchase History	
Customer ▾	Amount ▾
Susan Smith - ID:12345	\$400
John Stand - ID:67890	\$300

Desired Behavior

If user clicks on “Susan Smith – ID:12345”
then direct to “Customer Details” dashboard and pass ID=‘12345’
(effectively, extract the ID from the customer field, and use that token)

Multiple Actions (XML Only)

Purchase History	
Customer ↕	Amount ↕
Susan Smith - ID:12345	\$400
John Stand - ID:67890	\$300

```
<drilldown>
  <eval token="customer_id">substr($row.Customer$, -5)</eval>
  <link target="_blank">customer_details?form.customer_id=$customer_id$</link>
</drilldown>
```

Appendix B: Event Annotations

How To Configure Event Annotations?

- ▶ Dashboard XML only
- ▶ Driven by a secondary search
 - <search type="annotation">
 - If <earliest> and <latest> are not specified, then it will use the primary search
- ▶ Supported fields in search results
 - `__time` – required field to overlay on a time-series chart
 - `annotation_label` – optional field for display in the tooltip
 - `annotation_category` – optional field to differentiate types of annotations, by color
 - `annotation_color` – optional field to override color *(recommended to use charting `categoryColors` instead)
- ▶ XML charting options
 - `charting.annotation.categoryColors` – Override color palette for annotation categories

Scenario #1: Basic Overlay w/ Event Annotations

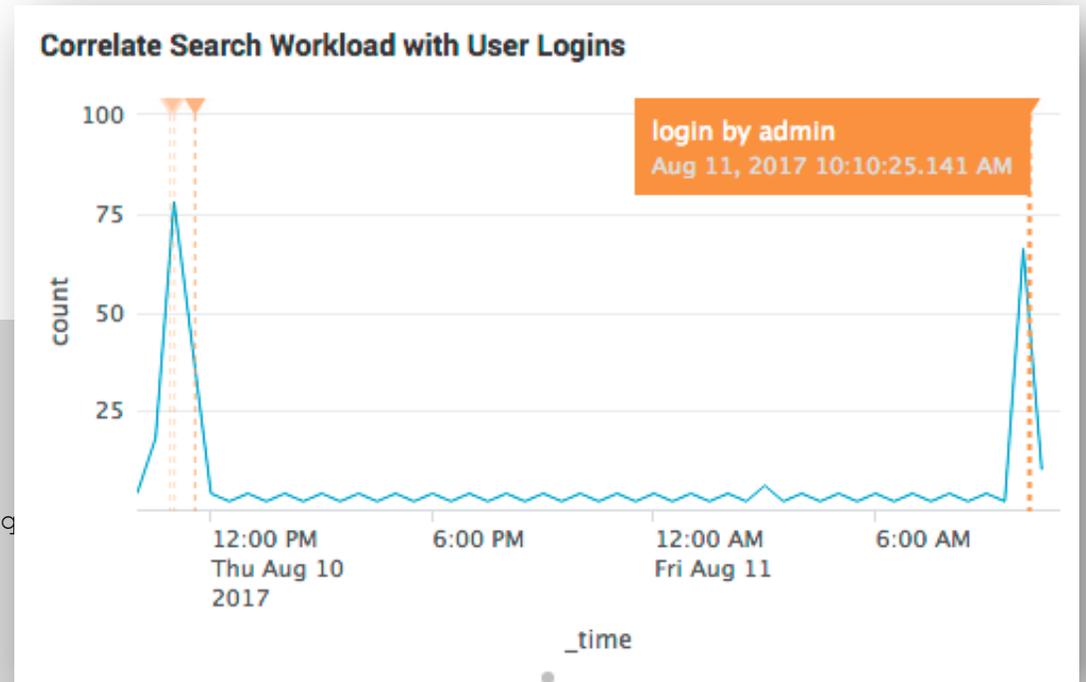
Example: Correlate search workload with user activity. Set label to include the login username.

```
<chart>
  <title>Correlate Search Workload with User Logins</title>

  <!-- Base search that drives the visualization -->
  <search>
    <query>index=_audit action=search result_count="*" | timechart count</query>
    <earliest>-24h@h</earliest>
    <latest>now</latest>
  </search>

  <!-- Secondary search that drives the annotations -->
  <search type="annotation">
    <query>index=_audit action="login attempt" | eval annotation_label = "login
  by " . user</query>
    <earliest>-24h@h</earliest>
    <latest>now</latest>
  </search>

  <option name="charting.chart">line</option>
  <option name="charting.drilldown">none</option>
  <option name="charting.legend.placement">none</option>
  <option name="charting.lineWidth">1</option>
</chart>
```



Scenario #2: Multiple Categories of Event Annotations

Example: Correlate search run time with various warning and error log events. Use category to differentiate log level, and label to display the log message.

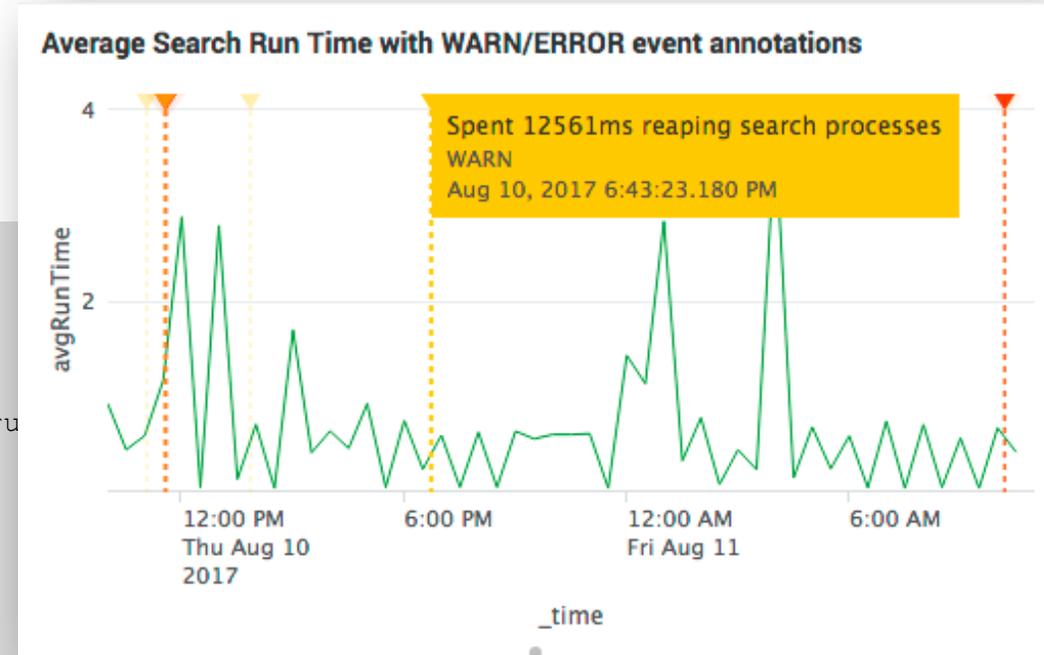
```
<chart>
  <title>Average Search Run Time with WARN/ERROR event annotations</title>

  <!-- Base search that drives the visualization -->
  <search>
    <query>index=_audit action=search result_count="*" | timechart avg(total_run
avgRunTime</query>
    <earliest>-24h@h</earliest>
    <latest>now</latest>
  </search>

  <!-- Secondary search that drives the annotations -->
  <search type="annotation">
    <query>index=_internal (log_level="WARN" OR log_level="ERROR") | eval annotation_label = message| eval
annotation_category = log_level</query>
    <earliest>-24h@h</earliest>
    <latest>now</latest>
  </search>

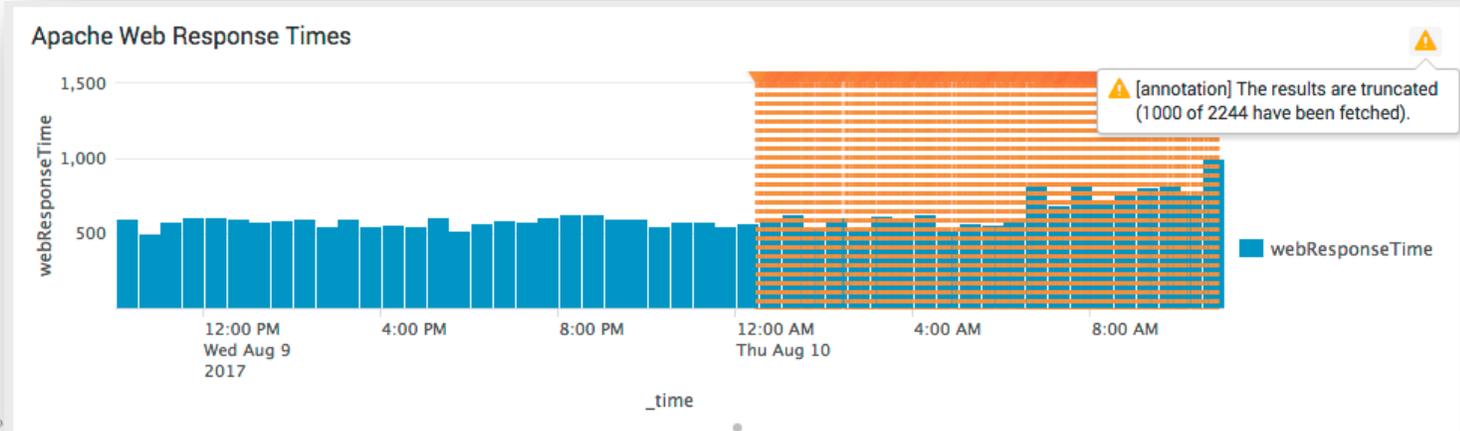
  <!-- Customize the event annotation colors based on category name -->
  <option name="charting.annotation.categoryColors">{"ERROR":"0xff3300","WARN":"0xffcc00"}</option>

  <option name="charting.chart">line</option>
  <option name="charting.drilldown">none</option>
  <option name="charting.legend.placement">none</option>
  <option name="charting.lineWidth">1</option>
  <option name="charting.seriesColors">[0x339933]</option>
</chart>
```



Important Details

- ▶ Currently, integrated with dashboard XML only
- ▶ Does not yet support “user” annotations
 - Search-driven annotations only
- ▶ Supports discreet events
 - Does not support “duration” event (ex. maintenance windows, etc)
- ▶ Performance
 - Does run an additional search on dashboard load time
- ▶ Result limit of 1000



Thank You

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

splunk> .conf2017