Best Practices and Better Practices

Burch

Sales Engineer @ Splunk, Inc.

splunk>

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make.

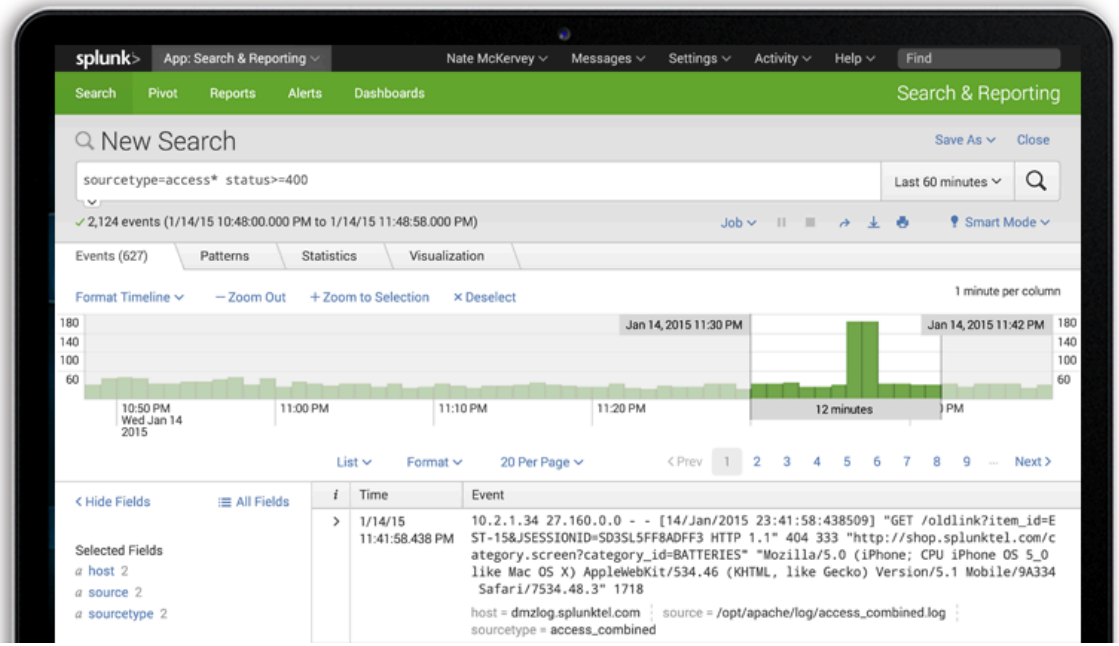
In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the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

Goal: Get Gooder!

- Interact
  - Ask Questions
  - Offer better ideas
  - Don't be "that guy"
- 2 Much Content
  - Download & reference this
  - Not covering everything
- This deck is updated

Agenda

1. Who are we?
2. References
3. Resources
4. Searching
5. Admin
6. Next Steps



Best Practices

Who are we?

What's a Burch?

- Sales Engineer in Boston
- Education
  - CS @ Boston University
  - MBA @ Northeastern University
- Splunk Customer
  - Middleware for 8 years (+splunk)
  - Splunk Admin for 1.5 years (splunk 4.3+)
- Certs: Knowledge, Admin, Architect
- @Splunk for 10 months
- AUTOmatic App

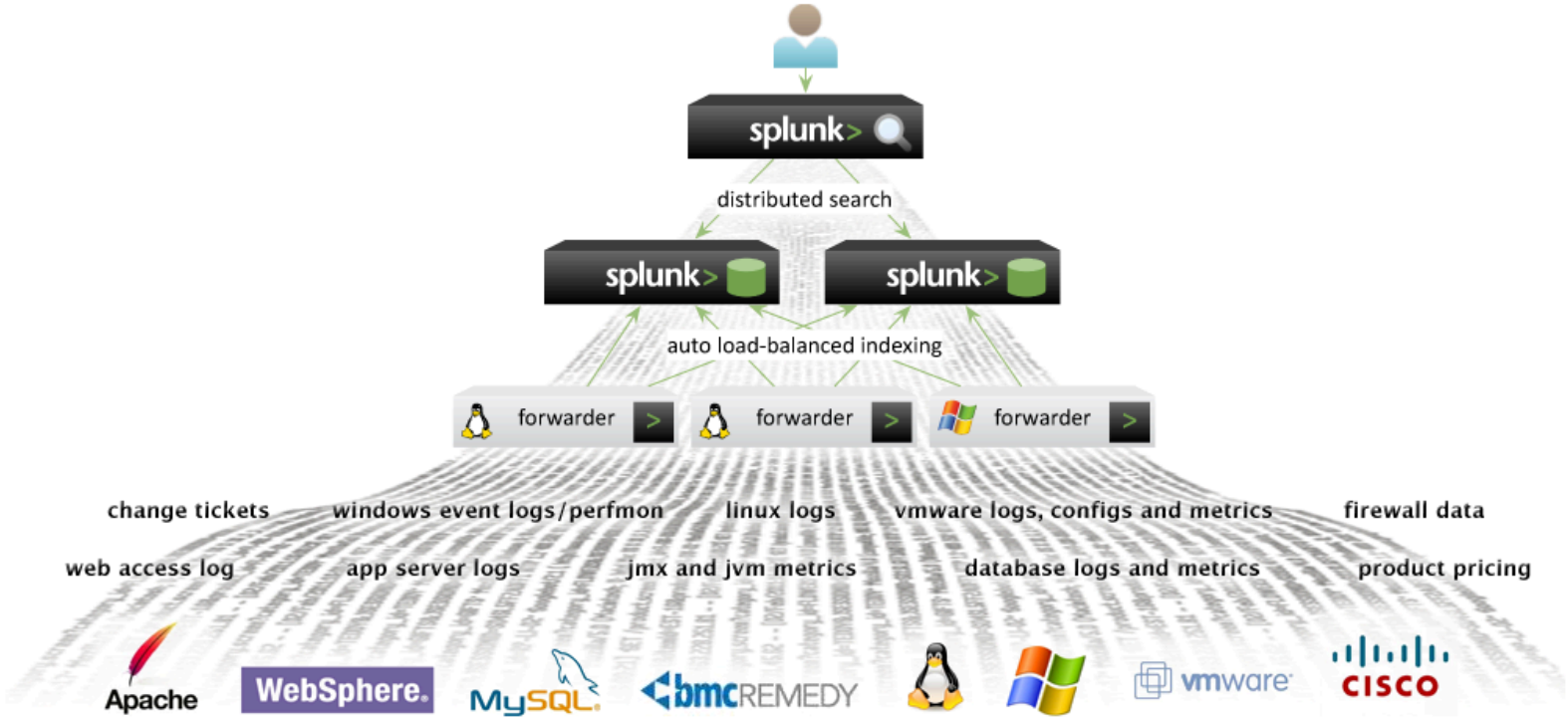
About you

- Name
- User?
- Power User?
- Admin?
- Groupie?

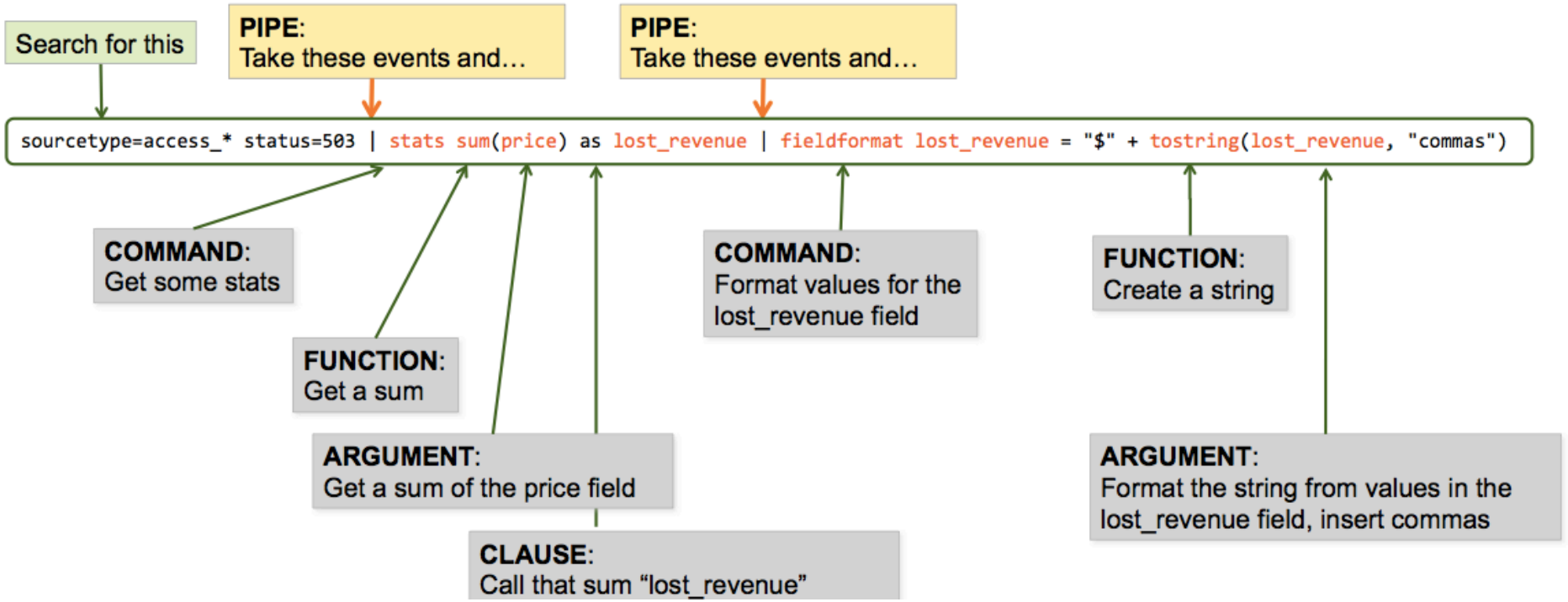
Best Practices

References

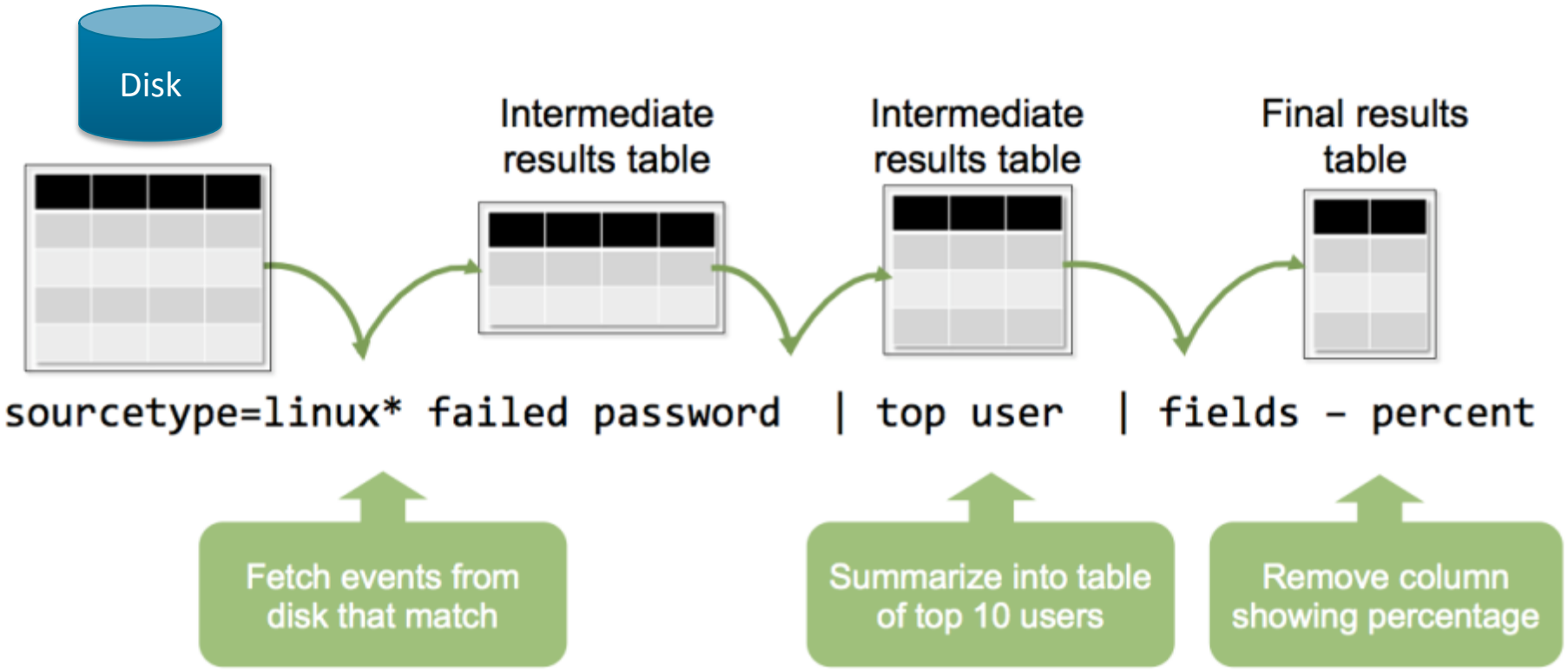
Splunk Review



Search Syntax Components



Anatomy of a Search



Acceleration Options

	Summary Indexing	Report Acceleration	Data Model Acceleration
Benefits	• Save disk space • Control on impact to system	• Backfill • Simple	• Backfill • Simple • Extensible • Search Agnostic
Limits	• Gaps • Intellectually difficult • Backfill	• Requires transforming • Specific to search	• Massive if misused

- Great article: Search documentation for “report acceleration”
- New Feature: Archive to Hadoop

Best Practices

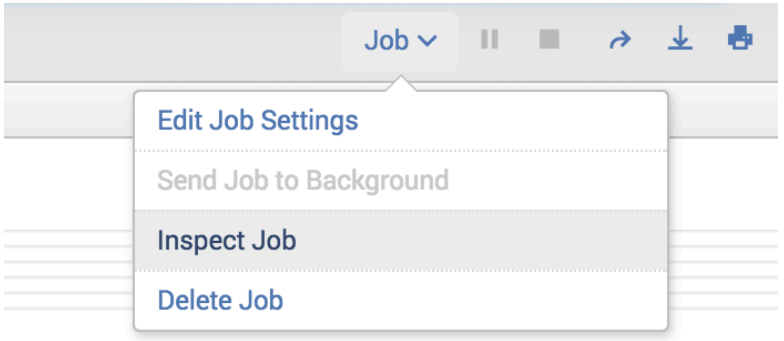
Resources

Reference

- Free Search Tutorial -> docs.splunk.com -> Search Tutorial
- Splunk Documentation -> docs.splunk.com
- Community Q&A -> answers.splunk.com
- Community Tips & Tricks -> wiki.splunk.com
- Splunk! The Book -> http://www.splunk.com/goto/book
- Apps -> splunkbase.splunk.com

Job Inspector

- Job Inspector
  - docs.splunk.com “Search Job Inspector”

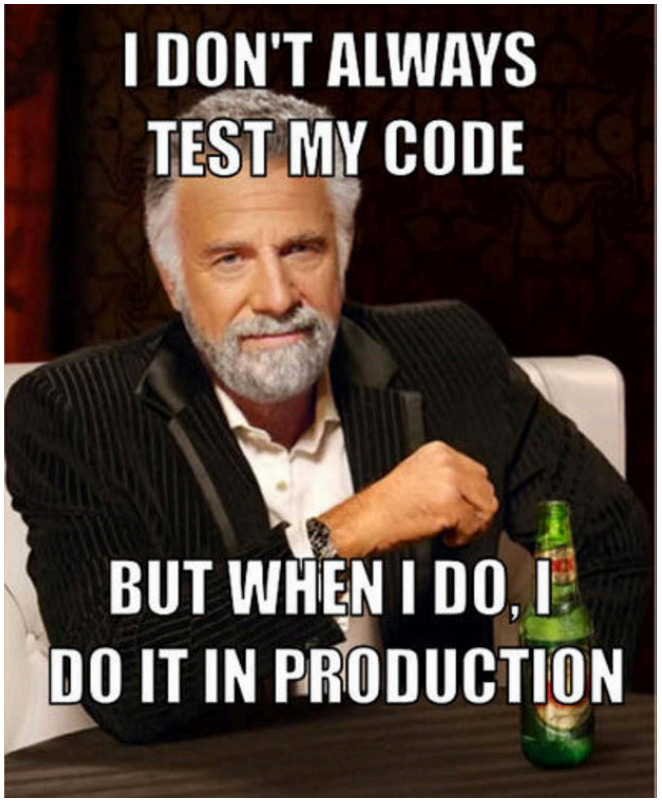


This search has completed and has returned **1,000** results by scanning **22,696** events in **1.049** seconds.

- events per second = events / seconds
- results per second = results / seconds

Play it safe

- Install splunk (free license) local
- Create ‘sandbox’ index
  - 1 day retention
- Bonus Points: VirtualBox + Splunk

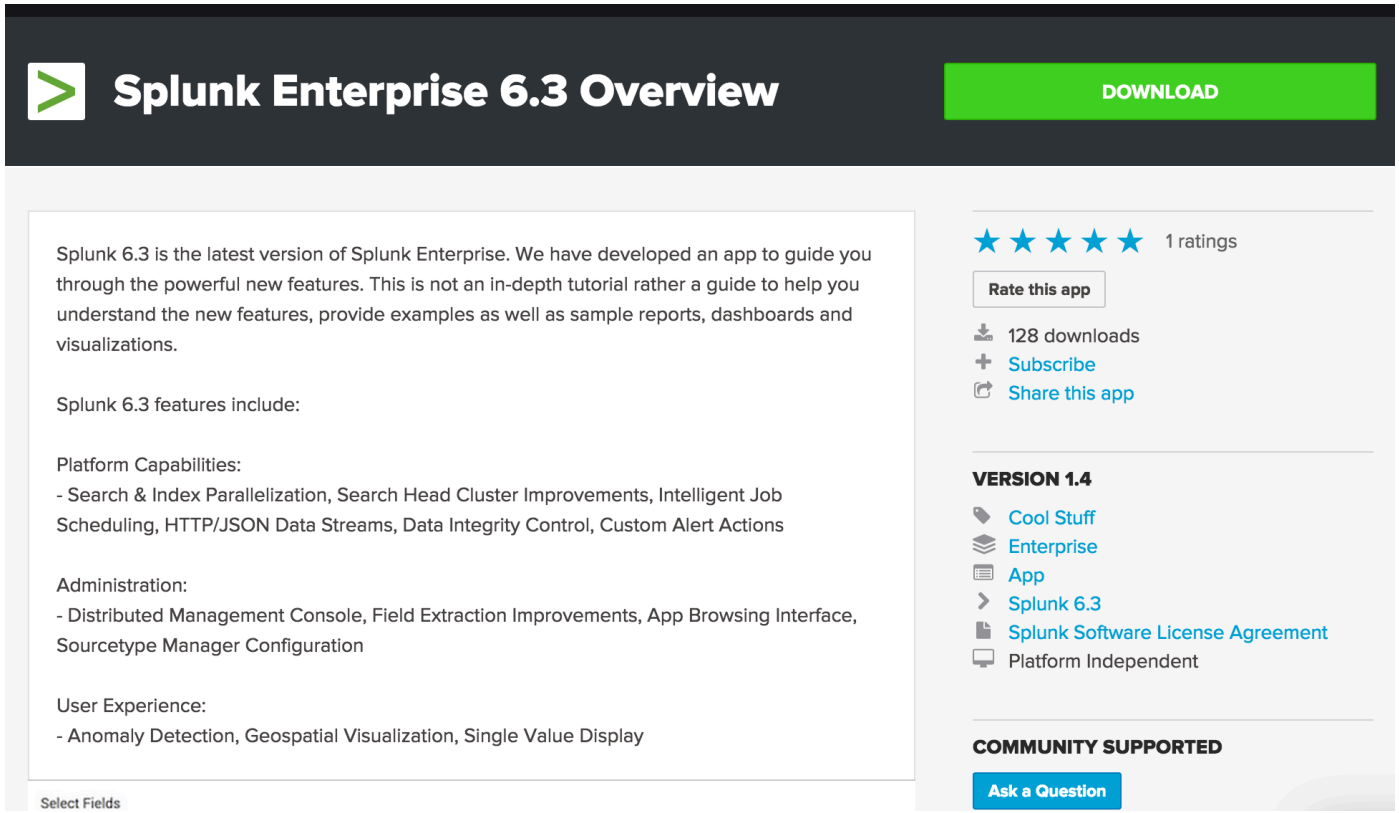


To btool, or not to btool

```
btool <configuration> list <stanza|> <--debug|>
```

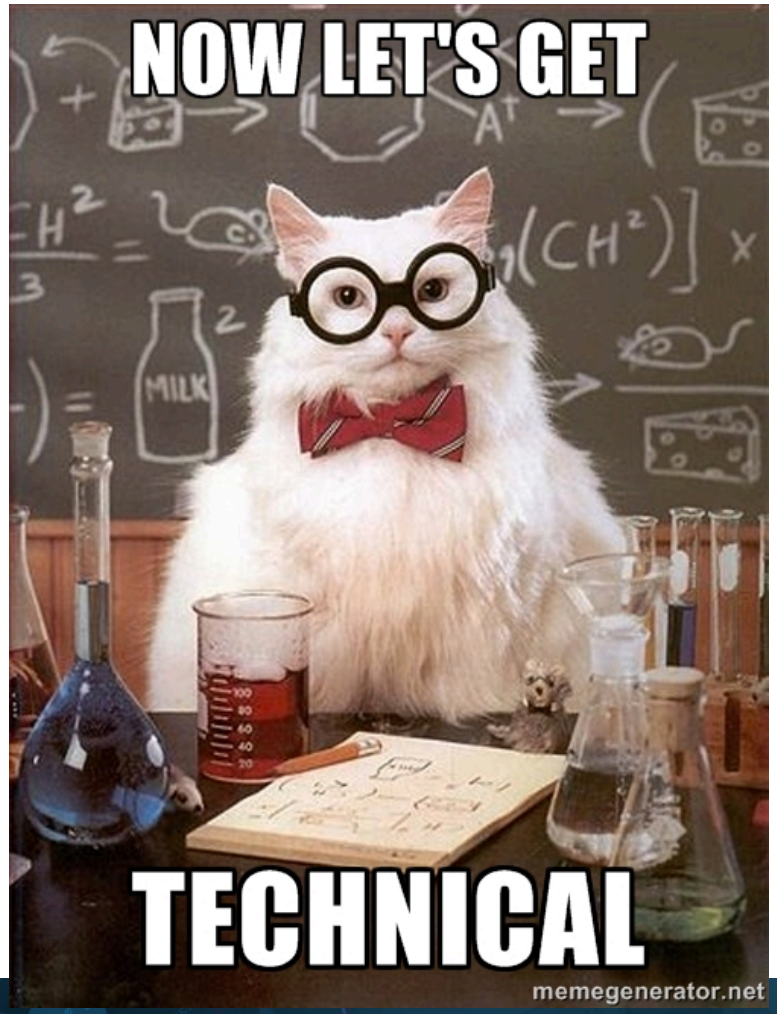
- Add to your env path! (source a profile file from an app)
  - Linux: `export LD_LIBRARY_PATH=$SPLUNK_HOME/lib`
  - Mac: `export DYLD_LIBRARY_PATH=$SPLUNK_HOME/lib`
- No ".conf"
- Use --debug with | `grep –v "system/default"`
- Not current runtime

New Stuff

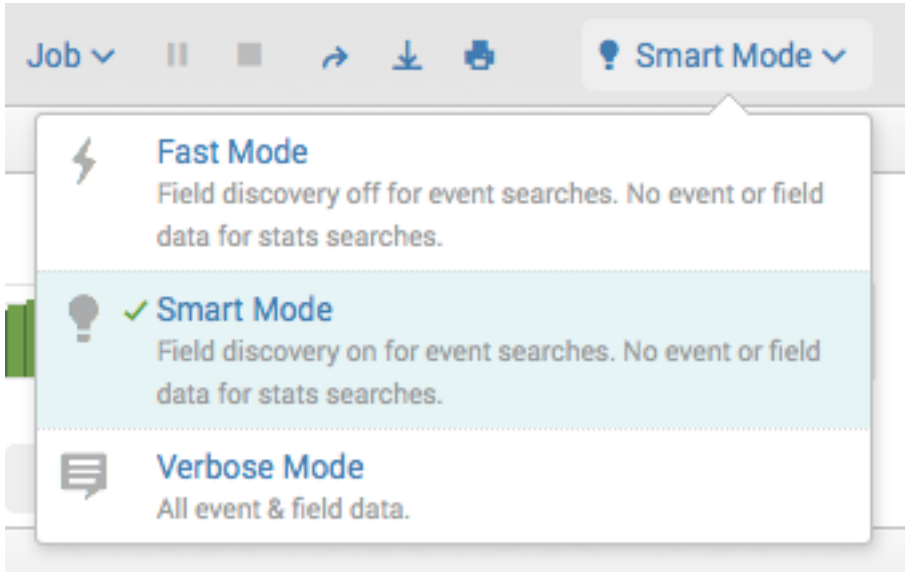

Splunk Enterprise 6.3 Overview
DOWNLOAD
Splunk 6.3 is the latest version of Splunk Enterprise. We have developed an app to guide you through the powerful new features. This is not an in-depth tutorial rather a guide to help you understand the new features, provide examples as well as sample reports, dashboards and visualizations.
Splunk 6.3 features include:
Platform Capabilities:
- Search & Index Parallelization, Search Head Cluster Improvements, Intelligent Job Scheduling, HTTP/JSON Data Streams, Data Integrity Control, Custom Alert Actions
Administration:
- Distributed Management Console, Field Extraction Improvements, App Browsing Interface, Sourcetype Manager Configuration
User Experience:
- Anomaly Detection, Geospatial Visualization, Single Value Display
Select Fields
1 ratings
Rate this app
128 downloads
Subscribe
Share this app
VERSION 1.4
Cool Stuff
Enterprise
App
Splunk 6.3
Splunk Software License Agreement
Platform Independent
COMMUNITY SUPPORTED
Ask a Question

Best Practices

Searching


NOW LET'S GET
TECHNICAL
memegenerator.net

Search Speed



Pretty Searches: Keep it kosher

Weak:

```
… | rename machine as "host for later" | rename net as Subnet |
sort "host for later" | timechart count by "host for later"
span=1h
```

Strong:

```
…| timechart span=1h count by machine
  | sort machine
  | rename machine AS "host for later",
net AS Subnet
```

- new pipe = new line + space + pipe
- | <command> <params> <processing>
- cosmetics at end
- combine multiple renames and rexs

Faster Searching: Less is more

Weak:

```
iphone
  | stats count by action
  | search action=AppleWebKit
```

Strong:

```
iphone action=AppleWebKit
  | stats count
```

Faster Searching: Require Fields

Weak:

```
iphone
  | stats count by action
```

Wrong Results:

Pulls both phone=iphone and user_agent=*iphone*

Strong:

```
phone=iphone action=*
  | stats count by action
```

Faster Search: Be specific

Weak:

```
iphone
  | stats count by action
```

Strong:

```
index=oidemo host=dmzlog.splunktel.com sourcetype=access_combined
source=/opt/apache/log/access_combined.log iphone
user_agent="*iphone*”
  | stats count by action
```

Time selector and eventtypes/tags!

Faster Searching: stats vs dedup/transaction

Weak:

```
… phone=*
  | dedup phone
  | table phone
  | sort phone
```

```
… phone=*
  | transaction host
  | table host, phone
```

Strong:

```
… phone=*
  | stats count by phone, host
  | fields - count
```

Pro Tip:

- Table is cosmetic
- Fields is reducing

Pretty Searches: foreach is clean

Weak:

```
…| timechart span=1h limit=0 sum(eval(b/pow(1024,3))) as size by
st
```

Strong:

```
…| timechart span=1h limit=0 sum(b) by st
  | foreach * [ eval <<FIELD>> = '<<FIELD>>' / pow( 1024 , 3 ) ]
```

Pretty Searches: coalesce's cooler than if

Weak:

```
…| eval size = if( isnull(bytes) , if( isnull(b) , "N/A" , b ) ,
bytes )
```

Strong:

```
…| eval size = coalesce( bytes , b , "N/A" )
```

Faster Searching: Avoid Subsearches

Weak:

```
index=burch | eval blah=yay
 | append [ search index=simon | eval blah=duh ]
```

Strong:

```
( index=burch … ) OR ( index=simon …)
 | eval blah=case( index==”burch" , "yay" , index==”simon" ,
"duh" )
```

Faster Searching: NOT NOTs

Weak:

```
index=burch NOT blah=yay blah=cool
```

Strong:

```
index=burch blah=duh
```

```
index=burch blah!=yay
```

Search Commands: Transaction

Weak:

```
…| transaction host
```

Mo data, Mo problems!

Strong:

```
…| transaction maxspan=10m maxevents=100 …
```

Search Commands: Time and Units

Weak:

```
…| eval new_time = <ridiculous string edits>
```

Strong:

```
…| convert ctime(duration)
```

```
…| bin span=1h _time
```

```
…| eval pause = tostring( pause , "duration" )
```

```
…| rename new_time as _time
```

Search Commands: metadata

Weak:

```
index=*
 | stats count by host
```

Strong:

```
 | metadata index=* type=hosts
```

Search Commands: eventcount

Weak:

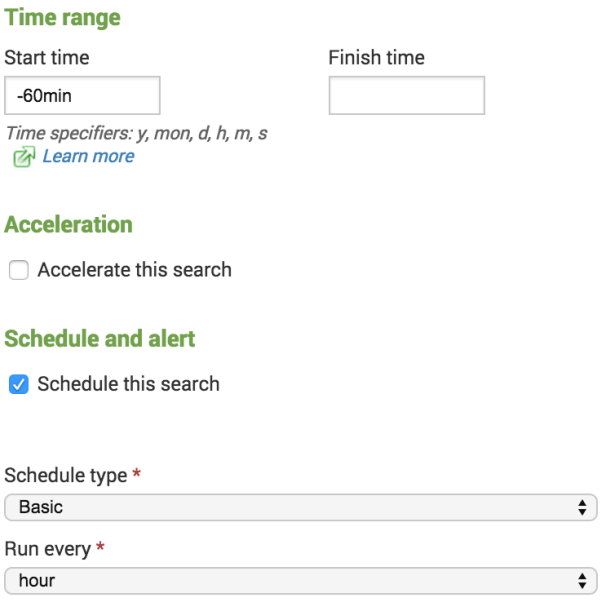
```
index=*
  | stats count by index
```

Strong:

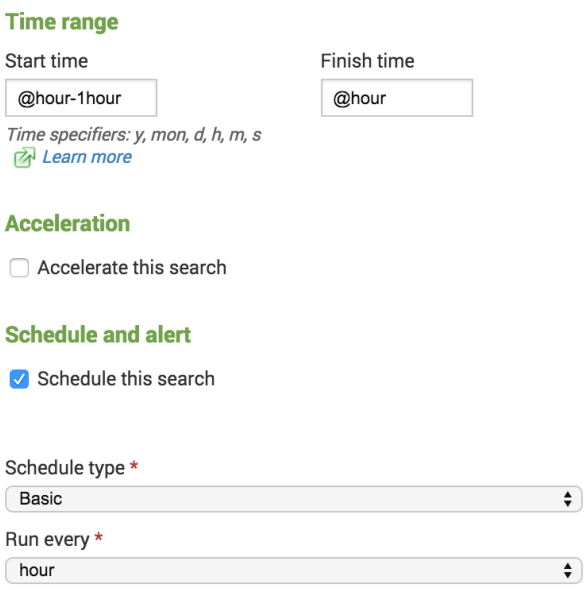
```
 | eventcount summarize=false index=*
```

Accurate Results: Snap-To Times

Weak



Strong



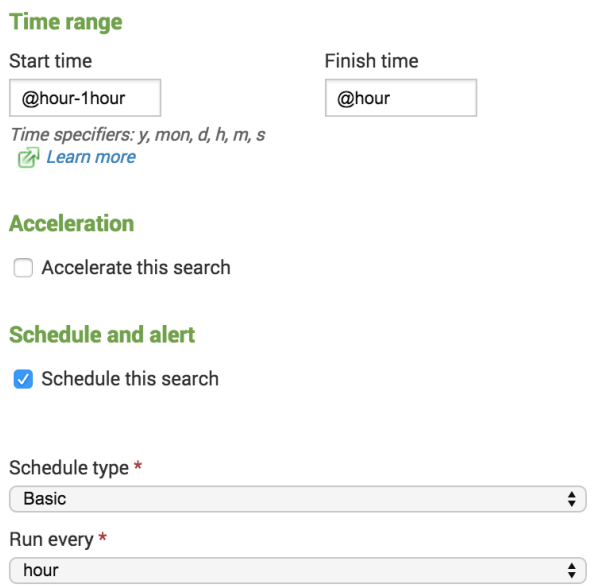
Accurate Results: Time Fields

Weak

Search

```
earliest=-24hours latest=now
...
```

Strong



Accurate Results: Realistic Alerts

Weak

- Static conditions
  - | where count>10
- Spam
  - Avg

Strong

Actionable:

- stddev
- percXX

Find anomalies when outside statistical “normal”

Plug: Tom LaGatta

Best Practices

Administration

Configuration Distribution Recap

In a mature environment

Deployment Server	Deployer	Master Node
Forwarders	Search Head Cluster	Index Cluster

Separate Installs:

- Scalability
- Avoid reload deploy-server on restart
- Cheap VMs
- Not in critical path

Bonus points:

DS -> Master -> IDXC

DS -> Deployer -> SHC

Bootstrap

1. Install splunk binaries
2. Point to DS/Master/Deployer
3. Download config and purpose config
4. Download app with scripted input

Installing Splunk

- Boostrap to DS
  - Segregates install from config
  - Empowers admin with config

- Scripted input to
  - place: local-log.cfg
  - disable local auth (passwd)
  - .ui_login

- Global Config
  - Disable splunkweb
  - Set ports
  - authentication

Remember:

- Transparent Huge Memory Pages
- Source Control

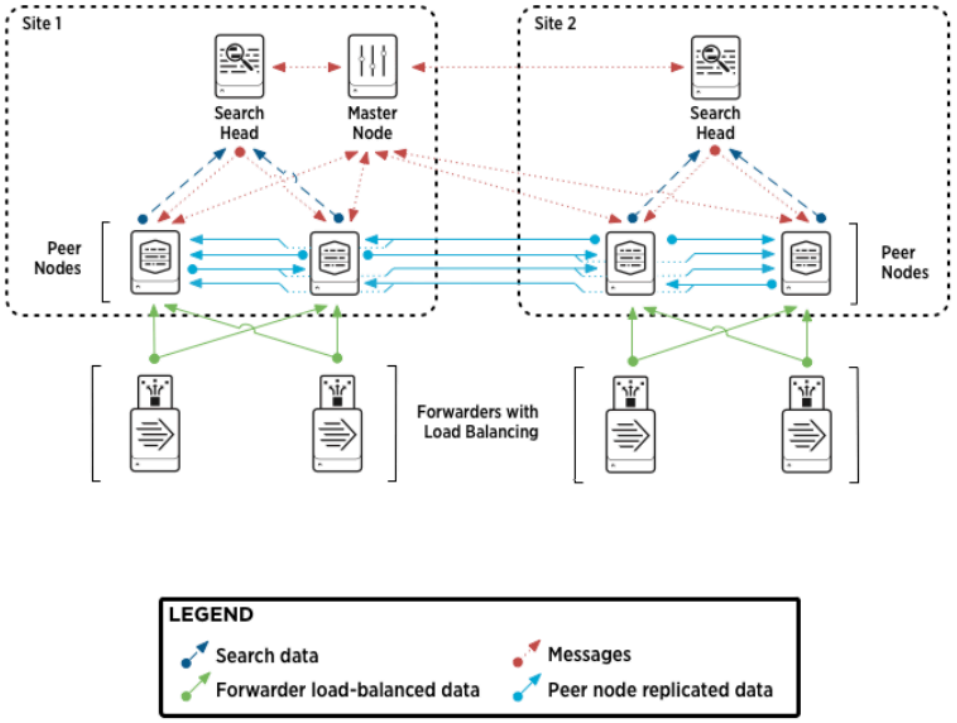
Keep It Clean: Naming Conventions

Template: <summary|>_<company>_<function>_<environment>

- <company>
  - Yours or from a 3rd party/splunk app
- <function>
  - Nothing that changes (i.e. organization/teams)
- <environment>
  - PROD, DR, QA, TEST, DEV, etc…
- <summary|>
  - Exists as a modifying of corresponding index

Architecture: Data Management

- Non PROD data -> PROD SPLUNK!
  - Or Search Head traverses envs
- Logical Separation:
  - Role Based Access Control
  - Separate indexes per env
  - Use event types/tags



Architecture: Cluster of One

- Replication & Search Factor of 1
- Same disk space as non-cluster
- Allows replication on old data
- Seamless scalability

Dangerous Capabilities

Weak

- Scheduled Search
- Real Time Search
- Acceleration
  - Summary Indexing
  - Report Acceleration
  - Data Models

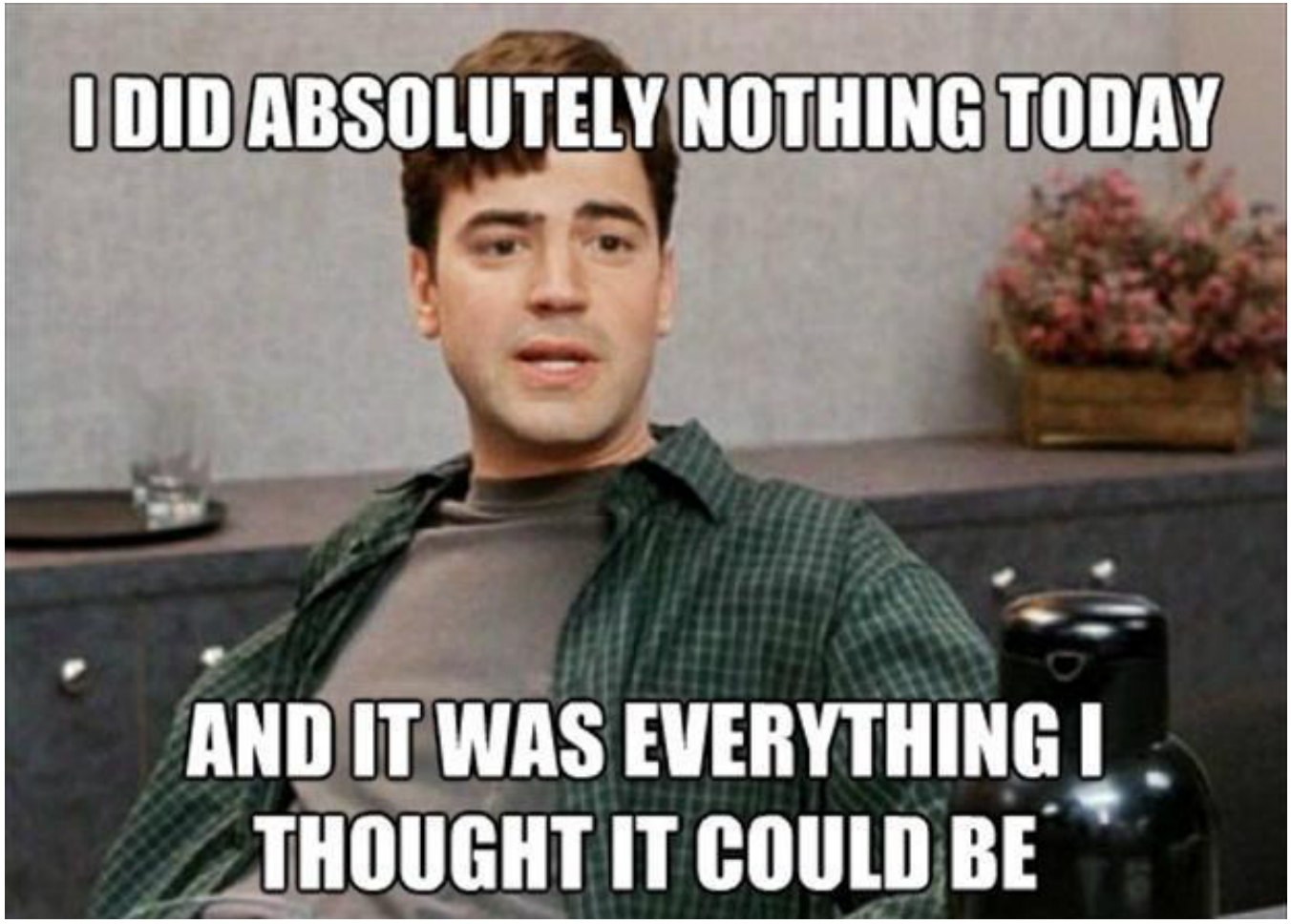
Strong

- Everyone a ‘user’
- Capabilities only for ‘power’+
- Work with you to implement and learn best practices
- Identify & coach & promote to power
- Don’t be a data butler

Log Management

"If you log it, then you should Splunk it"

- Waste of resources:
  - App/System performance to write logs
  - Disk to store logs
- Move cronjobs/scheduled tasks to Splunk
  - scripted inputs
  - Standard output/error captured


I DID ABSOLUTELY NOTHING TODAY
AND IT WAS EVERYTHING I THOUGHT IT COULD BE

Logging Made Easy

- ❑ Use clear key-value pairs
- ❑ Create events humans can read
- ❑ Use developer-friendly formats
- ❑ Use timestamps for every event
- ❑ Use unique identifiers (IDs)
- ❑ Log in text format
- ❑ Log more than debug events
- ❑ Use categories
- ❑ Identify the source
- ❑ Minimize multi-line events

Forwarding & Search Heads

- Forward all instances to indexers
  - All indexes – including summary
  - All instances:
    - * Forwarders
    - Search Heads
    - Deployment Server
    - License Server
    - Cluster Master
    - Deployer

Indent Config

Example:

```
[general]
pass4SymmKey = $1$ShiC+P0X
serverName = elBurcho
    sessionTimeout = 30m
```

Benefit

- Easily see system vs hand edits
- Detect hand config updated by system

Search Head limits.conf

Example:

```
[scheduler]
max_searches_perc = 90
```

Benefit

- Defaults to 50
- Ad Hod takes precedent
- Additional controls for scheduling

New Feature: Indexer Discovery

Example:

Master Node's server.conf

```
[indexer_discovery]
indexerWeightByDiskCapacity = true
```

Forwarder's outputs.conf

```
[indexer_discovery:master1]
master_uri = https://masterhost:8089
[tcpout:group1]
indexerDiscovery = master1
[tcpout]
defaultGroup = group1
```

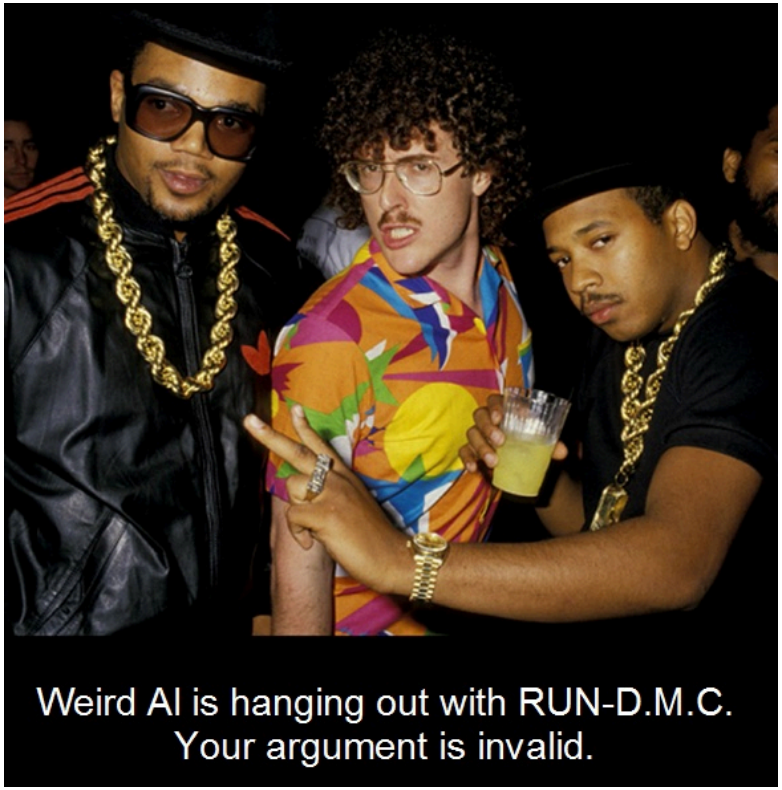
Benefit

- Great for indexers with different volume sizes
- Requires network traffic to master node
- Search docs.splunk.com for "indexerdiscovery"

- Don't forget about Volumes in indexes.conf

Run DMC

- Manage Splunk 6.2+ environments
- Replaces Deployment Monitor App
- Incorporates SOS app prior to 6.2+



Best Practices

Next Steps

Questions?

- Burch @ IoT Panel @ 5:15
- Download these slides
- Questions?
  1. now
  2. find me after (IT Ops Booth)
  3. burch@splunk.com
- Sessions
  - Search for “Optimization” or “Best”
  - Search for “Machine Learning”
  - Go to 320 @ noon

THANK YOU
(rate this)

splunk>