T2
Test Data Management
Thursday, May 3rd, 2018
9:45 AM

Machine Data Is Everywhere: Use It for Testing!

Presented by:

Tom Chavez

Splunk

Brought to you by:

350 Corporate Way, Suite 400, Orange Park, FL 32073
888-268-8770 ⋅ 904-278-0524 - info@techwell.com - http://www.stareast.techwell.com/

Tom Chavez

Splunk

Tom Chavez has more than twenty years of experience as a manager and product manager in the software development tools field. Tom works in product marketing as senior manager of developer marketing at Splunk, the leader in operational intelligence. Tom has worked across the Silicon Valley in California at companies including Apple, Sun, PalmSource, and Intuit, delivering tools for web, Mac, Java, PalmOS, Linux, and Android development and testing. He speaks frequently at industry conferences and meet-ups on topics including web app performance, testing at large scale, mobile continuous integration and testing, automated mobile testing tools, and big data analytics for business value. Follow Tom on Twitter @TomChavez.

Machine Data is EVERYWHERE!

Use it for Testing (and beyond!)

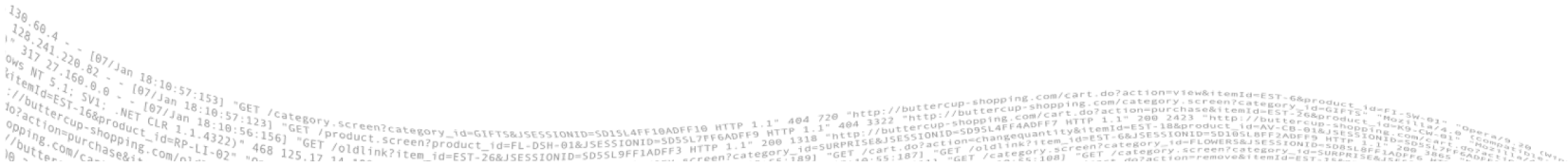
Tom Chavez | Sr. Manager, Developer Marketing
tchavez@splunk.com @TomChavez

May 2018

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> listen to your data

Agenda

Machine Data is EVERYWHERE – Almost drowning in it!

- What is Machine Data?
- Where do you find it?
- How do you use it in testing?
- What else can you do with it?

What is Machine Data?

splunk> listen to your data

What is Machine Data?

System and app data that is created and collected

- Digital Exhaust
  - Log files
  - Log data written to database
  - Log streams
  - stdout / stderr
- Data about systems, devices, apps, network
  - collectd – system metrics/stats and application perf metrics
  - App crash logs, stack traces / dumps, core dumps
  - Wire data – conversations between endpoints
- Data from an application, service via REST API, webhook

What Produces Machine Data?

- Hardware does –
  - Time, temperature, memory, devices, physical state …
- Operating systems do –
  - Filesystem, CPU utilization, memory allocation, threads, stack, …
- Software stacks do –
  - App server, web server, database, Java, …
- Apps do –
  - Log and event data, …
- Devices do –
  - Any sort of log or event data from a device, sensor, …

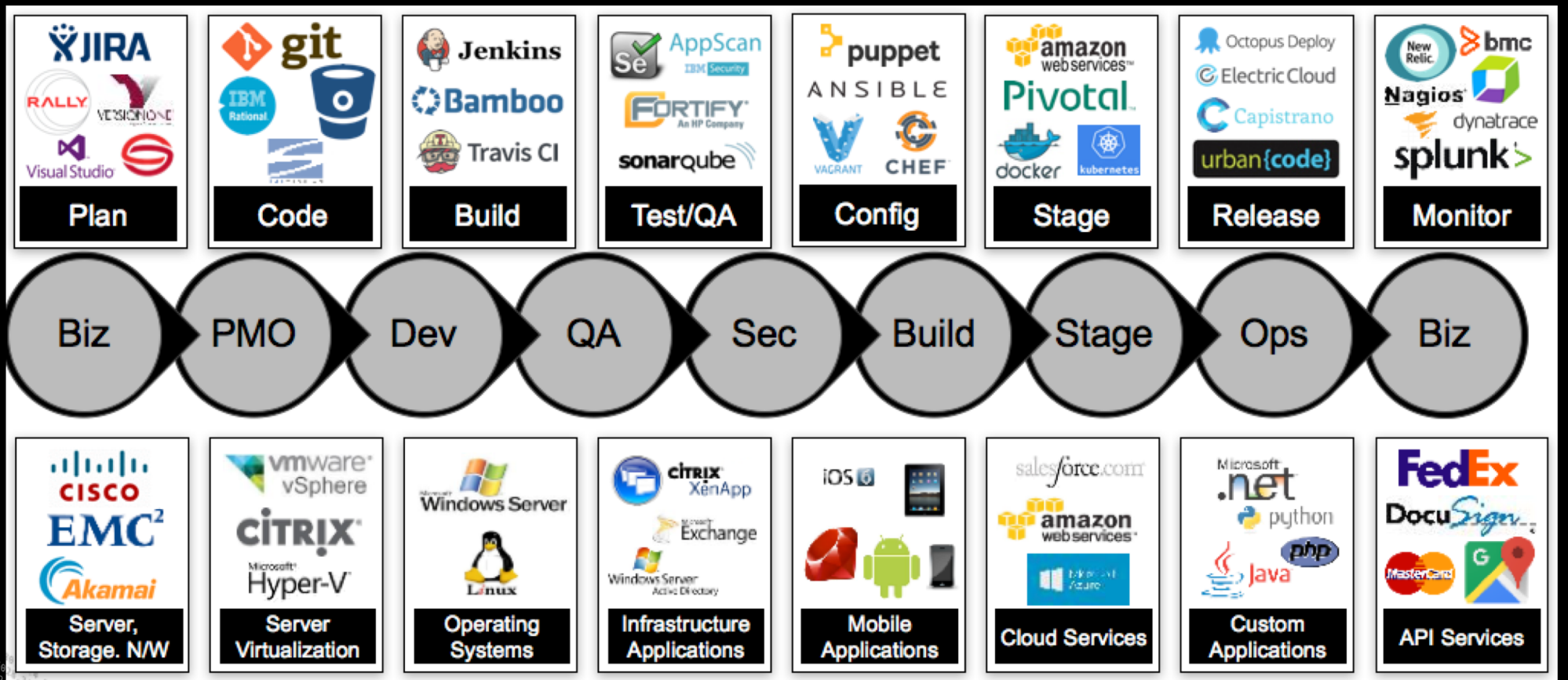
What Does Machine Data Look Like?

Sample Log Entries

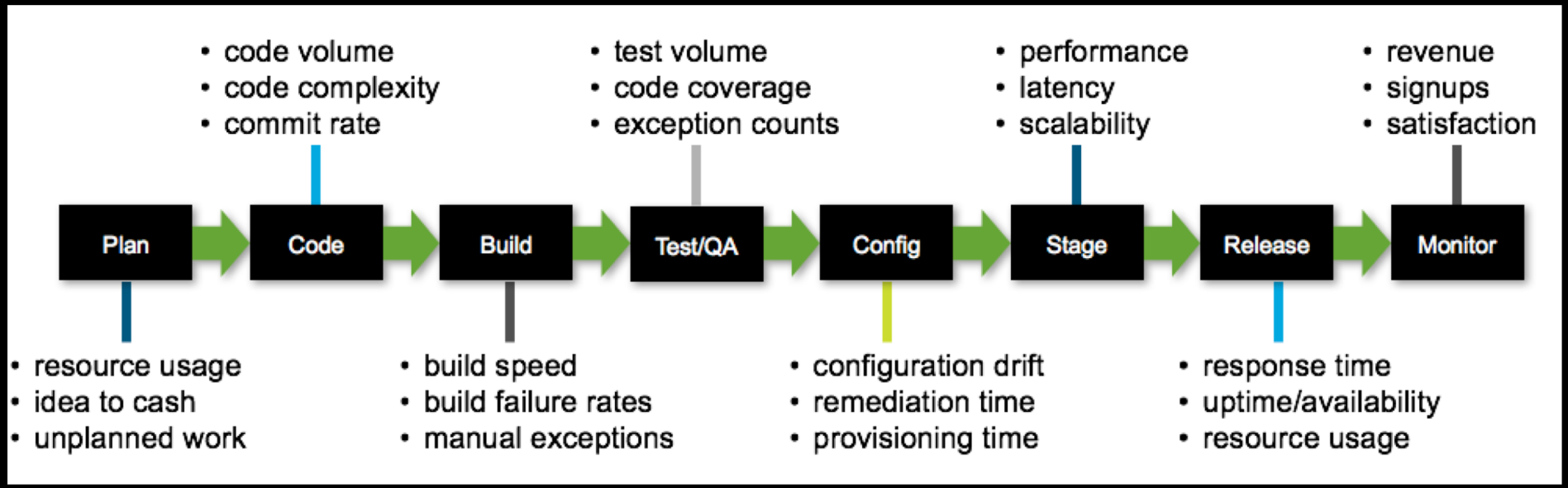
- nginx log entry
  - `150.128.102.148 - - [07/Aug/2014:00:59:52 +0000] \"GET /images/web/2009/banner.png HTTP/1.1\" 200 52315 \"http://www.semicomplete.com/blog/articles/week-of-unix-tools/day-1-sed.html\" \"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36\`
- Java Service Error Log
  - `2017-07-12T18:14:50.661995+00:00 ERROR [com.zillow.db.InnerPool] Connection hard closed due to exception:java.sql.SQLException: Invalid state, the Connection object is closed. src:{ call dbo.XXXXXXXXXX(XXXXXXXXXX) } on jdbc:jtds:sqlserver://XXXXXXXXXX`
- Twitter tweet
  - `{actor:{displayName: “Go Boys!!”,followersCount:1366,friendsCount:789, link:http://dallascowboys.com/,location:{displayName:“Dallas, TX”,objectType:“place”}, body: “Can’t buy this device from @ACME. Site doesn’t work! Called, gave up on waiting for them to answer! RT if you hate @ACME!!”, objectType:“activity”, postedTime: “2016-05-21T16:39:40.647-0600”}`

How does Machine Data provide value?

Machine Data Provides Visibility Across Dev and Ops



Machine Data Drives Analytics for Every Phase



How does Machine Data apply to Testing?

- Machine data gives a view into an application while it is executing and after
  - When did it start, when did it terminate
  - What happened in the application while it was running
- And of the environment in which the application was run
  - What was the state of the environment in which the application was running
  - What other systems were running at the time
  - How did the environment change while the application was running
- Which makes it easier to test, and understand, the app under test

How should we create Machine Data?

Four Rules of Machine Data

- It has to exist – you are logging, right?
- It has to be correctly formatted and usable
- It has to be consumable by your log capture tools
- It has to be complete

Example Project Logging Rules

- Log at the beginning and end of a function denoting its start and successful end.
- Log at the beginning and end of any external utility/function being called.
  - Logging Frameworks do this

- If there is an error, log it only at the lowest level where the error actually happened, otherwise just return.
- At the higher levels just return the error.

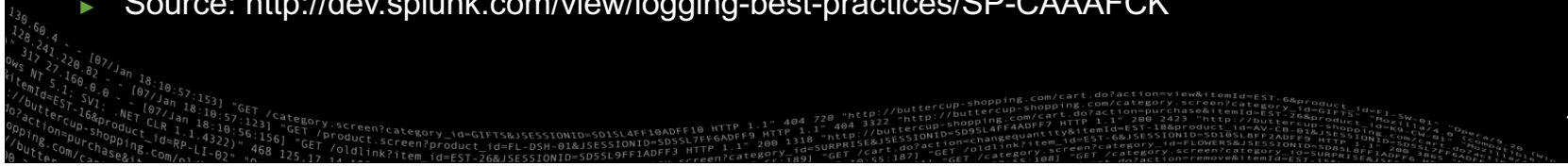
- While logging errors, always provide a error message too.

Logging Best Practices

Tips for what to include in data you log

- Use timestamps for every event
  - Use the most verbose time granularity possible – time should be rendered to microseconds
  - Put the timestamp at the beginning of the line, so not to confuse it with other data
  - Include a four-digit year
  - Include a time zone, preferably a GMT/UTC offset
- Examples:
  - `00:00:00.000 Start of test` – seen in performance testing
    - On what date did this test start? At what time? File creation date might tell, might not.
    - Bad: cannot correlate by time with data from other systems.

- Source: http://dev.splunk.com/view/logging-best-practices/SP-CAAAFCK



Logging Best Practices

- Create events that humans can read
  - Avoid using complex encodings. If logs are binary, provide tools to convert to ASCII
  - Don’t log a binary file but log the meta information for it: file locator + relevant data
    - E.g. for a JPG file, include image size, creation tool, username, camera, GPS location, etc.

- Use unique identifiers (IDs) such as transaction IDs and user IDs
  - Provide additional / greater granularity than time alone
  - Can be shared across services, system, machines, networks for easier correlation

Logging Best Practices

- Use clear key-value pairs
  - `key1=value1, key2=value2, key3=value3`
  - If values contain spaces, quote them: `username="bob smith"`
- Use a developer-friendly, structured format
  - JavaScript Object Notation (JSON):
    - `{ "sender" : "michael"`
      `"recipient": { "name" : "michael", "name" : "andrea", "name" : "itay" }`
      `subject:"I heart logs" }`
- Even CSV is nice

Logging Best Practices

- Identify the source
  - Class, function or file name
- Use categories
  - E.g. severity: `INFO, WARN, ERROR, DEBUG`
- Log more than just debugging events
  - Audit trails, what users are doing, transactions, timing information, etc.

How do you consume Machine Data in testing?

Capturing and Consuming Machine Data

- Free tools for gathering machine data
  - Splunk – commercial big data logging and analytics platform, free up to 500Mb data/day
  - DataDog – commercial monitoring and analytics tools, free tier
  - ELK/Elastic – Open source project to gather data from any source
  - Sumo Logic – machine data analytics, free tier
  - Loggly – unified log analysis & log monitoring, free lite tier
  - Others….
- Some things to consider:
  - On premises? In Cloud? Hybrid?

Capturing and Consuming Machine Data

- Getting data into your machine data tool
  - Look for a marketplace – pre-built add-ons or plug-ins to push, pull, and ingest data
  - Code samples to transform data from raw to JSON, or other formats
  - Webhooks and REST APIs to push and pull data
  - Google for (tool-name) and (data-type), e.g. Jenkins Splunk

What can you do with Machine Data?

Graph it, Analyze it, Track it, Test it

Information from Jenkins

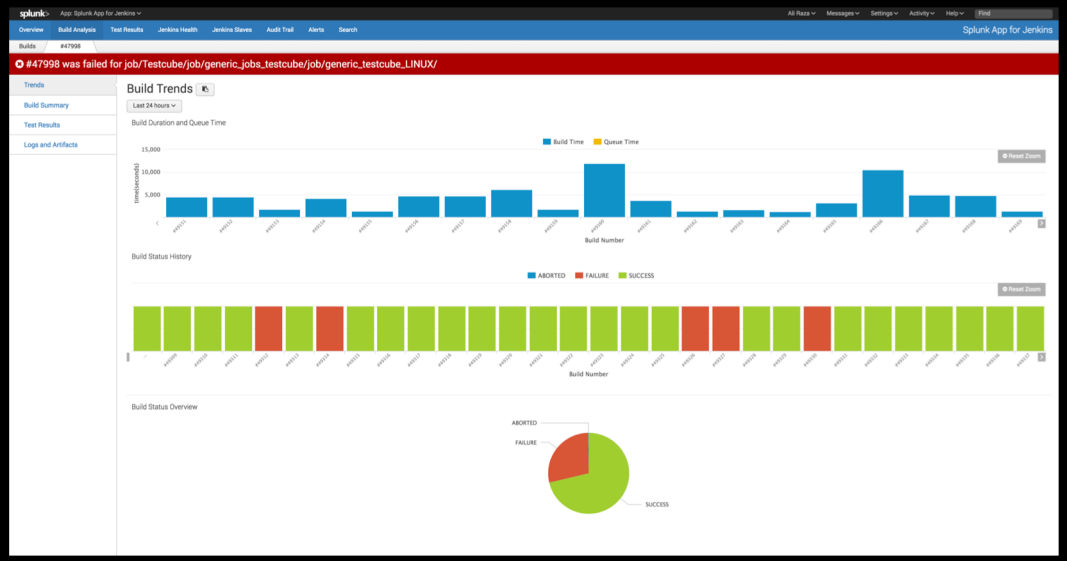
- Build information
  - Queue time
  - Build duration
  - Job status
  - Who started the job
  - Console logs
  - Artifacts
  - Environment / injected variables
- Testing
  - X-unit style report processing
  - TestNG reports
  - Cucumber reports
  - Data to make test result triaging (how long is a test failing, test duration, etc.)
  - Code Coverage
- Build name, id, etc
- Audits (who changed what)
  - Job configs
- Plugin configs
- Master config
- Logs
- Slaves logs
- Master logs
- o Build/console logs
- Health
- o JVM information
- o Queuing information
- o Slave health stats
- o Build distributions across slaves
- o Label distribution across slaves

Single View of Jenkins Infrastructure

- Visualize multiple masters and associated slaves in a single page
- View build status trends and drill down and get details information about any build

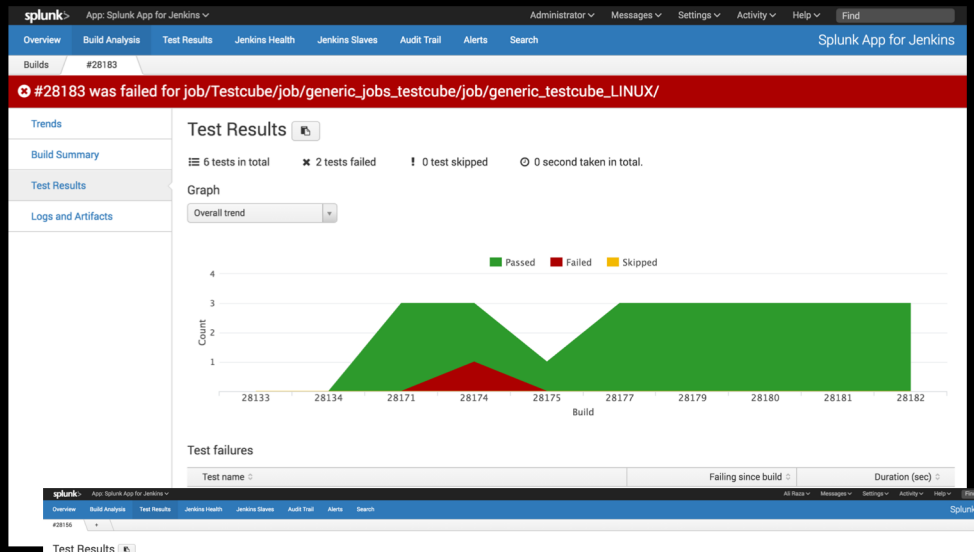
Build Analysis

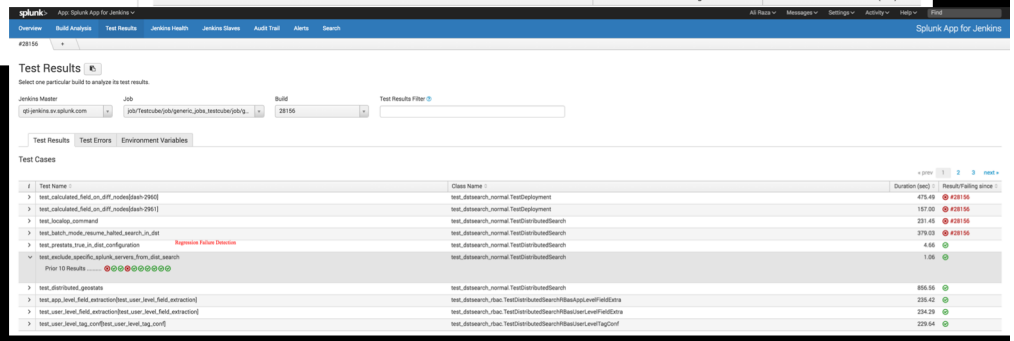
- Find any Jenkins build using a variety of easy to use filters
- View build summary or drill down to see:
  - build status trends
  - build time and queue time analysis
  - tests pass/fail trends
  - test runtime distribution
  - console logs couple with Splunk's powerful search interface



Test Results

- Test Results:
  - Shows all the failing tests with stack traces
  - flags regression failures
  - groups test failures by errors
  - captures Jenkin's environment variables
  - provides nifty filters to find tests with long run times, particular errors, testsuites, etc.





Murex talk from .conf2017

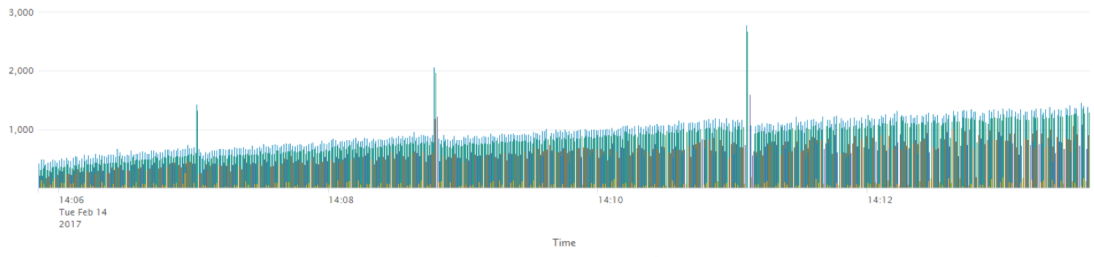
Major Financial Trading Platform

© 2017 SPLUNK INC.

MUREX™

"If you Graph it, they will see it"

- When you graph an issues it can become more obvious
- Sometimes "Average" and "Mediums" and "Maximums" are not enough to see issues…
- In the below graph an obvious increase in the "class.method" over time can be seen
- This might not have been obvious with out a visualization



TPS Real-Time Stats TPS_Class = * [For the time-period selected in the chart above]

fullyQualifiedMethod	average	count	stdev	median	perc95	perc99.5	min
murex.risk.control.excesses.client.ExcessWorkspace#addEvent	810.30	15330	268.3	789	1230	1422	223
murex.risk.control.excesses.client.ExcessWorkspace#commit	38.80	15301	21.1	34	80	128	14
murex.risk.control.excesses.controller.session.ExcessControllerServiceSession#getExcessByKey	314.60	15330	181.9	288	630	878	34
murex.risk.control.excesses.controller.session.ExcessControllerServiceSession#getOrCreateExcessKeyId	346.10	15325	152.4	325	613	886	62
murex.risk.control.excesses.service.DefaultExcessService#addEvent	689.60	15330	268.5	670	1111	1297	162
murex.risk.control.excesses.service.DefaultExcessService#commit	0.10	15301	0.6	0	1	1	0

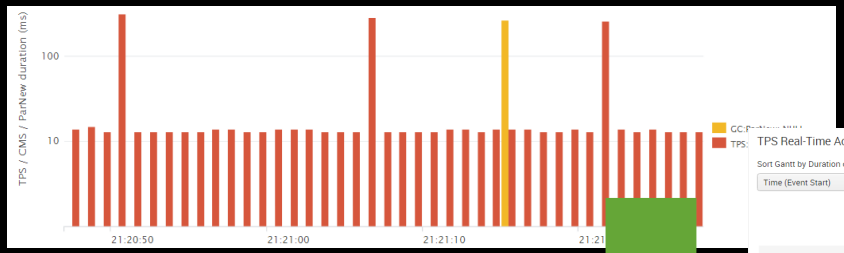
splunk> .conf2017

splunk> listen to your data

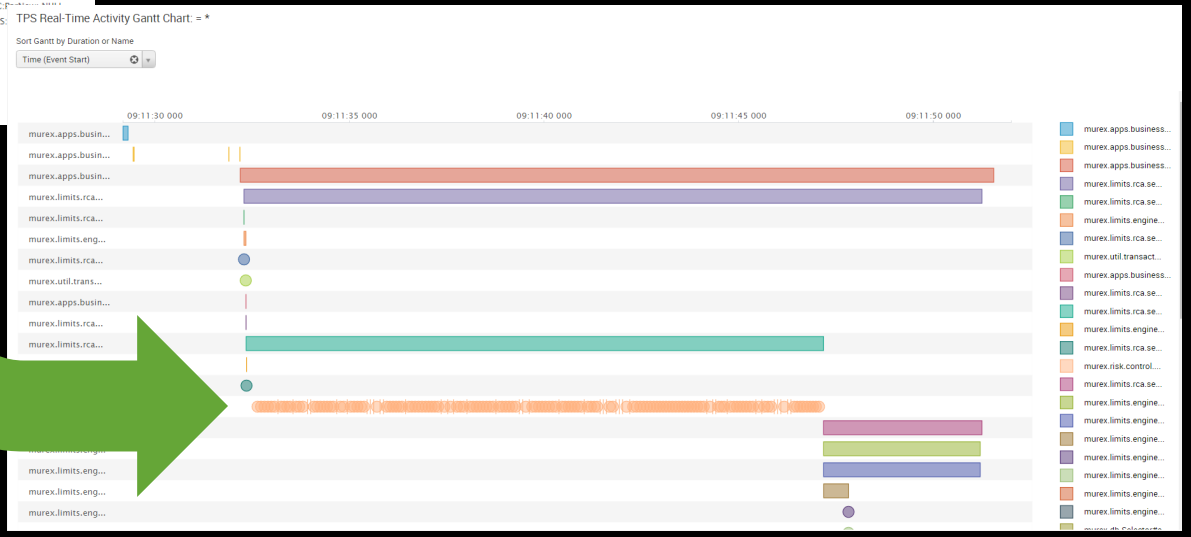
Murex talk from .conf2017

Major Financial Trading Platform

- How you display machine data – as separate events, look fine



- But over time, they can really add up!



Analyze the Test Results

- After a test is complete, use the ***search*** in your machine data logging tool to validate that the test was complete, e.g.
  - Were all required tests performed?
  - Did results fall within acceptable parameters?
  - Did tests perform as expected?
  - Did tests create the expected outcomes?

- And analyze that the tests were validly performed
  - **Was the test environment set up to defined specifications?**

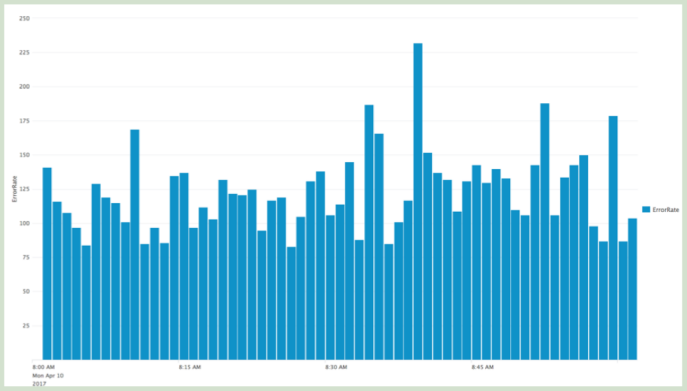
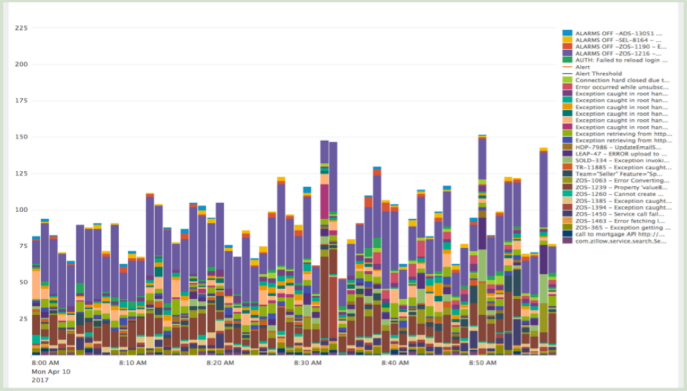
Splunk Test Automation

- Data about your test environment
  - Configuration information – changes since last tests
  - Logging and data from the environment during the test – looking for changes

Automatically Categorize Your New Bugs

- Zillow at .conf2017

Needed a solution

to turn this	into this… automatically
	

- http://conf.splunk.com/sessions/2017-sessions.html#search=jira&

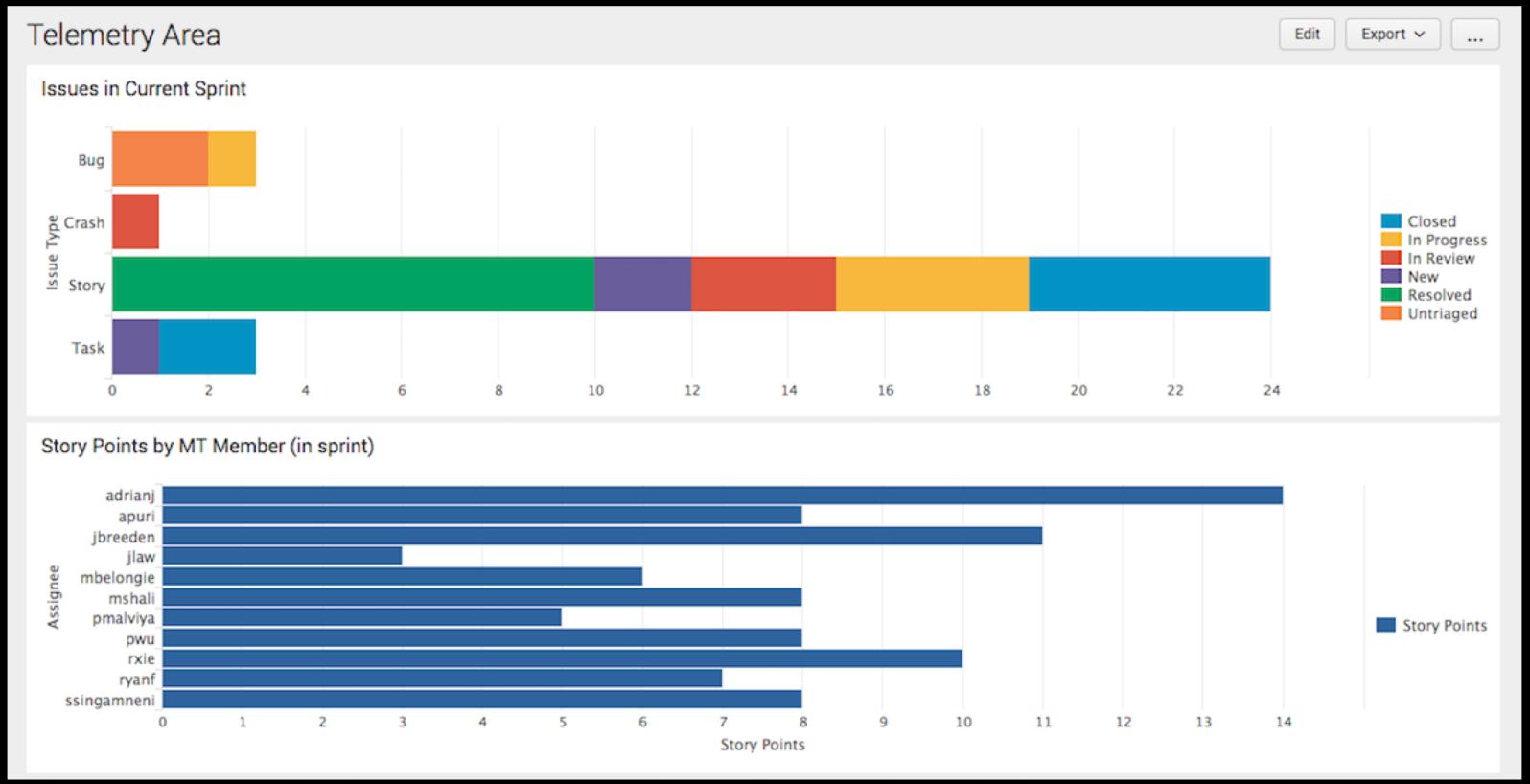
Splunk Test Automation

Getting Data out of Jenkins and Jira

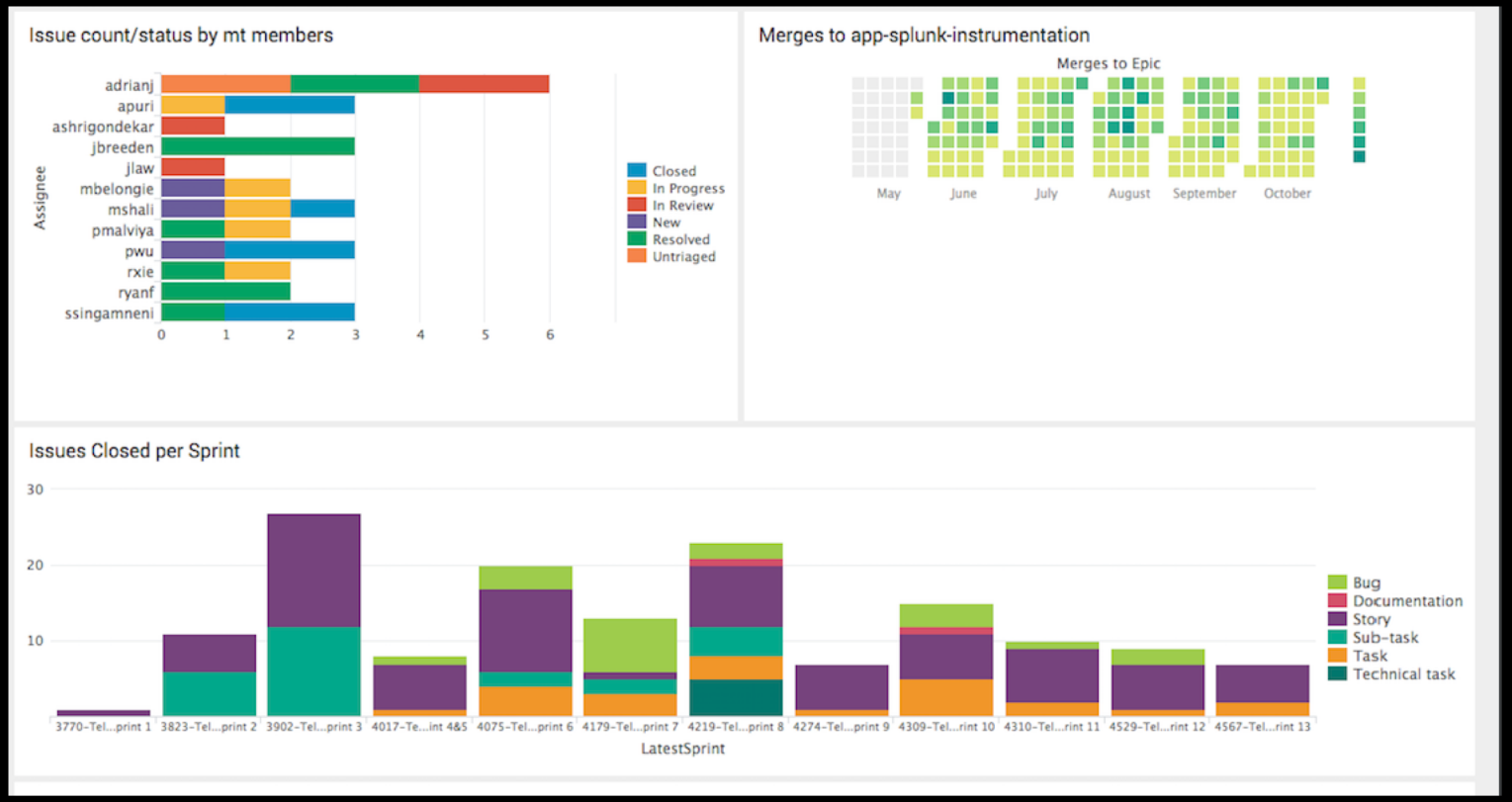
- Test results over time
- Rate of bugs found over time
- Rate of bugs closed over time
- Average time to close a bug
- Time per engineer to close a bug

- Sprint Data
  - Stories assigned, closed by staff member
  - Overview of multiple teams for manager, director, VP view

Getting Data from Jira

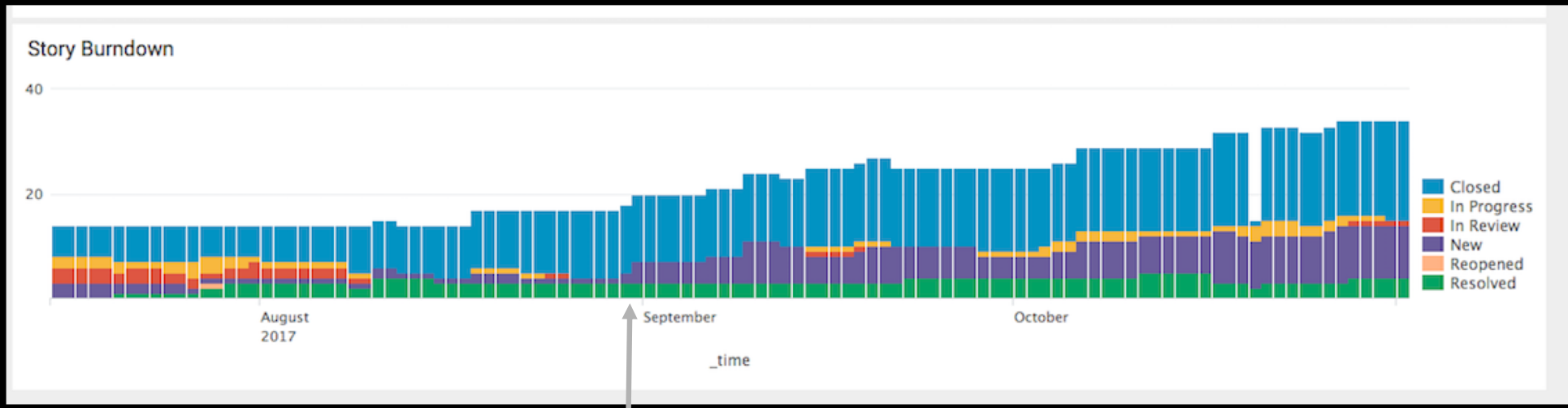


Getting Data from Jira



Getting Data from Jira

Monitoring for appropriate behavior



Branch to create new work

Resources

Where to go now?

- Free data logging & analytics products:
  - **Splunk**: splunk.com
  - **Elastic**: elastic.co
  - **Datadog**: datadoghq.com
- Getting Data into tools
  - **Splunk**: splunkbase.com
  - Or Google (toolname) and (datatype/source)!

Public Presentations

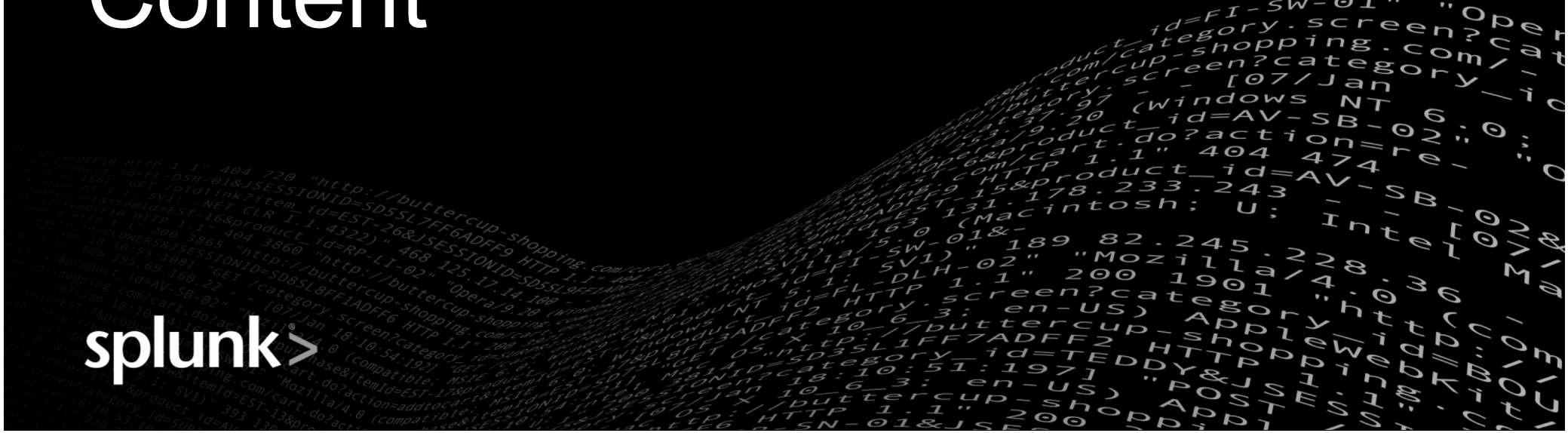
- Zillow and auto-categorizing of Jira defect tickets
  - Finds regressions & automatically identifies up to 80% of daily defects
  - http://conf.splunk.com/sessions/2017-sessions.html#search=jira
- If You Graph It, They Will See It: Identifying Root Issues from Product Testing to Production Crisis. (Splunk@Murex For Test and Development)
  - Visualizing Data to find Insights
  - http://conf.splunk.com/sessions/2017-sessions.html#search=murex&

Thank you!

Machine Data is EVERYWHERE

splunk>

Additional Content



Splunking Jenkins

Getting Awesome Insights about Your Jobs and Infrastructure

Tom Chavez | Sr. Manager, Developer Marketing

@TomChavez

August 30, 2017

splunk>

More DevOps Metrics that *Might* Matter

Culture	Process	Quality	Systems	Activity	Impact
• Retention	• Idea-to-cash	• Tests passed	• Throughput	• Commits	• Signups
• Work hours	• MTTR	• Tests failed	• Uptime	• Test run	• Checkouts
• Callouts	• Deliver time	• Best/worst	• Build times	• Releases	• Revenue

Jenkins Platform Accelerates Software Delivery

- Key Benefits
  - Simplifies software delivery
  - Automates builds and tests
  - Extensible with a vast ecosystem
  - Adopted widely

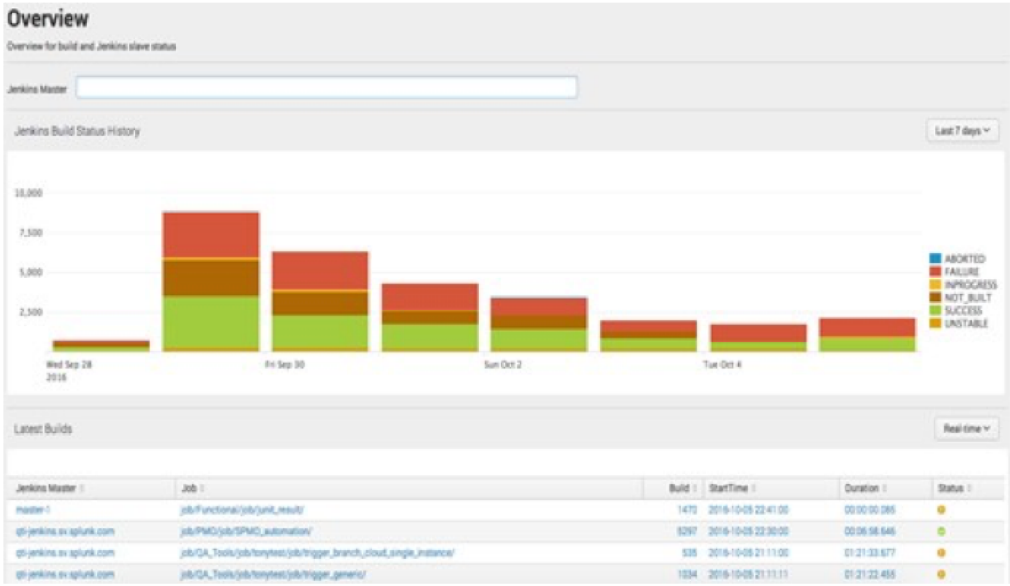
But Monitoring CI/CD Can Be Challenging!

- Processing large amounts of data (test and build results, logs artifacts)
- Troubleshooting and triaging can be complex
- Correlating data across multiple data sources not easy
- Proactive alerting can be difficult

Splunk App for Jenkins

Real-time insight into your CI/CD pipelines

- Continuous visibility into builds progress
- Gain instant visibility into test results
- Monitor the health of Jenkins infrastructure



splunk> listen to your data

Successful Businesses Use Splunk for DevOps

ENERNOC

Cerner

NBCUniversal

APOLLO EDUCATION GROUP

SNAP INTERACTIVE

TiVO

Zillow

StubHub

redhat.

VERISIGN

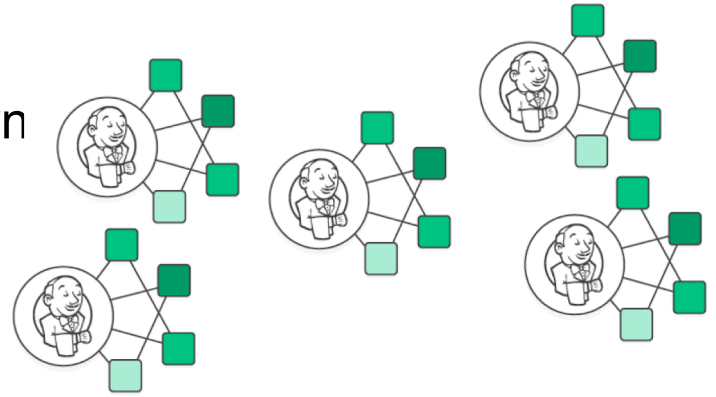
AMAYA

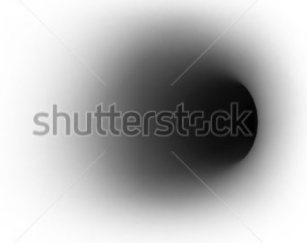
Agenda

1. The story of Jenkins at Splunk
2. How we tamed our Jenkins infrastructure
3. What you can do to tame your Jenkins!

Moving to Jenkins Automation Wasn't Easy.

- Jenkins sprawl – Every team was running their own
- Ownership – Who owns all the Jenkins slaves?
- Jenkins Health – Hardware and software problems
- Data overload -- Too much Jenkins data for engineers
- Jenkins ROI – "Black hole" to upper management





More Jenkins Didn't Solve the Problem

>6	>400	>2400	7000+	250k
Jenkins Masters	Slaves per Master	Slaves during Peak usage	Builds per day	Tests per day

<u>3 to 4 weeks</u> to go through all the Jenkins data to certify the build !!

splunk> listen to your data

We Set Goals to Manage Jenkins

- 1. Centralize Jenkins data and Splunk EVERYTHING !!!!
- 2. Use Splunk Alert workflow for monitoring critical conditions
- 3. Shorten build certification times
- 4. Get engineers out of Jenkins
  - Deep analytics for engineers
  - Real-time KPIs for managers

- Deliver Jenkins + Splunk solution for the general community
  - Must see ROI in 5 minutes!

Our Solution Required Two Components

- 1. **Jenkins plugin** to get data into Splunk
  - Intercepts and sends all sorts of Jenkins data to Splunk
  - Highly configurable -- send as little or as much data as you want
  - Installs and configures in 5 minutes!

- 2. **Jenkins App** for Splunk
  - Typical Splunk app with a bunch of pre-configured dashboards
  - Easy way to view Jenkins infrastructure, builds, test results, console logs, etc.
  - Create powerful Splunk Alerts on any problematic pattern
    - Job failing, critical tests failing, slave offline, queue size too big, etc.

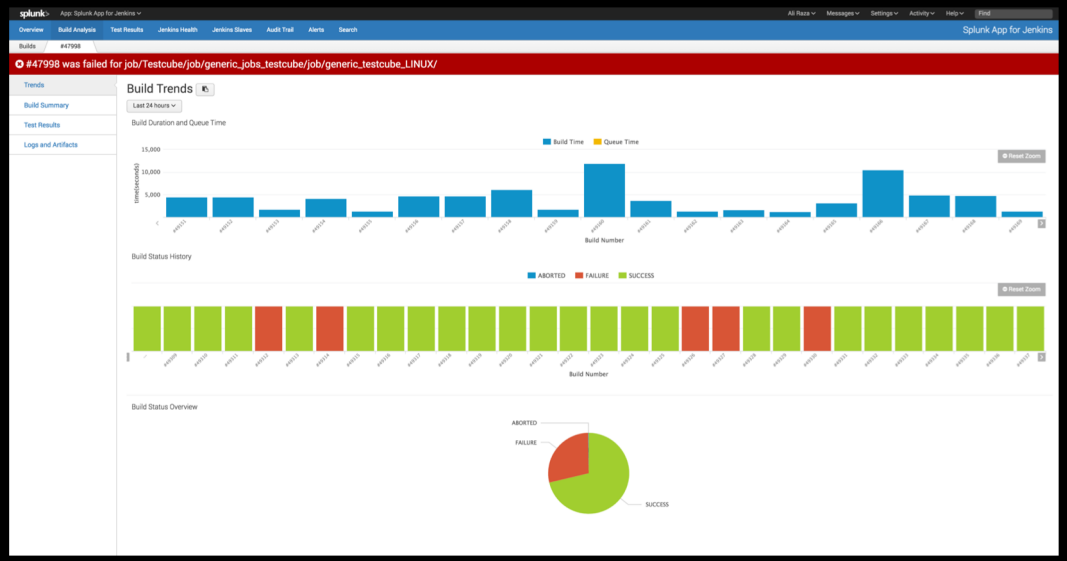
splunk> listen to your data

Single View of Jenkins Infrastructure

- Visualize multiple masters and associated slaves in a single page
- View build status trends and drill down and get details information about any build

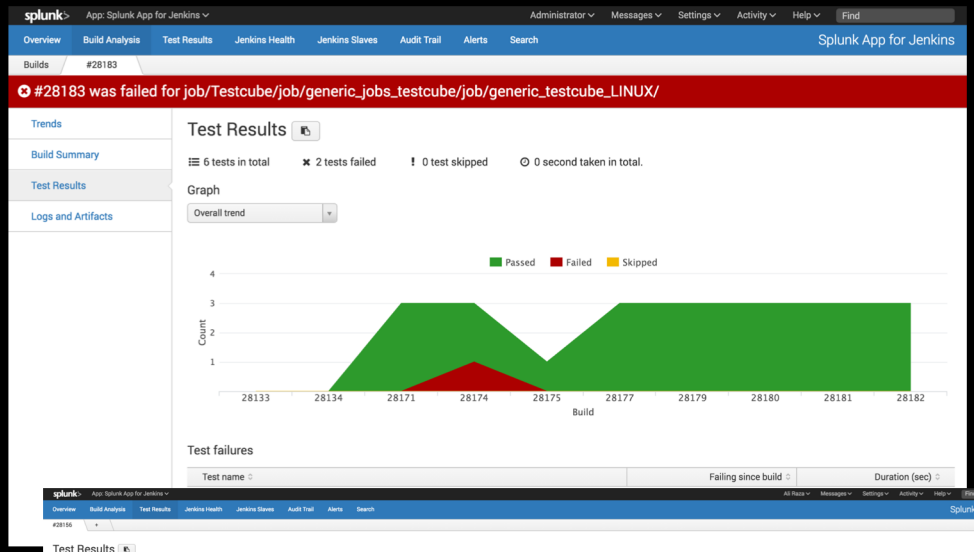
Build Analysis

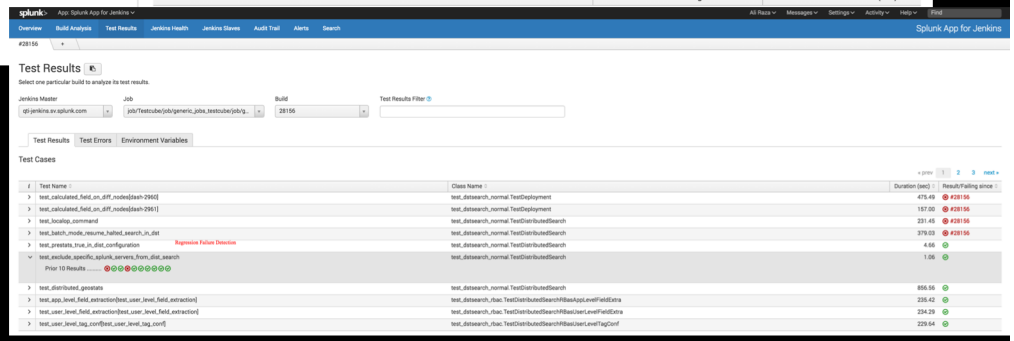
- Find any Jenkins build using a variety of easy to use filters
- View build summary or drill down to see:
  - build status trends
  - build time and queue time analysis
  - tests pass/fail trends
  - test runtime distribution
  - console logs couple with Splunk's powerful search interface



Test Results

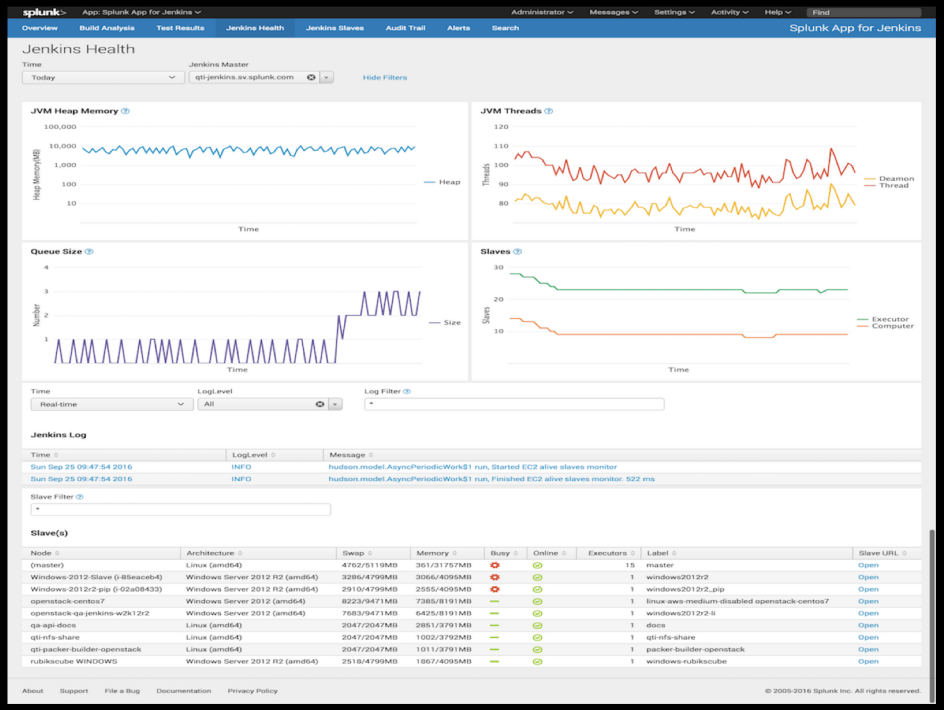
- Test Results:
  - Shows all the failing tests with stack traces
  - flags regression failures
  - groups test failures by errors
  - captures Jenkin's environment variables
  - provides nifty filters to find tests with long run times, particular errors, testsuites, etc.





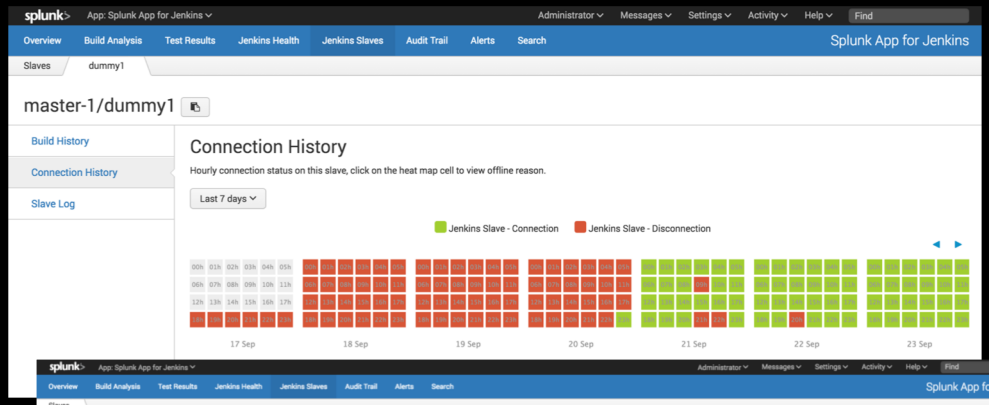
Jenkins Health

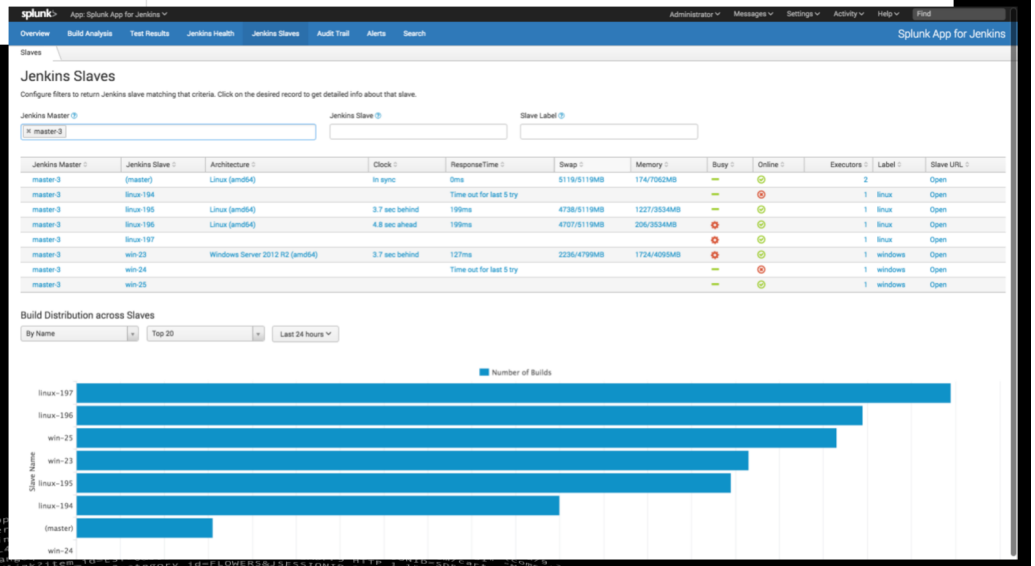
- Splunk Jenkins App captures Jenkins internal JVM information as well as keys metrics like queue size, executors and slaves stats, Jenkins master logs, and Jenkins slave stats.
- Information is captured in real-time, allowing you to quickly discover hard to find issues and fix them before they become a bottleneck for development teams.
- No more ssh-ing into Jenkins systems to find issues.



Jenkins Slaves

- Analyze all activity on a particular slave.
- View builds executed on a slave, view real-time slave logs, build activity across all slaves, and check connection history to find out unstable Jenkins slaves.
- Identify problematic components in a Jenkins cluster and optimize your team's throughput.

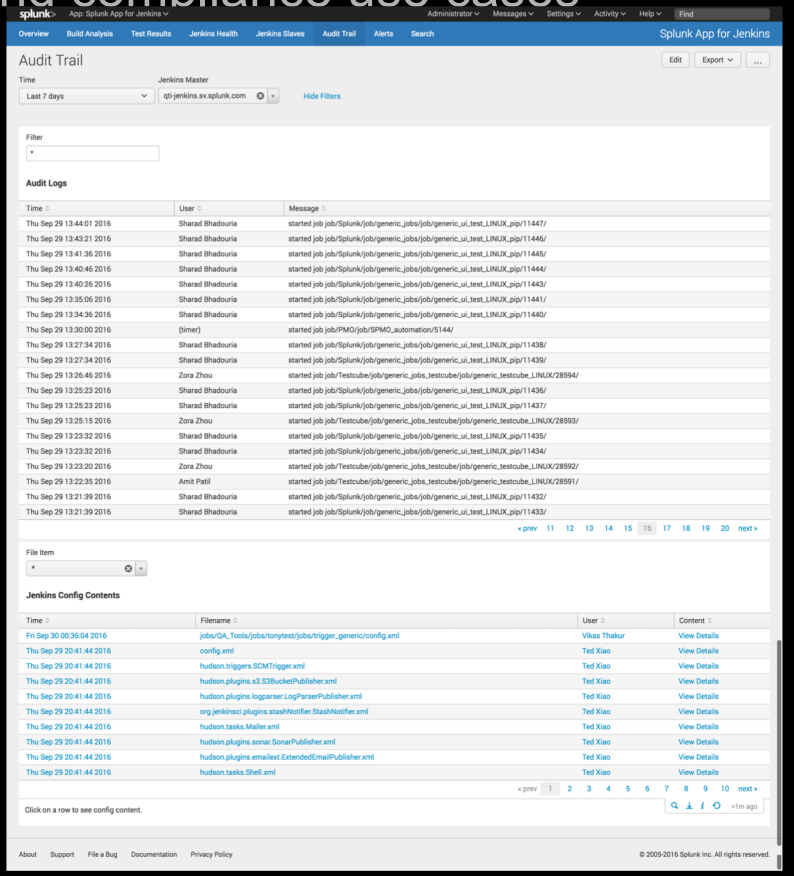




Audit Trail

Useful for organization with security and compliance use cases

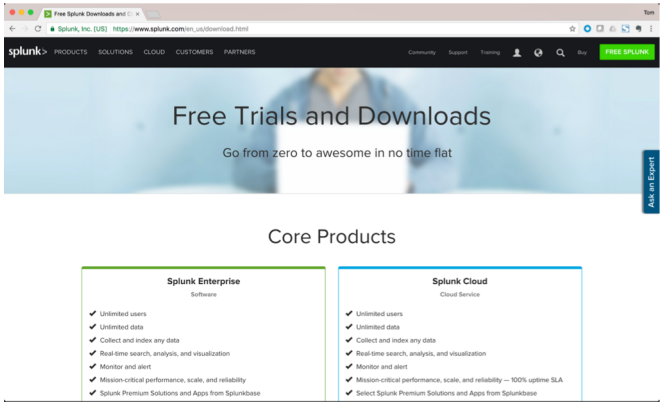
- See who has logged into your Jenkins system and performed any activity like starting, aborting, changing jobs.
- Examine which configs have been changed by some user and can view the config xml directly in Splunk.

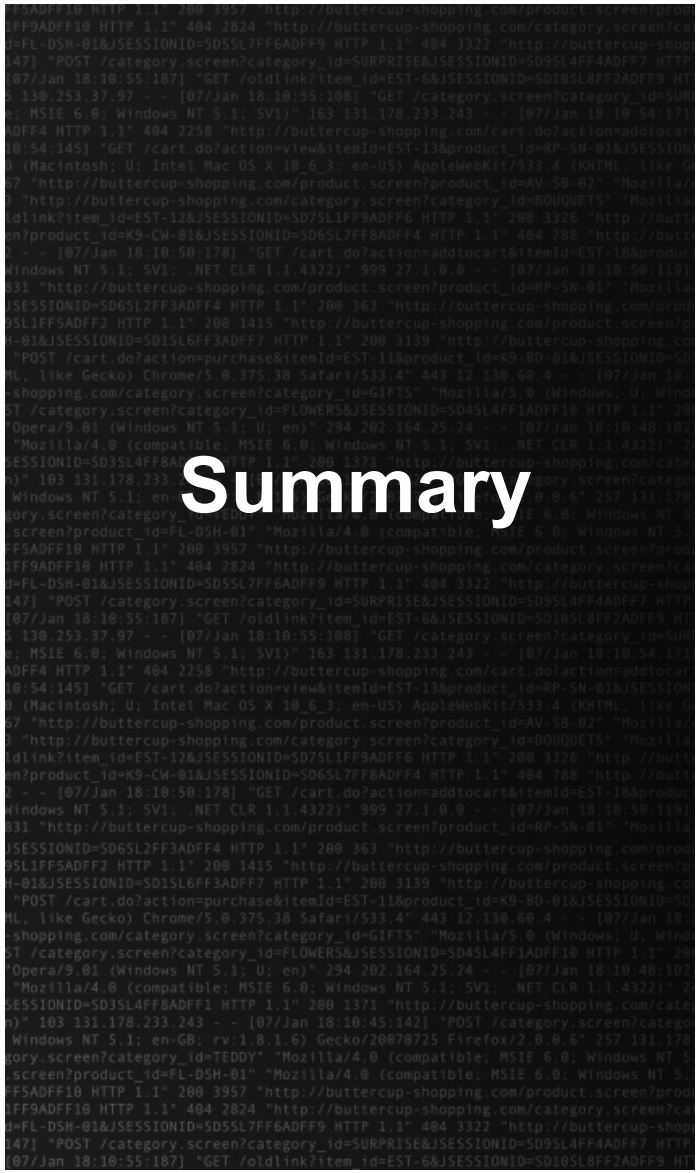


You Can Get Started *for Free!*

- First: Download free Splunk trial: www.splunk.com/download
- Then: Download Splunk App for Jenkins: splunkbase.splunk.com/app/3332

- DevOps solutions on Splunk for DevOps webpage: www.splunk.com/devops
- Free Splunk DevOps ecosystem apps: splunkbase.splunk.com
- Splunk Community: www.splunk.com/community





Summary

- The story of Jenkins at Splunk
- How we tamed our Jenkins infrastructure
- What you can do to tame your Jenkins!