

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, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk> and Turn Data Into Doing 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. © 2022 Splunk Inc. All rights reserved.

Reducing Time-to-Resolution with Interactive Runbooks

OBS1663C

Mike Kruze

Principal Engineer | Splunk



splunk> .conf22



Mike Kruze

Principal Engineer | Splunk® Observability

About Me

I'm a backend software engineer working on Splunk® Observability

I'm part of the on-call rotation

I'm a service owner, so I can get paged even when I'm not on-call

What is a Runbook?

A list of steps to debug and remediate a problem, usually to be run by an on-call engineer when they get paged

On-call engineers may not be familiar with all system components

Often a static document or wiki that must be maintained

Runbooks the Old Way

The dreaded 2 AM wall of text:

Quantizer dropped points
<https://mon.signalfx.com/#detector/v1/D1Fh4DsAsAA/edit>

Not to be confused with the "Ingest TStrouter data plane drops" detector, there is an interactive dashboard to walk you through the steps for debugging the detector, and post a message in #se-observability (slack channel). You should walk through the steps outlined there and contact the #se-observability.

It could be due to some org sending data in a way they should not. Wait for a few minutes, and post a message in #se-observability (slack channel).

When notifying #se-observability, they will need to know:

1. The customer we are dropping for: <https://mon.signalfx.com/#chart/v2/D2ISHRSAsAA>
2. The reason we are dropping for (look for a correlated spike in one of the reason charts)
3. When did the drops start? (zoom out on the above charts)
4. Example MTSES that are being dropped. Read how to parse the quantizer logs: <https://mon.signalfx.com/#chart/v2/Ds1KtYBAKAA>

Quantizer logs only every 1000000th dropped datapoint, and since generally only a few MTSES would correct realm name and aws_tag_service filter) <https://mon.signalfx.com/#chart/v2/Ds1KtYBAKAA>

Dropped 34800000th point : 0 (2019.03.09-21:35:04.532)

where is the MTS ID being dropped and is the token ID for the org who is sending the data

For AWS EKS realms, try running:

```
kubectli logs (pod-name) -e (realm-name) -n (namespace) > /tmp/quant-logs && grep -i drop /tmp/quant-logs
```

For Kubernetes realms try running:

```
kubectli logs (pod-name) -e (realm-name) > /tmp/quant-logs && grep -i drop /tmp/quant-logs
```

To fix this, increase the value of `sf.quantizer.server.batch.quantization.policy.per.vnode` in both quantizer paths. No restarts are required. The alert should clear 10 minutes after setting the new value. If you are using the sfc CLI instructions are provided below, you can also use the sfc UI if you want.

Quantizer CPU Utilization % Detector

Check if a quantizer died or recently restarted. If so, this is expected on the pod that restarted or 2-3 other pods that a dead node's vnodes shuffled to. The spike should last 5-15 minutes and then clear. If the alert doesn't clear within 15 minutes, page the metric data team escalation on-call. Nothing more to do.

If no quantizers have died or restarted recently, check if it's just 1 quantizer with high CPU or several. If it's just 1, most likely the underlying node will need to be replaced. This is easily done with a Jenkins job: <https://ci-deploy.signalfx.com/view/mon0/job/roll-eks-nodes-mon0>. (Be sure to use the job for the correct realm, mon0 given here as an example. There are 2 important settings to use for this job:

- For TARGETS be sure to specify the **node** name for the affected quantizer, not the pod name. For example `ip-10-25-130-247.ec2.internal`.
- You can get the node name from the output of `kubectli get po -o wide -l sfx_cluster=quantizer-bravo --context mon0 -n o1ly-imm` (change the realm and path accordingly)
- For TARGET_TYPE select `k8s_nodes`

This job will replace the underlying node and restart the pod. You can expect to see a spike in CPU after the pod restarts, but this will only last a few minutes and then CPU should return to normal. If it does, then nothing more to do.

Sometimes a quantizer is being restarted as part of a regular code push and we forget to mute this detector. The detector should clear within a few minutes. Check with the IMM metric data team if this is the case.

It's also possible this detector can trigger if there are zookeeper problems. Usually there will be a flurry of other alerts at the same time. If a bunch of stuff is blowing up you can always escalate to the metric data team escalation on-call.

```
# Replace "us1-sfc" with the actual affected sfc
# Get existing limit for alpha
$ us1-sfc config get -c quantizer-alpha /config/_quantizer-alpha/sf.quantizer.server.batch.quantization.policy.per.vnode
# Get existing limit for bravo
$ us1-sfc config get -c quantizer-bravo /config/_quantizer-bravo/sf.quantizer.server.batch.quantization.policy.per.vnode
# Set new limit for alpha path
$ us1-sfc config set -c quantizer-alpha sf.quantizer.server.batch.quantization.policy.per.vnode 1000000 /config/_quantizer-alpha/sf.quantizer.server.batch.quantization.policy.per.vnode
# Set new limit for bravo path
$ us1-sfc config set -c quantizer-bravo sf.quantizer.server.batch.quantization.policy.per.vnode 1000000 /config/_quantizer-bravo/sf.quantizer.server.batch.quantization.policy.per.vnode
# Don't forget to claim your change in the #olly-con
```

After doing that, notify @autobahn-society on slack that you did this.

Runbooks the Old Way

One of the most common lines in our old runbooks:

"There are multiple reasons this detector could have triggered..."

Runbooks the Old Way

"Is the problem happening on just one node or all nodes?"

"This can happen sometimes when a host unexpectedly restarts"

"Or maybe all threads are exhausted because of a large query"

"Database write errors can also cause this"

And of course it's a different remediation step depending on the reason!

Runbooks the Old Way

Having multiple reasons for an alert requires looking at a different set of metrics

Often done with static (broken!) links to dashboards

Confusion == Escalation



There Must Be a Better Way

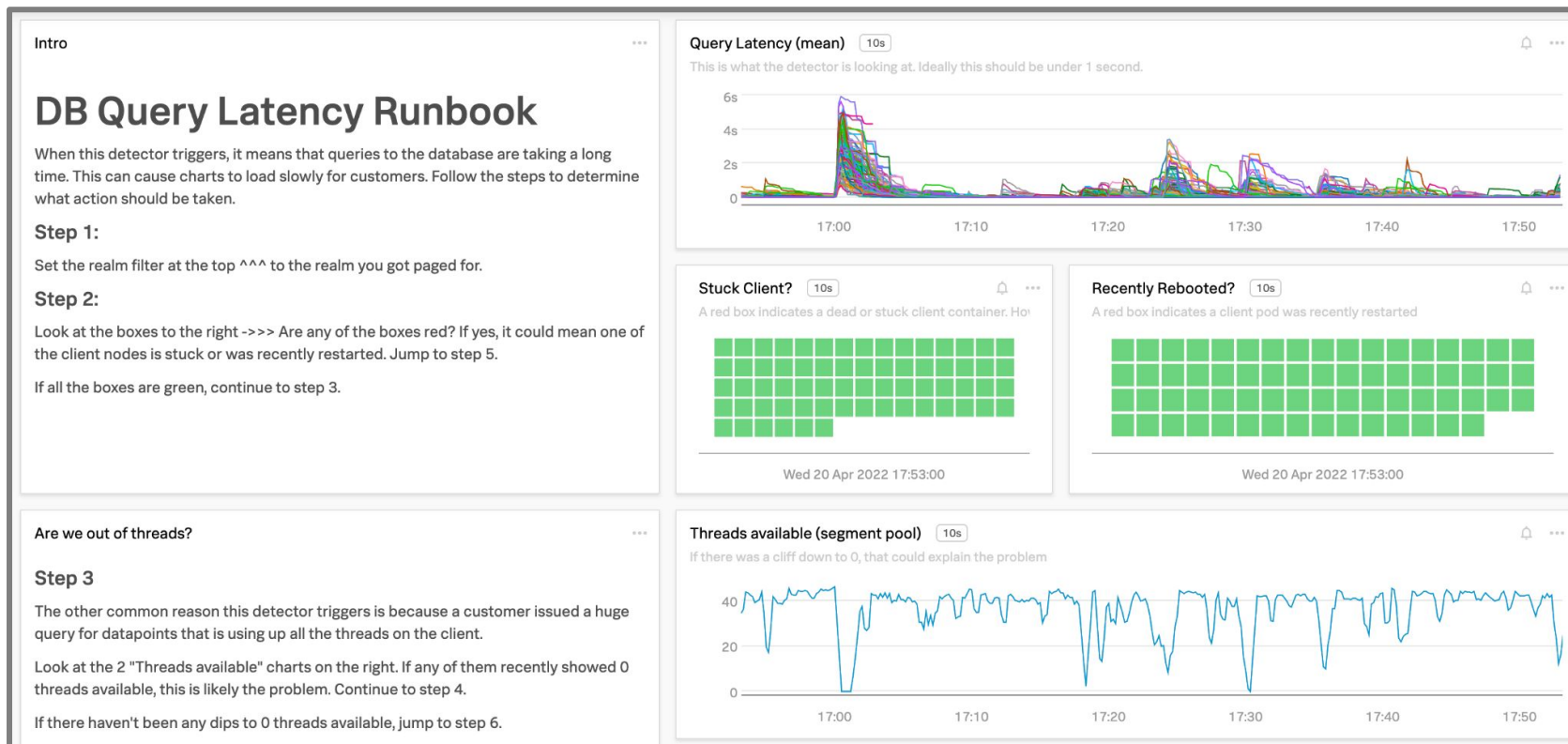
How can we improve the on-call experience for engineers?

How can we reduce mean-time-to-resolution for incidents?

How can we reduce impact to customers?

Introducing Interactive Runbooks

What if we could combine runbooks with the real-time metrics available in Splunk® Observability?

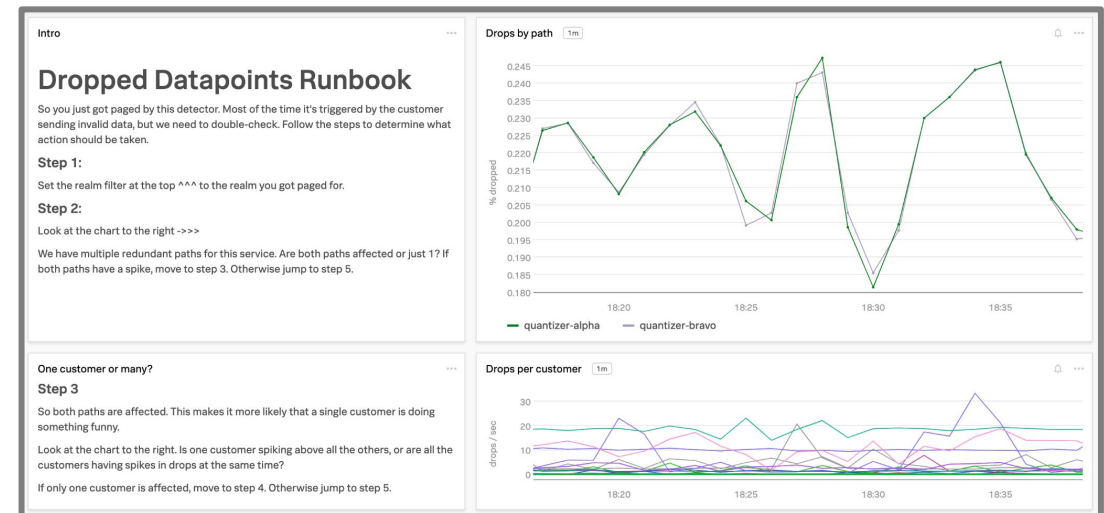
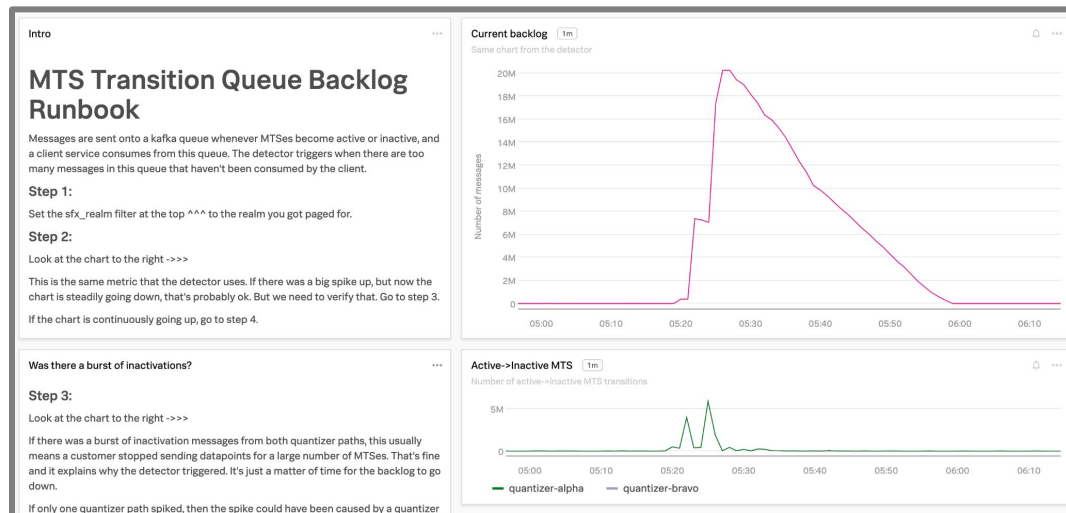


Interactive Runbooks in Summary

Increase developer efficiency and lower stress- our on-call engineers love them!

Decrease total mean-time-to-resolution

Provide a better customer experience



Thank You

