splunk> .conf2017

Measuring HEC Performance For Fun and Profit

Itay Neeman | Director, Engineering, Splunk

Clif Gordon | Principal Software Engineer, Splunk

September 2017 | Washington, DC

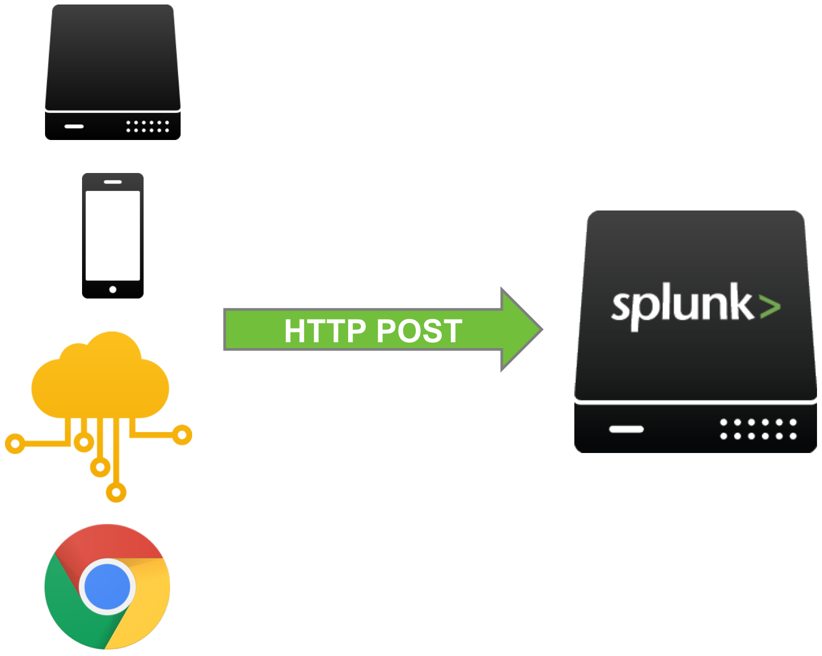
Forward-Looking Statements

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.

What the HEC Is the HTTP Event Collector?

- A simple HTTP endpoint for pushing data into Splunk
- Send events **directly** from anywhere (servers, mobile apps, IoT)
- Easy to configure and secure
- Highly scalable and performant
- Advanced features like specifying sourcetype, index, requesting ACK, etc.

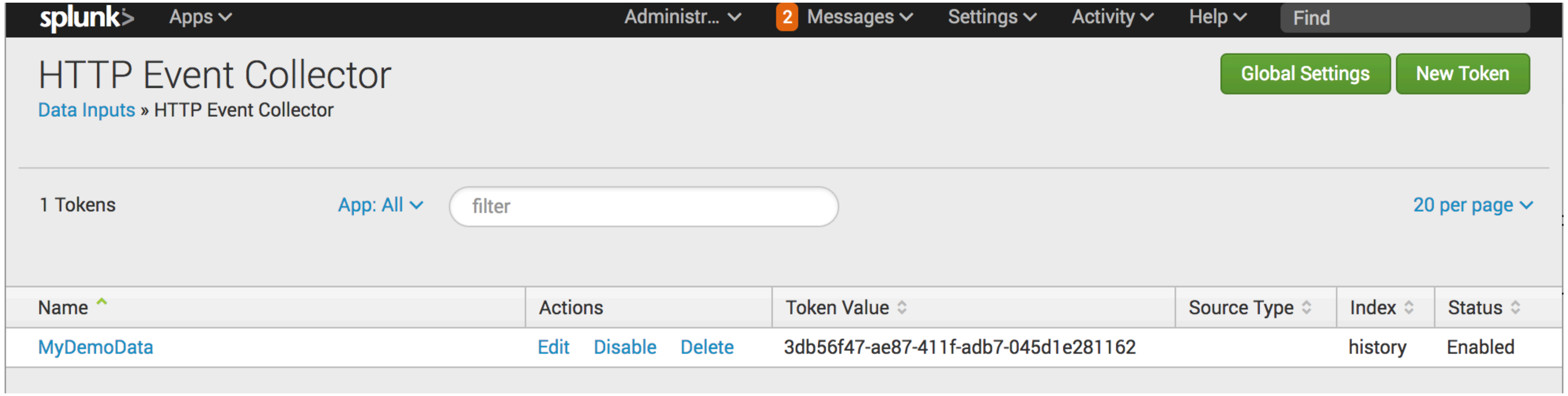


HEC in the Real World

- Customers have seen as much as 5x improvements in performance when switching from syslog to HEC
- Flexibility of HEC allows customers to ingest data from Kafka or AWS Lambda easily
- Management overhead can be greatly reduced by replacing many forwarders with just a few HEC endpoints

Enabling HEC

- Enable the HTTP Event Collector endpoint through Data Inputs (it’s disabled by default)
- Generate an authorization token



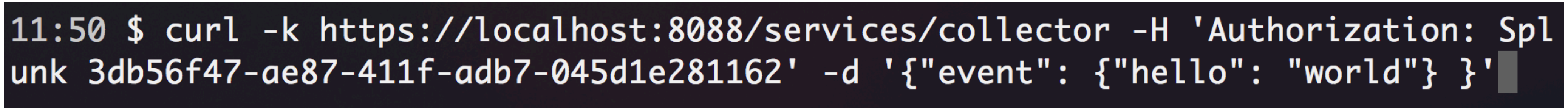
HTTP Event Collector

Data Inputs » HTTP Event Collector

1 Tokens

Name	Actions	Token Value	Source Type	Index	Status
MyDemoData	Edit Disable Delete	3db56f47-ae87-411f-adb7-045d1e281162		history	Enabled

- Use Splunk logging libraries to send data, or simply craft your own HTTP POST



```
11:50 $ curl -k https://localhost:8088/services/collector -H 'Authorization: Spl
unk 3db56f47-ae87-411f-adb7-045d1e281162' -d '{"event": {"hello": "world"} }'
```

Event Collector Performance Metrics

- The only metric that really matters is **Events Per Second** (eps)

- Secondary metrics are tracked to drive investigations and help us understand the performance of the system
  - Client network throughput (kbps)
  - Splunk CPU usage
  - Splunk Memory usage
  - Splunk Queue usage

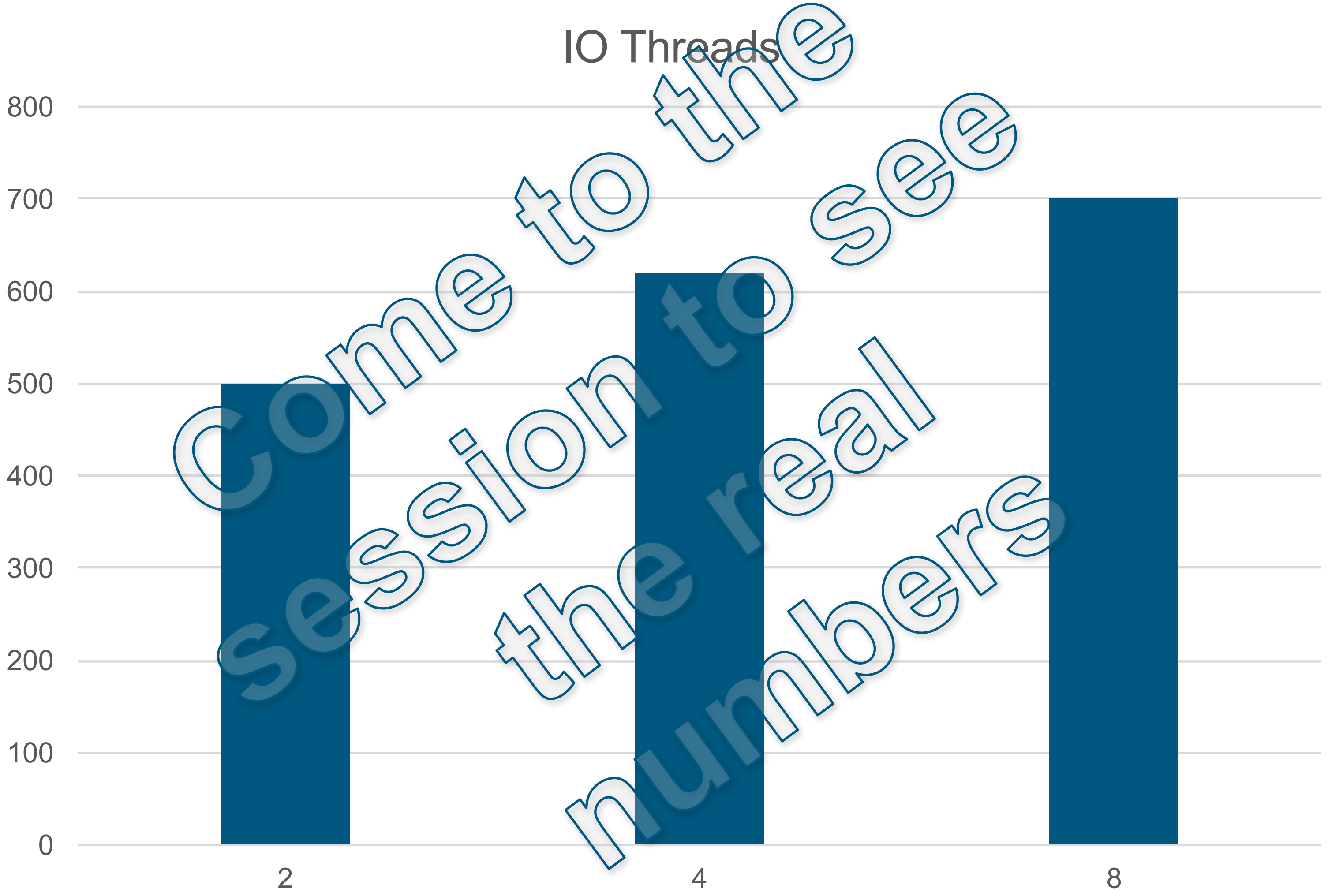
Tuning the Event Collector

- Splunk side tuning
  - Number of dedicated IO threads
  - Number of parallel ingestion pipelines
- Client-side tuning
  - Batching of events
  - HTTP Keep-Alive

Tuning HEC

Splunk Dedicated IO Threads

- By default, HEC uses one thread to handle all incoming HEC requests
- You can edit inputs.conf to raise the number of threads used by HEC to improve performance
- Recommendation: set to roughly the number of CPU cores on the machine



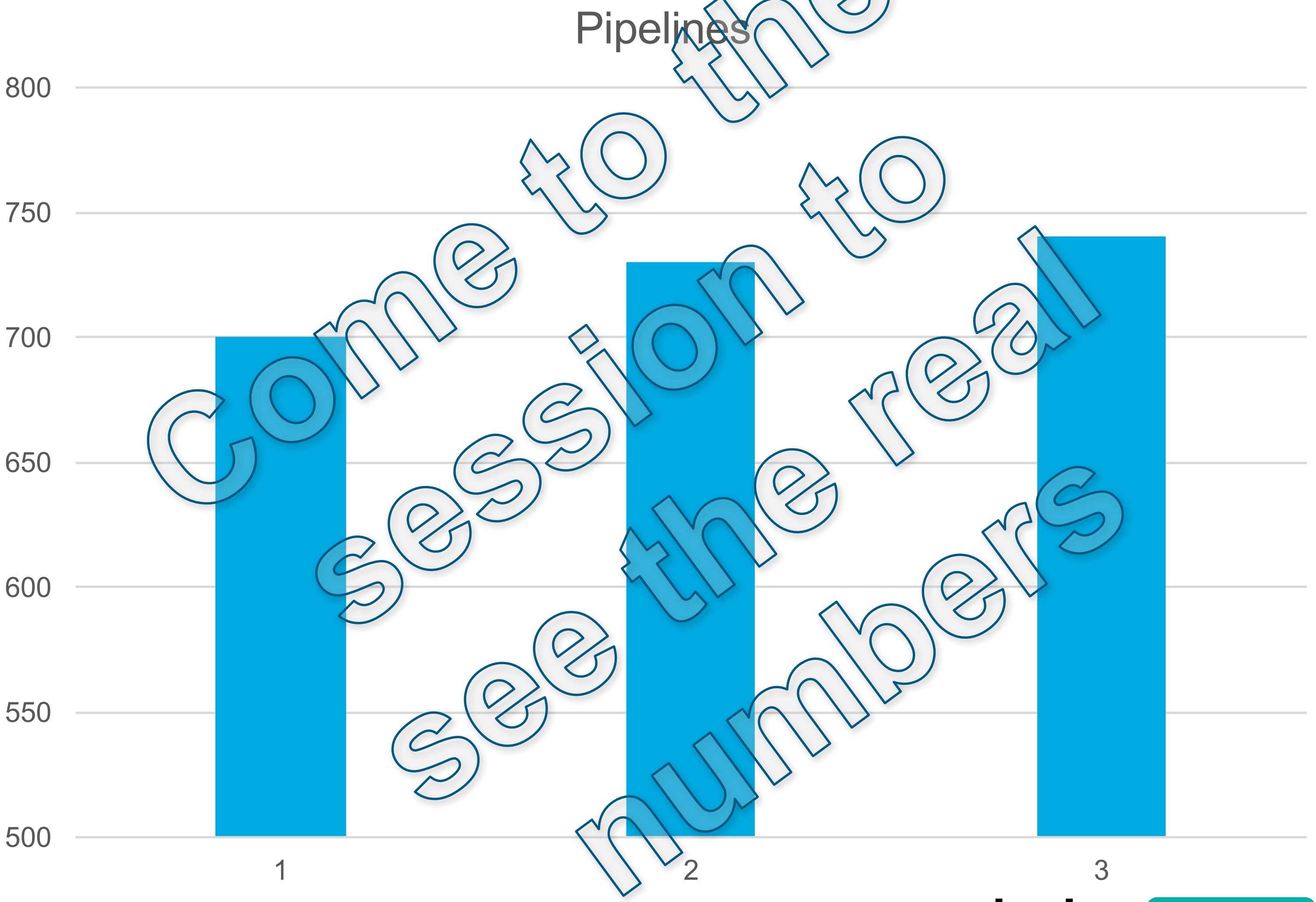
Come to the session to see the real numbers

splunk> .conf2017

Tuning HEC

Splunk Parallel Ingestion Pipelines

- By default Splunk has a single data pipeline that runs from receiving data all the way to writing to disk
- The number of pipelines can be increased in server.conf
- Especially useful if you're processing large events
- Recommendation: Depends on event type, but typically 2 pipelines

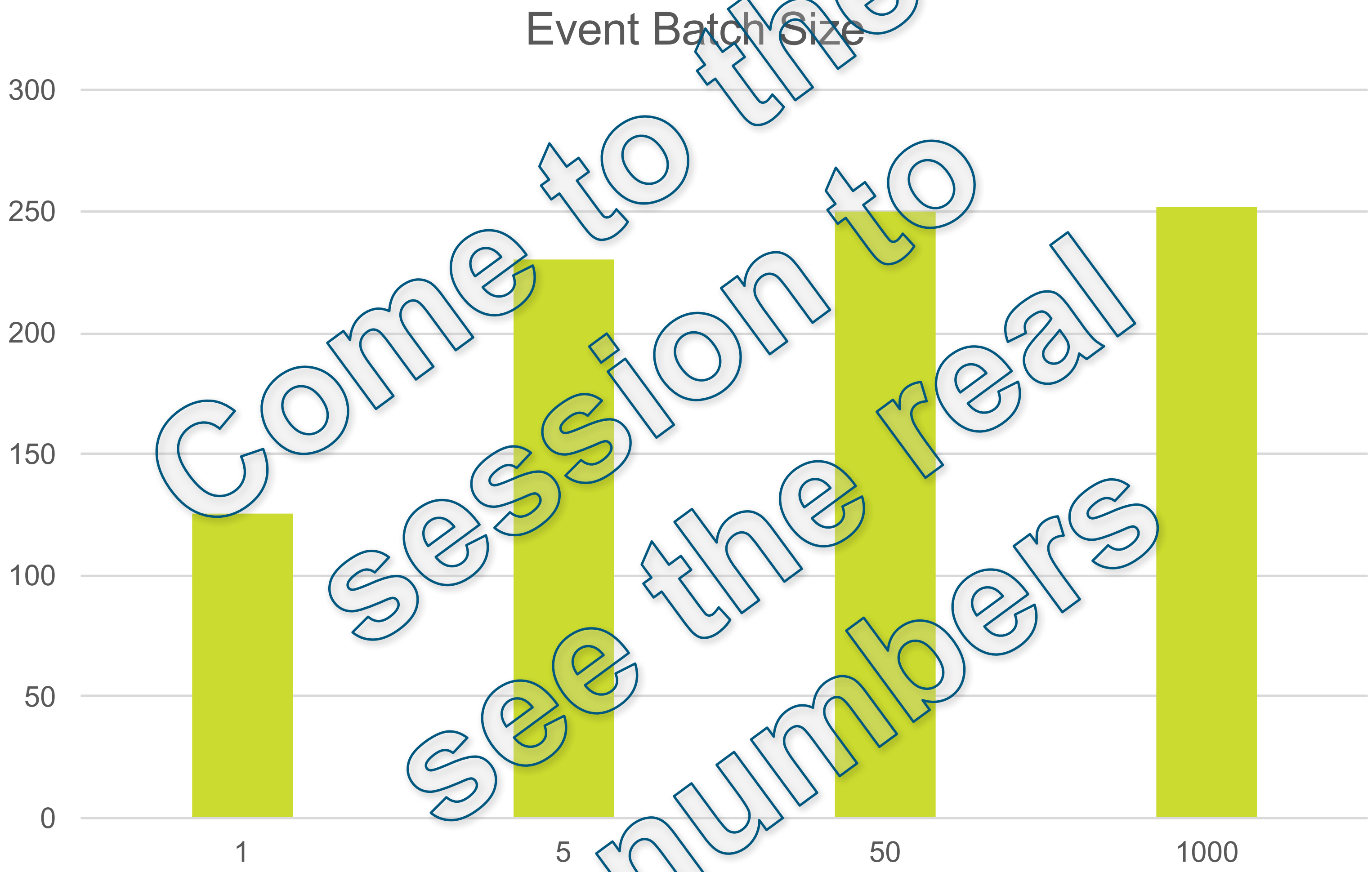


Come to the session to see the real numbers

Tuning HEC

Client Event Batching

- Batching has a significant impact on HEC performance
- The more events in a single request, the less wasted overhead
- Recommendation: Batch size between 5 and 50



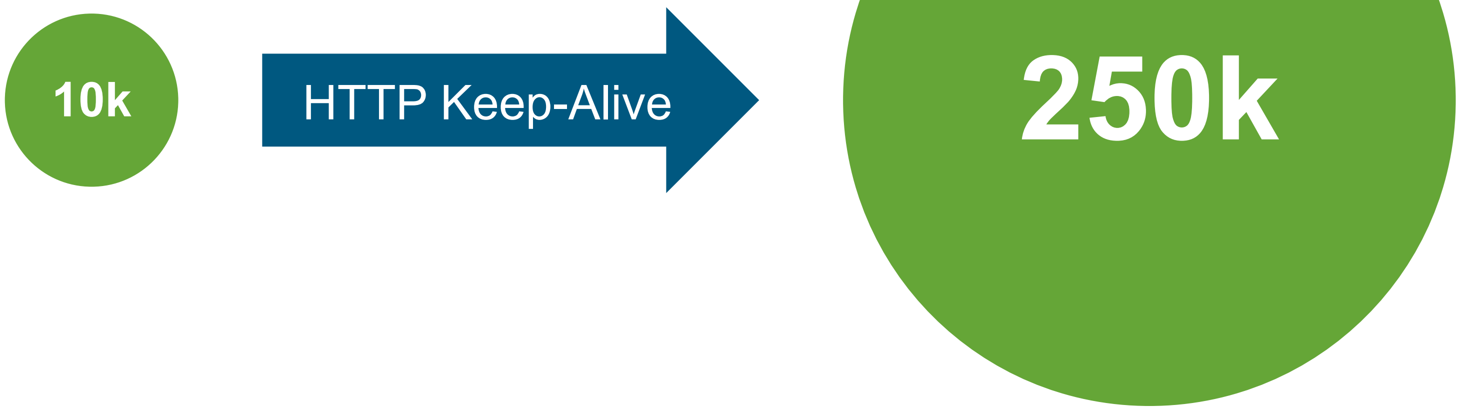
Come to the session to see the real numbers

Tuning HEC

Client HTTP Keep-Alive

- The setup and teardown of HTTP connections is expensive
- Enabling Keep-Alive allows us to reuse the same connection for multiple batches
- Recommendation: Enable HTTP Keep-Alive on clients

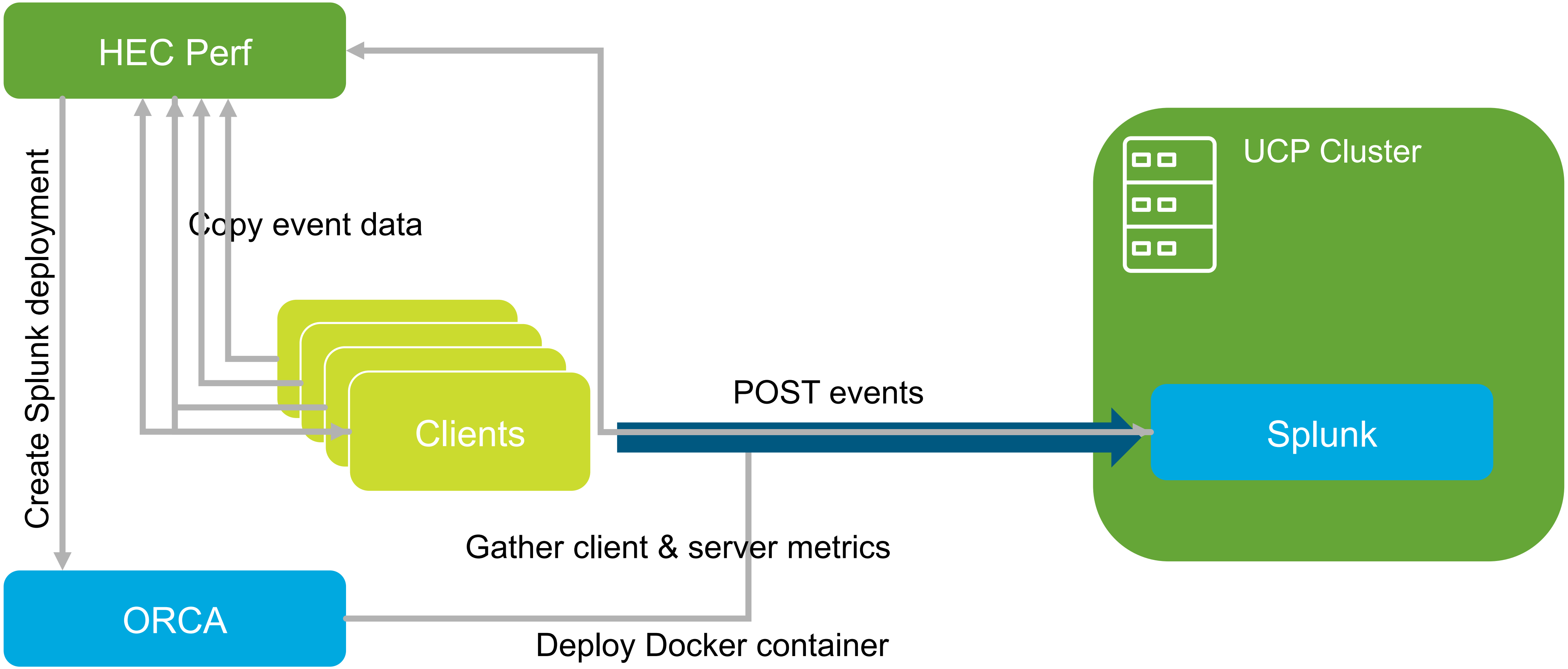
Events Per Second



HEC Performance Testing

- Nightly Splunk builds are packaged as Docker images
- Every build has a suite of performance scenarios run against it, leveraging virtualization and a cluster of high-performance test machines
- Metrics from both the client and Splunk side are gathered and stored in Splunk for analysis

HEC Performance Testing



HEC Perf

Create Splunk deployment

Copy event data

Clients

POST events

UCP Cluster

Splunk

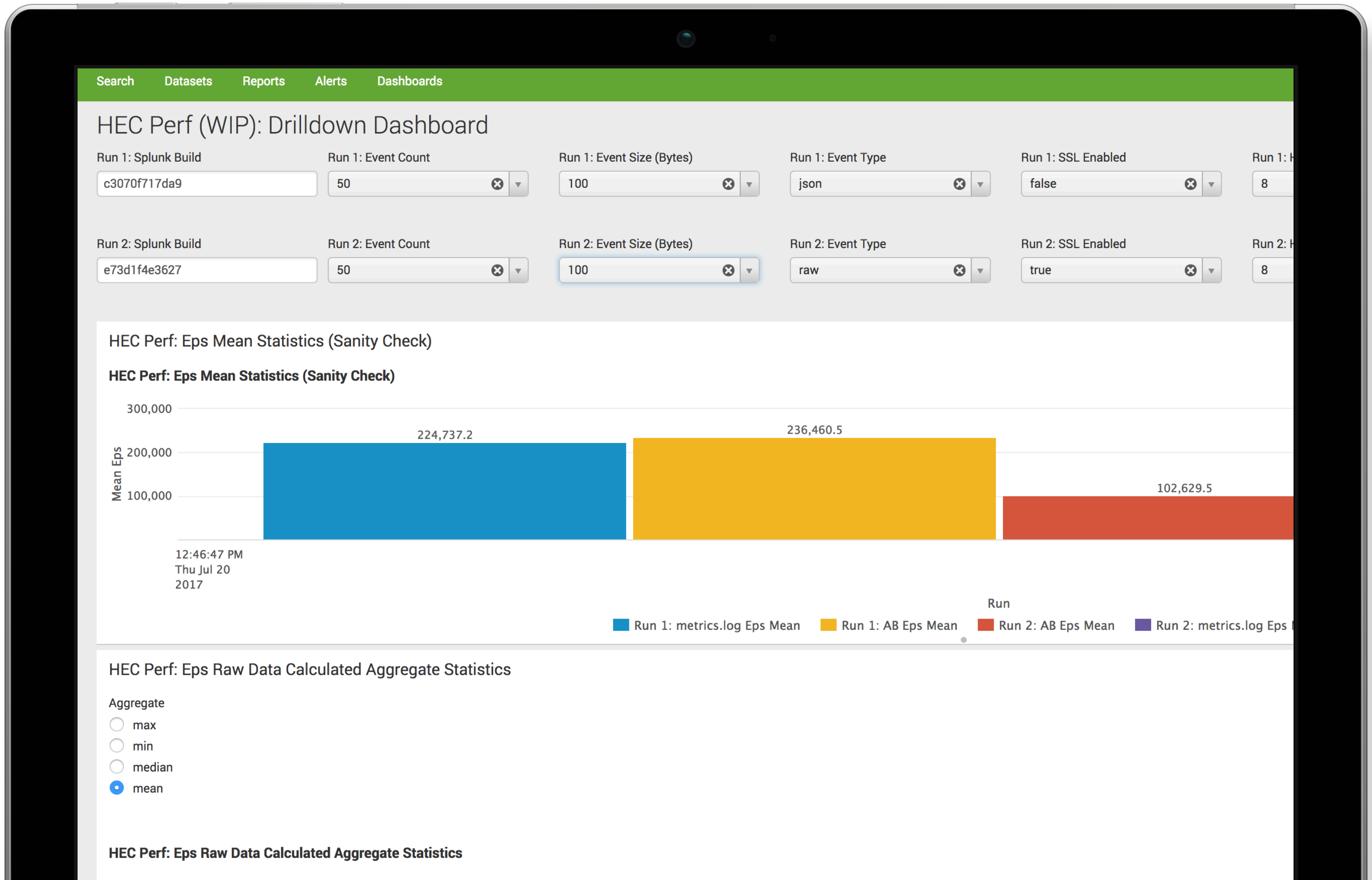
Gather client & server metrics

ORCA

Deploy Docker container

HEC Performance Dashboards

Demo



HEC Perf (WIP): Drilldown Dashboard

Run 1: Splunk Build	Run 1: Event Count	Run 1: Event Size (Bytes)	Run 1: Event Type	Run 1: SSL Enabled	Run 1: H
c3070f717da9	50	100	json	false	8

Run 2: Splunk Build	Run 2: Event Count	Run 2: Event Size (Bytes)	Run 2: Event Type	Run 2: SSL Enabled	Run 2: H
e73d1f4e3627	50	100	raw	true	8

HEC Perf: Eps Mean Statistics (Sanity Check)

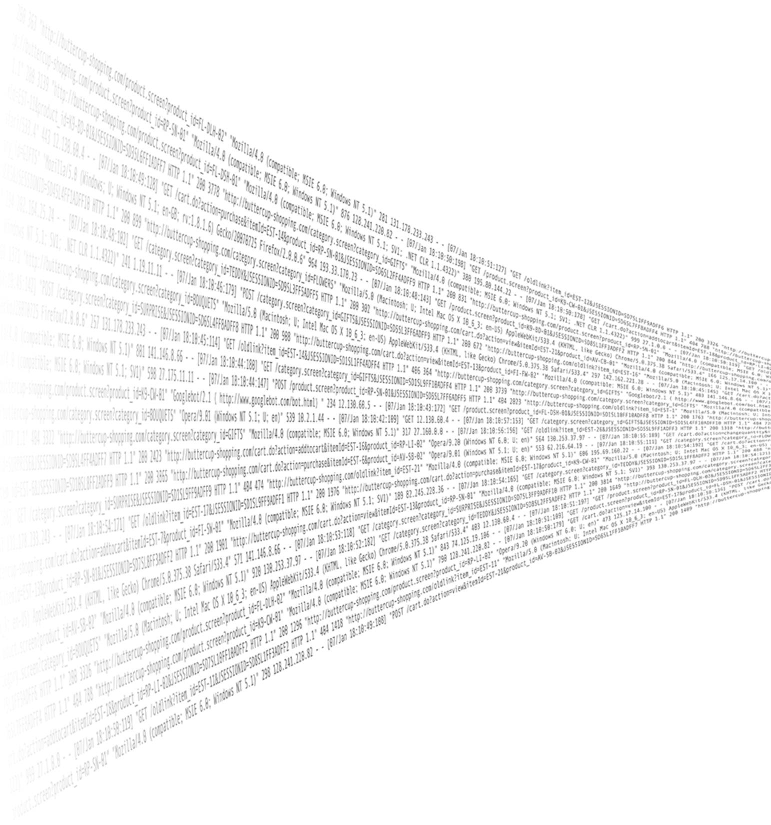
HEC Perf: Eps Mean Statistics (Sanity Check)

HEC Perf: Eps Raw Data Calculated Aggregate Statistics

Aggregate

○ max
○ min
○ median
● mean

HEC Perf: Eps Raw Data Calculated Aggregate Statistics

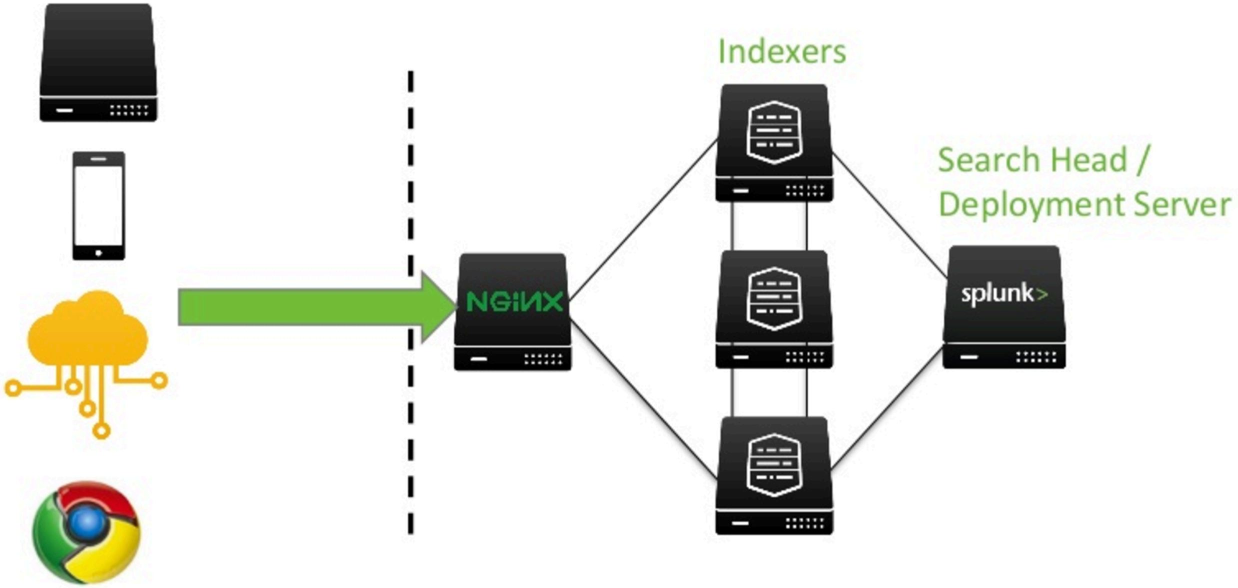


HEC Performance Recommendations

Client Batching	Client Keep-Alive	Splunk IO Threads	Splunk Pipelines
100 events/req	On	**8**	**2**

Scaling the Event Collector

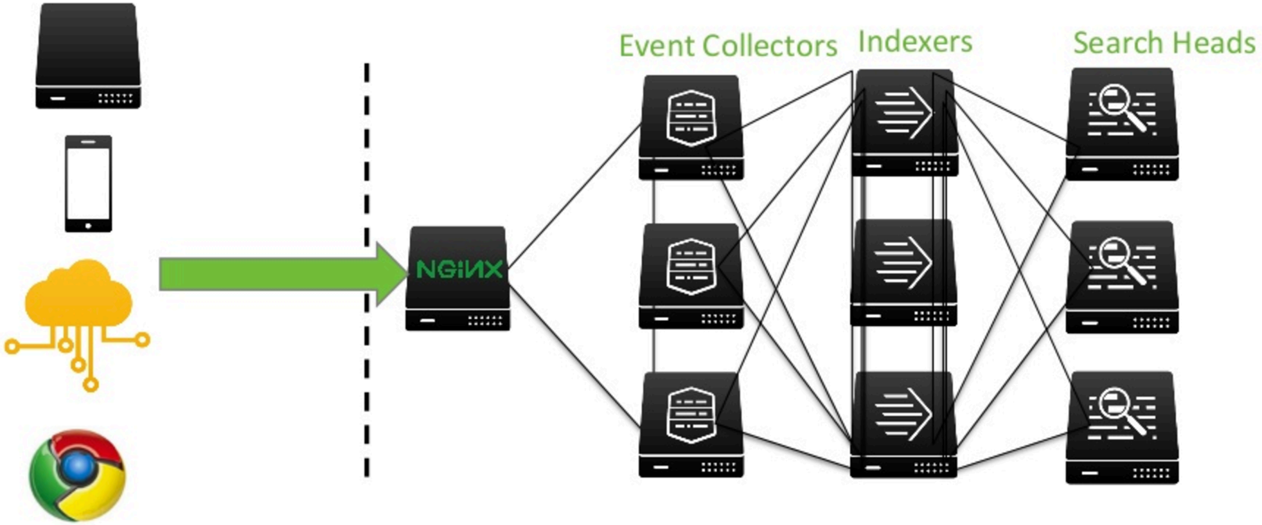
- HEC can be enabled directly on your Splunk indexers
- When you have an indexer cluster, a load balancer can distribute incoming events evenly
- However, this requires you to scale indexing to increase HEC capacity



Scaling the Event Collector

- Enabling HEC on a cluster of forwarders dedicated to data ingestion allows independent scaling

- This topology avoids any conflict between optimizations for event collection vs. data indexing and searching



Next Steps

Learning how you can leverage HEC and tune it for your needs!

- HEC Developer Docs: http://dev.splunk.com/view/event-collector/SP-CAAAE6M
- Configuring a pool of HEC forwarders: http://dev.splunk.com/view/event-collector/SP-CAAAE73

Q&A

Itay Neeman | Director, Engineering, Splunk

Clif Gordon | Principal Software Engineer, Splunk

Thank You

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

splunk> .conf2017