# Streaming Telemetry Really cool stuff or just another buzzword?

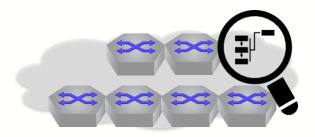
NLNOG Camp – June 2, 2018 Joris Claassen <joris@arista.com>





### Today's Telemetry Trends

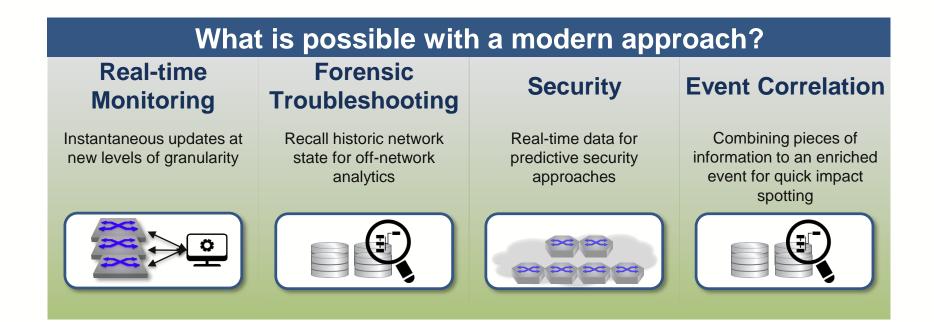




Traditional / Legacy Approach	Cloud Telemetry Requirements
1990's networking	Cloud DC Architectures
Polling Approach (5 min)	Real-time streaming
State scope limited to MIB definition	Complete state history
Per-Switch Per Device	Network-wide scope
Static, discrete events. Manually correlated	Dynamic event correlation

The Cloud has driven new telemetry approaches....

### **Telemetry Use-Cases**



### Improved visibility is broadly applicable

### **Streaming Telemetry and Analytics**

### State Streaming Infrastructure

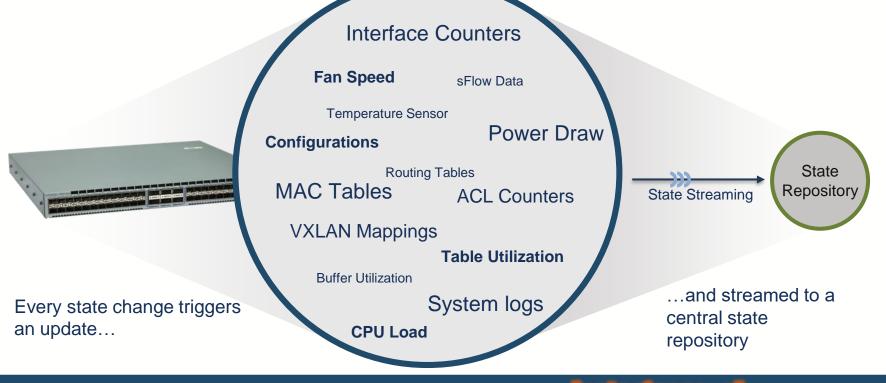
Real-time streaming of events from devices w/ Open Standards

# 2 Analytics Engine

State repository providing analytics and API's

**Telemetry Visualization** Device, Event, Metric, Topology views

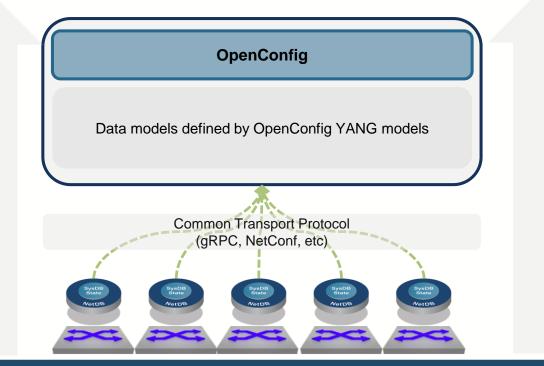
### What is State Streaming?



### Every state change. From every device. Instantaneously.

# Foundation for State Streaming

1



### **Open & Standards-based APIs.**



### Three Components to the Backend Infrastructure



High-throughput & Highly available pub/sub engine

Built on proven, scalable open source technology

Analytics Engine

Versions, aggregates, and filters raw state into actionable information:

- Track trends
- Correlate data
- Detect anomalies

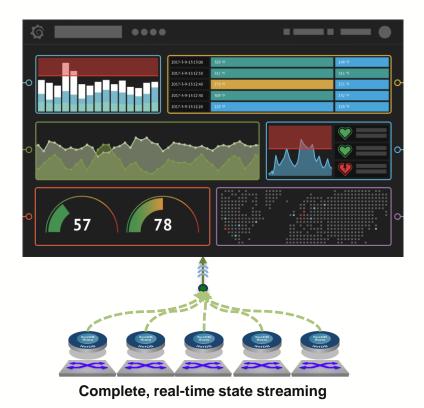
API Server

Standard APIs accessed via REST, Websocket, or gRPC

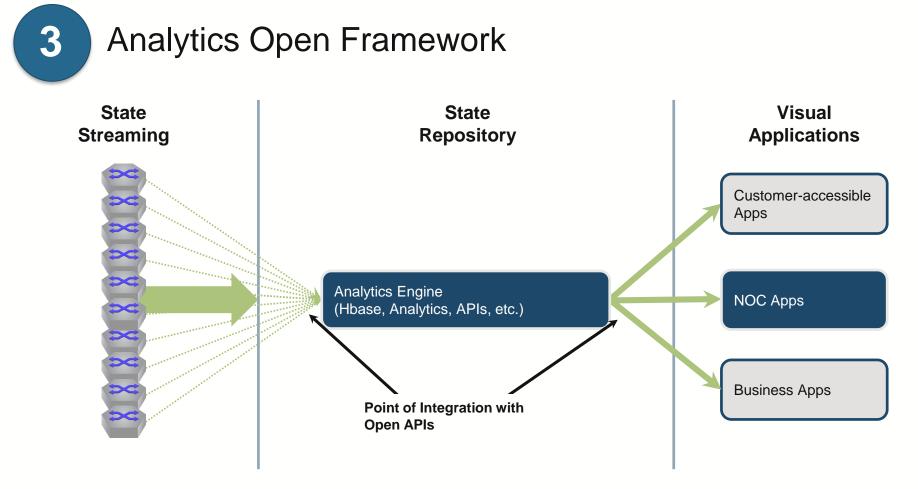
Query historical state and subscribe to streaming updates

# **Telemetry Visualization**

3

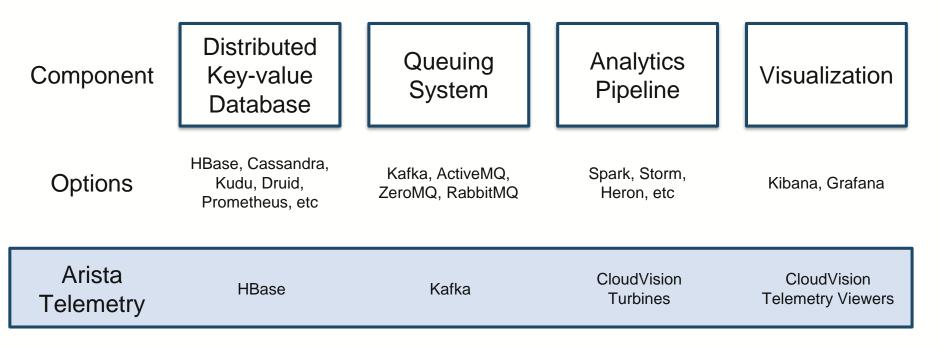


- Telemetry Apps provide front-end for visibility network state
  - Correlation of network-wide data
  - Views: Event, Device, Metric, and more
  - Timeline view for better historic troubleshooting
  - APIs for customer & partner apps



### **Build Your Own Telemetry System**

(i.e. how a hyper-scale cloud operator might build a telemetry platform)



### **Telemetry based on cloud scale approaches**

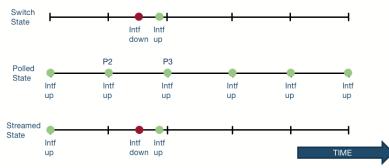
### So what to do with it?





### **Data Collection**

- Data being provided 'near real-time' (within seconds) instead of pre-defined polling intervals
- Retrieve all available data from the switch (or just the ones you like)
  - Device health (Temperature, fan, memory, CPU, power, etc.)
  - Network health (Optical levels, interface counters, ACL violations, QoS drops, etc.)
- Reduce load on collectors and network devices
  - No unnecessary information being repeatedly processed
- Keep historic values as detailed as you like
  - Aggregation of values of time is up to your collector/database, but not a must

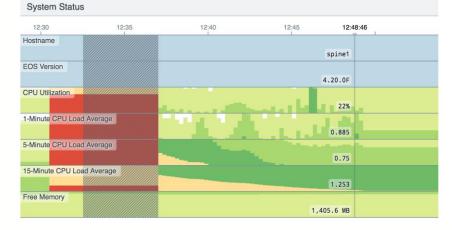


### **Data Collection**

#### A High CPU load average on spine1

Apr 10, 2018 12:48:46 CEST • a few seconds ago

Event on spine1: Device's 15 minute CPU load average exceeded threshold of 1.2



#### Processes Using More Than 5% CPU



### Monitoring / ACL counters

- Use Case
  - IXPs maintain a list of allowed/forbidden protocols on their exchange and protect the shared infrastructure with ACLs
- Reality
  - Once the customer is out of quarantine, his connection will be ACL'd but increasing counters are only being looked at when an issue occurs. This is also not something being monitored by existing SNMP solutions.
- Approach
  - Being proactively informed when a Production Customer is violating the ACLs and automatically inform him about it

### Monitoring / Microbursts

- Use Case
  - Especially with increased traffic you are likely to see more microbursts during 'release' windows.
- Reality
  - Interface counters (customer & backbone) are queried on a 1 to 5 minute average. Shorts bursts are flattened out and congestion of backbone interfaces might not be detected. This can cause severe impact to a large chunk of traffic.
- Approach
  - The Telemetry agent on the network device can provide more granular interface statistics. This can be brought down to 5 seconds per metric and enables Operations to detect congestion quickly.

# Monitoring / DDM/DOM monitoring

- Use Case
  - Over time optics may degrade on the transmit/receive side ('optic becomes blind') leading to uncontrolled outages on either the backbone- or customer-facing side.
- Reality
  - Not all vendors provide implementation of DDM-MIB on SNMP. Also due to the aggregation of data with conventional tools the usefulness is not really given.
- Approach
  - Telemetry can be combined with Anomaly Detection and/or Machine Learning technologies to provide prediction mechanisms on when an issue could arise.

### Monitoring / Proxy ARP detection

- Use Case
  - Misconfiguration of a customer interface with Proxy ARP can lead to network-wide issues and customers outages.
- Reality
  - It can be relatively easy to spot the misbehaving party, but it's hard to spot the issue in arrears. This is the case when the 'issue fixed itself'.
- Approach
  - With the historic information provided by the Telemetry database it is easy to 'go back in time' and pin down the rogue.

### Monitoring / Proxy ARP detection

Showing data from Apr 10, 2018 12:53:11. Compare data snapshots

IP Address	1	MAC Address	Interface 1	Host Route 1↓	Static Route 1
Q		Q	Q	Q	Q
172.16.112.201		2c:c2:60:d8:4e:73	Vlan12	Yes	No
172.16.200.1		2c:c2:60:56:df:93	Ethernet2	Yes	No
172.16.200.17		2c:c2:60:94:d7:6c	Ethernet3	Yes	No
192.168.0.2		2c:c2:60:ff:00:13	Management1	Yes	No
192.168.0.4		2c:c2:60:14:01:b5	Management1	Yes	No
192.168.0.5		2c:c2:60:68:de:c6	Management1	Yes	No
192.168.0.254		2c:c2:60:ff:00:36	Management1	Yes	No
					Showing 7 of 7 rows

- Use Case
  - Event generation can lead to an 'overflow of information' and takes an operator quite a while to actually find the root-cause and the customer impact.
- Reality
  - An event comes in, several commands are executed on the CLI to check customer impact and various other factors.
- Approach
  - Providing event-specific information (MAC addresses, optical levels of the interface, throughput, discards, etc.) around the device and network health with a timeline before and after the event helps to easily spot all relevant details for further troubleshooting and where to start.

#### A Syslog event detected: BGP peer changed state on leaf1

Apr 5, 2018 17:31:07 CEST • 5 days ago

Event on leaf1: BGP peer 172.16.200.1 (VRF default AS 65001) changed from Established to Idle due to Stop event.

#### **BGP** Overview

Showing metrics for VRF default

Local BGP Details					172.16.200.1 Details
17:15	17:20	17:25	17:31:07	17:35	17:15
BGP Status					BGP Peer State
			Enabled		
BGP Peers					BGP Peer Enabled State
			5 peers		
BGP Learned Paths					BGP Peer AS Number
			2 paths		
IPv4 BGP Installed Routes					BGP Peer Description
			2 routes		
IPv6 BGP Installed Routes					BGP Local Advertized Router ID
			N/A		
BGP AS Number					BGP Peer Via Local Address
			65101		
Configured BGP Router ID					
			192.168.0.14		

172.16.200.1 Details							
17	15	17:20	17:25	17:31:07	17:35		
BGP Peer State							
				Idle			
BGP Peer Enabled	State						
				Shutdown			
BGP Peer AS Numb	ber						
				N/A			
BGP Peer Description	on						
				N/A			
BGP Local Advertize	ed Router ID						
				N/A			
BGP Peer Via Local	Address						
				N/A			

BGP Peer Counts on Other Devices				
17:15	17:20	17:25	17:31:07	17:35
eaf1 (this device)				
			5 peers	
cvx01				
			N/A	
leaf2				
			3 peers	
leaf3				
			5 peers	
leaf4				
			4 peers	

Show all 7 graphs

#### BGP Learned Paths on Other Devices

	17:15	17:20	17:25	17:31:07	17:35
leaf1 (this	device)				
				2 paths	
cvx01					
				0 paths	
leaf2					
				0 paths	
leaf3					
				0 paths	
leaf4					
				0 paths	

Show all 7 graphs

#### A Syslog event detected: BGP peer changed state on leaf1

Apr 5, 2018 17:31:07 CEST • 5 days ago

Event on leaf1: BGP peer 172.16.200.1 (VRF default AS 65001) changed from Established to Idle due to Stop event.

#### Recent Routing Table Changes

IPv4	More
Change	Time
172.16.0.1/32 modified	Apr 5, 2018 17:01:17
172.16.0.2/32 modified	Apr 5, 2018 17:01:19
172.16.0.1/32 removed	Apr 5, 2018 17:01:36
172.16.0.2/32 removed	Apr 5, 2018 17:01:37
172.16.0.2/32 modified	Apr 5, 2018 17:01:45
172.16.0.1/32 modified	Apr 5, 2018 17:01:45
172.16.0.2/32 removed	Apr 5, 2018 17:20:32
172.16.0.1/32 removed	Apr 5, 2018 17:20:32
172.16.0.1/32 modified	Apr 5, 2018 17:20:34
172.16.0.2/32 modified	Apr 5, 2018 17:20:34
	Showing 10 of 10 rows

IPv6

More...

Change	Time
::/96 modified	Feb 20, 2018 21:00:30
::1/128 modified	Feb 20, 2018 21:00:30
fe80::/10 modified	Feb 20, 2018 21:00:30
::1/128 modified	Apr 4, 2018 10:46:40
fe80::/10 modified	Apr 4, 2018 10:46:40
::/96 modified	Apr 4, 2018 10:46:40
::1/128 modified	Apr 4, 2018 10:46:40
fe80::/10 modified	Apr 4, 2018 10:46:40
	Showing 8 of 8 rows

Showing 10 of 10 rows

#### (i) System reboot on leaf1

Apr 10, 2018 12:30:45 CEST • 20 minutes ago

#### Event on leaf1: Device leaf1 Reloaded

**Device Trends** 

Name	Before	After	Trend
IPv4 Route Count	22	21	-4.5%
IPv6 Route Count	(unknown)	(unknown)	-
MAC Addresses Learned	2	1	-50%
ARP Table Size	7	6	-14%
Port Channels	<u>1</u>	1	-
VXLAN Interfaces	<u>1</u>	1	-
Configured VLANs	3	3	

#### Processes

Processes Using More Than 5% CPU

No graphs to display.

Processes Using More Than 50 MB of Memory					
	12:15	12:20	12:25	12:30:45	12:35
Aaa (PID 2663)					
				122.4 MB	
Aaa (PID 2695)					
				96.2 MB	
Acl (PID 2800)					
				122.9 MB	
Acl (PID 2824)					
				124.9 MB	
AgentMonitor (P	ID 2570)				
				1.5 MB	

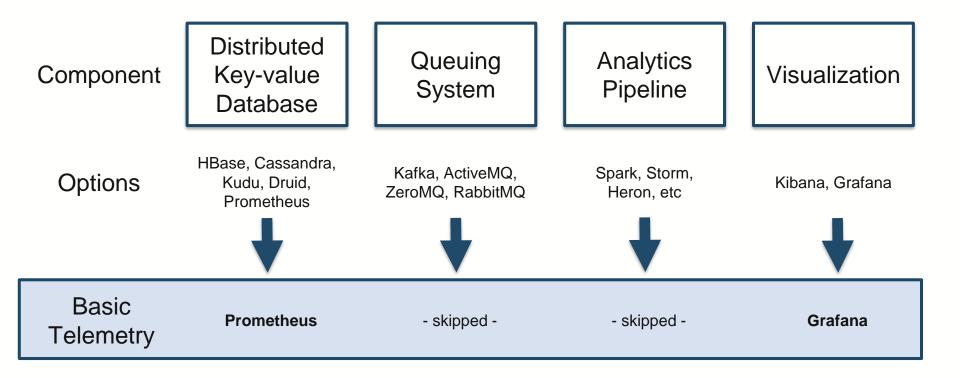
Show all 136 graphs

### How to build a (simple) Telemetry system?





### Back to the drawing board

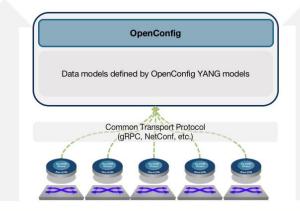


### Providing the metrics

- Prerequisites
  - You NEED a device/firmware which supports streaming in whatever way
  - Disk space and processing power on the collector
  - An idea what metrics you want to collect (KPIs)
- Things to look out for
  - Inform your self about the implementation on your device/vendor of choice!
    - Some vendors 'transform' internal data from another format into streaming telemetry (CLI -> Streaming or SNMP -> Streaming), others support it 'out of the box' from the switch state database.
    - $\gg$  Data might be then just as 'outdated' as SNMP in those cases
  - Licensing fees
  - Load on the device (Telemetry can be CPU-hungry)

### Providing the metrics

- Readable format to state repository
  - Convert the metrics to a format your solution can understand
- Push or Pull
  - Whilst 'push' would be the desired method, some monitoring solutions prefer 'pull' (like Prometheus)
- 'Source of Truth' should be always the same
  - One Agent should provide the switch metrics to
    - $\,\gg\,$  A system who understands the metrics as they are
    - » A converter (exporter) to a different format



### Converting the metrics to a Prometheus-readable format

- Only provide necessary metrics
  - Ability to define granular metrics you really need to not bloat your state repository
- Metrics will be provided via *http://<switch>:8080/metrics*

#### subscriptions:

- /Sysdb/environment/archer/cooling/status
- /Sysdb/environment/archer/power/status
- /Sysdb/environment/archer/temperature/status
- /Smash/counters/ethIntf
- /Smash/interface/counter/lag/current/counter
- /Sysdb/hardware/archer/xcvr/status

#### metrics:

- name: intfCounter
- path: /Smash/counters/ethIntf/FocalPointV2/current/(counter)/(?P<intf>.+)/statistics/(?P<direction>(?:in|out))(Octets|Errors|Discards)
  help: Per-Interface Bytes/Errors/Discards Counters
- name: intfLagCounter
  - path: /Smash/interface/counter/lag/current/(counter)/(?P<intf>.+)/statistics/(?P<direction>(?:in|out)) (Octets|Errors|Discards)
- help: Per-PortChannel Bytes/Errors/Discards Counters
- (...)

### **Deploying Prometheus / Grafana**

- This demo uses a 'ready-to-go' Prometheus/Grafana docker stack
- Only need to edit 'prometheus/prometheus.yml'

	ttps://github.com/veg	asbrianc/prometheus.git					
() \$ cd promethe	us						
	us/prometheus.yml						
()							
\$ docker swar	m init						
()							
\$ HOSTNAME=\$ (	hostname) docker stac	k deploy -c docker-compos	se.yml prom				
()							
	k ps prom   grep Run						
ybxe20abekqd	prom_cadvisor.bpo4ex	9k1pgdlknkkxvwh6qv0	google/cadvisor:latest	labvm Rı	unning	Running 2 hours ago	
q6x35kj8wuy9	prom_node-exporter.b	po4ex9k1pgdlknkkxvwh6qv0	prom/node-exporter:latest	labvm Rı	unning	Running 2 hours ago	
hoag8nj3gncv	prom prometheus.1		prom/prometheus:v2.1.0	labvm Rı	unning	Running 2 hours ago	
lcxocx172v2i	prom alertmanager.1		prom/alertmanager:latest	labvm Rı	unning	Running 2 hours ago	
sikfj95q1hmc	prom_grafana.1		grafana/grafana:latest	labvm Ru	unning	Running 2 hours ago	
\$ docker ps							
CONTAINER ID	IMAGE					COMMAND	
CREATED	STATUS	PORTS	NAMES				
888d3bd183f2	prom/prometheu	s@sha256:7b987901dbc44d17	7a88e7bda42dbbbb743c161e3152	2662959acd	9f35aeefb9a3	"/bin/prometheus" 2	
hours ago	Up 2 hours	9090/tcp p	prom prometheus.1.hoag8nj3gr	ncv3lohrfqr	mdtrhb		
()			_				

### Retrieving the metrics

- Define the targets (switches) in 'prometheus.yml'
- Define scraping intervals
- Prometheus will connect to the switch and retrieve all defined metrics

scrape_configs:	
- job_name: 'arista'	
scrape_interval: 5s	
static_configs:	
<pre>- targets: [`leaf1:8080',</pre>	`leaf2:8080']

Prometheus Alerts Graph	Status - Help			
Targets				
O Only unhealthy jobs				
arista (6/6 up) show less				
Endpoint	State	Labels	Last Scrape	Error
Enapoint	State	Labels		
				2.1.01
http://leaf1:8080/metrics	UP	instance="leaf1:8080"	3.595s ago	
http://leaf1:8080/metrics http://leaf2:8080/metrics	UP	Instance="leaf1:8080" instance="leaf2:8080"		
			3.595s ago	
http://leaf2:8080/metrics	UP	Instance="leaf2:8080"	3.595s ago 1.071s ago	
http://leaf2:8080/metrics http://leaf3:8080/metrics	UP	Instance="lesf2:8090" Instance="lesf3:8080"	3.595s ago 1.071s ago 2.331s ago	

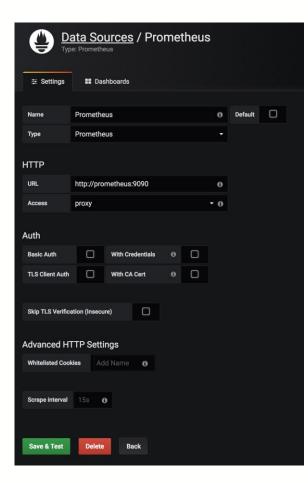
### Retrieving the metrics

Prometheus Alerts Graph Status - Help					
C Enable query history					
intfLagCounter					
Execute - insert metric at cursor - \$	Total time series: 6				
Graph Console					
Element	Value				
intfl.agCounter{direction="in*,instance="leaf1:8080",intf="Port-Channel4",job="arista",unnamedLabel1="counter*,unnamedLabel4="Discards"}	0				
intfLagCounter(direction="in",instance="leaf1:8080",intf="Port-Channel4",job="arista",unnamedLabel1="counter",unnamedLabel4="Errors"}	0				
intfLagCounter{direction="in",instance="leaf1:8080",intf="Port-Channel4",job="arista",unnamedLabel1="counter",unnamedLabel4="Octets"}	1200528				
intfLagCounter(direction="out",instance="leaf1:8080",intf="Port-Channel4",job="arista",unnamedLabel1="counter",unnamedLabel4="Discards")	0				
intfl.agCounter{direction="out*,instance="leaf1:8080*,intf="Port-Channel4*,job="arista",unnamedLabel1="counter*,unnamedLabel4="Errors"}	0				
intfLagCounter{direction="out",instance="leaf1:8080",intf="Port-Channel4",job="arista",unnamedLabel1="counter",unnamedLabel4="Octets"}	1737100				
	Remove Graph				
intfLagPktCounter	Load time: 84ms Resolution: 14s Total time series: 6				
Execute - insert metric at cursor - \$					
Graph Console					
Element	Value				
intfLagPktCounter(direction="in",instance="leaf1:8080",intf="Port-Channel4",job="arista",type="Broadcast",unnamedLabel1="counter",unnamedLabel5="Pkt"}	80				
intfLagPktCounter{direction="in",instance="leaf1:8080",intf="Port-Channel4",job="arista",type="Multicast",unnamedLabel1="counter",unnamedLabel5="Pkt"}	9131				
intfl.agPktCounter{direction="in",instance="leaf1:8080",intf="Port-Channel4",job="arista",type="Ucast",unnamedLabel1="counter",unnamedLabel5="Pkt"}	0				
intfLagPktCounter{direction="out",instance="leaf1:8080",intf="Port-Channel4",job="arista",type="Broadcast",unnamedLabel1="counter",unnamedLabel5="Pkt"}	0				
intfLagPktCounter{direction="out",instance="leaf1:8080",intf="Port-Channel4",job="arista",type="Multicast",unnamedLabel1="counter",unnamedLabel5="Pkt"}	13490				
intfLagPktCounter{direction="out",instance="leaf1:8080",intf="Port-Channel4",job="arista",type="Ucast",unnamedLabel1="counter",unnamedLabel5="Pkt"}	18				

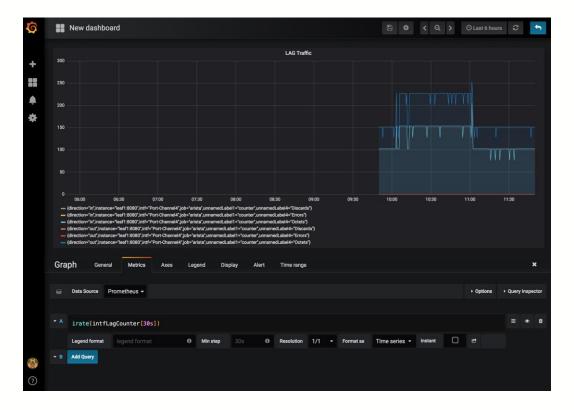
### Visualizing the metrics

- Grafana supports Prometheus natively as a data source
- Besides Prometheus a lot of other Data Sources are supported by Grafana as well

	CloudWatch
₽ Settings	Elasticsearch Graphite
	InfluxDB
Name	MySQL OpenTSDB
Hume	PostgreSQL
Туре	✓ Prometheus



### Visualizing the metrics



- Configure your dashboard(s) with the available metrics
- Auto-completion for metrics and functions is available
- If you have multiple vendors, make sure that the counters are named the same

### Vendor solutions vs. Open Source

- Essentially it depends on the man power available
- Vendor solutions provide detailed and profound understanding of events for their own devices and can correlate them 'out of the box'
- Open Source solutions can support multiple vendors in the same UI, but 'intelligence' on metrics and correlation has to be built by the end user

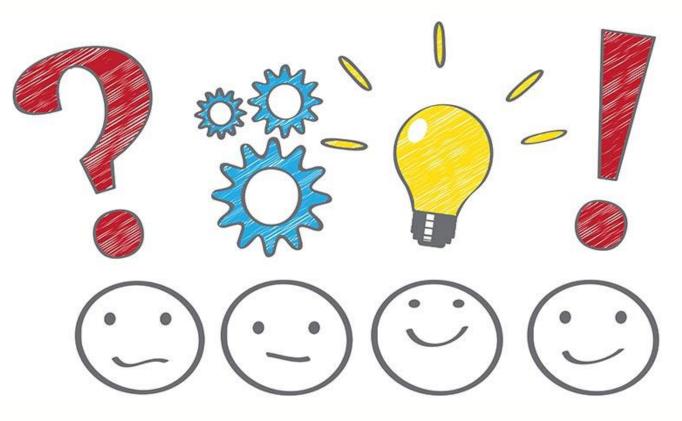
Local BGP Details					172.16.200.1 Details				
17:15	17:20	17.25	17.31.07	17:35	17:15	17:20	17:25	17:31:07	17:35
GP Status					BGP Peer State				
SGP Peers			Enabled		BGP Peer Enabled State			Idle	
Adv Heels			5 peers		bar ren craseo state			Shutdown	
3GP Learned Paths					BGP Peer AS Number				
			2 paths					N/A	
Pv4 BGP Installed Routes					B3P Peer Description				
			2 routes					N/A	
Pv8 BGP Installed Routes					BGP Local Advertized Router ID				
3GP AS Number			N/A		B3P Peer Via Local Address			N/A	
SSP AS Number			65101		BGP Peer Via Local Address			N/A	
Configured BGP Router ID			65101					N/A	
			192,168,0,14						
			192.168.0.14						
BGP Peer Counts on Othe	r Devices		192.168.0.14		BGP Learned Paths on Oth	er Devices			
BGP Peer Counts on Othe					BGP Learned Paths on Oth				
17:15	r Devices	1725	192.168.0.14	17,35	17:15	er Devices	17.25	1731:07	17:26
17:15 aft (this device)		1725	17:21:07	17:25	17:15 leaf1 (this device)		17.25	1	17:05
		1725		17.25	17:15		17.25	17:31:07 2 paths	17:35
17:15 aft (this device)		1725	17:21:07	17.35	17:15 leaf1 (this device)		17.25	1	1725
17:15 aft (this device)		1725	1721.07 5 peers	17.35	17:15 leaf1 (this device)		17.25	2 paths	17:25
17:15 aff (this device) w01 af2		1725	1721.07 5 peers	17.35	17:15 leaft (his device) cve01 leaf2		17.25	2 paths	17:25
17:15 aft (this device)		17.25	1731.97 5 peers N/A 3 peers	17.35	17:15 leaf1 (this device) ove01		17.85	2 paths 0 paths 0 paths	17:25
17:15 aft (this device) w01 af2 af3		17:25	1731.97 5 peers 8/A	17.35	17:15 lealt (this device) ovi61 leal2 leal3		17,25	2 paths 0 paths	17:25
17:15 aff (this device) w01 af2		17,25	1731.97 5 peers N/A 3 peers	17.35	17:15 leaft (his device) cve01 leaf2		17,25	2 paths 0 paths 0 paths	17.25



### References

- OpenConfig 'Streaming telemetry' definition
  - <u>http://www.openconfig.net/projects/telemetry/</u>
- Database 'connectors'
  - <u>https://github.com/aristanetworks/goarista/tree/master/cmd</u>
- Prometheus/Grafana Docker Stack
  - <u>https://github.com/vegasbrianc/prometheus</u>

### Questions?



# Thank You

Joris Claassen Systems Engineer (e) joris@arista.com (m) +31 6 48538869



