OpenNTI

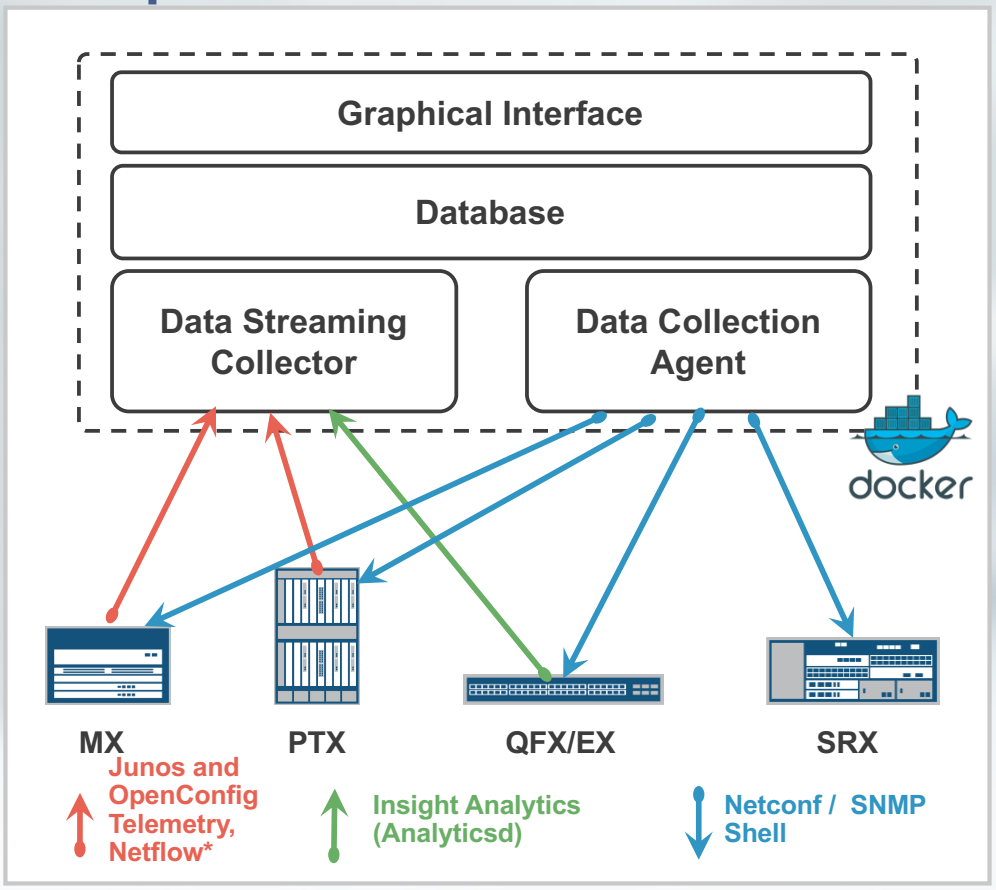
Collect and visualize KPI from Networks devices

Open Network Telemetry Insights

Efrain Gonzalez (efrain@juniper.net)
Pablo Sagrera (psagrera@juniper.net)

Version 3.0 / Oct 2017

OpenNTI / Dashboard – Collector



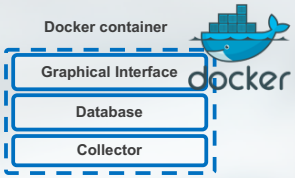
Pre-configured with all tools and with default dashboard ..
Send it data, it will graph it
Pre defined template for Cli commands

Packaged with Docker,
Easy to deploy, easy to maintain
Can run on server, on laptop .. on the device itself

Proof of concept
Accept multiple sources of Data (Netconf/ JTI / Analyticsd / OpenConfig telemetry / SNMP, Netflow *)
Can send to multiple collector / database

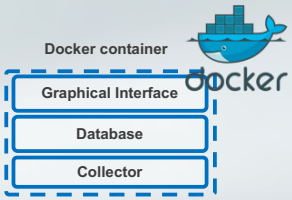
Docker Deployment options

On a server



Physical or Virtual Server w/ docker

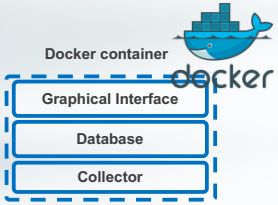
On box Guest VM



Guest VM w/ docker

QFX10K w/ JDM

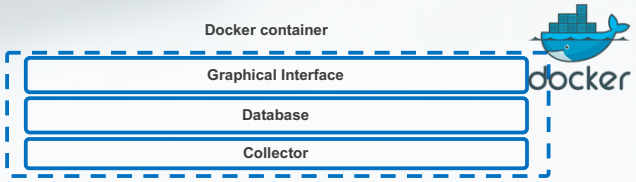
On a laptop



Docker Machine

Windows or Mac laptop

On a cluster



Physical or Virtual Server w/ docker

Physical or Virtual Server w/ docker

Use cases

POC/Demo

PS/RE

Testing/Dev

Troubleshooting

What OpenNTI is and isn't

What it's

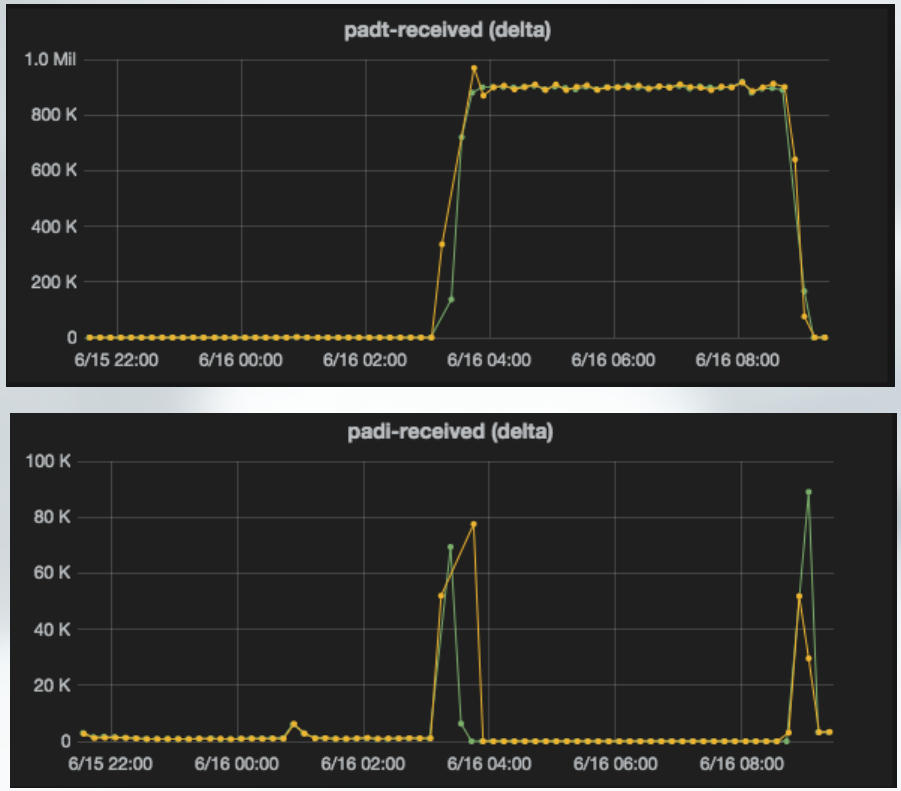
- An open source project
- Supported by the community
- Tool to collect and graph time series data
- Tool to demonstrate easily the value of telemetry

What it's NOT

- A Juniper "product"
- Officially supported by Juniper (No JTAC)
- A configuration management solution
- An analytics solution
- A replacement for CAE

The value of graphical representation

Discover how some features really works in Junos

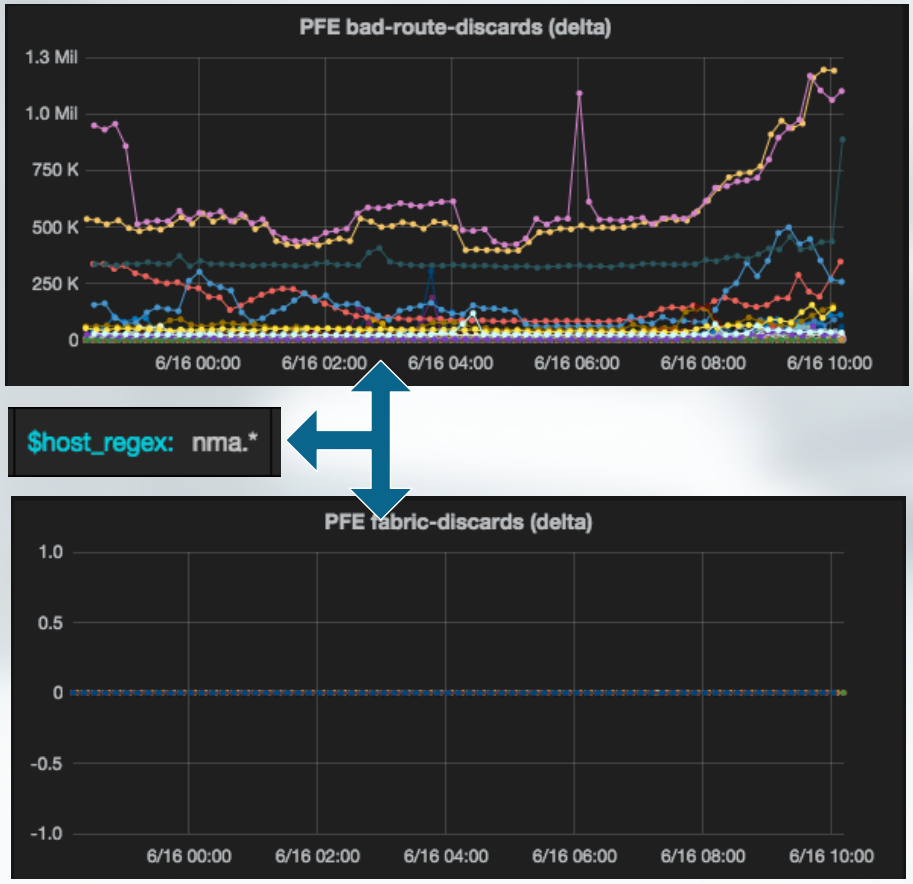


- In a BNG environment when there is a huge breakdown in the access network, a PADT and PADI storm arises.
- DDOS begun to discard based on it configuration, but we found out that there is a implicit priority for PADT over PADI, so service could not be restored
- So as workaround we redefined DDOS policers

Compare same kpi on different routers using regex

- Sometimes just one kpi does not tell the whole story, in this BNG scenario there was a massive problem in the access network.
- If we just look one kpi on one router we could not be sure where the problem resides (access network or BNG)
- If we compare same kpi among all BNG in the POP (just using a regex), we easily conclude the problem is not our BNG

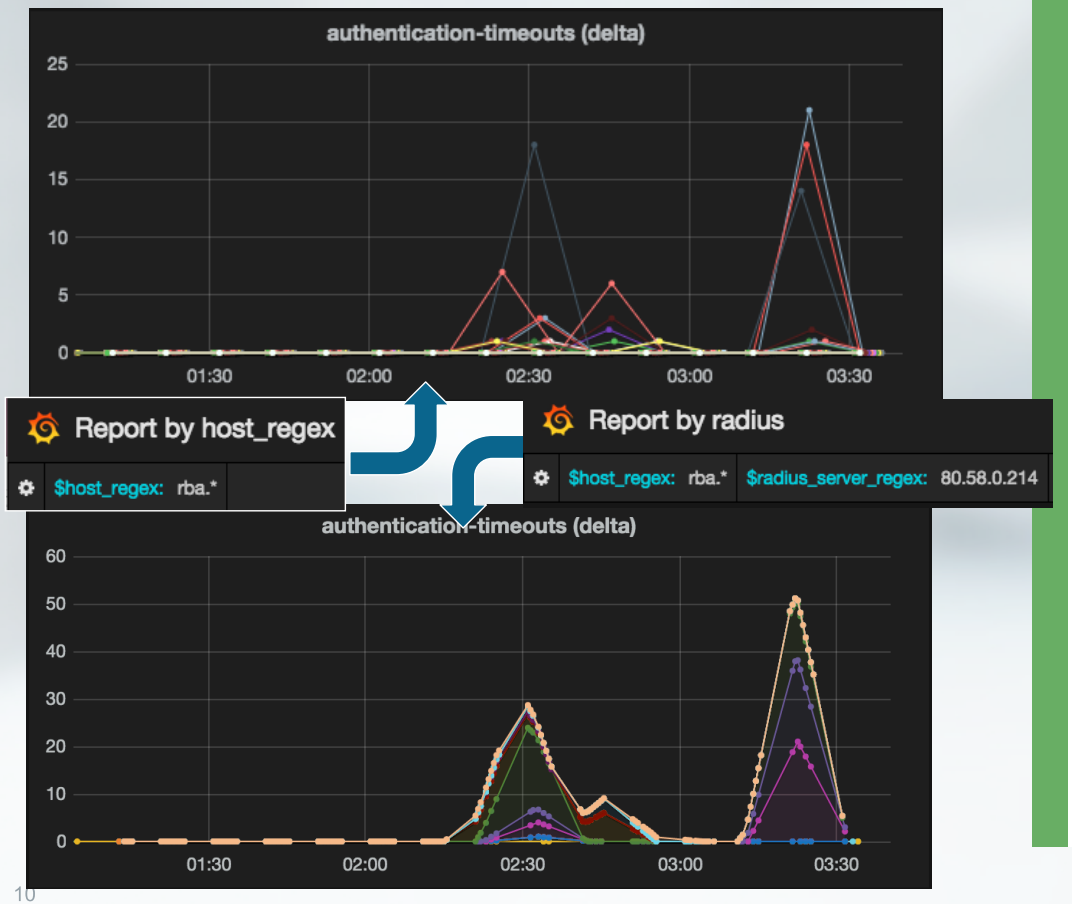
A quick overview of the whole network



- In order to speed up any kind of analysis (audit or troubleshooting), first we need the ‘big picture’.

- In this case using a different regex, we matched all business PE, just to confirm that although there were many ‘normal discards’ that need to be reviewed, there is NO fabric drops (a.k.a blackholing)

KPI analysis from different perspectives



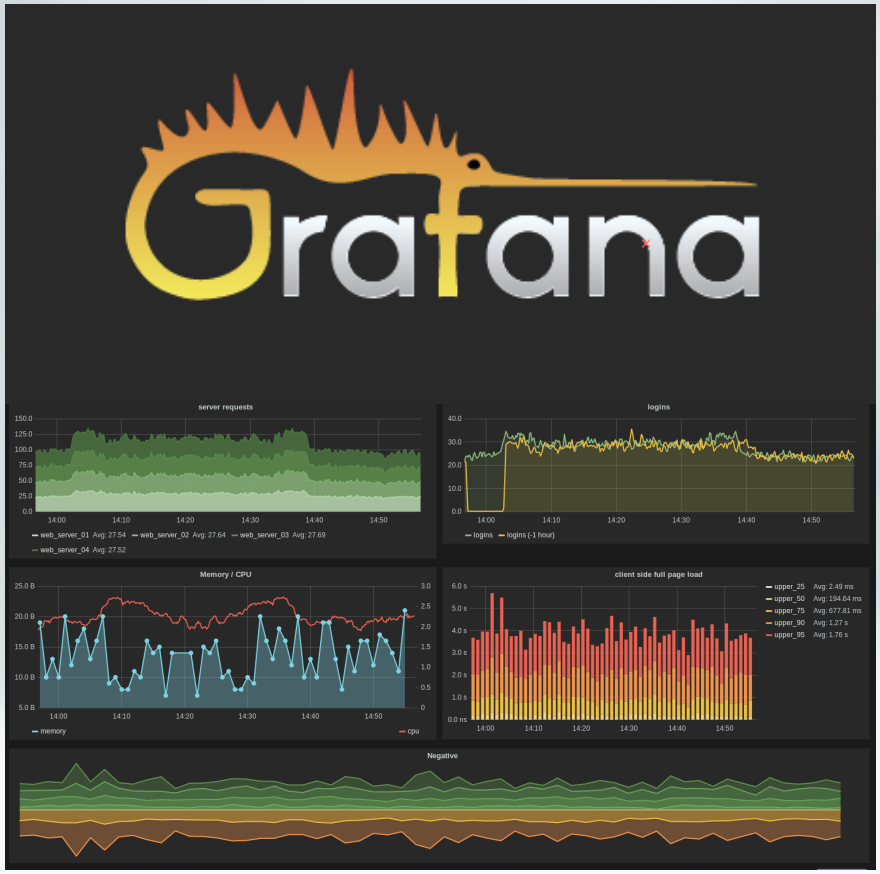
- First chart report problems in different BNGs, but so far we could not conclude anything yet, because there is anything in common.
- The second chart belong to a report that shows the same kpi but in this case group by radius server.
- Now we can conclude that there is a problem with just one radius server, that is having timeouts will many BNG

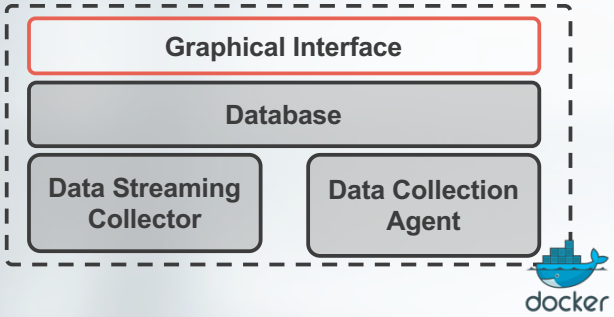
Correlate events and data

- Overlay main events (Commit or Flaps) to telemetry data can quickly help to understand the root cause.

Components

Graphical Interface – Grafana



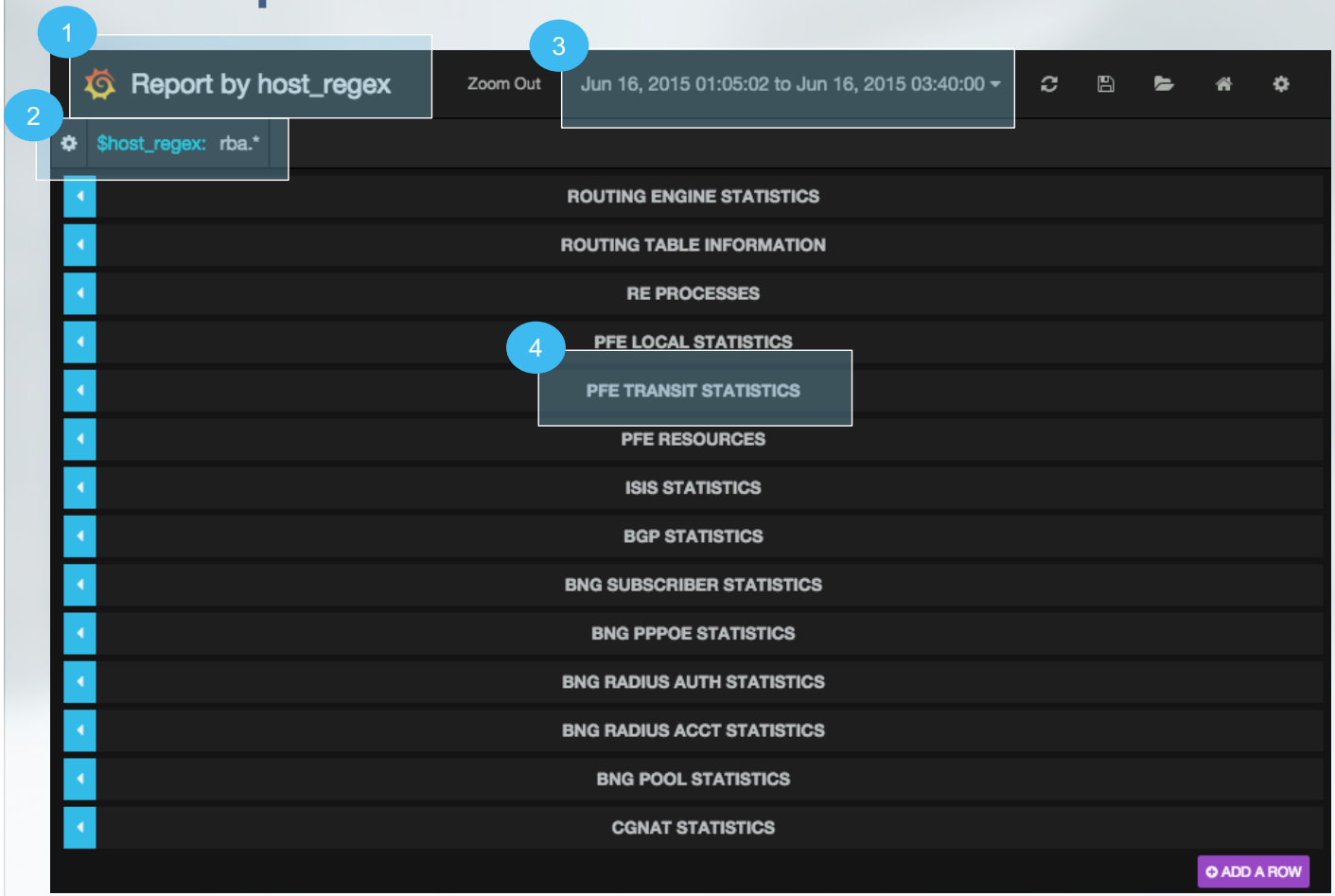


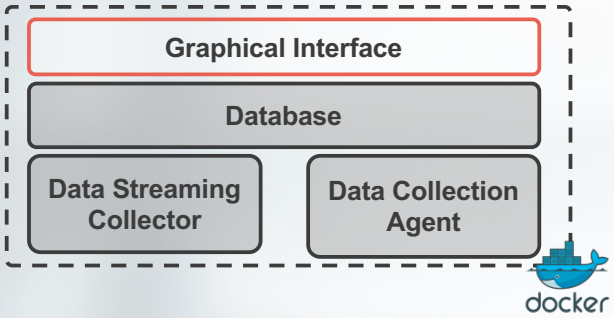
Grafana provides a powerful and elegant way to create, explore, and share dashboards and data with your team and the world

Grafana supports multiple data sources: Graphite, Elasticsearch, InfluxDB, OpenTSDB, KairosDB etc

Grafana supports RestAPI

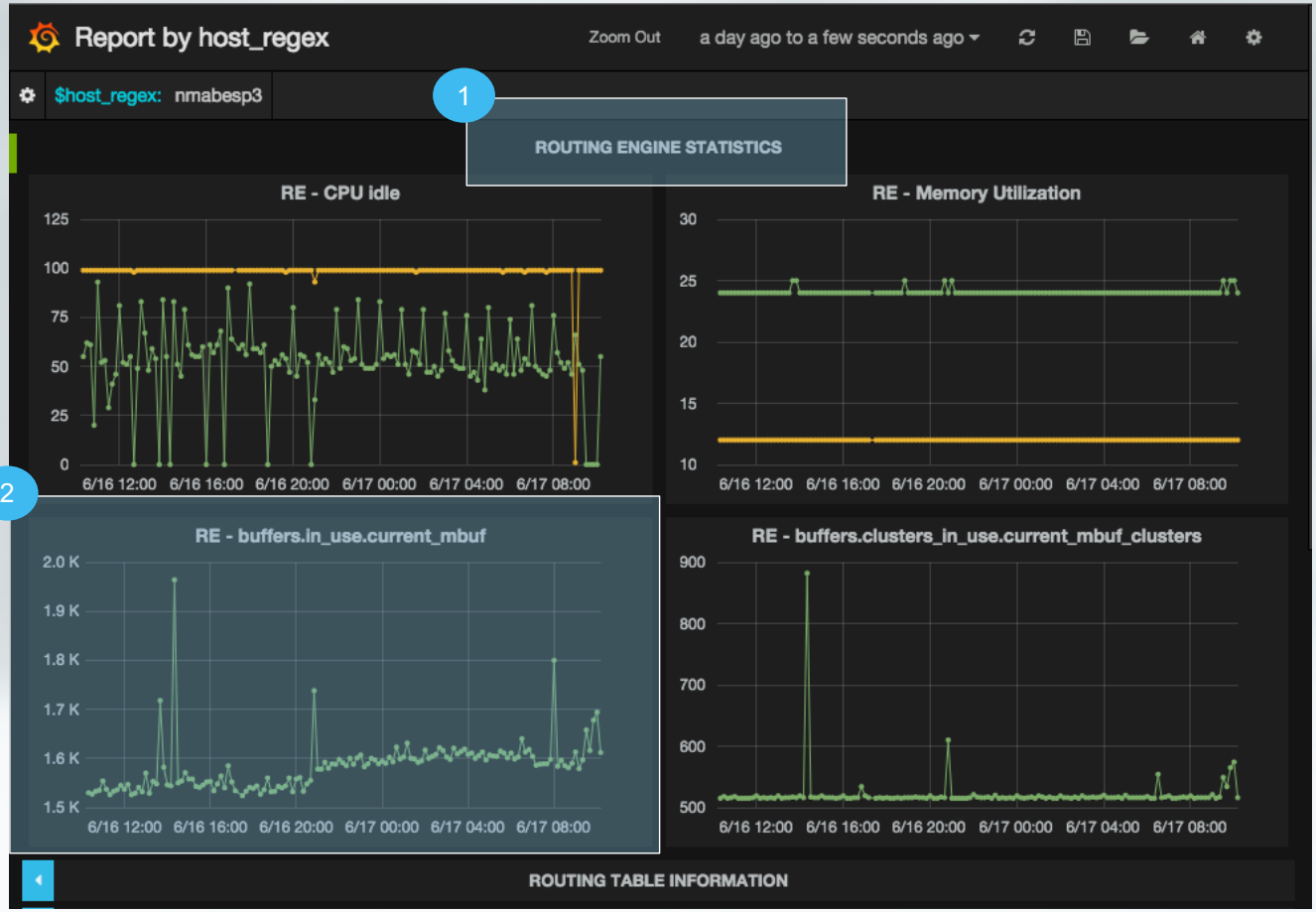
Graphical Interface - Grafana

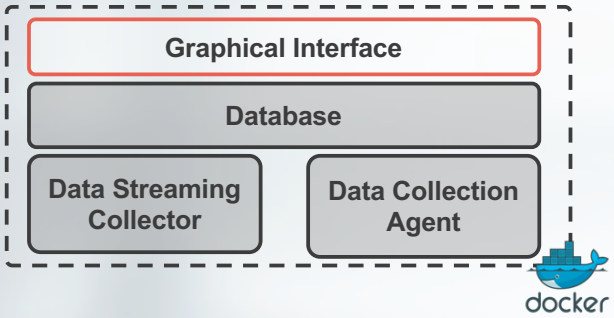




1. Report Name
2. Regex filter
3. Time range
4. KPI are organized by sections

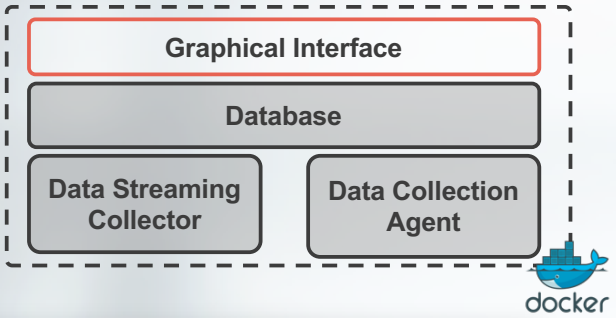
OpenNTI Web UI: Grafana





1. Expand the report section by clicking on it title
2. Each graph represent a query to the DB

OpenNTI Web UI: Grafana



Overlay events on top of your data

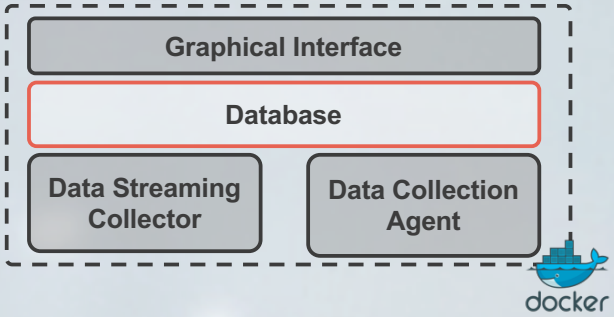
In this example :

- Commit are represented in **Blue**
- Error are represented in **Red**

Events can be added :

- With a syslog message
- With a REST call (shell, Ansible, Junos)

Database – Influxdb

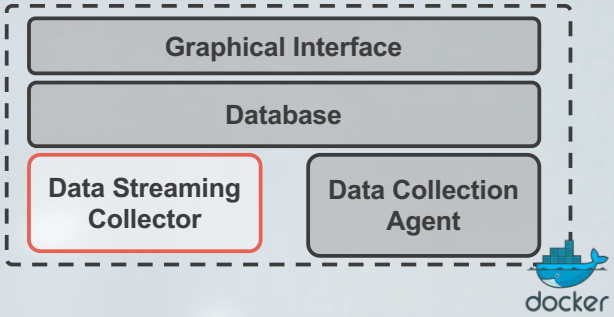


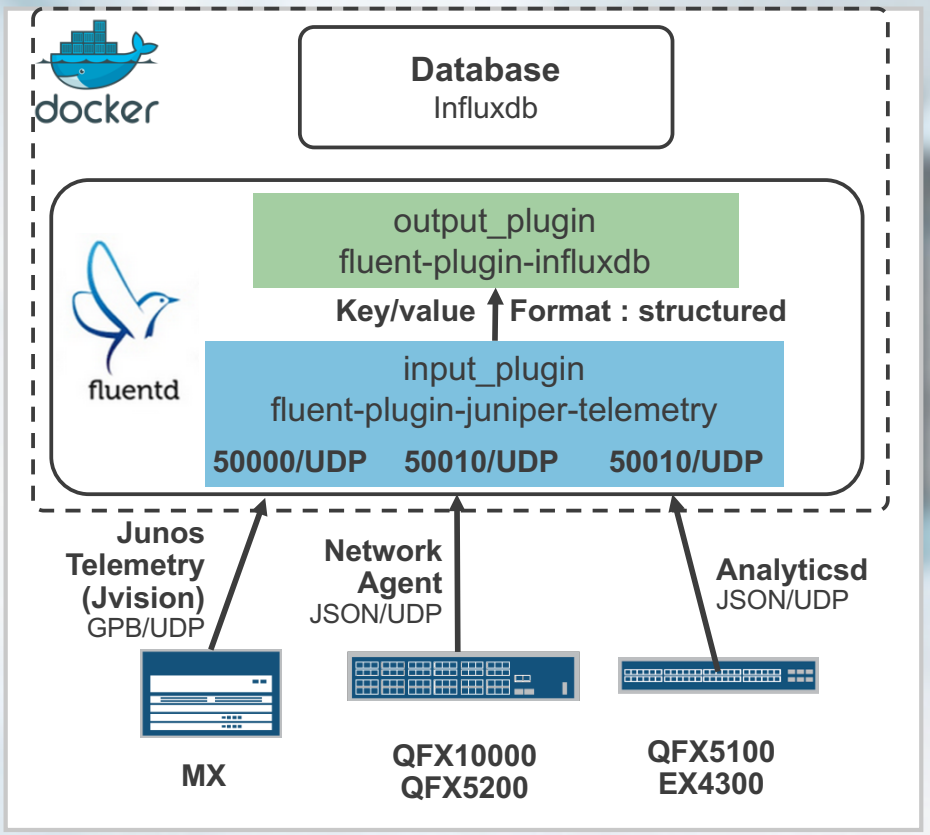
- Open Source time series database
- Written in Go
- Store time series data and events down to ms accuracy
- Schema less
- Data are stored with “tags”

- Design to scale-out within a cluster
- Very popular since 2014
- Natively expose a full REST API (read/write)

https://github.com/influxdata

Data Streaming Collector – Fluentd





Fluentd accept different type of streaming data from MX, PTX, QFX & EX :

- Junos Telemetry (GPB/UDP)
- Analyticsd (JSON/UDP)
- Network Agent (JSON/UDP)

Data are converted into Key/Value pairs and injected into Influxdb as “tags”

Fluentd plugins used

Fluent-plugin-juniper-telemetry

This plugin include 3 input parsers and 3 output-format

Input Parser :

- Juniper_jti - default on port 50000/UDP
- Juniper_na - default on port 50010/UDP
- Juniper_analyticsd - default on port 50020/UDP

Output_format :

- Structured - *Compatible with Influxdb plugin*
- Flat - *Compatible with Graphite plugin*
- Statsd

https://github.com/JNPRAutomate/fluent-plugin-juniper-telemetry

Fluent-plugin-influxdb

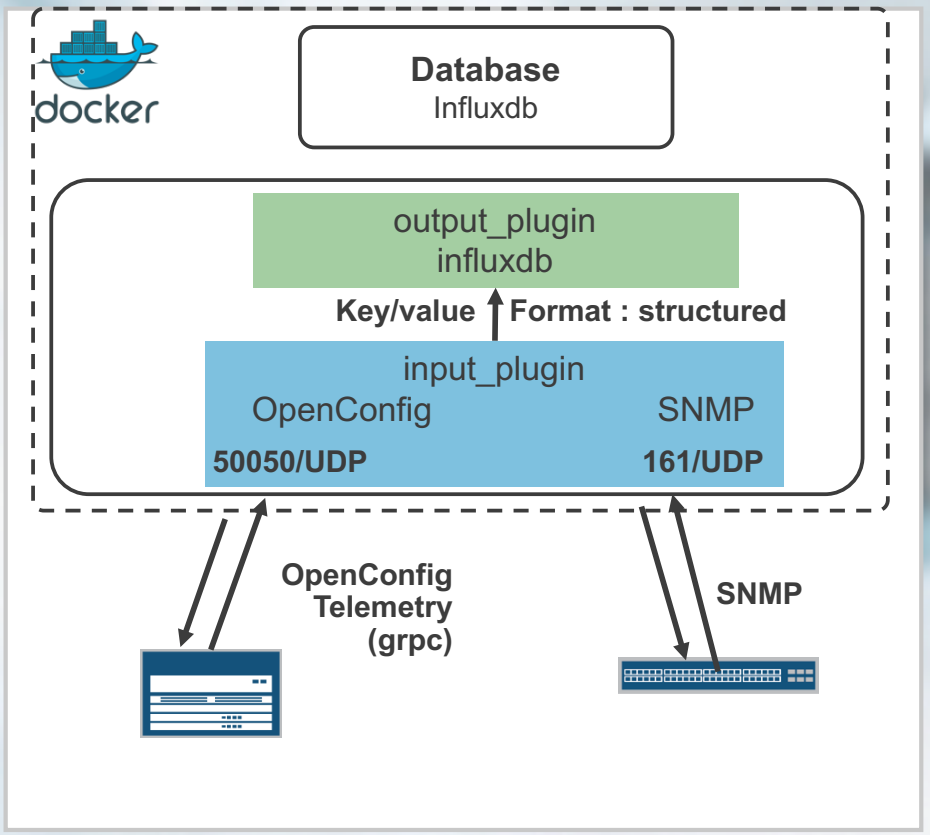
Accept data in Structured format

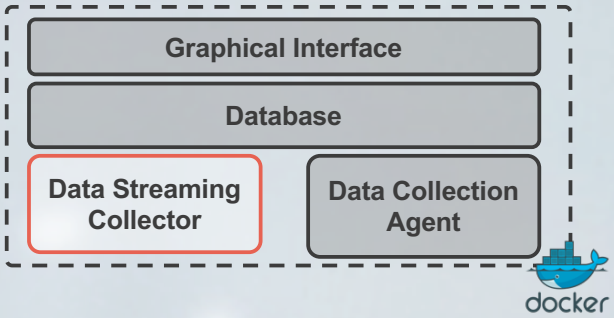
Push data to influxdb over HTTP using tags

Integrate a buffer, by default send data out every 5 seconds

https://github.com/fangli/fluent-plugin-influxdb

Telegraf based collectors – SNMP/OpenConfig



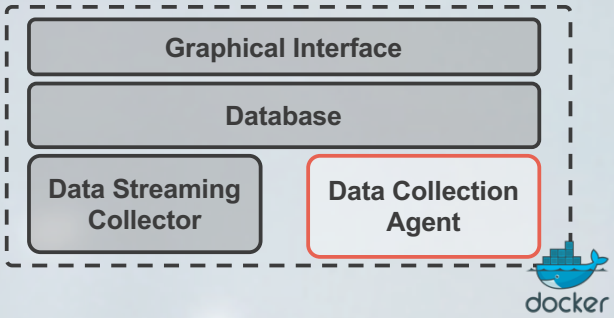


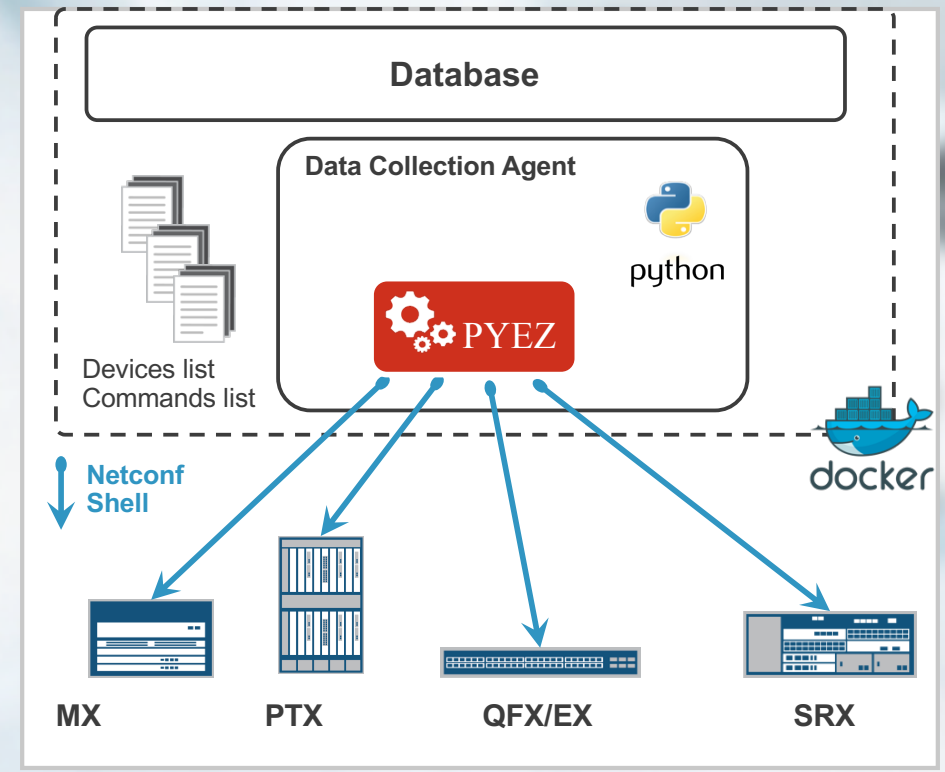
Telegraf different type of streaming data from MX, PTX, QFX & EX :

- OpenConfig Telemetry
- SNMP

Data are converted into Key/Value pairs and injected into Influxdb as “tags”

Data Collection Agent – PyEZ





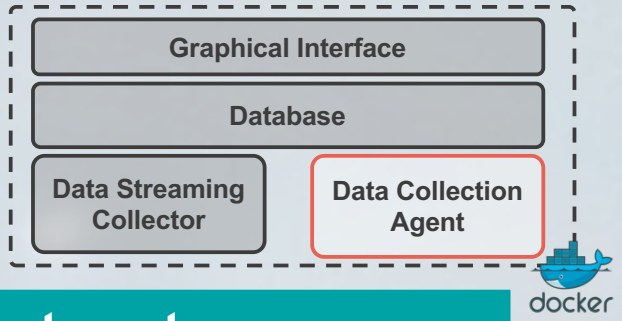
Tested with 9k KPI / min from 450 routers

Collect information over CLI or Shell
Compatible with all Junos devices and third party
Information collected either in XML or in Text

Devices are classified using “**tags**”
Leverage a collection of “**parser**”

Written in Python leveraging PyEZ library

Data Collection Agent – PyEZ



Classify devices by tags

	By role	By type	By location	By feature
mx-edge011:	edge	mx	madrid	bgp mpls
mx-edge012:	edge	mx	madrid	bgp mpls
mx-edge021:	edge	mx	munich	bgp mpls
mx-edge022:	edge	mx	munich	bgp mpls
mx-agg011:	agg	mx	madrid	bgp isis
mx-agg012:	agg	mx	madrid	bgp isis
qfx-agg021:	agg	qfx	munich	bgp
qfx-agg022:	agg	qfx	munich	bgp
qfx5100-01:	tor	qfx	madrid	isis
qfx5100-02:	tor	qfx	madrid	isis

Use tags to

Define credentials

```
simple_cred:
        username: simple
        password: mytor
        community: mytor
        tags: tor
```

```
secure_cred:
        username: root
        password: myedge123!?=
        community: edge123!?=
        tags: edge agg
```

Define commands to execute

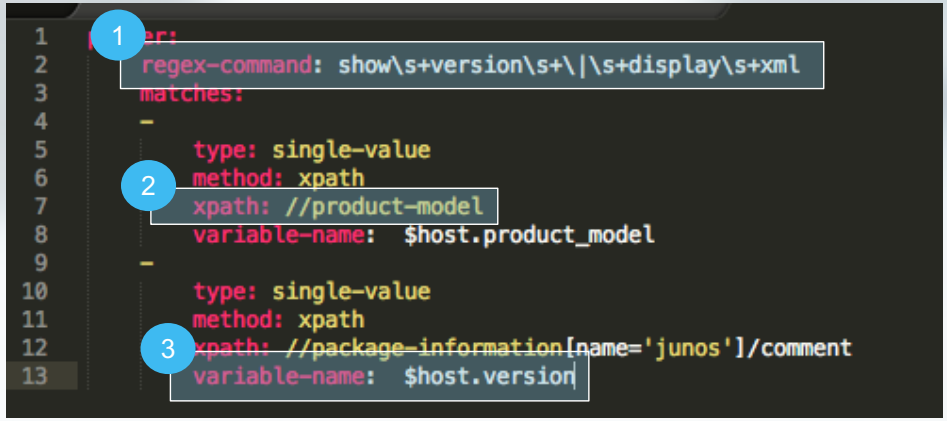
```
base_command:
  commands: |
    show system processes extensive
    show chassis routing-engine | display xml
  tags: agg edge tor

bgp_commands:
  commands: |
    show route summary | display xml
    show bgp summary | display xml
  tags: bgp
```

Data Collection Agent - Parser

- In order to extract KPIs from a junos command output, a related ‘parser’ must exist before.
- There are parsers for xml and for non-xml output (using regex)
- Below there is an example parser for ‘show version | display xml’ command.
  - This parser extract 2 values (the product-model and the version)

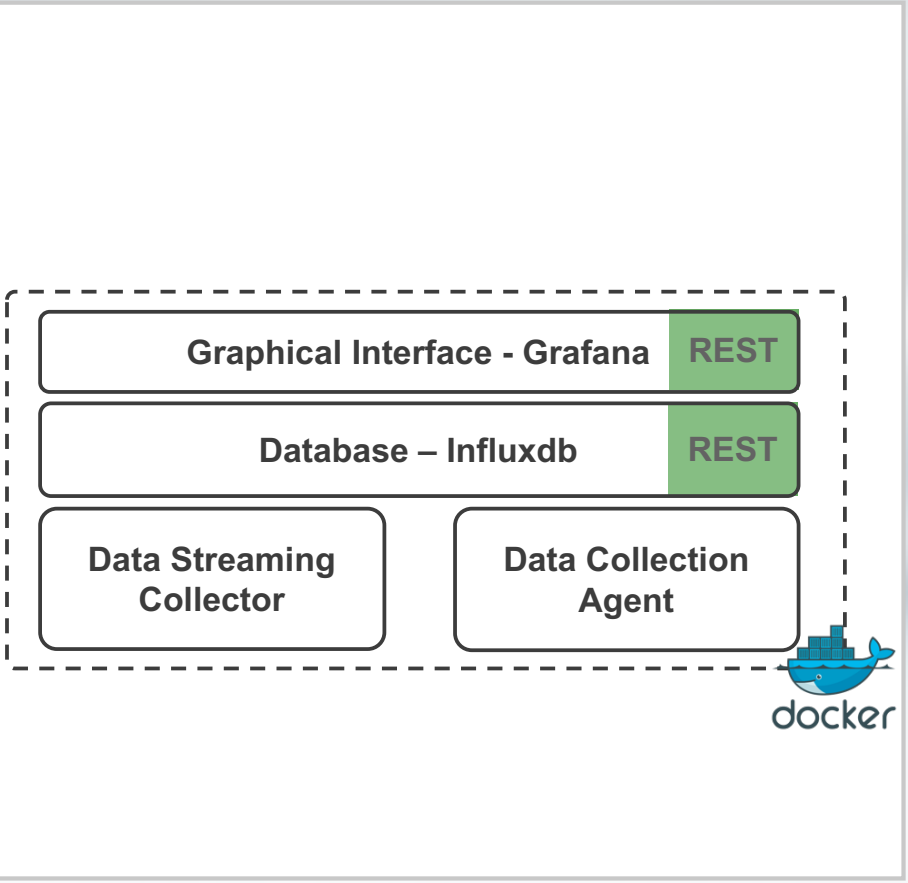
show-version.parser.yaml



1. The regex-command relates the parser with the executed command
2. The xpath to be execute to extract the KPI
3. The KPI will be stored in the DB under the variable with this name.

Interact with OpenNTI in a programmatic way

Programmatic interfaces



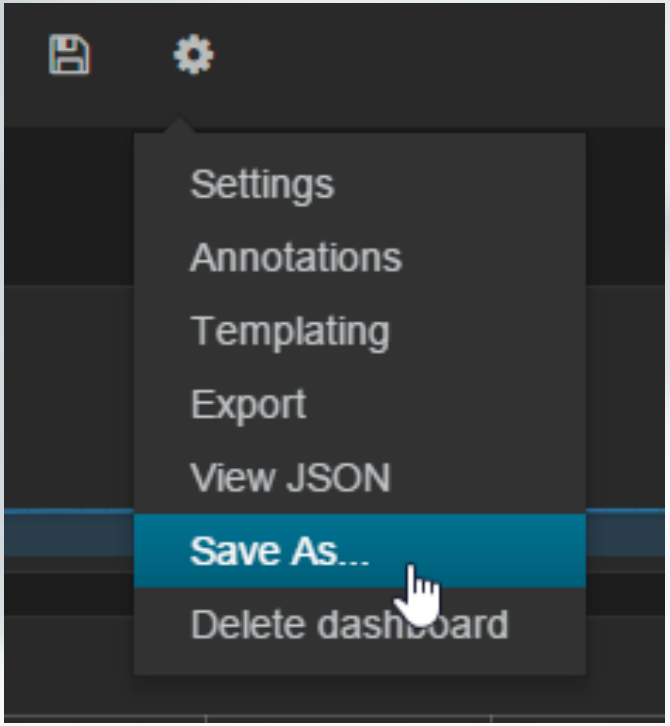
Grafana - REST Interface

The GUI has a REST interface available that can be used to publish dashboard automatically.
A dashboard is just a JSON file
It's possible to generate Dashboard from Ansible and publish them automatically

Influxdb - REST Interface

The database has a REST interface what allow anyone to push or access data.

How to automate creation of dashboards?



- Dashboards can be imported/exported
- Format is JSON
- Dashboard creation is part of the same Ansible project which is used to automate Junos analytics configuration on Junos devices
- RestAPI is used to push generated dashboards to Grafana

Contribute and Participate

Projects open to contribution

Open NTI
https://github.com/Juniper/open-nti

[Public] Fluentd plugin
https://github.com/JNPRAutomate/fluent-plugin-juniper-telemetry

[Public] Capture file replay
https://github.com/dgarros/docker-tcpreplay

How can you contribute

- Improve documentation
- Report issue
- Improve dashboard
- Add command parser
- Suggest enhancement
- Write code/plugin

Contribution is not just about code


JUNIPER
NETWORKS
Thank You