Deploying a “push” model Prometheus

Hristo Mohamed on behalf of the LHCb collaboration
CHEP 9-13 July 2018, Sofia
Our infrastructure

- 3000+ metrics collectible devices – including switches, servers, storage appliances
- Infrastructure expected to grow in the upcoming upgrade
- Different network segments, some lacking two way connectivity (segment A can reach B, but not vice versa)
The need to adopt modern metrics system

- Need to support easy tagging of metrics – infrastructure will grow, metrics analysis will only get more complex
- Need to have active community
- Need to be ‘snappy’ - human operators do not tolerate waiting for more than a few seconds
- Ability to consume the same metrics multiple times
- Low deployment/operations cost
Why Prometheus?

- Time Series Database (TSDB)
- Open Source & Community Driven
- Very powerful query language
- Rich metadata model
What is Prometheus?
Exporting metrics and scraping jobs

• Service Discovery? Yes (dns,kubernetes,gce,azure,consul)
• Are the targets always reachable? Pushgateway – lose instance health monitoring, single point of failure, time series needs to be deleted through API
• Reusing metrics → Write your own Prometheus remote write adapters
• Operation problems with Prometheus => TSDB data loss
• Things are getting complicated :(

Hristo Mohamed, Deploying "push" model Prometheus, CHEP 9-13.07.2018, Sofia, Bulgaria
Going back to 2014
Going back to 2014

Diagram:

- Log shipper
- Log shipper
- Log shipper
- Kafka Cluster
- Logstash
- Logstash
- ElasticSearch Cluster
Wait a moment…

Metrics and logs kinda look the same in the Prometheus world?

prometheus_tsdb_tombstone_cleanup_seconds_bucket{le="0.025"} 0

Jun 25 17:45:25 hlta0105 root: This is a message for a presentation

At least they did to me.
Going back to 2014

Hristo Mohamed, Deploying "push" model Prometheus, CHEP 9-13.07.2018, Sofia, Bulgaria
Metrics shipper(possibly) that can talk with Kafka

- Statsd – built for Graphite, with hacking can be made to talk with Kafka, last updated 2016
- Collectd – Alive and well since 2015, can talk with Kafka, partial tag support
- Telegraf – Designed with Tags in mind, can talk bothways with Kafka (produce/consume), project under InfluxDB, grows very fast in popularity => more than 80 supported plugins, single binary

We picked Telegraf
Decoupling

- Producing metrics
- Collecting logs in a distributed pipeline
- But what about the Prometheus collection part? Don’t want to write a producer/exporter your self? Telegraf to the rescue
Some numbers

- Telegraf on a client => 14.6MB
- Telegraf exporting metrics =>
  ~ 50k metrics → 177MB
  ~ 1.5 seconds scrape time
In conclusion

- Adding a message queue can greatly make any Prometheus deployment cheaper
- It is not expensive, resource and money wise
- It allows greater flexibility with metrics usage
Questions?