Prometheus Long Term Metrics Storage With Thanos

Roberto Valverde
CERN IT - Storage and Data Management
EOS WORKSHOP - 15/03/2024

Prometheus

- In Storage and Data Management group we rely on prometheus for many services:
    - Ceph, EOS, cback...
- Pull-based, time-series data model
- PromQL query language: `count by(cluster, eos_version)(eos_node_info)`
- Central server that `scrapes` targets, called `exporters`
    - There is an exporter for that! exporters (https://prometheus.io/docs/instrumenting/exporters/)
    - And there is also an EOS exporter!

EOS EXPORTER

- https://github.com/cern-eos/eos_exporter (https://github.com/cern-eos/eos_exporter)
- Developed/maintaned/Used by EOS CERN Operators
    - 8 releases last year.
- Agnostic (we try)
- More info EOS Workshop 2023 (https://indico.cern.ch/event/1227241/contributions/5326727/)

Problem/Motivation:

- Prometheus is very good for live metrics, short-term retention
- In storage services, long-term metrics is desirable:
    - Capacity planning
    - Historical evolution of services
- We need a long-term metric storage solution

Thanos (https://thanos.io/)

Open source, highly available Prometheus setup with long term storage capabilities.

- Used extensively around prometheus community (many cross-service contributors)
- Integrates seamsly into an existing prometheus server environment.
- Thanos will send this metrics indefinetely in S3. Thanos will also compact, create

downsampled versions (5m, 1h) and serve this metrics using a Prometheus-compatible API.

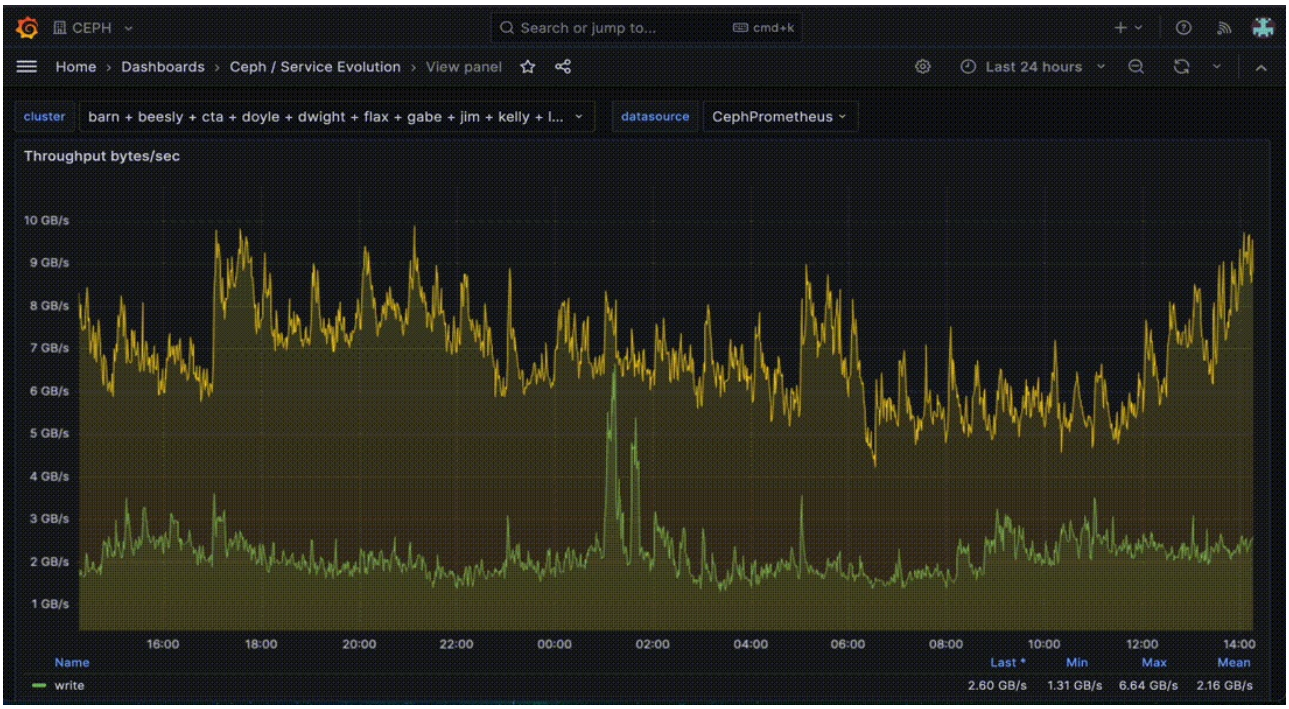
Thanos (https://thanos.io/)

- 1 binary (Golang), stateless. Four main roles:
    - **Thanos-sidecar**: Prometheus server companion, send metrics to S3 everytime prometheus flushes the metrics to disk (by default, every 2h). Exposes API for accessing metrics still on memory.
    - **Thanos-store**: S3 Storage gateway.
    - **Thanos-query**: Endpoint used as datasource in Grafana, process queries reading metrics from `thanos-store` and `thanos-sidecar` endpoints.
    - **Thanos-compactor**: Compacts TSDB blocks and creates downsampled versions of the metrics stored in the S3 bucket.

Cold or Hot?

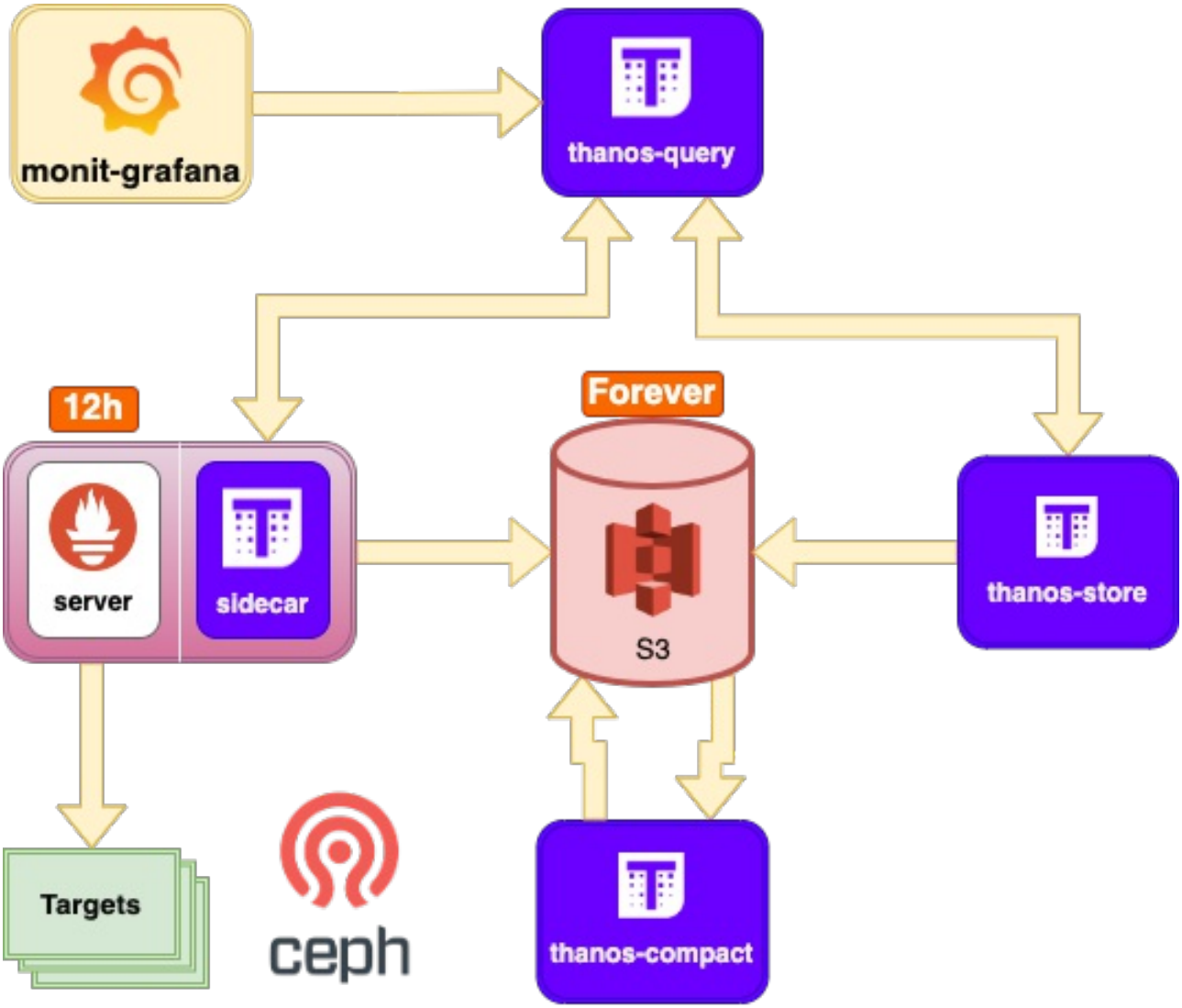
- Thanos creates `additional` downsampled versions of the RAW metrics.
- RAW metrics are always kept by default, allowing to `zoom-in` into the original resolution at any time.
- Grafana will choose use the adequate downsampling acording to the chosen interval.

Demo



Demo 1 (https://monit-grafana.cern.ch/d/wiKW7TiSz/ceph-service-evolution-in-numbers?orgId=49)

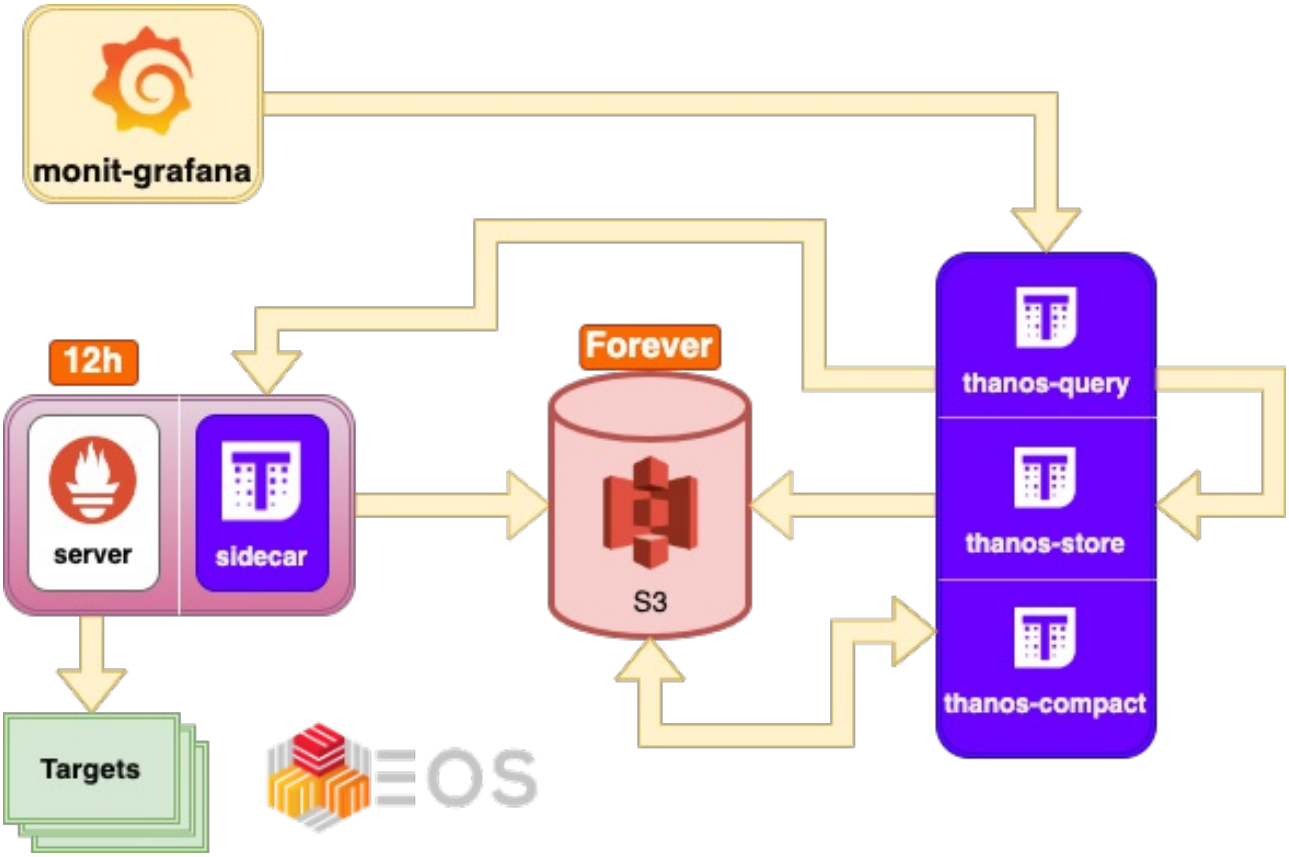
Architecture of Ceph Monitoring



Expand Thanos usage for EOS

- In production since 1st November.
- More compact and streamlined setup (2 VMS)
    - 1 x VM for thanos (thanos-{store, compact, query}) - `eosthanos.cern.ch`
    - 1 x VM for prometheus server + `thanos-sidecar` - `eos-prom.cern.ch`
- Using new [thanos (https://gitlab.cern.ch/ai/it-puppet-module-thanos)](https://gitlab.cern.ch/ai/it-puppet-module-thanos) module added to the CERN puppet infrastructure.

Architecture of EOS Monitoring



Demo

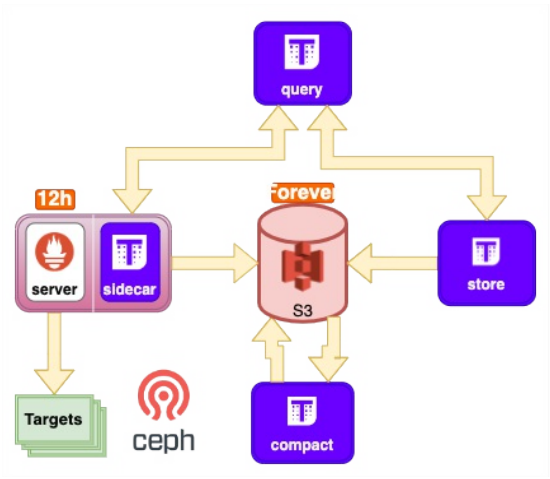
Prometheus + Thanos vs Graphite

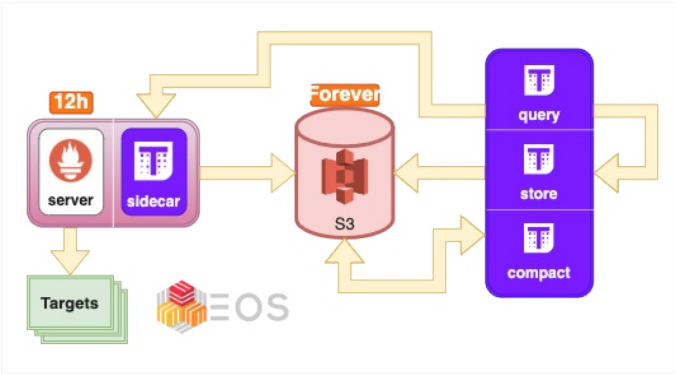
<img src="https://codimd.web.cern.ch/uploads/upload_6c55ac96a33c090b3ffdbc45bf4a97ef.png" alt="Girl in a jacket" style="width:450px;height:400px;">

<img src="https://codimd.web.cern.ch/uploads/upload_6238e6befd8f27c8d7e8a90c874c04e6.png" alt="Girl in a jacket" style="width:450px;height:400px;">

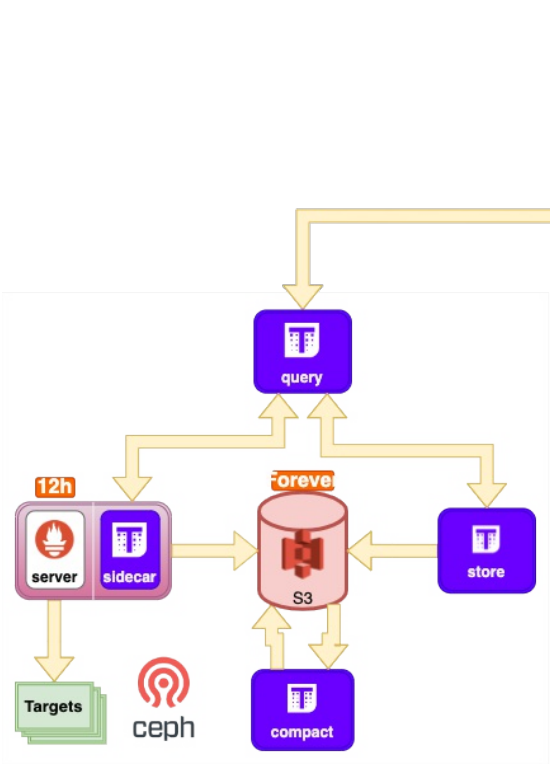
- Prometheus+Thanos (https://monit-grafana.cern.ch/d/7kC6Gwl7z/eos-general-overview?from=1702425600000&orgId=22&to=1702511999000&viewPanel=164) VS Filer-Carbon (https://filer-carbon.cern.ch/grafana/d/000000036/eos-control-tower?orgId=1&viewPanel=20&from=1702425600000&to=1702598399000)

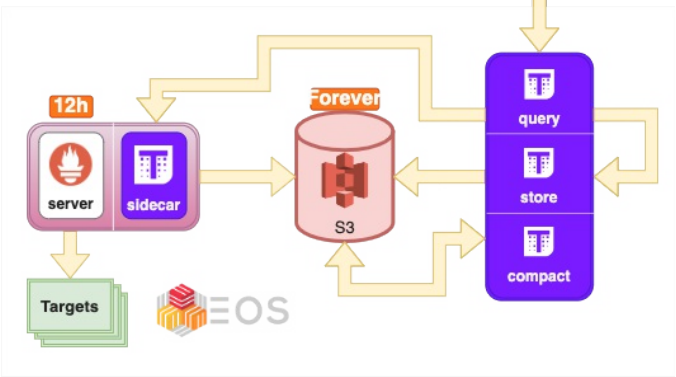
Bonus: Cross-service queries.



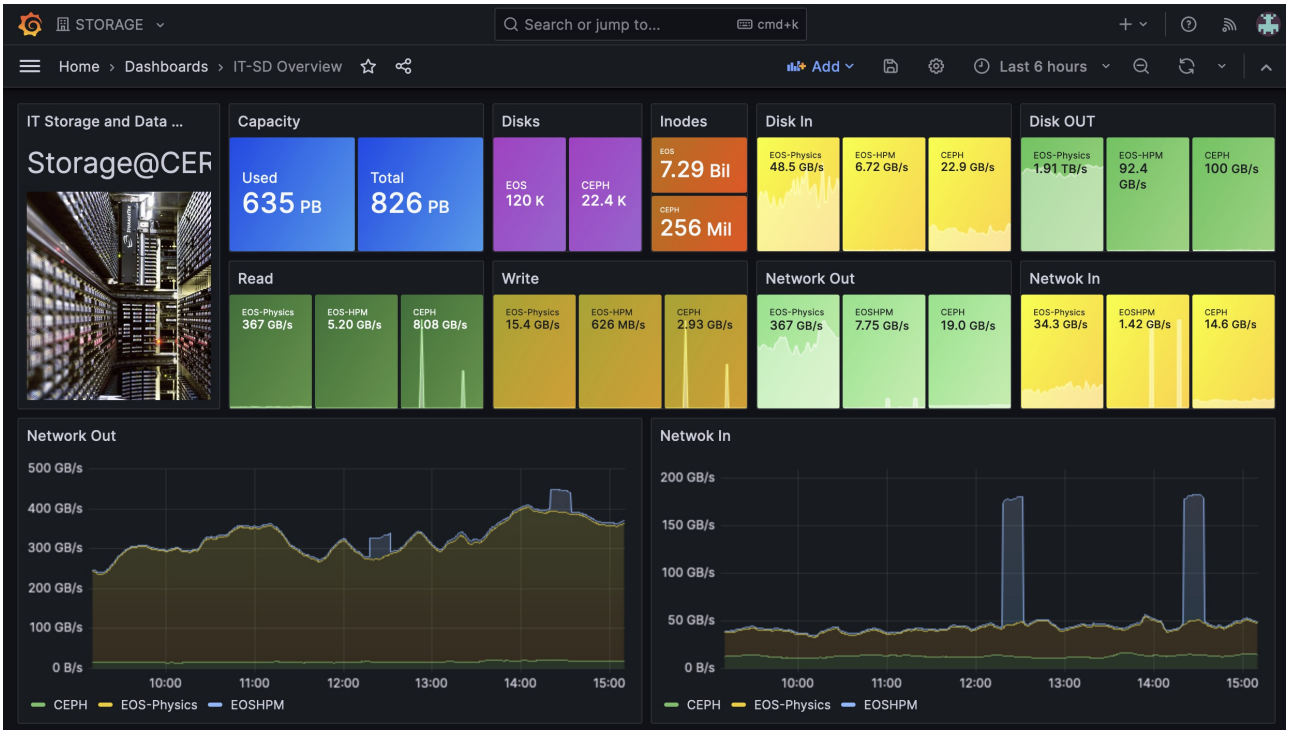


Bonus: Cross-service queries.





Bonus: Cross-service queries.



Conclusion

- Thanos is a pretty nice solution to have long-term metric storage for prometheus.
    - Transparent integration with Grafana
    - No need to re-write queries
    - Scalable
- Some internal notes if you want to try it out [Prometheus+Thanos (Puppet) (https://gitlab.cern.ch/dss/guilds/observability/-/wikis/%5BGuide%5D-Prometheus-+-Thanos-Configuration)](https://gitlab.cern.ch/dss/guilds/observability/-/wikis/%5BGuide%5D-Prometheus-+-Thanos-Configuration)

Thanks
