

Deploying Prometheus

Filippo Giunchedi - Operations Engineer
filippo@wikimedia.org



WIKIMEDIA
FOUNDATION

Agenda

- Introduction
- What we have and what we need
- Why Prometheus?
- How does it look like in production?
- What Prometheus does (and will do) for us

Wikipedia & co

Wikipedia and sister projects did

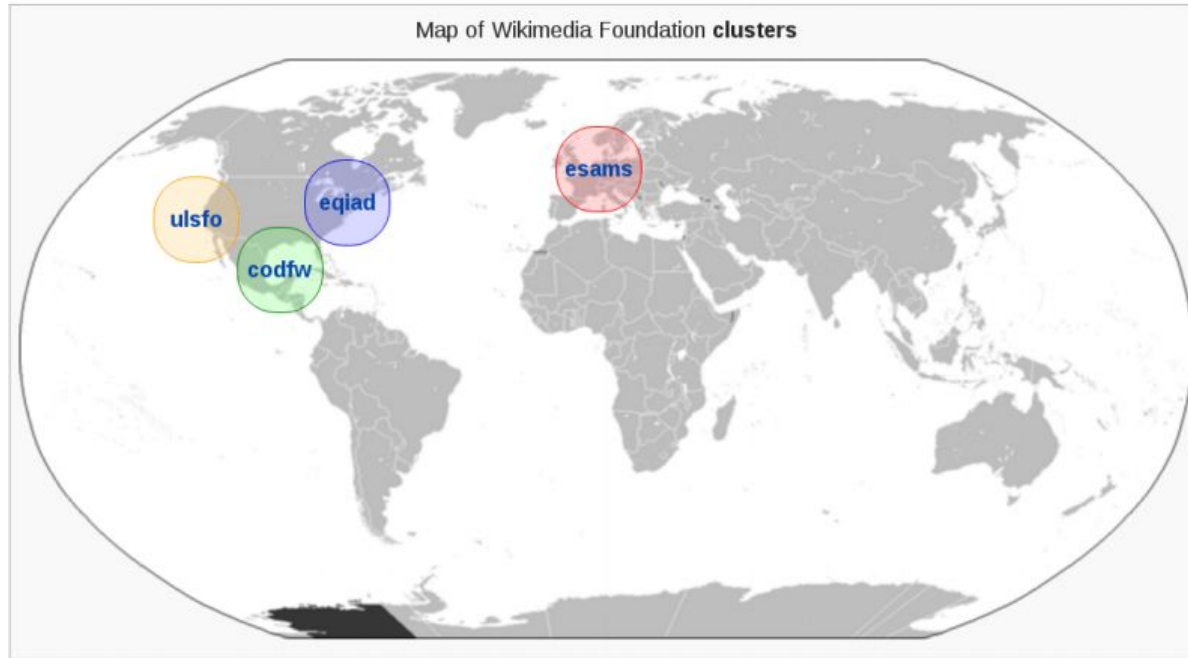
- 16 billion pageviews / month
- 13 thousand new editors / month
- 41 million articles
- 34 million multimedia files

More data on <https://reportcard.wmflabs.org>

Infrastructure

- 4 sites: 2 datacenters, 2 caching PoPs
- 1400 bare metal machines
- 125k req/s (HTTPS)
- 32Gb/s outbound to clients

Infrastructure



WIKIMEDIA
FOUNDATION

Monitoring landscape at WMF

Over time we have been *adding* monitoring systems but *removing* none

- Ganglia - aggregated & individual machine stats
- Graphite/diamond/statsd - machine & service stats
- Grafana - dashboards
- Tendril - MySQL
- LibreNMS - network & power stats
- Torrus - power stats
- Smokeping - network latency & availability
- Icinga/Shinken - alerting

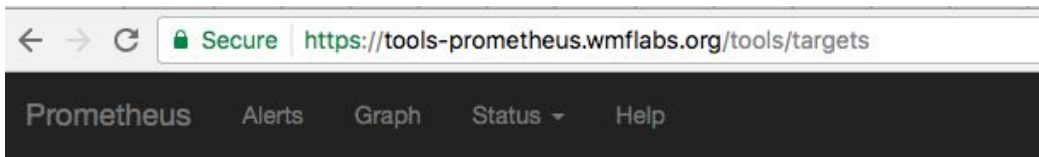
Enter Prometheus

- Powerful data model and query language
- Prometheus as a *toolkit*
- Multi tenancy
- Reliable
- Efficient resource usage
- Metric flow easy to understand and debug

Before production

- Virtualized environment: *WMF Labs*
- Runs community's software: *tools, bots, etc*
- Also a playground for production users
- Used to validate Prometheus: use cases, performance, etc
- Publicly available
 - <https://beta-prometheus.wmflabs.org/beta/targets>
 - <https://tools-prometheus.wmflabs.org/tools/targets>
 - <https://grafana-labs.wikimedia.org>

Before production



Targets

etcd

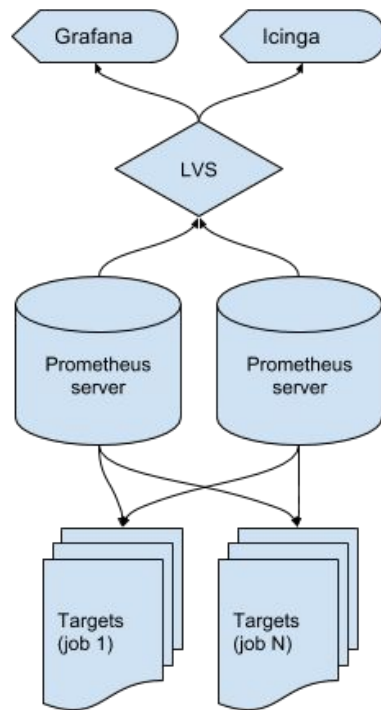
Endpoint	State	Labels
http://tools-flannel-etcd-01:9051/metrics	UP	none
http://tools-flannel-etcd-02:9051/metrics	UP	none
http://tools-flannel-etcd-03:9051/metrics	UP	none
http://tools-k8s-etcd-01:9051/metrics	UP	none
http://tools-k8s-etcd-02:9051/metrics	UP	none
http://tools-k8s-etcd-03:9051/metrics	UP	none

k8s-api

Site deployment

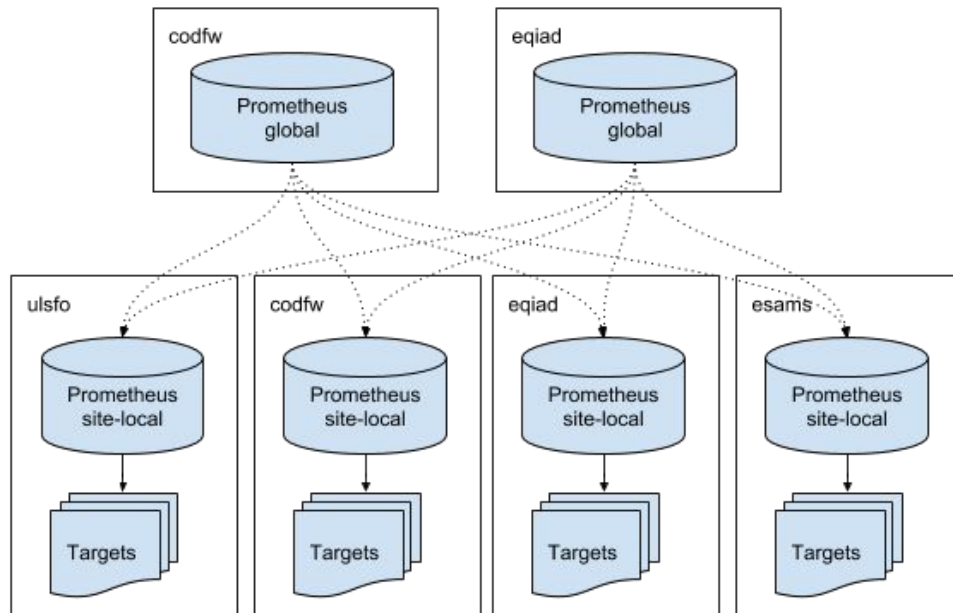
- 1+ bare metal Prometheus machines
- 1+ Prometheus instances per machine
- HA via identical machines per site + LVS-DR
- Local Nginx: access control, reverse proxy
- Configuration: Puppet + autogenerated yaml files

Gory details at <https://github.com/wikimedia/operations-puppet>
and <https://wikitech.wikimedia.org/wiki/Prometheus>

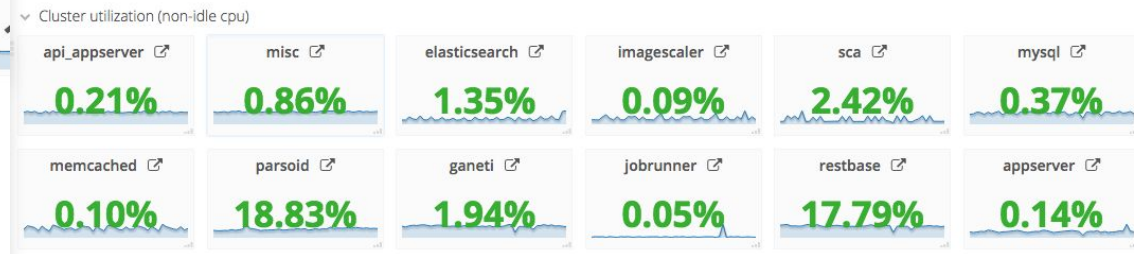
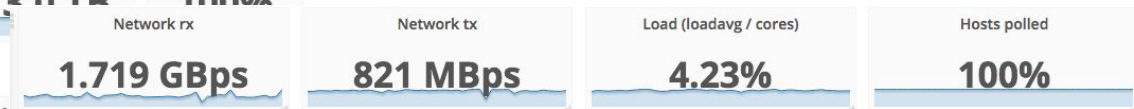
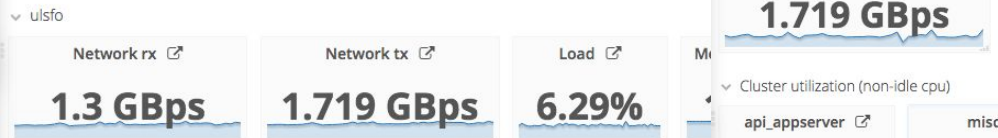
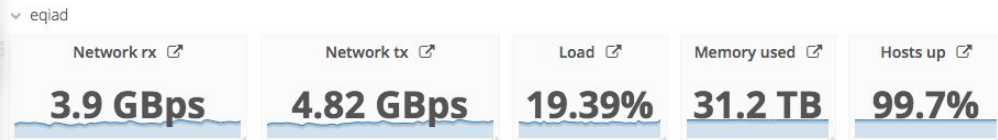
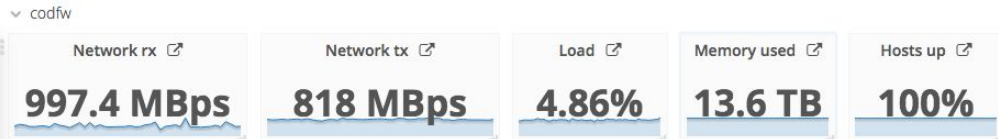


Site-local and global

- Federation via global instance
- Global overview via dashboards
- Drilldown on local instances



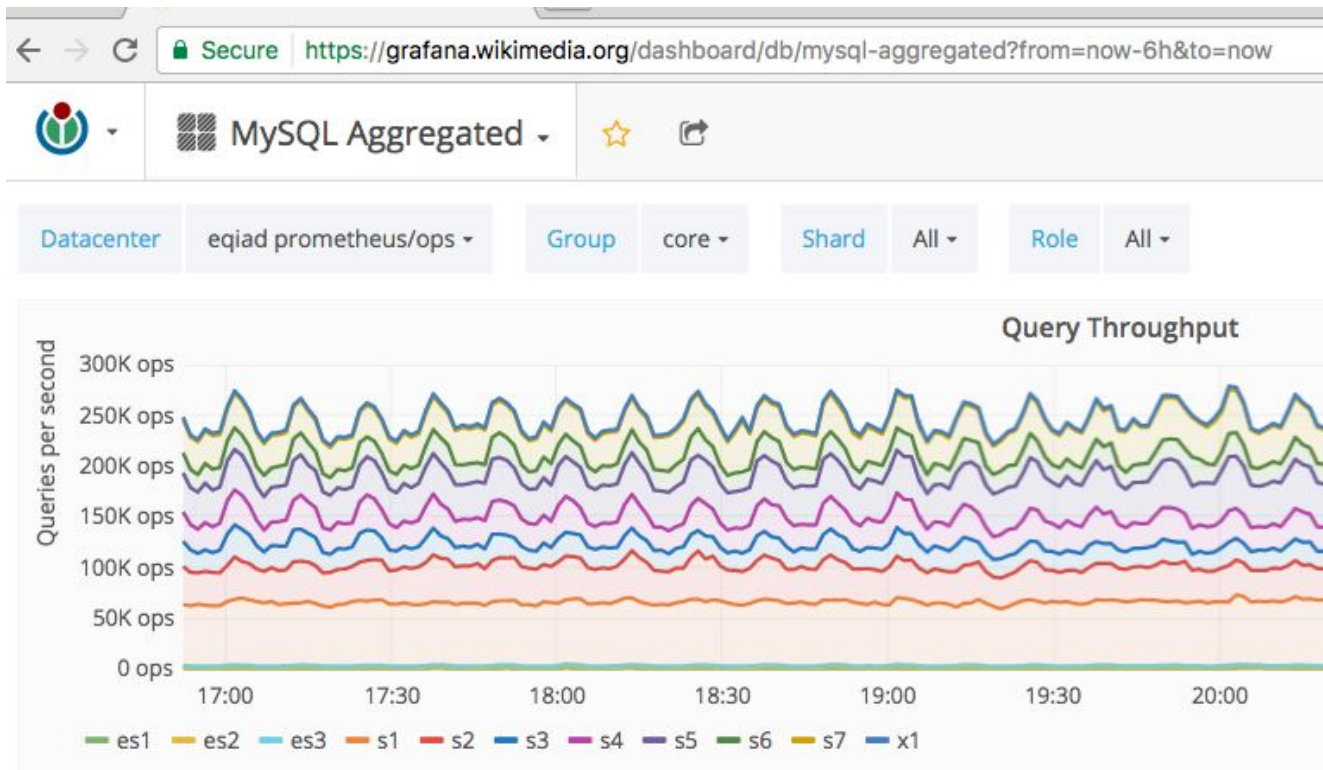
Site-local and global



Database monitoring

- First Prometheus use case in production
- ~ 180 DB machines across two datacenters
- 7 main clusters, 21 clusters total
- MariaDB 10.0
- Private data: internal monitoring tool, *Tendril*
- Public data: `mysqld-exporter` + Prometheus + Grafana

Aggregated metrics



Replacing Ganglia

- Ganglia used to inspect *service clusters* health
- Health: machine-level and service-level
- Used for aggregated / overview data
- Audit and replace standard and custom Ganglia plugins

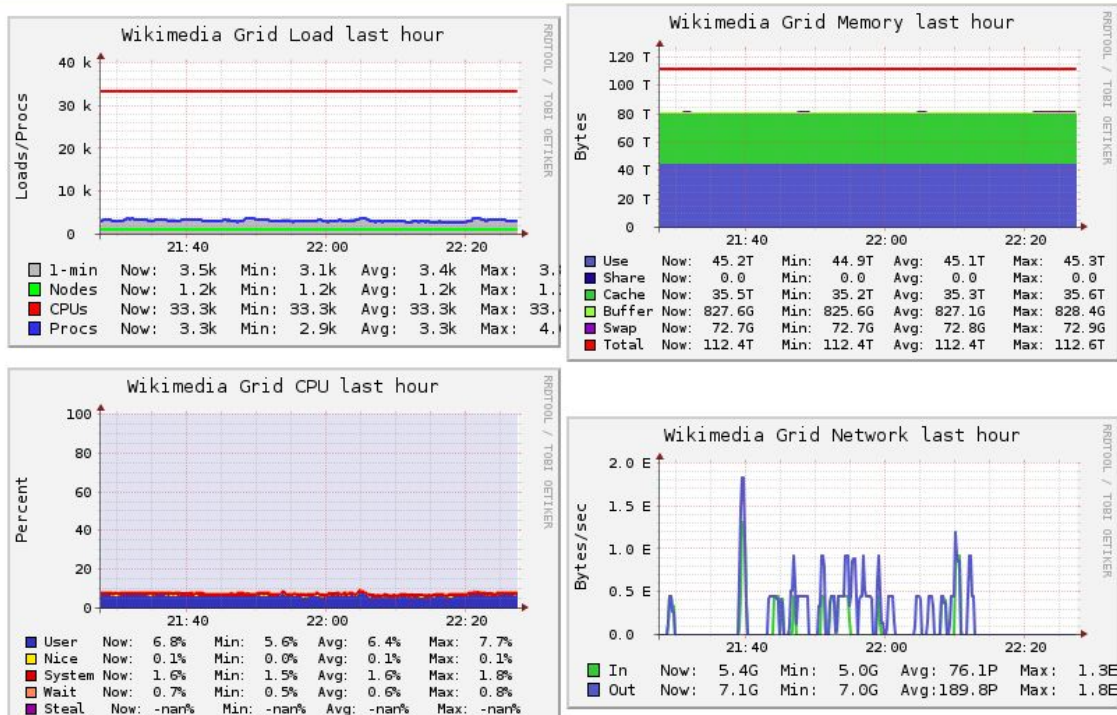
Gory details at <https://phabricator.wikimedia.org/T145659>

Exabytes?

Wikimedia Grid (79 sources) [\(tree view\)](#)

CPU's Total: **33342**
Hosts up: **1216**
Hosts down: **3**

Current Load Avg (15, 5, 1m):
10%, 10%, 10%
Avg Utilization (last hour):
10%



Porting metrics

- Custom Ganglia plugin replaced with an exporter
- Happy case: exporter already in Debian
- Unhappy case: write and package the exporter (e.g. HHVM)
- Some cases covered by `node-exporter + textfile`
- Exporter minimal configuration via Puppet
- Add Prometheus job
- Build Grafana dashboards

Future

- Onboard more teams
- Native instrumentation for services
- Kubernetes production monitoring
- More exporters
- Alerting
- Retire Graphite ?

Takeaways

- Prometheus is helping Wikimedia Foundation's monitoring
- Deploying to production was fun
- ... and the gains well worth it
- Multi dimensional metrics are awesome