Deploying Prometheus

Filippo Giunchedi - Operations Engineer
filippo@wikimedia.org
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 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
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
- Smokepeing - 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

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>State</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://tools-flannel-etcld-01:9051/metrics">http://tools-flannel-etcld-01:9051/metrics</a></td>
<td>UP</td>
<td>none</td>
</tr>
<tr>
<td><a href="http://tools-flannel-etcld-02:9051/metrics">http://tools-flannel-etcld-02:9051/metrics</a></td>
<td>UP</td>
<td>none</td>
</tr>
<tr>
<td><a href="http://tools-flannel-etcld-03:9051/metrics">http://tools-flannel-etcld-03:9051/metrics</a></td>
<td>UP</td>
<td>none</td>
</tr>
<tr>
<td><a href="http://tools-k8s-etcld-01:9051/metrics">http://tools-k8s-etcld-01:9051/metrics</a></td>
<td>UP</td>
<td>none</td>
</tr>
<tr>
<td><a href="http://tools-k8s-etcld-02:9051/metrics">http://tools-k8s-etcld-02:9051/metrics</a></td>
<td>UP</td>
<td>none</td>
</tr>
<tr>
<td><a href="http://tools-k8s-etcld-03:9051/metrics">http://tools-k8s-etcld-03:9051/metrics</a></td>
<td>UP</td>
<td>none</td>
</tr>
<tr>
<td>k8s-api</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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](https://github.com/wikimedia/operations-puppet) and [https://wikitech.wikimedia.org/wiki/Prometheus](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

<table>
<thead>
<tr>
<th>Site</th>
<th>Network rx</th>
<th>Network tx</th>
<th>Load</th>
<th>Memory used</th>
<th>Hosts up</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODFW</td>
<td>997.4 MBps</td>
<td>818 MBps</td>
<td>4.86%</td>
<td>13.6 TB</td>
<td>100%</td>
</tr>
<tr>
<td>AGJAD</td>
<td>3.9 GBps</td>
<td>4.82 GBps</td>
<td>19.39%</td>
<td>31.2 TB</td>
<td>99.7%</td>
</tr>
<tr>
<td>ESAMS</td>
<td>3.0 GBps</td>
<td>4.29 GBps</td>
<td>6.71%</td>
<td>3.0 TB</td>
<td>100%</td>
</tr>
<tr>
<td>UFSO</td>
<td>1.3 GBps</td>
<td>1.719 GBps</td>
<td>6.29%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster utilization (non idle cpu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_appserver: 0.21%</td>
</tr>
<tr>
<td>misc: 0.86%</td>
</tr>
<tr>
<td>elasticsearch: 1.35%</td>
</tr>
<tr>
<td>imagescaler: 0.09%</td>
</tr>
<tr>
<td>sca: 2.42%</td>
</tr>
<tr>
<td>mysql: 0.37%</td>
</tr>
<tr>
<td>memcached: 0.10%</td>
</tr>
<tr>
<td>parslod: 18.83%</td>
</tr>
<tr>
<td>ganeti: 1.94%</td>
</tr>
<tr>
<td>jobrunner: 0.05%</td>
</tr>
<tr>
<td>restbase: 17.79%</td>
</tr>
<tr>
<td>appserver: 0.14%</td>
</tr>
</tbody>
</table>
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: `mysqlld-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](https://phabricator.wikimedia.org/T145659)
Exabytes?
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