prometheus

Abel Camarillo <acamari@verlet.org>

19 de enero de 2023

Agenda

- ¿Quién soy?
- ¿Qué es prometheus?
- Arquitectura
- Integración

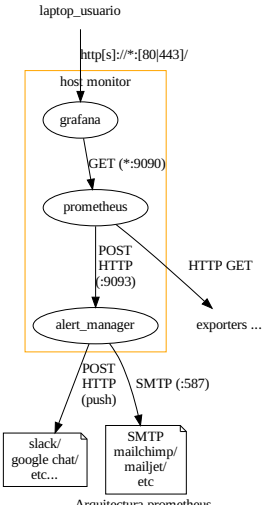
¿Quién soy?

- Desarrollador de software desde el 2008 - OpenBSD, perl, C, sh, js
- Lead developer en Neuroservices Communications durante 6 años.
- Tech Lead en Vordem SA de CV (2018).
- Desarrollador freelance desde el 2015 - Verlet.
- Maintainer de 21 paquetes en el árbol oficial de OpenBSD - http://openports.se/bbmaint.php?maint=acamari@verlet.org
- Interés en UNIX, carpintero, ~arte~ (teatro, poesía, fotografía), cocina, etc...

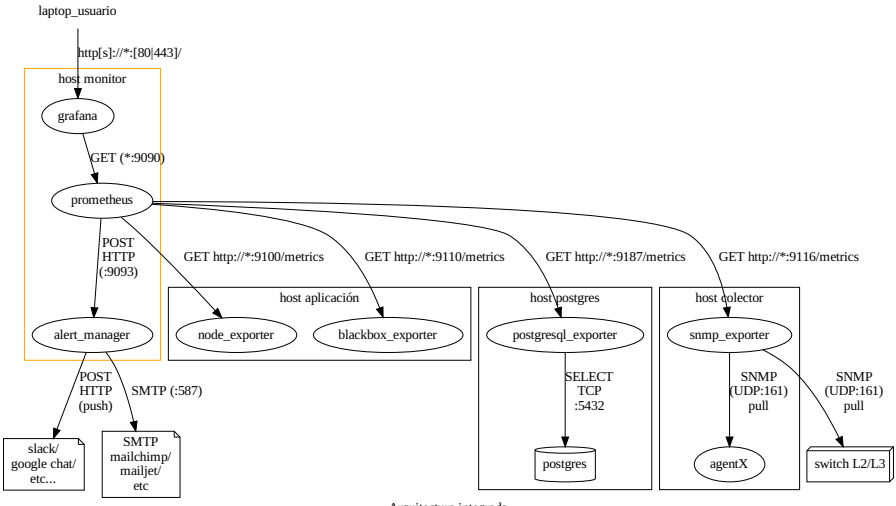
Qué es prometheus

- Producto open source de comunidad
- Timeseries database for metrics
- indexa llaves, no valores
- Prometheus is an open-source systems monitoring and alerting toolkit with an active ecosystem. It is the only system directly supported by Kubernetes and the de facto standard across the cloud native ecosystem.
  (verbatim de https://prometheus.io/docs/introduction/faq/)

Arquitectura



Arquitectura prometheus



Arquitectura integrada

Formato de métricas

V4. Definido en github de prometheus (\ agregados para legibilidad):

https://github.com/prometheus/docs/blob/master/content/docs/instrumenting/exposition_formats.md

Ejemplo:

```
$ curl http://*:9100/metrics;
...
# HELP node_forks_total Total number of forks.
# TYPE node_forks_total counter
node_forks_total 1.8757377e+07
# HELP node_load1 1m load average.
# TYPE node_load1 gauge
node_load1 1
# HELP node_load15 15m load average.
# TYPE node_load15 gauge
node_load15 1.37
# HELP node_load5 5m load average.
# TYPE node_load5 gauge
```

```
node_load5 1.15
# HELP node_uname_info Labeled system information as provided by   \
        the uname
system call.
# TYPE node_uname_info gauge
node_uname_info{domainname="(none)",machine="x86_64",nodename="db4"\
    ,release="4.15.12-x86_64-linode105",sysname="Linux",\
    version="#1 SMP Thu Mar 22 02:13:40 UTC 2018"} 1
# HELP process_cpu_seconds_total Total user and system CPU time \
    spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 2111.27
# HELP node_cpu_seconds_total Seconds the cpus spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 2.980243359e+07
node_cpu_seconds_total{cpu="0",mode="iowait"} 8497.06
node_cpu_seconds_total{cpu="0",mode="irq"} 0
node_cpu_seconds_total{cpu="0",mode="nice"} 20709.44
```

Formato de métricas

- Una métrica puede tener varios valores si hay etiquetas
- Las etiquetas pueden tener longitud arbitraria
- Etiquetas pueden variar entre ejecuciones: fs nuevos, tarjetas de red o IPs entran y salen, etc

Administración

- Configuración vía YAML
- Se puede configurar rotación de datos por omisión:
  `/usr/bin/prometheus --storage.tsdb.retention=180d`
- Respaldos vía copia directa del directorio completo de data de prometheus `/var/lib/prometheus`. Puede ser en vivo.
- Todo o nada, no hay respaldos segmentados por host o intervalo de tiempo.
- No hay escalabilidad multihost integrada

Integración

- Logueo tradicional: syslog, newsyslog, logrotate, etc
- Manejo de señales: SIGHUP, SIGINT, etc
- Manejo atómico de archivos:
  - No queremos escribir archivo metrics a medias, porque en el resto del stack no se podría distinguir la falta de métrica vs archivo a medias
  - Usar archivo temporal /var/www/pronodebsd/.metrics
  - `rename("./.metrics", "./metrics")`

¿Preguntas?