Monitoring with Prometheus

Shiao-An Yuan
2017/04/22

Requirements

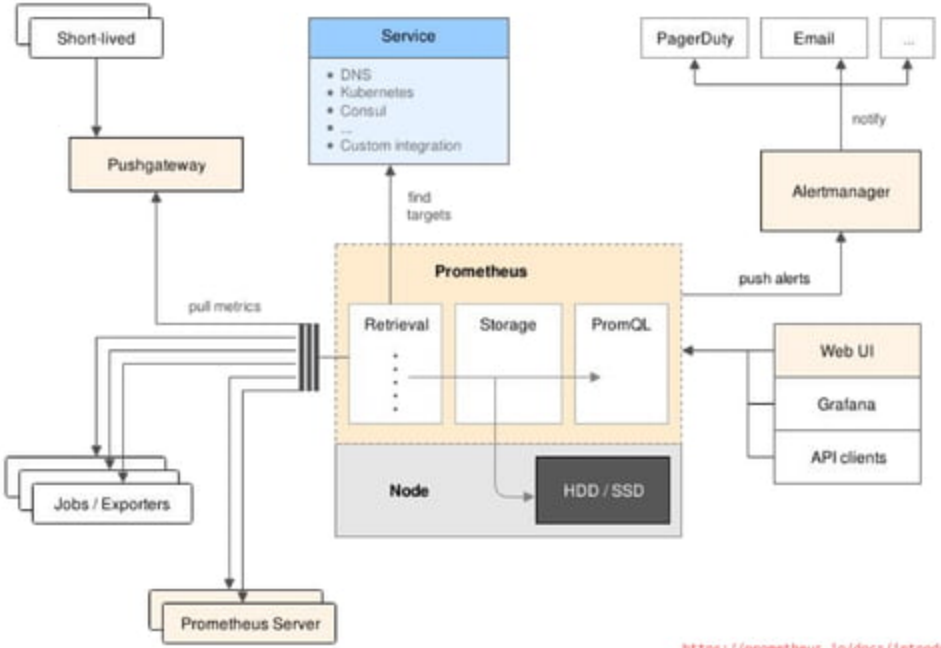
1. Able to “see” the abnormal on the dashboard
2. Notify me when abnormal happen

http://play.grafana.org/

Time Series Databases

A time series database (TSDB) is a software system that is optimized for handling time series data, arrays of numbers indexed by time -- from Wikipedia

- InfluxDB
- OpenTSDB
- Graphite
- Prometheus



https://prometheus.io/docs/introduction/overview/

Node Exporter

```
$ curl http://localhost:9100/metrics
```

```
...
# HELP node_filesystem_avail Filesystem space available to
non-root users in bytes.
# TYPE node_filesystem_avail gauge
node_filesystem_avail{mountpoint="/"} 6.301462528e+09
...
```

Metric Name *Labels* *Value*

Scrape Configs

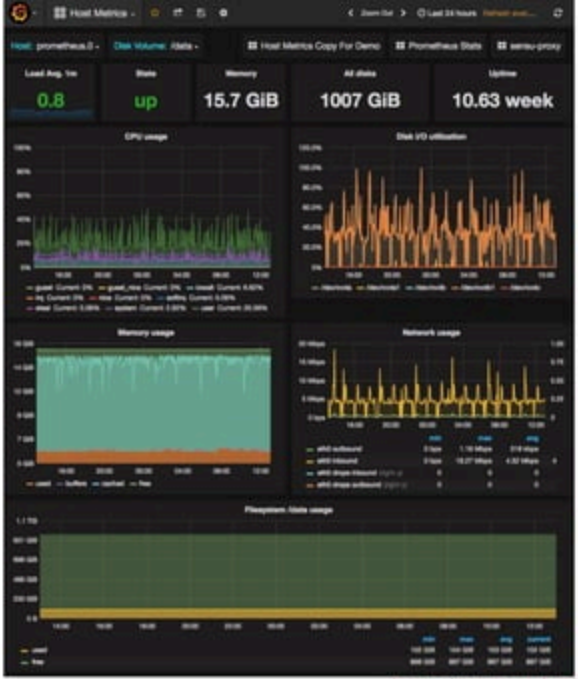
```
scrape_configs:
  - job_name: "node"
    scrape_interval: "1m"
    static_configs:
    - targets: ["localhost:9100"]
```

Before scrape:

```
node_filesystem_avail{mountpoint="/"}
```

After scrape:

```
node_filesystem_avail{instance="localhost:9100",job="node",mountpoint="/"}
```



https://prometheus.io/blog/2016/09/21/interview-with-compose/

Exporters

https://prometheus.io/docs/instrumenting/exporters/

- Lots of official & 3rd-party exporters
  - Node Exporter
  - JMX Exporter
  - ...
- Directly instrumented software
  - Kubernetes
  - cAdvisor
  - ScyllaDB (C++ implementation of Cassandra)
  - ...
- Client Libraries
  - Go, Java/Scala, Python, Ruby (official)
  - 12+ other languages (3rd-party)

Cassandra - by JMX Exporter

Writing Exporter

- Metric Types
  - Counter
  - Gauge
  - Histogram
  - Summary

Kafka Offset Exporter

https://github.com/sayuan/kafka-offset-exporter

PromQL

```
> node_filesystem_free
```

node_filesystem_free{instance="host1:9100",mountpoint="/"}	11111
node_filesystem_free{instance="host1:9100",mountpoint="/root"}	22222
node_filesystem_free{instance="host2:9100",mountpoint="/"}	33333
node_filesystem_free{instance="host2:9100",mountpoint="/root"}	44444

Instant Vector

Instant Vector Selector

```
> node_filesystem_free{instance="host1:9100"}
```

node_filesystem_free{instance="host1:9100",mountpoint="/"}	11111
node_filesystem_free{instance="host1:9100",mountpoint="/root"}	22222

Instant Vector

Range Vector Selector

```
> node_filesystem_free{instance="host1:9100"}[3m]
```

node_filesystem_free{instance="host1:9100",mountpoint="/"}	11111 11112 11113
node_filesystem_free{instance="host1:9100",mountpoint="/root"}	22222 22223 22224

Range Vector

Aggregation Operator

```
> sum(node_filesystem_free{instance="host1:9100"})
```

{}	33333

Scalar

Aggregation Operator

```
> sum(node_filesystem_free) by (instance)
```

{instance="host1:9100"}	33333
{instance="host2:9100"}	77777

Instant Vector

Data Types & Selectors

- Data Types
  - Instant Vector
  - Range Vector
  - Scalar
  - String (unused)
- Selectors
  - Instant Vector Selectors
  - Range Vector Selectors
  - Offset Modifier
    - `node_filesystem_free offset 5m`

Operations

- Arithmetic operators
  - +, -, *, /, %, ^
- Comparison operators
  - ==, !=, >, <, >=, <=
- Logical/set operators
  - and, or, unless
- Aggregation operators
  - sum, min, max, avg, stddev, stdvar
  - count, count_values, bottomk, topk, quantile

Functions

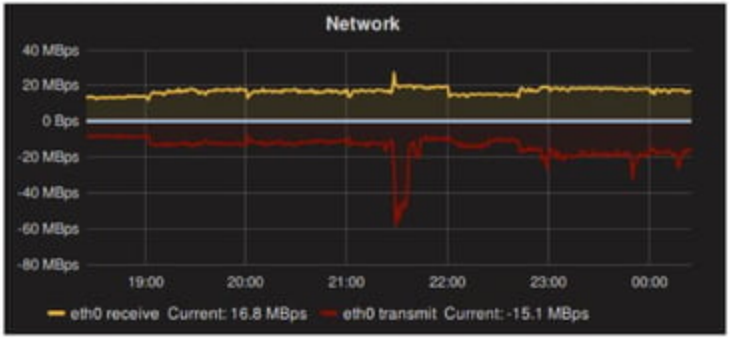
- `day_of_month()`, `day_of_week()`, `days_in_month()`, `hour()`, `minute()`, `month()`, `time()`, `year()`
- `abs()`, `ceil()`, `exp()`, `floor()`, `ln()`, `log10()`, `log2()`, `round()`, `sqrt()`
- `absent()`, `changes()`, `clamp_max()`, `clamp_min()`, `count_scalar()`, `delta()`, `deriv()`, `drop_common_labels()`, `histogram_quantile()`, `holt_winters()`, `idelta()`, `increase()`, `irate()`, `label_replace()`, `predict_linear()`, `rate()`, `resets()`, `scalar()`, `sort()`, `sort_desc()`, `vector()`, `<aggregation>_over_time()`

Query Example: Disk Usage

```
1 - ( node_filesystem_avail{instance="localhost:9090"}
    / node_filesystem_size{instance="localhost:9090"} )
```

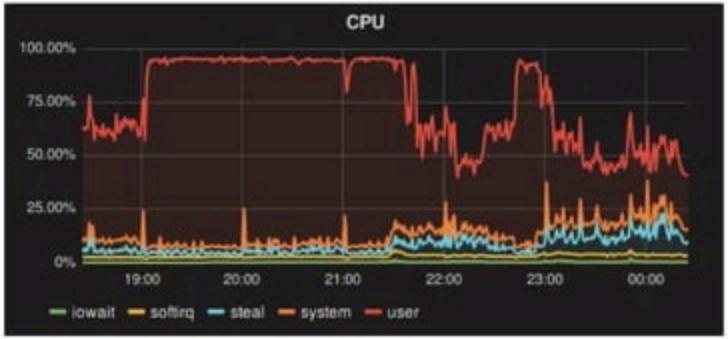
Query Example: Network Traffics

```
irate(node_network_receive_bytes{instance="localhost:9100"}[5m])
-irate(node_network_transmit_bytes{instance="localhost:9100"}[5m])
```



Query Example: CPU Usage

```
avg by (mode)(
  irate(node_cpu{instance="localhost:9100", mode!="idle"}[5m]))
```



Alert Rules

```
ALERT DiskUsageOver80Percent
  IF node_filesystem_avail / node_filesystem_size < 0.2
  FOR 5m
```

Alert Rules

```
ALERT DiskUsageOver80Percent
  IF node_filesystem_avail / node_filesystem_size < 0.2
  FOR 5m
  LABELS { severity = "critical" }
  ANNOTATIONS {
    description = "{{ $labels.instance }} disk usage has over 80%."
    link = "<Grafana URL>"
  }
```

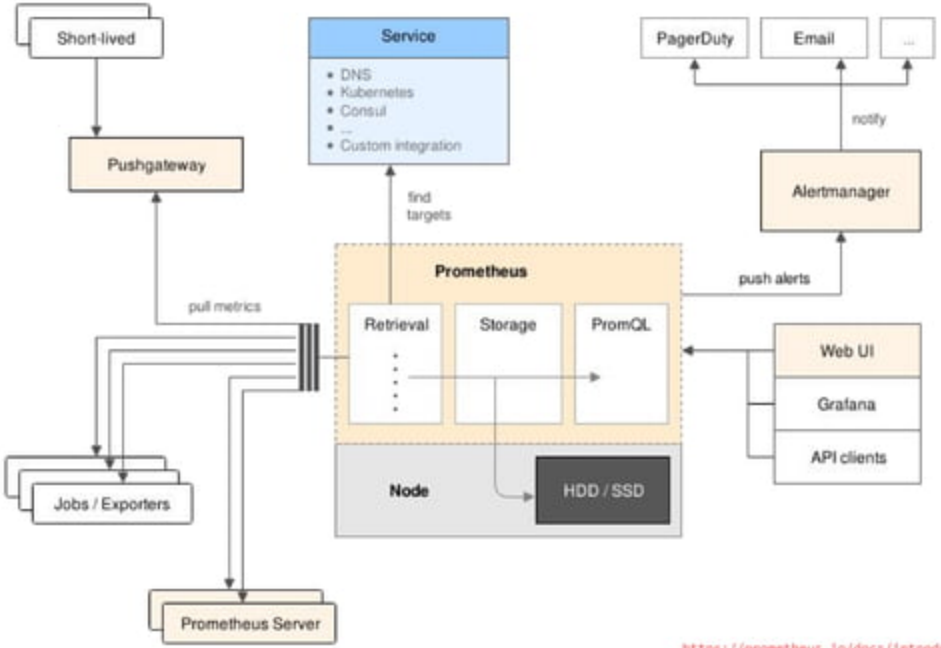
Alert Rules

```
ALERT DiskWillFillIn4Hours
  IF predict_linear(node_filesystem_free[1h], 4*3600) < 0
  FOR 5m
```

https://www.robustperception.io/reduce-noise-from-disk-space-alerts/

Alert Rules for Kafka

- Are brokers ingesting messages?
- Does the traffic increase/decrease rapidly than last hour/yesterday?
- Does the consumer lag over 30 minutes?



https://prometheus.io/docs/introduction/overview/

Alert Manager

```
route:
  routes:
  - match:
      severity: 'warning'
    receiver: 'email'
  receiver: 'slack'
receivers:
- name: 'slack'
  slack_configs:
  - channel: '#alert'
    api_url: 'https://hooks.slack.com/services/...'
    text: '{{ .CommonAnnotations.description }} {{ .CommonAnnotations.link }}'
…
```

Alert Manager

- Routing
- Grouping
- Inhibition
- Silences
- Receivers
  - Slack, Email, Webhook, ...

“*My Philosophy on Alerting*”

Rob Ewaschuk (former Site Reliability Engineer at Google)

- Pages should be urgent, important, actionable, and real.
- Over-monitoring is a harder problem to solve than under-monitoring.
- Cause vs. symptom
- Every page should require intelligence to deal with.

https://docs.google.com/a/boxever.com/document/d/199PqyG3UsyXlwieHaqbGiWVa8eMWi8zzAn0YfcApr8Q/edit

Pull doesn't scale?

- Prometheus is not an event-based system
- No spawning subprocess
- A single big Prometheus server can easily store 800,000 incoming samples per second
- Federation

https://prometheus.io/blog/2016/07/23/pull-does-not-scale-or-does-it/
https://www.robustperception.io/scaling-and-federating-prometheus/

Alert Alternatives

- Nagios
- Grafana
- TICK (Telegraf, Influxdb, Chronograf, Kapacitor)
- ELK+Beats (ElasticSearch, Logstash, Kibana)

Conclusion

- Prometheus is easy, but monitoring is difficult
- Read all documents on the official site/blog
- Keep improving the monitoring & alert rules