SRE Monitoring with Prometheus

PromQL and Service Level Objectives

Jack Neely
jjneely@42lines.net

June 25, 2020

42 Lines, Inc.

Service Discovery and `up`

No one cares if one instance is down:

```
up{job="foo"} == 0
```

Are at least 3 instances up?

```
sum(up{job="foo"}) < 3
```

A ratio of healthy instances to total:

```
avg(up{job="foo"}) < 0.8
```

When service discovery takes your lunch:

```
sum(up{job="foo"}) < 3 OR absent(up{job="foo"})
```

Who's Bouncing Your Apps?

No one cares if an instance restarted 5 minutes ago:

```
time() - process_start_time_seconds < 300
```

Instances in a restart loop:

```
changes(process_start_time_seconds[1h]) > 3
```

All of your instances are restarting:

```
avg WITHOUT (instance) (
 changes(process_start_time_seconds[1h]) > bool 3
) > 0.5
```

Using build_info Like a Pro

Every application should export a build_info metric:

```
prometheus_build_info{goversion="go1.12.5",revision="<hash>",version="2.10.0"} 1
```

Shove the version label into other metrics:

```
go_memstats_alloc_bytes * on (instance)
 group_left(version) prometheus_build_info
```

Median memory usage across all instances:

```
quantile(0.5,
 go_memstats_alloc_bytes * on (instance)
 group_left(version) prometheus_build_info
) WITHOUT (instance)
```

Cheater's Guide to Histograms and SLOs

Make the SLO a parameter to the application and export it!

```
foobar_slo_seconds 10
```

Use these 3 Counters:

```
foobar_pipeline_events_total{handler="foo",client="iOS"} 42
foobar_pipeline_events_errors_total{handler="foo",client="iOS"} 3
foobar_pipeline_events_over_slo_total{handler="foo",client="iOS"} 1
```

SLO alert for 90% of events processed under 10 seconds for last 24 hours:

```
1 - rate(foobar_pipeline_events_over_slo_total[1d]) /
 rate(foobar_pipeline_events_total[1d])
< 0.9
```

Burn Down Your SLOs

SLO: Ratio ≥ 0.95 of events are successful for a 30 day period

Burn Rate: How fast you are consuming the error budget during Δt relative to the SLO.

$$BurnRate = \frac{ErrorRatio\Delta t}{1 - SLO}$$

$$BudgetConsumed = \frac{t}{period} \times BurnRate$$

Can we alert if 5% of the budget is consumed in the last hour? Yes! Solve for *BurnRate*.

```
job:slo_errors_per_request:ratio_rate1h > 36 * 0.05
```

Takeaways

Boolean metrics in Prometheus are very powerful.

Use Prometheus to build dynamic alerting on the health of your applications.

Building alerts on customer experience is very possible understanding some of Prometheus's caveats.

Consistent use of Prometheus best practices enable you to build customer experience alerting around an A/B or canary release process.

Thank You!

42lines.net

operations.fm