Anomaly detection with Prometheus

Jorge Arco

127.1

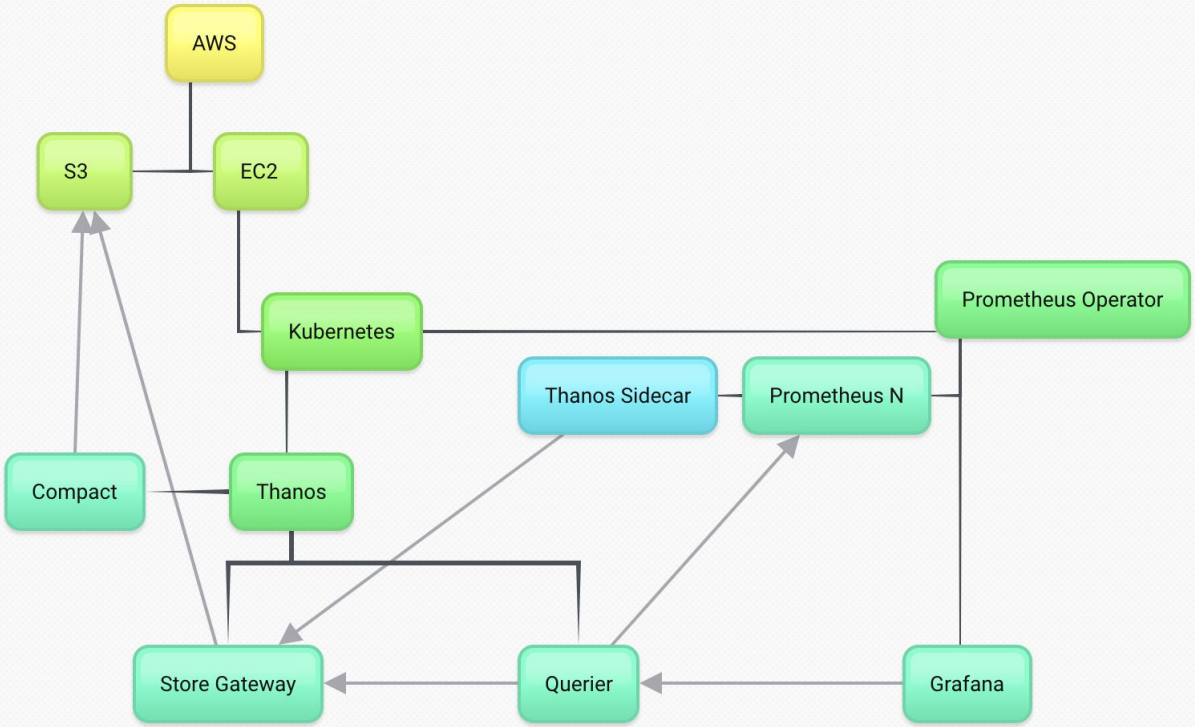
Jorge Arco

Lead SysAdmin @ Kodify.io

github.com/jorge07

Stack

- Kubernetes in EC2
- Prometheus Operator
- Thanos
- Grafana



Context

Adult industry

High volume

Traffic seasonality

Important number of attacks per week

React faster to possible mistakes

kodify MEDIA GROUP

Deeper.COM

PornTube®

TUSHY BLACKED

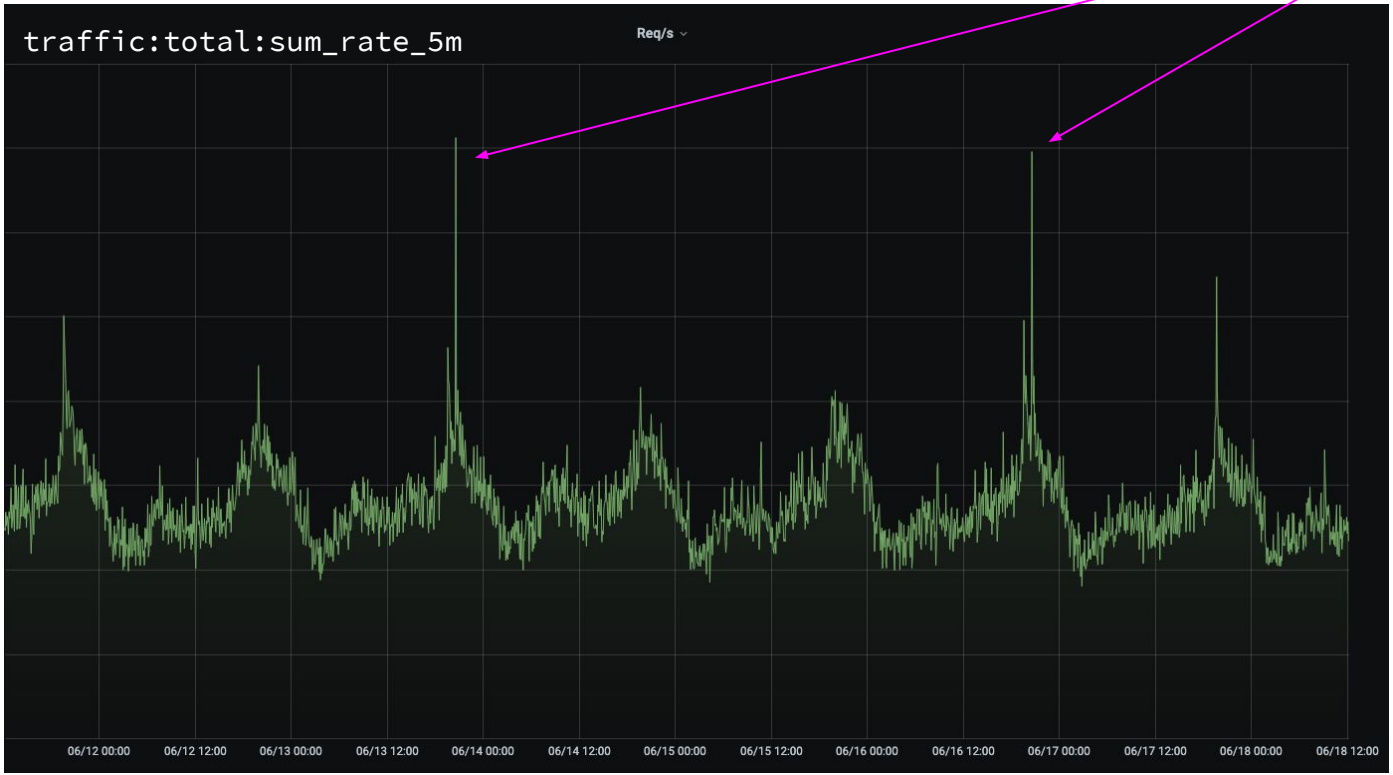
VIXEN BLACKED RAW

Problem

“We want to automatically detect

anomalies in our traffic”

Understanding the problem



Those are NOT anomalies

Traffic fluctuates over the year

Marketing campaigns are heavy and can trigger a low traffic alert when they finish

We may be comparing compare against prev months

Addressing the problem

Z-Score

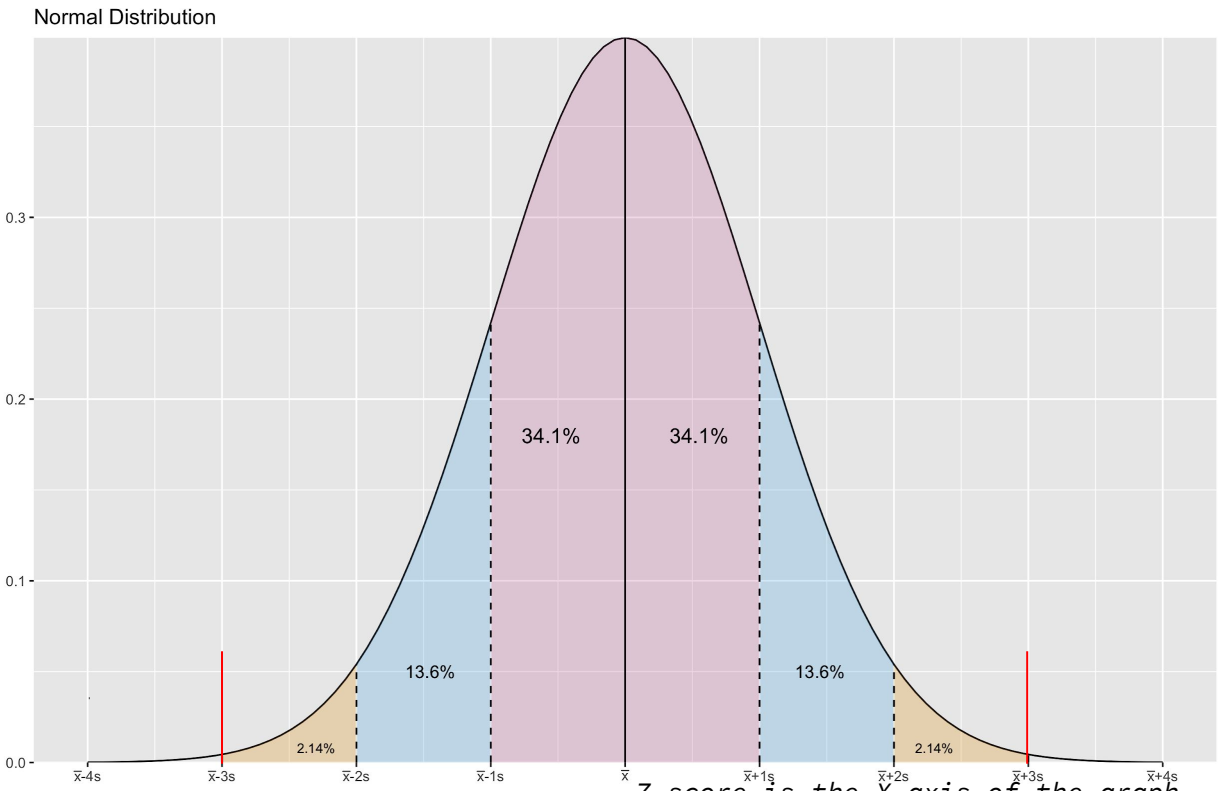
The z-score is measured in the number of standard deviations from the mean

We can assume that any value that falls outside of the range of roughly **+3 to -3 is an anomaly**

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation



Z-score is the X axis of the graph

Aggregation

- **Long term metrics require proper aggregation to avoid expensive queries, use less memory and waste less storage**.
- We use **traefik as Ingress controller** so we’re going to use this metrics to monitor our traffic

For this example we’re going to **exclude all the labels except backend** and aggregate 5 minutes

(backend = www.blacked.com | members.blacked.com)

```
record: traffic:total:sum_rate_5m
expr: sum by(backend) (rate(traefik_backend_requests_total{job="traefik-frontend-prometheus"}[5m]))
```

Z-Score calculation in prometheus

```
- record: traffic:total:sum_rate_5m
  expr: sum by(backend) (rate(traefik_backend_requests_total{job="traefik-frontend-prometheus"}[5m]))

- record: traffic:total:stddev_over_time_1w
  expr: stddev_over_time(traffic:total:sum_rate_5m[1w])

- record: traffic:total:avg_over_time_1w
  expr: avg_over_time(traffic:total:sum_rate_5m[1w])

- record: traffic:total:zscore_1w
  expr: (traffic:total:sum_rate_5m-traffic:total:avg_over_time_1w)/traffic:total:stddev_over_time_1w
```

Z-score results

Sum, week avg and week stddev

Z-Score

Seasonality

Growth trend per week:
Subtracting the rolling one-week average for last week from the rolling one-week average for now. Take 5 minutes frames from this week and the previous one to generate a prediction. You'll need to play with higher time frames depending of your normal distribution curve inclination *(How variable is you is your traffic timeframe).*

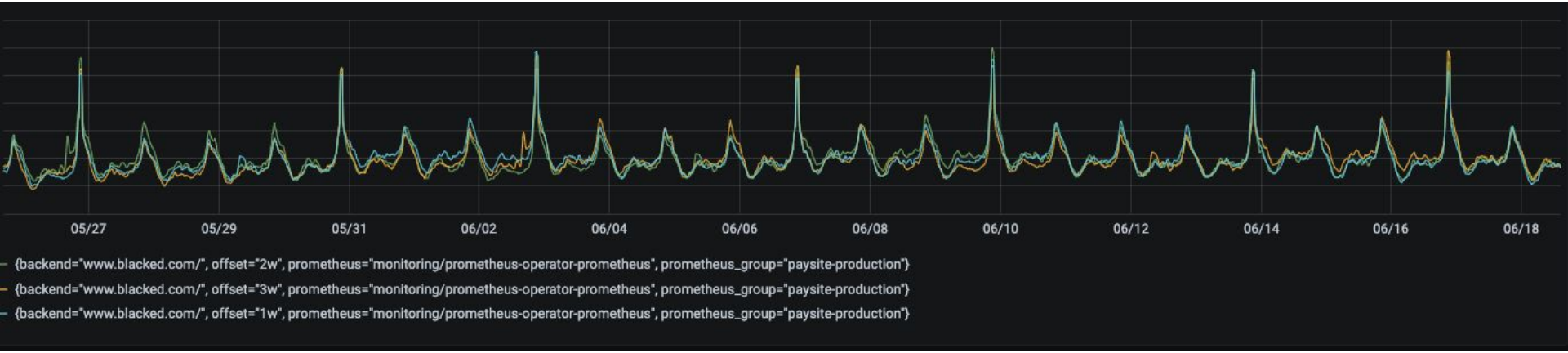
```
- record: traffic:total:sum_rate_5m_prediction_1w
  expr: >
    label_replace(
        traffic:total:sum_rate_5m{backend="www.blacked.com/"} offset 1w +
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 1w
    , "offset", "1w", "", "")

- record: traffic:total:sum_rate_5m_prediction_2w
  expr: >
    label_replace(
        traffic:total:sum_rate_5m{backend="www.blacked.com/"} offset 2w +
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 2w
    , "offset", "2w", "", "")

- record: traffic:total:sum_rate_5m_prediction_3w
  expr: >
    label_replace(
        traffic:total:sum_rate_5m{backend="www.blacked.com/"} offset 3w +
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 3w
    , "offset", "3w", "", "")

- record: traffic:total:sum_rate_5m_prediction_1h_1w
  expr: >
    label_replace(
        avg_over_time(
            traffic:total:sum_rate_5m{backend="www.blacked.com/"}[1h] offset 1w
        ) +
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
        traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 1w
    , "offset", "1w", "", "")
```

Seasonality / Trend



1, 2 and 3 week trend

Seasonality / Trend 1h time span

Let’s calculate the median for last 3w and validate results

```
- record: traffic:total:sum_rate_5m_prediction_for_3w
  expr: >
    quantile(0.5,
        traffic:total:sum_rate_5m_prediction_1w
        or
        traffic:total:sum_rate_5m_prediction_2w
        or
        traffic:total:sum_rate_5m_prediction_3w
    ) without (offset)
```

Metrics: `traffic:total:sum_rate_5m{backend="www.blacked.com/"}`

Metrics:

```
quantile(0.5,
  label_replace(avg_over_time(traffic:total:sum_rate_5m{backend="www.blacked.com/"}[1h]offset 1w)
+ traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 1w, "offset", "1w", "", "")
  or
  label_replace(avg_over_time(traffic:total:sum_rate_5m{backend="www.blacked.com/"}[1h]offset 2w)
+ traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 2w, "offset", "2w", "", "")
  or
  label_replace(avg_over_time(traffic:total:sum_rate_5m{backend="www.blacked.com/"}[1h]offset 3w)
+ traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 3w, "offset", "3w", "", "")
) without (offset)
```

Step auto 2.9s

Step auto 2.9s

+ Add query Query history

Graph

Taking 1h timeframe down work for us as fluctuation occurs is short time spans

traffic:total:sum_rate_5m{backend="www.blacked.com/", prometheus="monitoring/prometheus-operator-prometheus", prometheus_group="paysite-production"}

{backend="www.blacked.com/", prometheus="monitoring/prometheus-operator-prometheus", prometheus_group="paysite-production"}

Seasonality / Trend 5m time span

```
quantile(0.5,
  label_replace(traffic:total:sum_rate_5m{backend="www.blacked.com/"} offset 1w +
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 1w, "offset", "1w", "", "")
  or
  label_replace(traffic:total:sum_rate_5m{backend="www.blacked.com/"}offset 2w +
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 2w, "offset", "2w", "", "")
  or
  label_replace(traffic:total:sum_rate_5m{backend="www.blacked.com/"} offset 3w +
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} -
traffic:total:avg_over_time_1w{backend="www.blacked.com/"} offset 3w, "offset", "3w", "", "")
) without (offset)
```

Step auto 3.1s

Add query Query history

Ouh mama

Graph

06/16 16:00 06/16 20:00 06/17 00:00 06/17 04:00 06/17 08:00 06/17 12:00 06/17 16:00 06/17 20:00 06/18 00:00 06/18 04:00 06/18 08:00 06/18 12:00

traffic:total:sum_rate_5m{backend="www.blacked.com/", prometheus="monitoring/prometheus-operator-prometheus", prometheus_group="paysite-production"}

{backend="www.blacked.com/", prometheus="monitoring/prometheus-operator-prometheus", prometheus_group="paysite-production"}

With the algo validated let's generate the alerts and present this information properly

Alerting

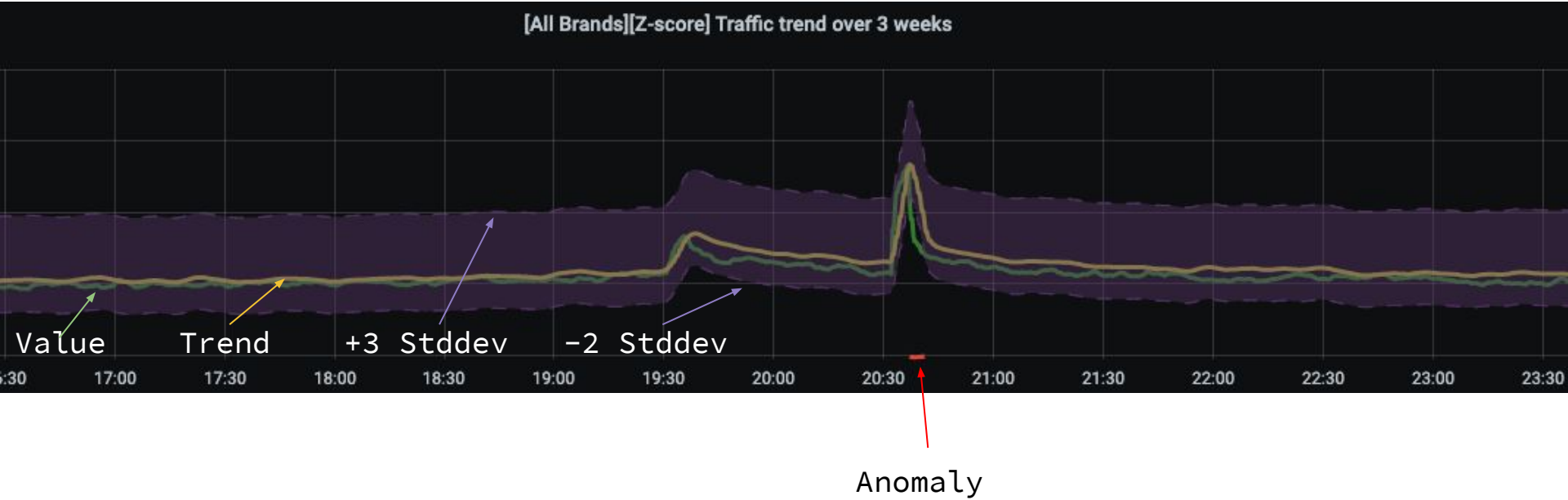
```
- record: traffic:total:zscore_for_3w_prediction
  expr: >
    (traffic:total:sum_rate_5m - traffic:total:sum_rate_5m_prediction_for_3w )
    / traffic:total:stddev_over_time_1w
```

We're going to take +-3 as our z-score limit

```
- alert: AnomalyInRequestRateInIngress
  expr: >
   abs(
     traffic:total:zscore_for_3w_prediction
   ) > 3
  for: 10m
  labels:
    severity: warning
    team: systeam
  annotations:
    summary: Requests for Ingress Rule {{ $labels.backend }} are outside normal
```

It doesn't need to be symmetric

Presenting the information



Caveats

- Long term storage retention in Prometheus cloud be very expensive (Thanos Ruler and Compact components helps a lot here)
- 2-3 Days retention in Prometheus is enough if you move the alert to Ruler component, this saves huge amount of memory
- Ruler component read path is distributed, it depends on network availability and Store Gateway to work, may not fit in all cases.

Resources

Prometheus

Thanos - Highly available Prometheus setup with long term storage capabilities

How to use Prometheus for anomaly detection in GitLab

Moving Z-Score

QA

T.Hanks