Grafana Labs

Practical Anomaly Detection at Scale With PromQL

Jorge Creixell
Principal Engineer

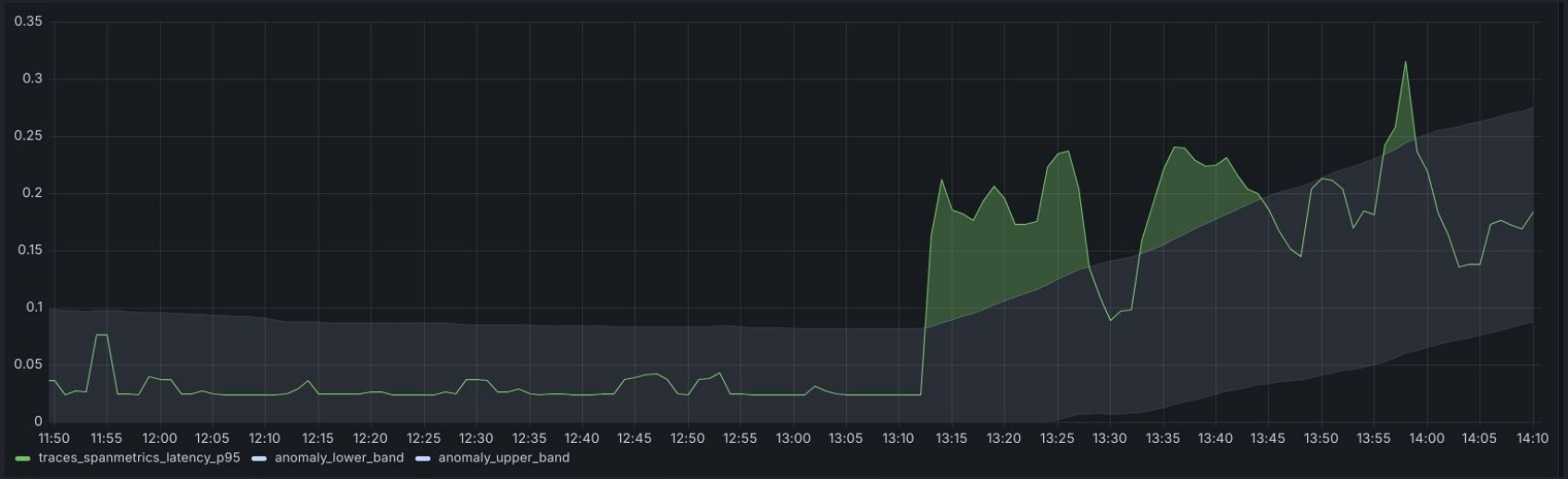
Manoj Acharya
VP Engineering (O11y)

Anomaly Detection

While investigating an incident, we noticed the following latency pattern. Is it **normal**?

Anomaly Detection

How about bringing the necessary context to the graph itself?



Disclaimer

There are many ways of solving this problem

Desired Properties

No External Systems

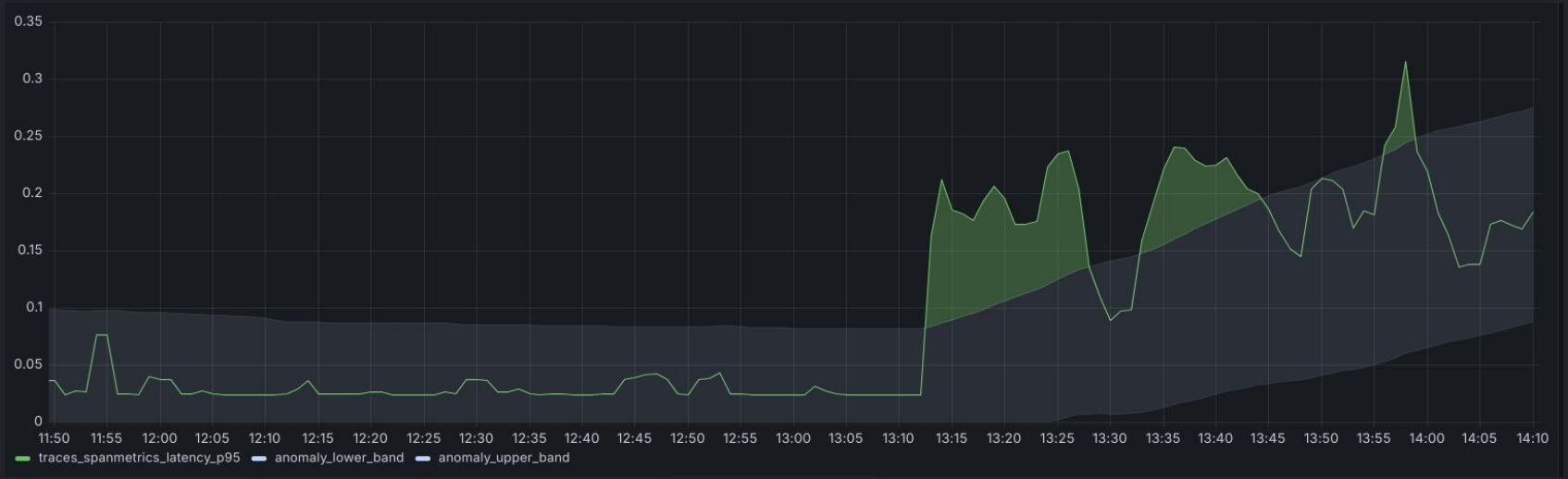
Performant at Scale

No Magic

Let’s Start!

The Idea

baselines = average ± stddev * multiplier



The Average

```
- record: avg_1h
  expr: avg_over_time(metric[1h])
```

The Standard Deviation

```
- record: stddev_26h
  expr: stddev_over_time(metric[26h])
```

The Standard Deviation

```
- record: stddev_26h
  expr: stddev_over_time(metric[26h])
```

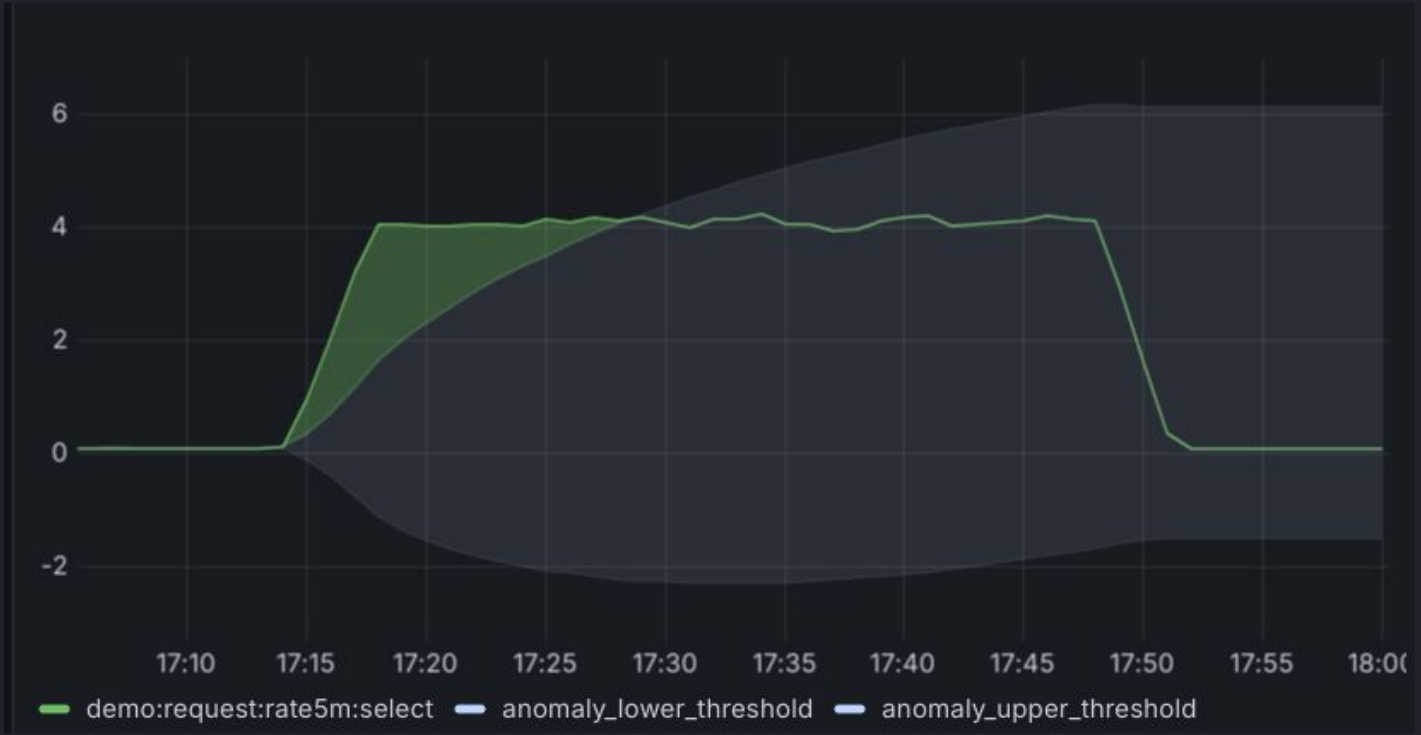
The Multiplier

```yaml
- record: stddev_multiplier
  expr: 2
```

Baselines: First Attempt

```
- record: upper_band_st
  expr: avg_1h + stddev_26h * on() group_left stddev_multiplier
```

Baselines: First Attempt



Problem 1: Extreme Outliers

Bands widen sharply in both directions in the presence of spikes

Idea: Smoothing Function

```
- record: stddev_1h
  expr: stddev_over_time(metric[1h])
```

```
- record: stddev_st
  expr: avg_over_time(stddev_1h[26h])
```

Problem 2: Low sensitivity

Bands take a long time to converge

Idea: High Pass Filter

stddev_1h increases in the presence of high variability

Idea: High Pass Filter

Filter out periods of low variability

Idea: High Pass Filter

What should be the threshold?

```
- record: stddev_1h:filtered
  expr: |
    stddev_over_time(metric[1h])
    > ?????
```

Idea: High Pass Filter

Coefficient of variation

```
- record: threshold_by_covar
  expr: 0.5
```

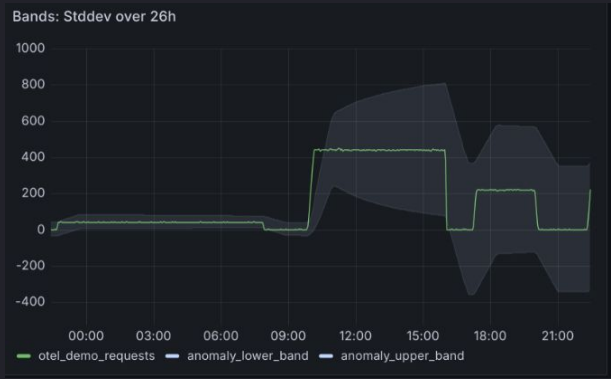
```
- record: stddev_1h:filtered
  expr: |
    stddev_over_time(metric[1h])
    > avg_1h * on() group_left threshold_by_covar
```

Solution : Smoothing + High Pass Filter

A good compromise between sensitivity and smoothness

Final Short Term Band

A side by side comparison



❌ ❌ 👍

Problem 3: Discontinuities

High pass filter removed bands for stable periods

Idea: Margin Bands

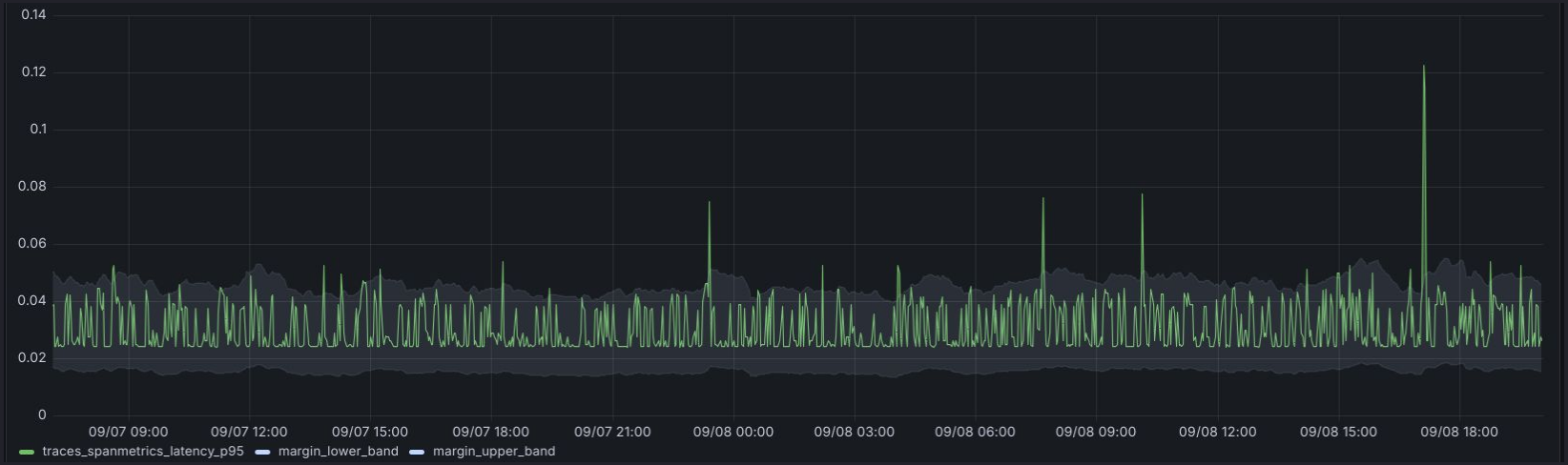
Minimum band width

```
- record: margin_multiplier
  expr: 2
```

```
- record: margin_upper_band
  expr: avg_1h + avg_1h * on() group_left margin_multiplier
```

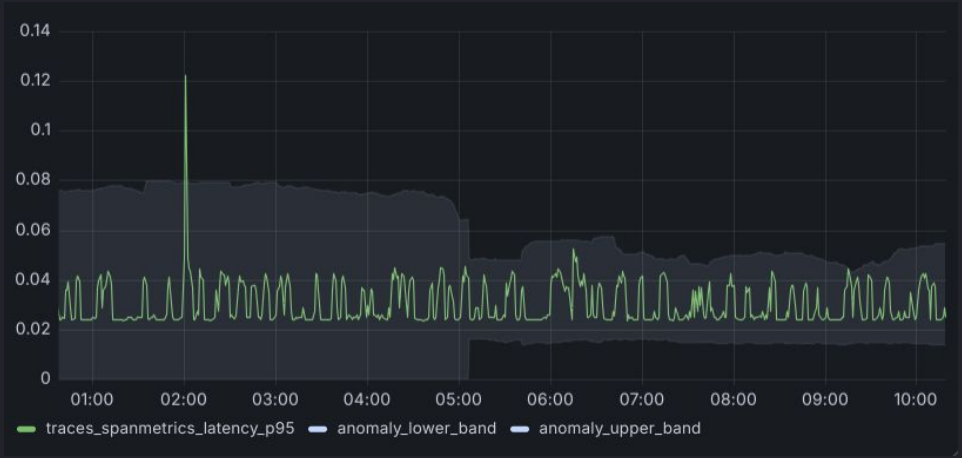
Solution: Margin Bands

Minimum tolerance bands



Combining the Bands

```
- record: upper_band
  expr: |
   max(
     margin_upper_band or
     upper_band_st
   )
```



Problem 4: Long Term Recurrent Patterns

Idea: Seasonality Bands

```
- record: upper_band_lt
  expr: |
    avg_1h offset 23h30m
    + stddev_1h offset 23h30m * on() group_left stddev_multiplier
```

Solution: Seasonality Bands

Putting It Together

```
- record: upper_band
  expr: |
   max(
     margin_upper_band or
     upper_band_st
     upper_band_lt
   )
```

The Framework

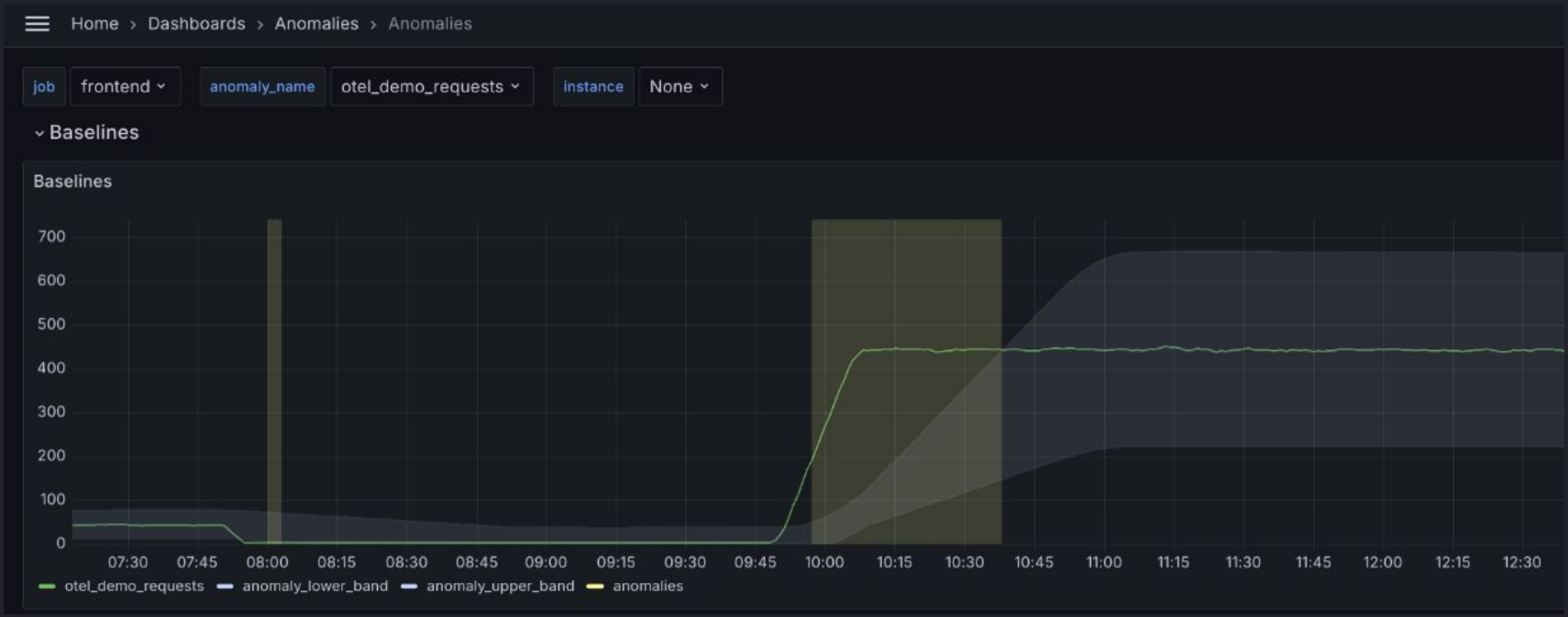
Tag your metrics

```
- record: anomaly:request:rate5m
  expr: sum(rate(duration_milliseconds_count[5m])) by (job)
  labels:
    anomaly_name: "otel_demo_requests"
    anomaly_type: "requests"
```

Alert Rule

```
- alert: AnomalyDetected
  for: 5m
  expr: |
    metric < lower_band
    or
    metric > upper_band
```

The Result


Home › Dashboards › Anomalies › Anomalies
job
frontend
anomaly_name
otel_demo_requests
instance
None
Baselines
Baselines
700
600
500
400
300
200
100
0
07:30
07:45
08:00
08:15
08:30
08:45
09:00
09:15
09:30
09:45
10:00
10:15
10:30
10:45
11:00
11:15
11:30
11:45
12:00
12:15
12:30
otel_demo_requests
anomaly_lower_band
anomaly_upper_band
anomalies

It’s all OSS!

github.com/grafana/promql-anomaly-detection

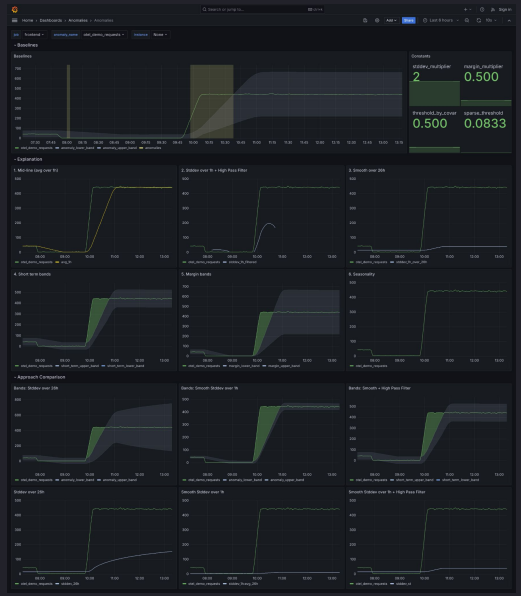
Recording Rules

Alerting Rules

Dashboard

Usage Examples

Demo



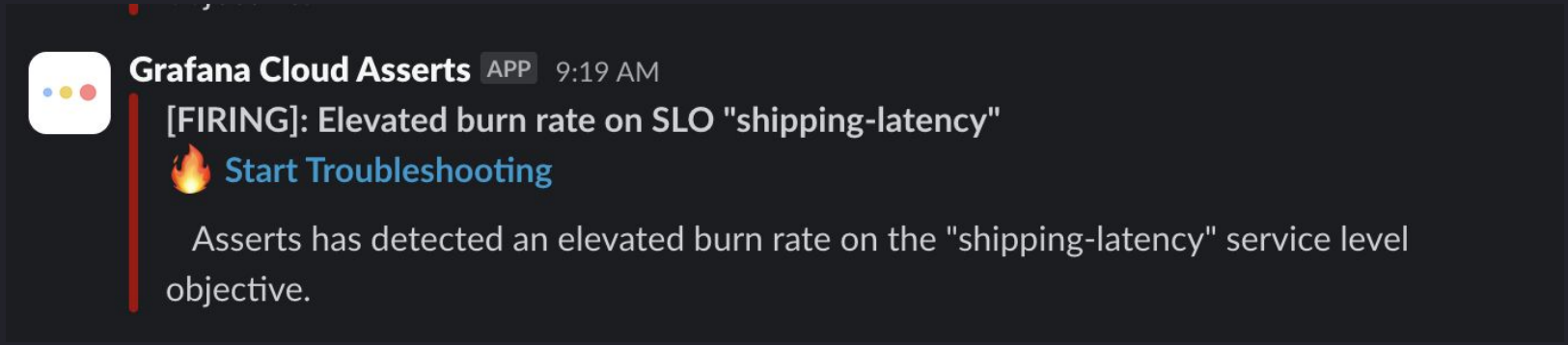
What now?

Should we page people at night on this Alert?

Maybe log warnings?

Don’t Page on Anomalies

Page on SLOs


Grafana Cloud Asserts
APP
9:19 AM
[FIRING]: Elevated burn rate on SLO "shipping-latency"
Start Troubleshooting
Asserts has detected an elevated burn rate on the "shipping-latency" service level objective.

Use anomalies to guide your attention

Home › Asserts › RCA workbench

RCA workbench

(6 Entities added) Clear

env prod × site app × 2024-09-10 11:52:38 to 2024-09-10 12:52:38

shipping connected services

Summary Timeline Graph Mind map

Click on timeline area to lock time

Sort By: Score 12:02 PM 09/10/2024 12:14 PM 12:22 PM 12:32 PM 12:42 PM

shipping
prod (app)
robot-shop

Service Version Updated
jvm:heap - JvmMajorGCCountHigh
jvm:heap - JvmMajorGCTimeHigh
error - ErrorLogRateBreach
/cities/{code} - Latencyp99high
/cities/{code} - LatencyAverageAnomaly
Logs Traces Metrics

Sep 10, 12:14:00 PM Logs Traces
Latency Average (ms): 2804.042567
Latency Average threshold (ms): 0-172.637422

3000 2000 1000 0

Latency Average (ms) Latency Average threshold (ms)

/health - LatencyAverageAnomaly

ingress-nginx-controller
prod (app)
nginx
Service

mysql
prod (app)
robot-shop
Service
buffer_pool_read_requests - RequestRateAnomaly
network:tx_bytes - ResourceRateAnomaly
cpu:usage - ResourceRateAnomaly

Open graph preview

Grafana Labs

Thank you!

Have more questions?

Join us at Grafana community slack

#promql-anomaly-detection

Creators of the framework

Jia Xu
Principal Engineer

Nandakumar Devi
Principal Engineer

Radhakrishnan Jankiraman
Director Engineering (O11y)

Get involved:

#promql-anomaly-detection

grafana/promql-anomaly-detection

community.grafana.com

Takeaways

- Baselines help you bring context to aid troubleshooting
- PromQL is all you need
- Do not page on anomalies
- Get started with our OSS framework

Appendix 1: stddev_1h

Bands expand and contract very fast. Unstable bands.

Turning it into a Framework

```
- record: metric
  expr: { anomaly_name!="", anomaly_select=”” } > 0
  labels:
    anomaly_select: 1
```