# Intro to Prometheus

Frederic Branczyk

Prometheus/Kubernetes @ CoreOS

@fredbrancz  |  frederic.branczyk@coreos.com

# What's Prometheus?

- **Open Source Monitoring system**

- **Founded at SoundCloud**

- **2nd CNCF Project**

- **What borgmon is to borg**

  - **Prometheus is to Kubernetes**

# cars_total

```
# HELP cars_total Total number of cars passed.
# TYPE cars_total counter
cars_total 4
```

# Dimensions

```
# HELP cars_total Total number of cars passed.
# TYPE cars_total counter
cars_total{color="white"} 2
cars_total{color="cyan"} 1
cars_total{color="orange"} 1
```

# Metadata

```
# HELP cars_total Total number of cars passed.
# TYPE cars_total counter
cars_total{color="white"} 2
cars_total{color="cyan"} 1
cars_total{color="orange"} 1
```

▼ **Collected from observer at "mainstreet"**

```
cars_total{color="white",street="mainstreet"} 2
cars_total{color="cyan",street="mainstreet"} 1
cars_total{color="orange",street="mainstreet"} 1
```

# Querying

```
# Is traffic congested?


rate(sum(cars_total)[5m]) < 10
```
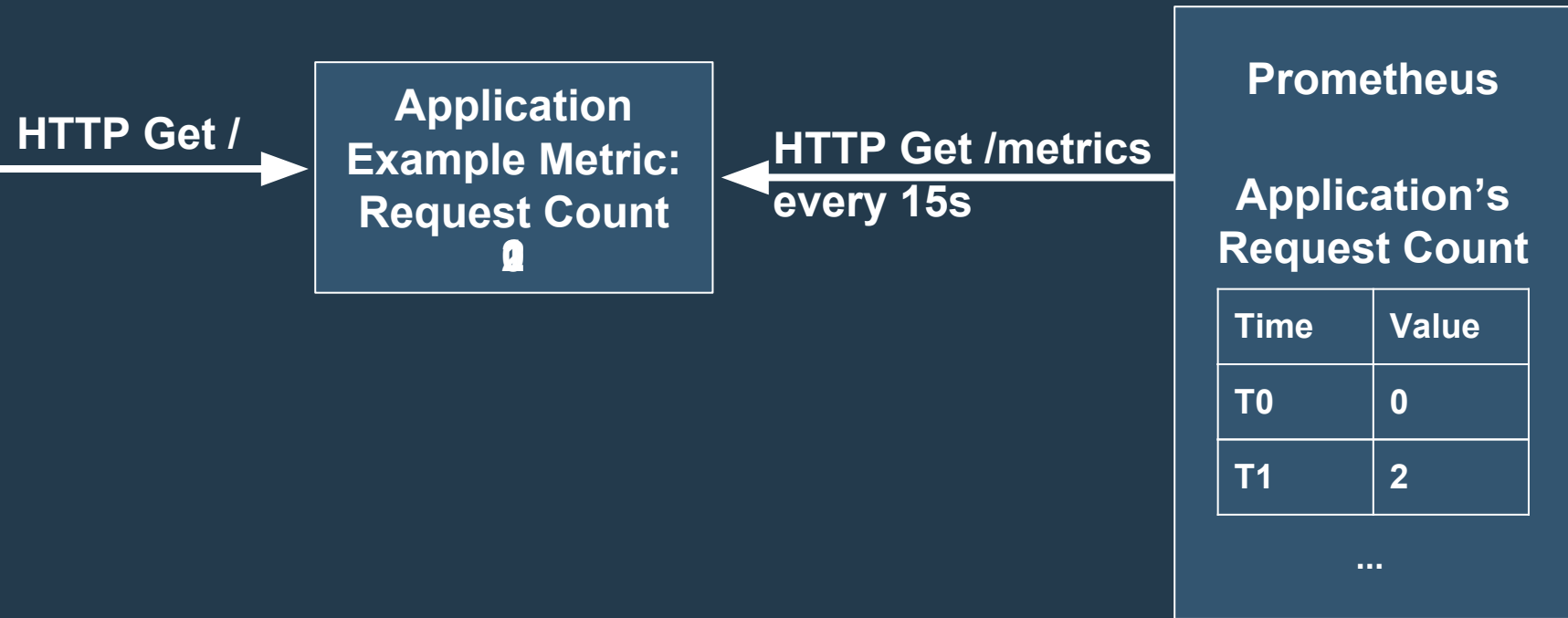
```
# Is traffic flowing?


rate(sum(cars_total)[5m]) > 10
```

# Ok, but how does it really work?

# Quick Prometheus intro

HTTP Get /  →  **Application Example Metric: Request Count**  ←  **HTTP Get /metrics every 15s**  **Prometheus**

**Application's Request Count**

| Time | Value |
|------|-------|
| T0   | 0     |
| T1   | 2     |

...

# What's a target?

- HTTP Server with /metrics endpoint

- Discovered by a SD mechanism

    - Static target list

    - DNS discovery

    - Kubernetes discovery

# Kubernetes Discovery

- Targets to discover

    - Pods

    - Nodes

    - Endpoints/Services

- Automatically reconfigure

    - Add, update, remove

# Metadata

```
# HELP http_requests_total Total number of HTTP requests made.

# TYPE http_requests_total counter

http_requests_total{code="200"} 8
```

⬇ **Enrich with target metadata**

```
http_requests_total{pod="pod-01",service="myservice",code="200"} 8
```
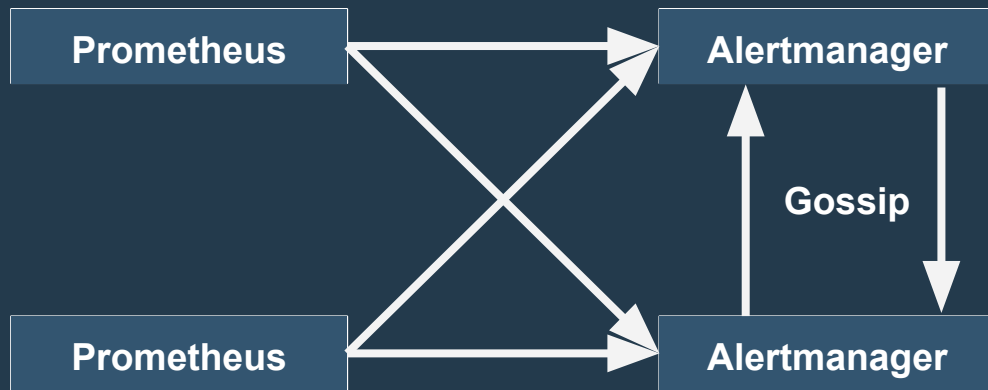
# Alert Evaluation

Rule 1    ●   **Evaluate Rule/Alert**

Rule 2    ●   **Fire alert against Alertmanager**

Rule 3

...

**Repeat in *rule evaluation interval***

# Alertmanger

- **Deduplicate alerts**

- **Group alerts**

- **Route alerts**

- **Send alerts**
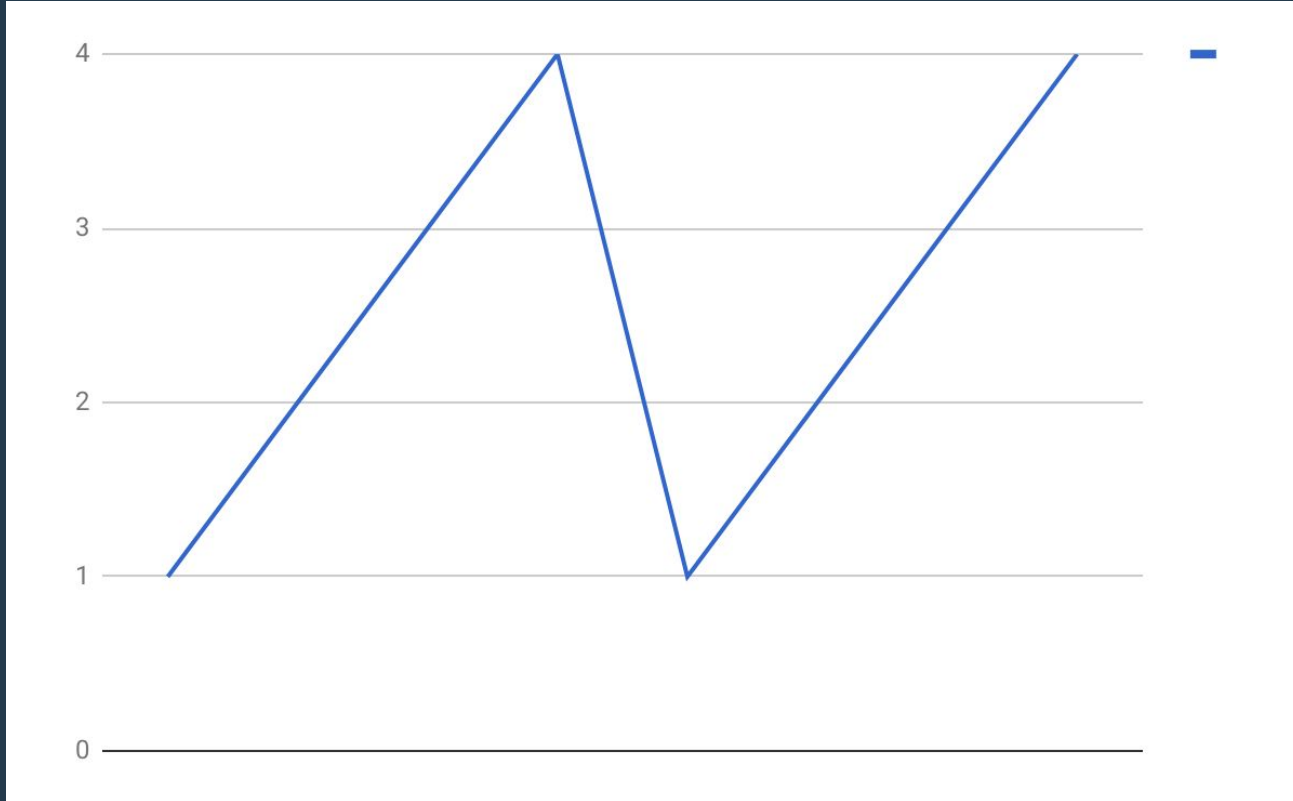
# Designing metrics

- **Labels identify a time-series**

- **Cardinality explosions**

  - **No variable data in metrics**

```
http_requests_total{instance="mytarget1:8080",job="myservice",code="200"} 8
```
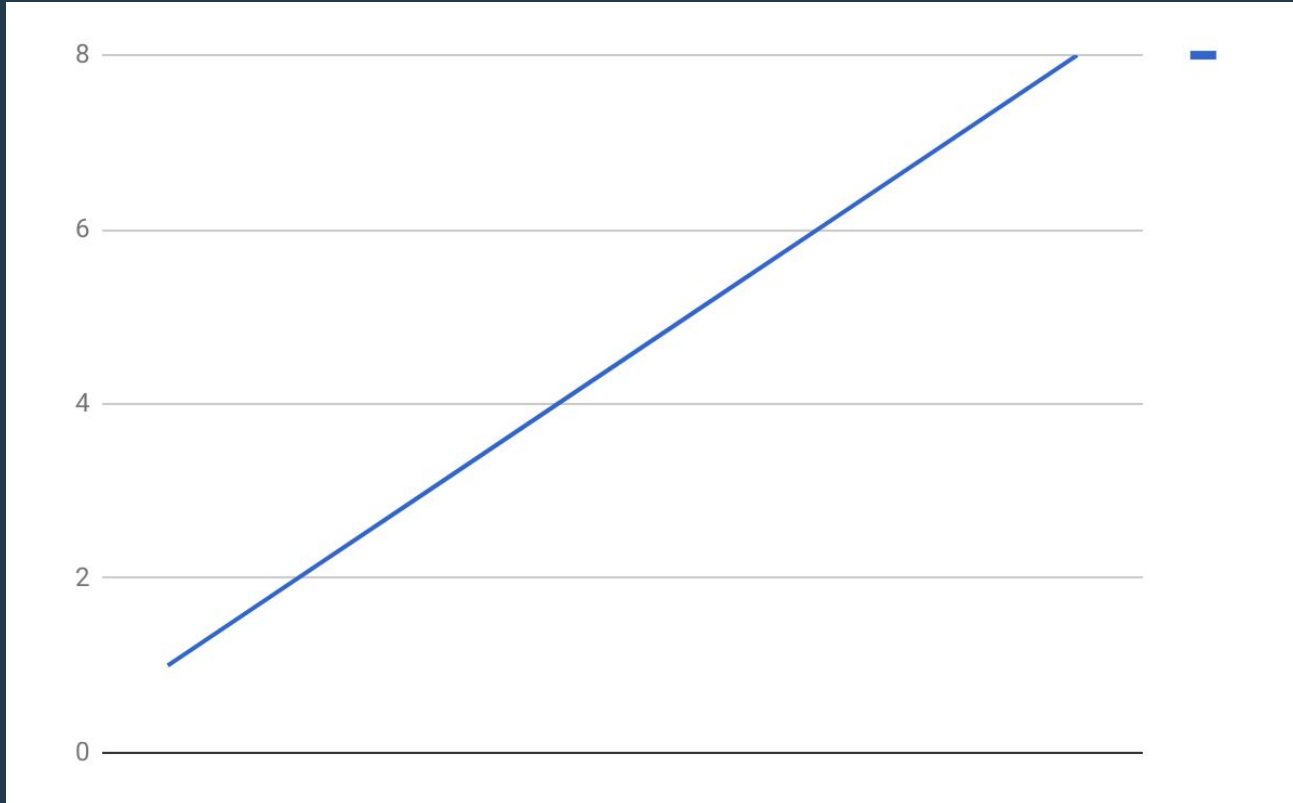
# Metric types

- **Counter - eg. Request Count**

- **Gauge - eg. Resource Usage**

- **Histogram - eg. Request Latency**

# Counter rate continuation

# Counter rate continuation

# Thank you!

Frederic Branczyk

@fredbrancz  |  frederic.branczyk@coreos.com