# The RED Method

*Patterns for instrumentation & monitoring.*

**Introduction**
Why does this matter?

**The USE Method**
Utilisation, Saturation, Errors

**The RED Method**
Requests Rate, Errors, Duration..

**The Four Golden Signals**
RED + Saturation

THIS PRESENTATION IS AN ADAPTION FROM A PRESENTATION
MADE BY GRAFANA LABS AND PUBLICLY AVAILABLE ON THE NET

# *The USE Method*

**For every resource, monitor**

**U**tilization — % time that the resource was busy

**S**aturation — amount of work resource has to do, often queue length

**E**rrors — the count of error events

| | Utilisation | Saturation | Errors |
|---|---|---|---|
| **CPU** | ✔ | ✔ | ✔ |
| **Memory** | ✔ | ✔ | ✔ |
| **Disk** | ✔ | ✔ | ✔ |
| **Network** | ✔ | ✔ | ✘ |

http://www.brendangregg.com/usemethod.html

**CPU Utilisation**:

```
1 - avg(rate(node_cpu{job="default/node-exporter",mode="idle"}[1m]))
```

**CPU Saturation**:

```
sum(node_load1{job="default/node-exporter"})
                        /
         sum(node:node_num_cpu:sum)
```
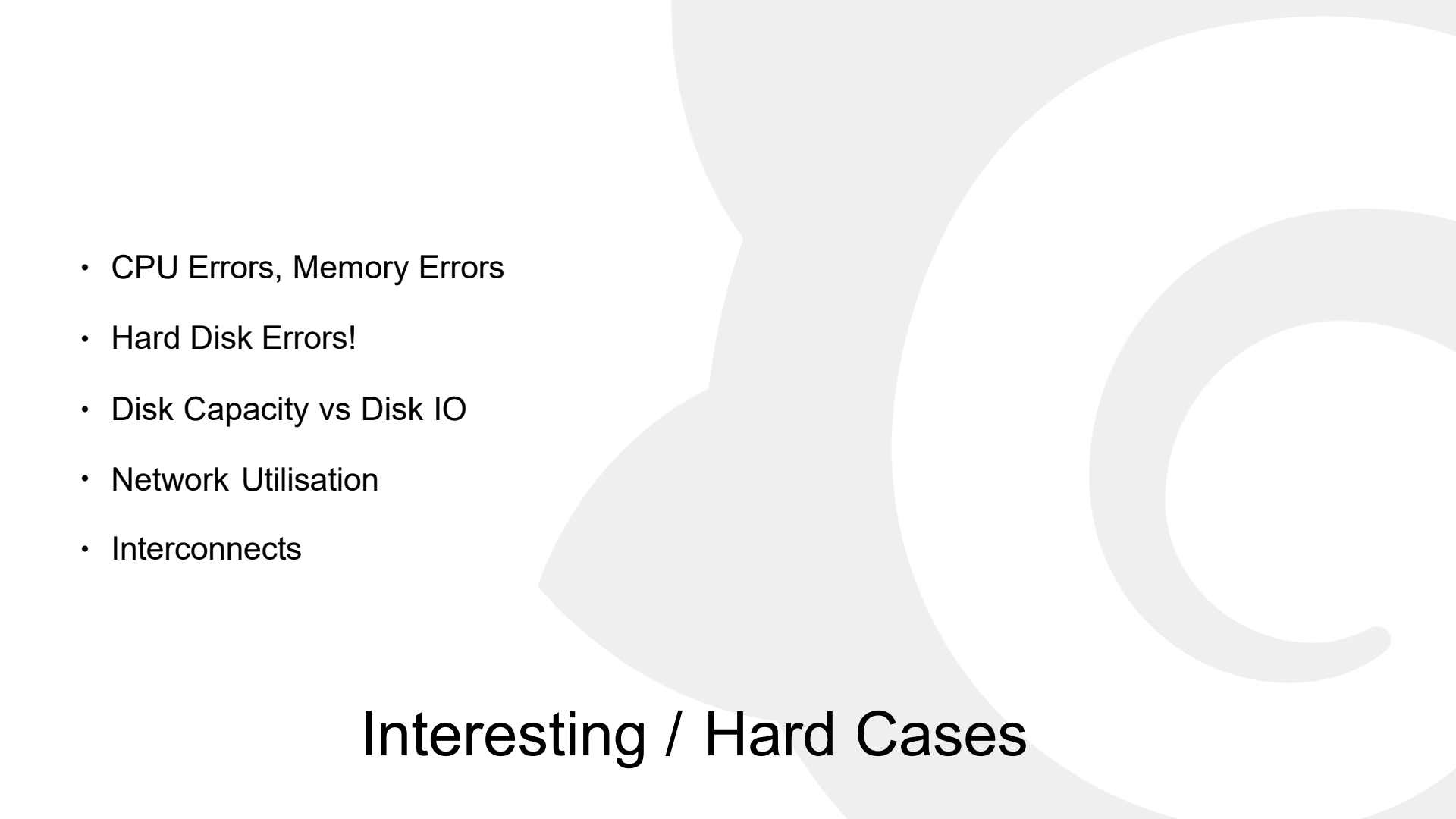
# CPU USE in Prometheus

**Memory Utilisation**:

```
1 - sum(
    node_memory_MemFree{job="…"} +
    node_memory_Cached{job="…"} +
    node_memory_Buffers{job="…"}
)
/ sum(node_memory_MemTotal{job="…"})
```

**Memory Saturation**:

```
1e3 * sum(
    rate(node_vmstat_pgpgin{job="…"}[1m]) +
    rate(node_vmstat_pgpgout{job="…"}[1m]))
)
```

# Memory USE in Prometheus

- CPU Errors, Memory Errors

- Hard Disk Errors!

- Disk Capacity vs Disk IO

- Network Utilisation

- Interconnects

# Interesting / Hard Cases

- "The USE Method" - Brendan Gregg

- Kubernetes Mixin - https://github.com/grafana/jsonnet-libs

# More Details

# The RED Method

For every service, monitor request:

- **Rate** - number of requests per second

- **Errors** - the number of those requests that are failing

- **Duration** - the amount of time those requests take

# The RED Method

**Rate**:

sum(rate(request_duration_seconds_count{job="…"}[1m]))

**Errors**:
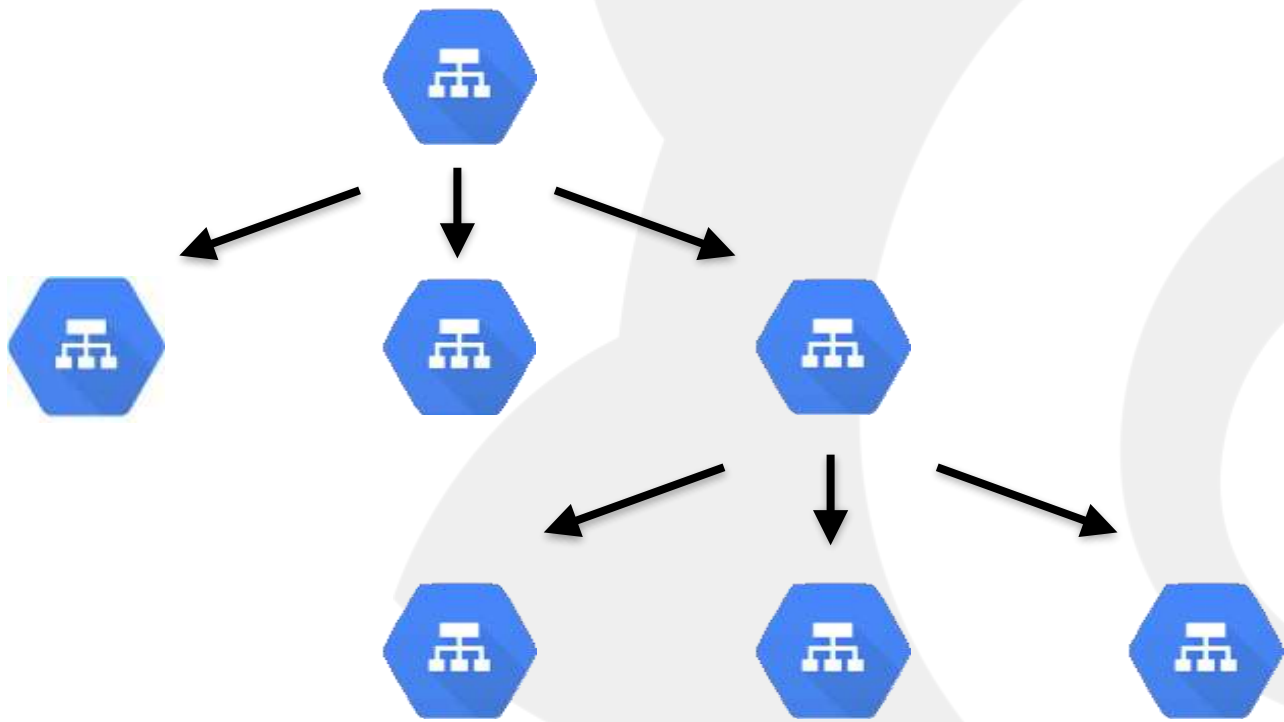
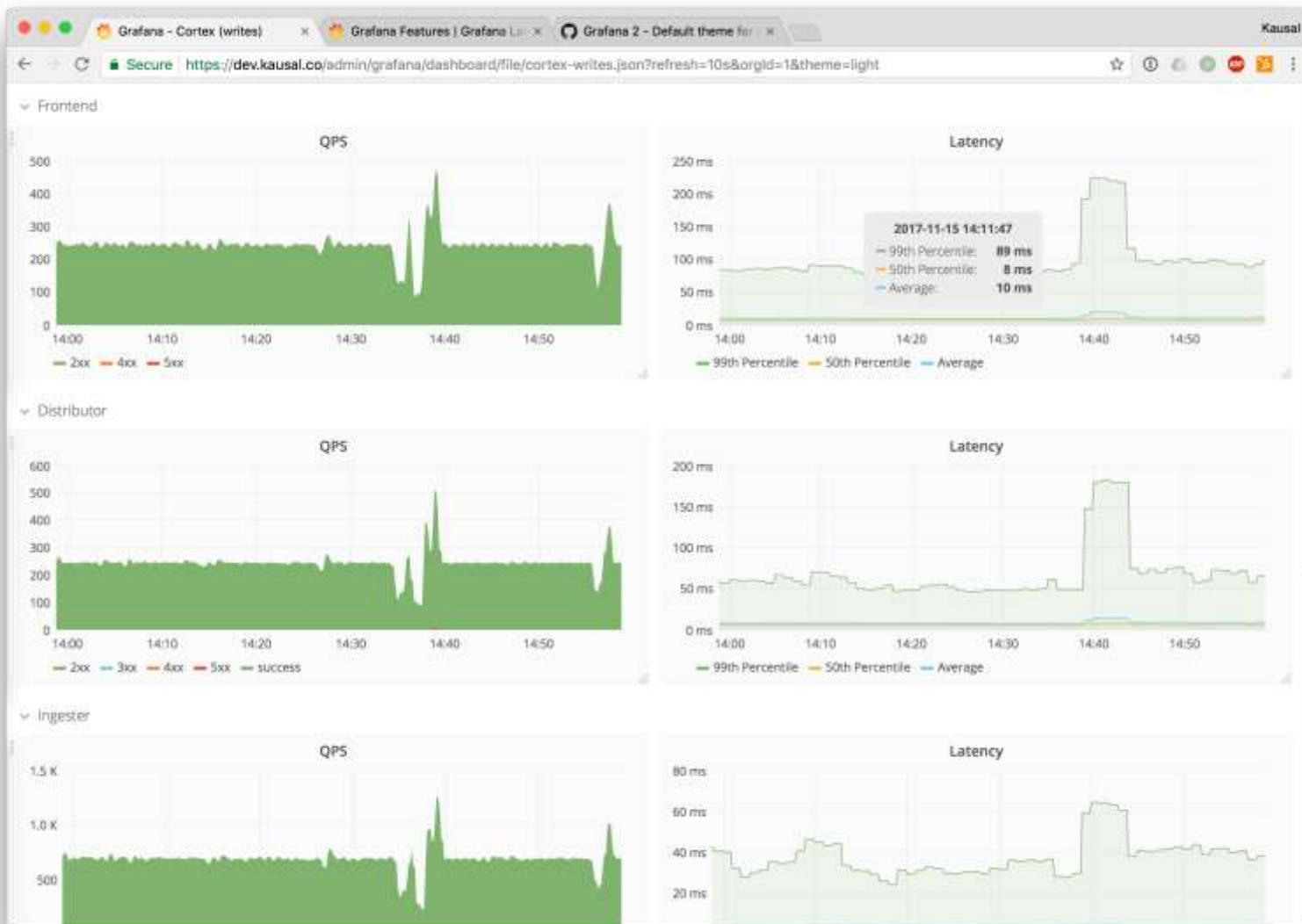sum(rate(request_duration_seconds_count{job="…",
status_code!~"2.."}[1m]))

**Duration**:
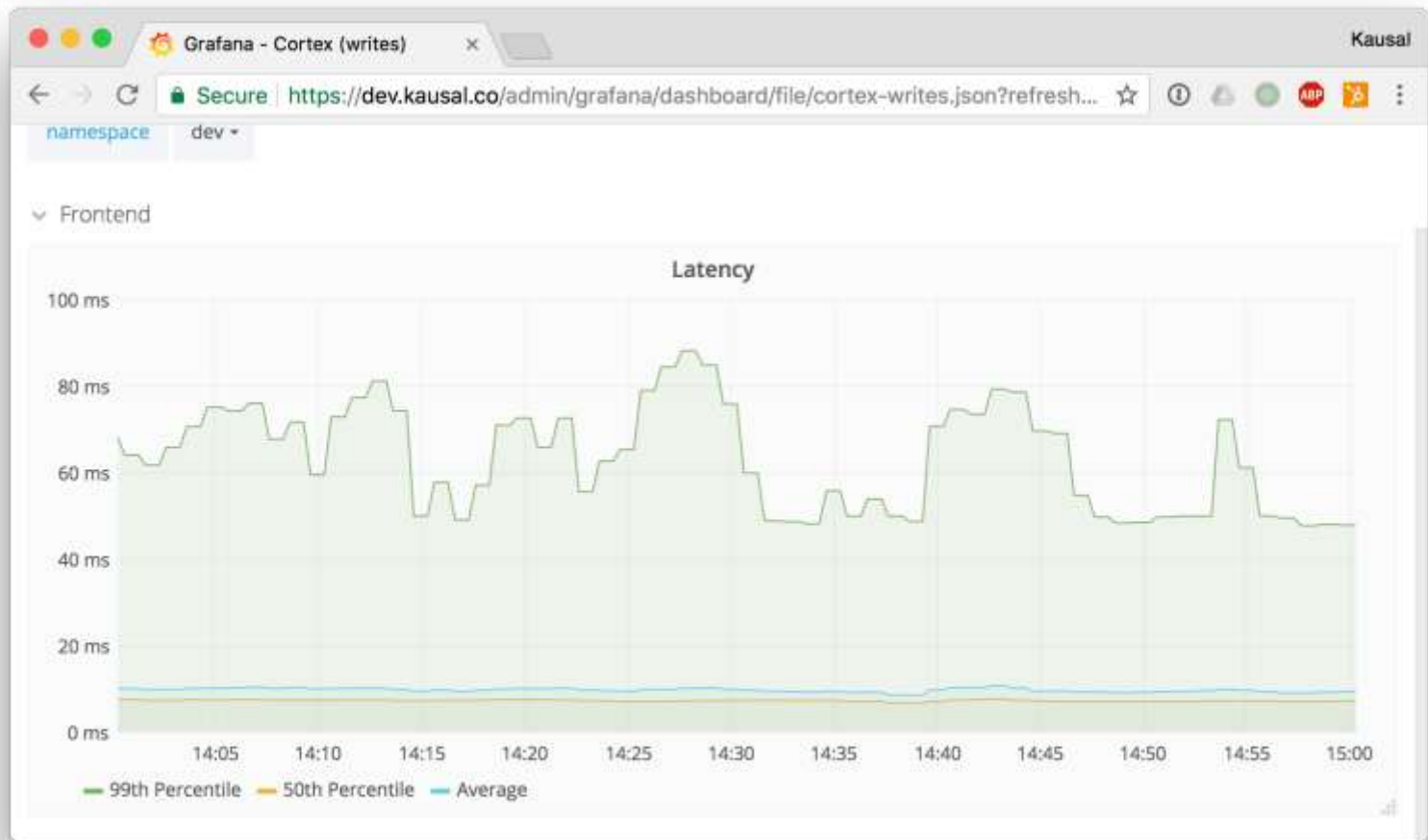
histogram_quantile(0.99,
sum(rate(request_duration_seconds_bucket {job="…}[1m])) by (le))

# Easy to query

DAG of Services

Latencies & Averages

- "[Monitoring Microservices](#)" - Weaveworks (slides)

- "[The RED Method: key metrics for microservices architecture](#)" - Weaveworks

- "[Monitoring and Observability with USE and RED](#)" - VividCortex

- "[RED Method for Prometheus – 3 Key Metrics for Monitoring](#)" - Rancher Labs

- "[Logs and Metrics](#)" - Cindy Sridharan

- "[Logging v. instrumentation](#)", "[Go best practices, six years in](#)" - Peter Bourgon

# More Details

# *The Four*
# *Golden Signals*

For each service, monitor:

- **Latency** - time taken to serve a request

- **Traffic** - how much demand is places on your system

- **Errors** - rate or requests that are failing

- **Saturation** - how "full" your services is

# The Four Golden Signals

- **Saturation** - how "full" your services is

- "The Four Golden Signals" - The Google SRE Book

- "How to Monitor the SRE Golden Signals" - Steve Mushero

# More Details