



# Prometheus + Grafana

Ровнягин Михаил Михайлович

5.11.2024



# А зачем вообще?

**Мониторинг** – это сбор, обработка, агрегирование и отображение количественных показателей системы в реальном времени

1. Мониторинг стоит денег
2. Требуется поддержки
3. Возможны ложные срабатывания
4. ...

# А зачем вообще?



1. Понимание как ведет себя приложение в **production**
2. Оповещение о проблемах
3. Долгосрочный и ретроспективный анализ

## Whitebox

Наблюдение  
показателей, доступных  
внутри системы

## Blackbox

Наблюдение  
поведения со стороны  
пользователя

## 4 golden signals

- Latency
- Traffic
- Errors
- Saturation

## RED

- Rate
- Errors
- Duration

## USE

- Utilization
- Saturation
- Errors

# Подходы. Latency

## Не берите среднее от latency!

00:30:00 - 200 ms

00:30:30 - 300 ms

00:31:00 - 1000 ms

00:31:30 - 300 ms

00:32:00 - 200 ms

$$(200 + 300 + 1000 + 300 + 200) / 5 = 400\text{ms}$$

## Берите квантиль от latency

99q = 972 ms

95q = 860 ms

00:30:00 - 200 ms

00:30:30 - 800 ms

00:31:00 - 1000 ms

00:31:30 - 700 ms

00:32:00 - 750 ms

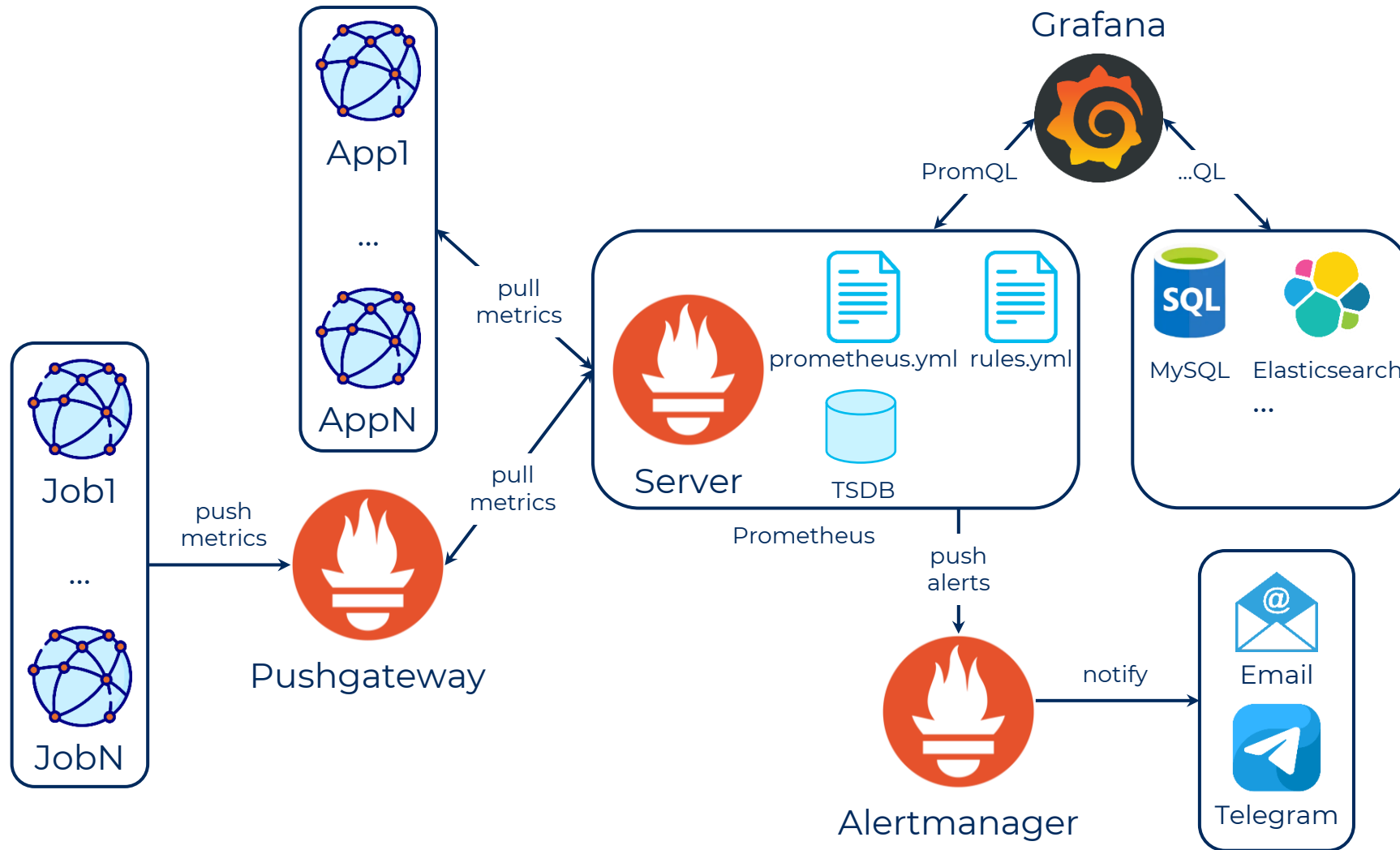
99q = 992 ms (против 972 ms)

95q = 960 ms (против 860 ms)

Квантиль не идеален

1. У удава толстый хвост
2. Мы не всегда знаем  
распределение запросов  
внутри квантиля

# Prometheus + Grafana





1. Данные хранятся в виде временных рядов
2. Каждый временной ряд идентифицируется именем и набором лейблов

```
http_requests_total{method="GET", host="example.com"}
```

```
http_requests_total{method="GET", host="example.com", query="/example/UUID1"}
```

```
http_requests_total{method="GET", host="example.com", query="/example/UUID2"}
```

```
http_requests_total{method="GET", host="example.com", query="/example/UUID3"}
```

# Prometheus. Типы метрик

- Counter
- Gauge

## Histogram

- распределяет значения по бакетам и считает их количество
- по гистограмме можно посчитать квантиль
- используется для замера **latency**

## Summary

- Результат агрегации гистограммы
- сразу считает квантили
- используется редко