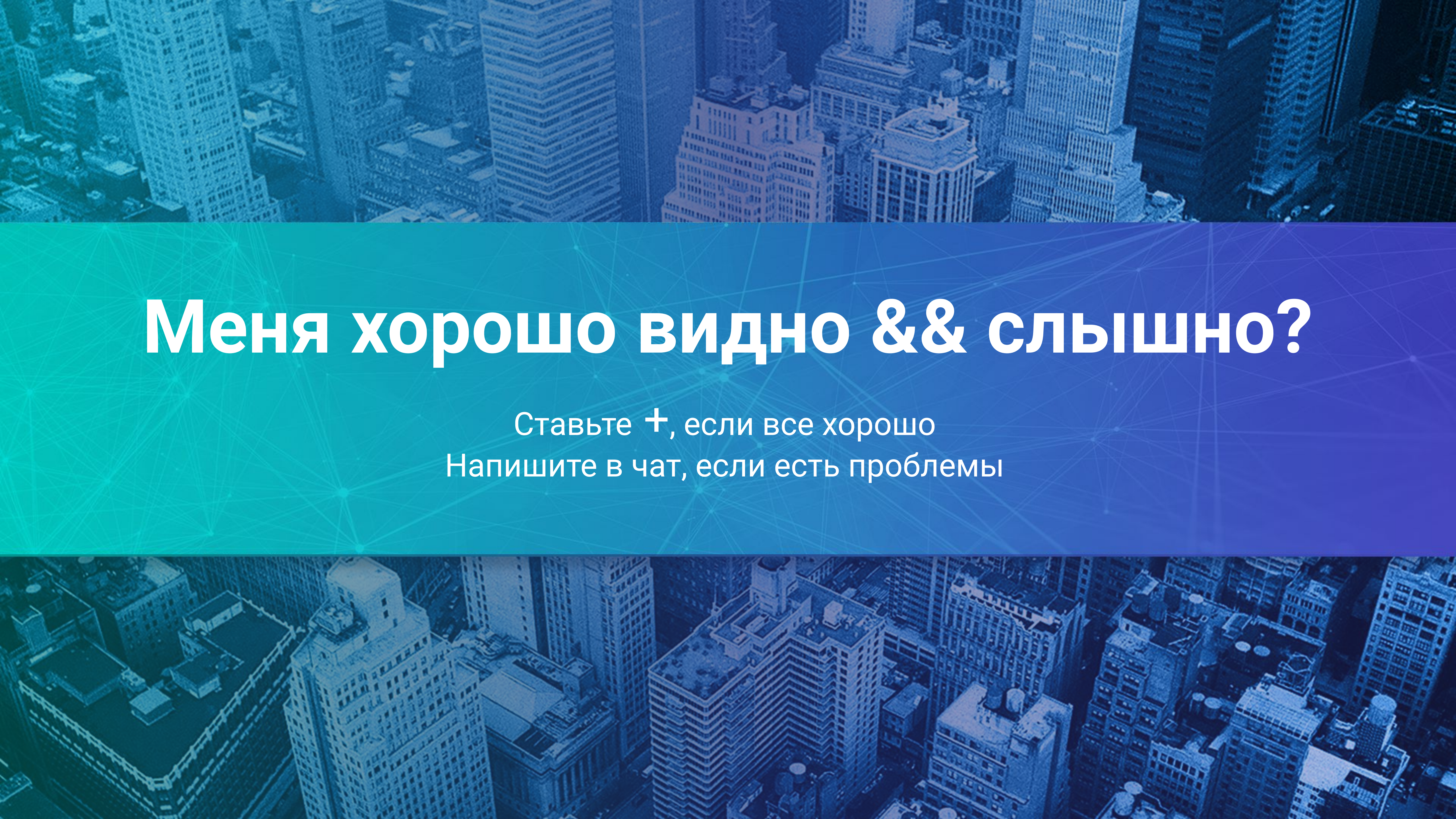
OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте +, если все хорошо
Напишите в чат, если есть проблемы

Проверить, идет ли запись!

OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Prometheus

Правила вебинара

Активно участвуем и делимся опытом

Задаем вопрос в чат или голосом

Off-topic обсуждаем в Slack

Вопросы вижу в чате, могу ответить не сразу

Цели и смысл вебинара

1 Из чего состоит экосистема Prometheus

2 Узнать как собирать и смотреть метрики

History

Open-source tool made by SoundCloud in 2012

Joined CNCF in 2016

Now completely separated from SoundCloud and developed by community

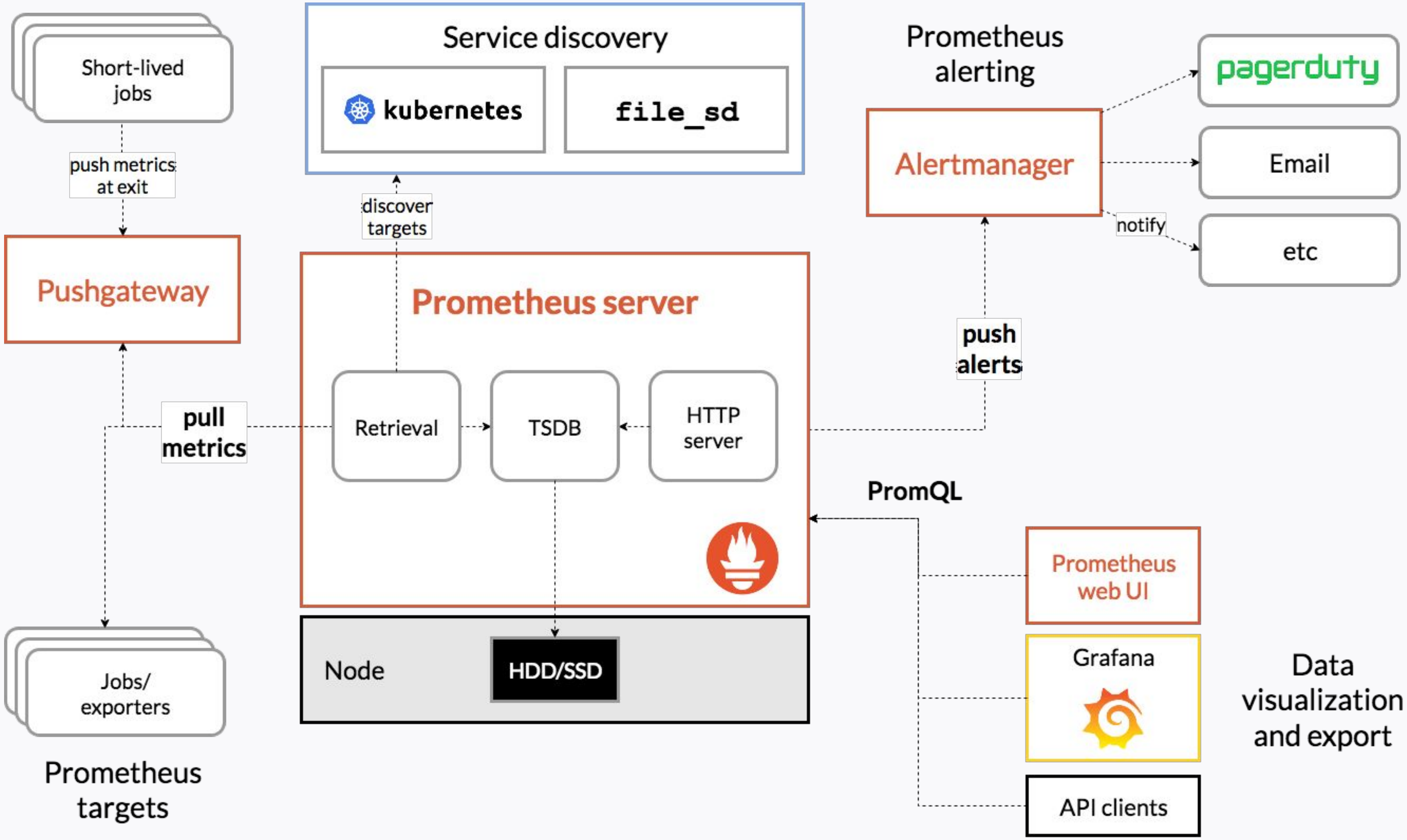
Features

- a multi-dimensional data model with time series data identified by metric name and key/value pairs
- PromQL, a flexible query language to leverage this dimensionality
- no reliance on distributed storage; single server nodes are autonomous
- time series collection happens via a pull model over HTTP
- pushing time series is supported via an intermediary gateway
- targets are discovered via service discovery or static configuration
- multiple modes of graphing and dashboarding support

Main Components

- the main Prometheus server which scrapes and stores time series data
- client libraries for instrumenting application code
- a push gateway for supporting short-lived jobs
- special-purpose exporters for services like HAProxy, StatsD, Graphite, etc.
- an alertmanager to handle alerts
- various support tools

Architecture



Metric types

Counter

Gauge

Histogram

Summary

Jobs and Instances

For each instance scrape, Prometheus stores a sample in the following time series:

- **`up{job="", instance=""}`**

  1 if the instance is healthy, i.e. reachable, or 0 if the scrape failed.
- **`scrape_duration_seconds{job="", instance=""}`**

  duration of the scrape.
- **`scrape_samples_post_metric_relabeling{job="", instance=""}`**

  the number of samples remaining after metric relabeling was applied.
- **`scrape_samples_scraped{job="", instance=""}`**

  the number of samples the target exposed.
- **`scrape_series_added{job="", instance=""}`**

  the approximate number of new series in this scrape.

Configuration

https://prometheus.io/docs/prometheus/latest/configuration/configuration/

Recording rules

```
groups:
  - name: example
    rules:
      - record: job:http_inprogress_requests:sum
        expr: sum by (job) (http_inprogress_requests)
```

Alerting rules

```yaml
groups:
  - name: example
    rules:
      - alert: HighRequestLatency
        expr: job:request_latency_seconds:mean5m{job="myjob"} > 0.5
        for: 10m
        labels:
          severity: page
        annotations:
          summary: High request latency
```

Querying Prometheus

https://prometheus.io/docs/prometheus/latest/querying/basics/

Federation

```yaml
scrape_configs:
  - job_name: 'federate'
    scrape_interval: 15s

    honor_labels: true
    metrics_path: '/federate'

    params:
      'match[]':
        - '{job="prometheus"}'
        - '{__name__=~"job:.*"}'

    static_configs:
      - targets:
          - 'source-prometheus-1:9090'
          - 'source-prometheus-2:9090'
          - 'source-prometheus-3:9090'
```

Alerting

Configuration

Alertmanager playground

Open Telemetry

OpenMetrics, a cloud-native, highly scalable metrics protocol

https://opentelemetry.io/docs/concepts/data-sources/

LAB

Lets make our own Prometheus monitoring!

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!

Константин Новаковский

Ведущий системный инженер
G-Core Labs
Telegram: @kostya_keeper