

Prometheus



The DevOps Monitoring Toolkit



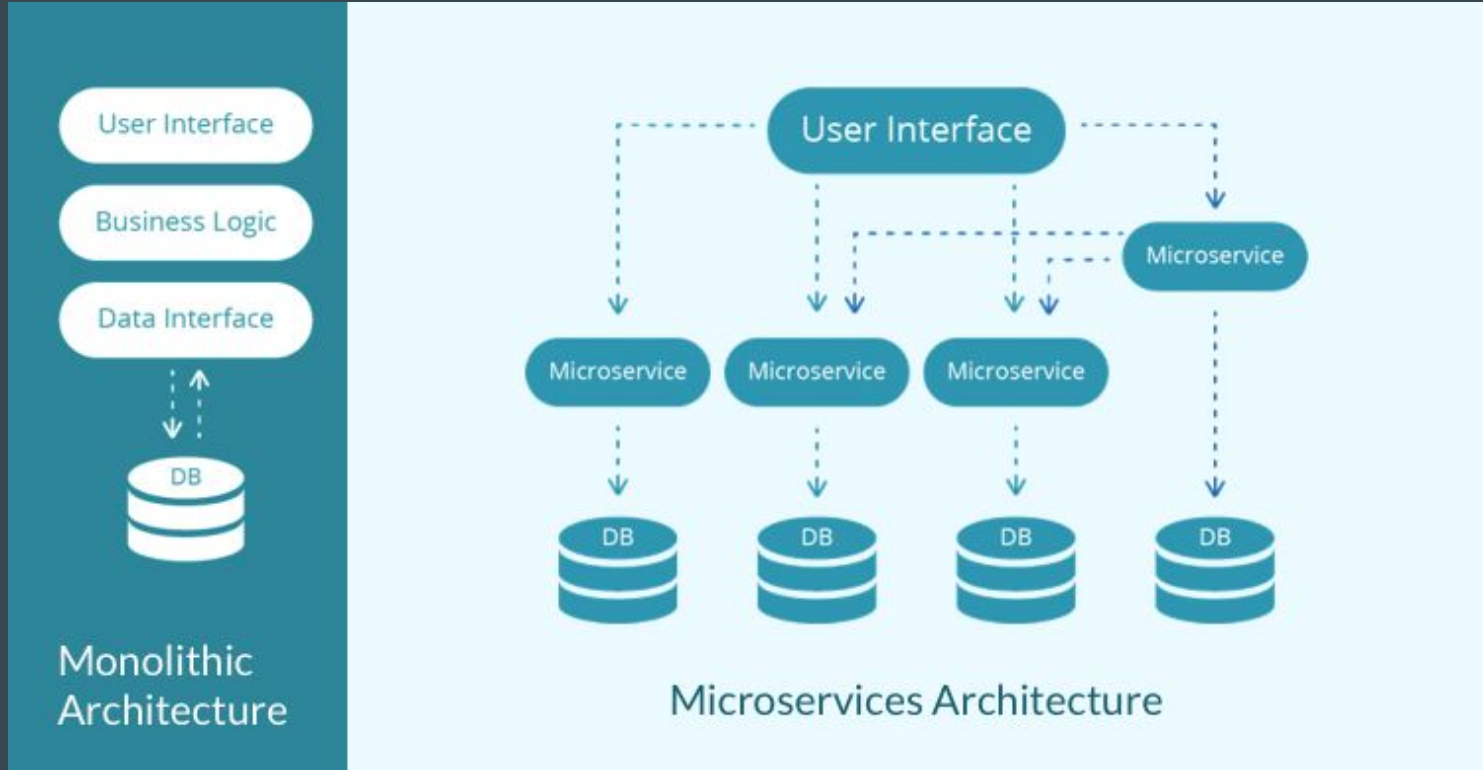
Axel Köhler - @axdotl
Rico Pahlisch - @rpahli

Agenda

- Architecture, Agility and Deployments
- Monitoring
- Prometheus
- Demo

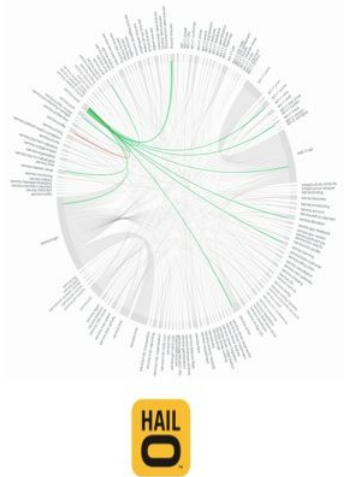
Architecture, Agility and Deployments

Mono and Micro



Death Star Architecture

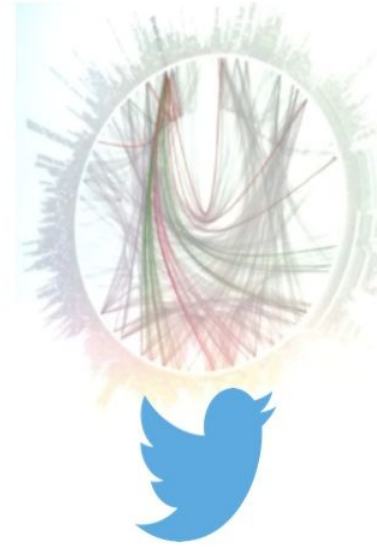
450 microservices



500+ microservices



500+ microservices



Source:

Netflix: <http://www.slideshare.net/BruceWong3/the-case-for-chaos>

Twitter: <https://twitter.com/adriano/status/441883572618948608>

Hail-o: <https://sudo.hailoapp.com/services/2015/03/09/journey-into-a-microservice-world-part-3/>

Agility und DevOps

- Agile Manifesto

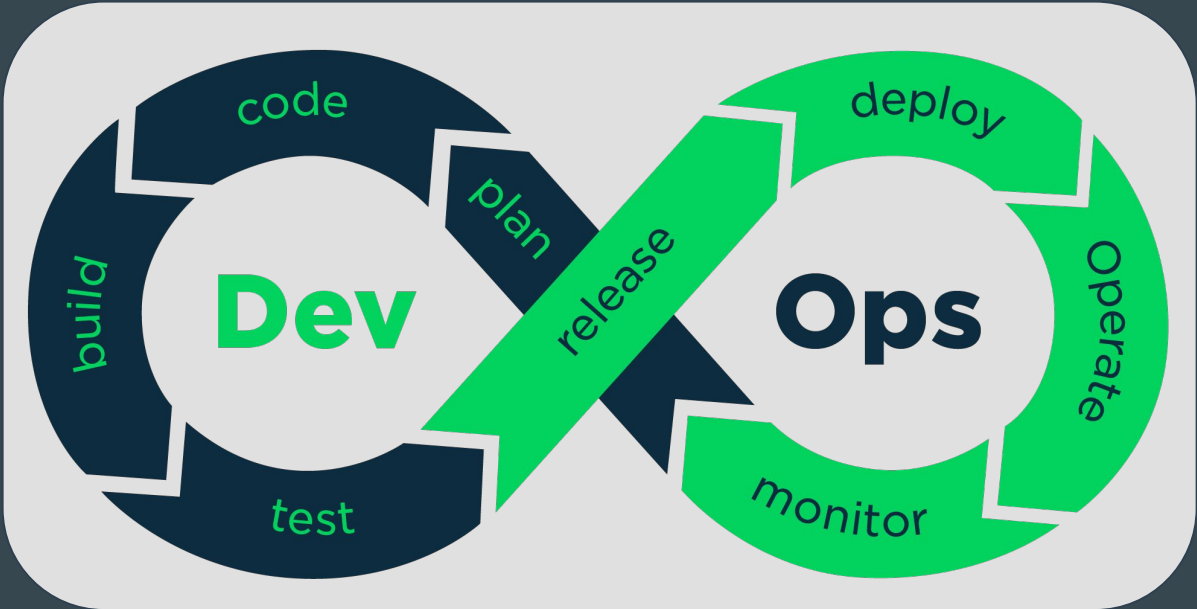
“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

“Deliver working software frequently [...] with a preference to the shorter timescale.”

- DevOps

“DevOps is a set of principles and practices — both technical and cultural — that can help organizations deploy better software, faster.”

DevOps Lifecycle



Deploy Phase

- High level of automation
- Reproducible
- Often
- “Self contained”
 - Contains everything needed to run and operate
 - Including monitoring configuration

Monitoring

Why?

To identify how system behaves in production, to:

- Identify bottlenecks and issues
- Derive actions (prevention)
- Retrieve results of experiments
- Identify potential for (cost) optimization

“Monitoring enhances communication and trust.” [1]

History

Monitoring Systems

- Nagios (1999)
- Zabbix (1998)
- Icinga (2009 Nagios fork)
- Shinken (2009)

Time-Series Databases

- Influx
- OpenTSDB
- Graphite

Why Prometheus?

- Build for scalable and dynamic infrastructures
- Several ways for Service Discovery
 - Consul
 - DNS
 - Kubernetes
- Built-in Alerting and Notification
- Mighty Query Language

Prometheus

“Operate and Monitor”

Facts

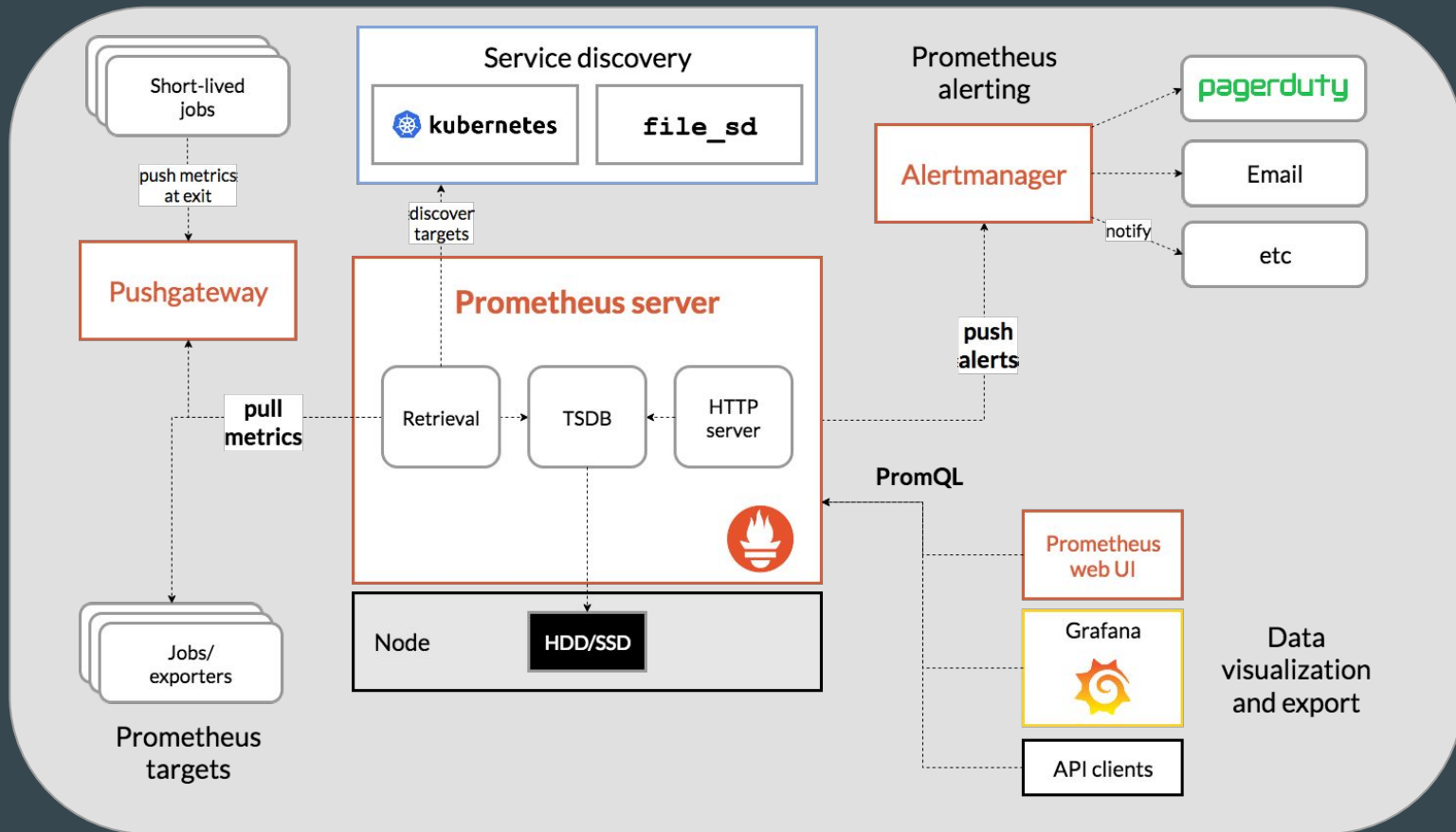
- Open-Source monitoring and alerting system
- Starts in 2012 by SoundCloud
 - v1.0 released in July 2016
- 26k stars on GitHub
- 390 contributors
- Latest Release 2.12.0 (28. August 2019)
- One of six CNCF Graduated projects



Data Model

- Everything is a time-series
- Every time-series has a unique name and a set of key-value pairs called *labels*
- Notation:
 - `<metric name>{<label name>=<label value>, ...}`
 - should have a single-word (application) prefix
 - `traefik_backend_requests_total{protocol="http"}`

Architecture



Jobs and Instances

- An endpoint you can scrape is called an *instance*
- A collection of instances (replicas) is called a *job*
- Prometheus attaches job and instance labels automatically to the scraped time series
 - `traefik_backend_requests_total{protocol="http", job="kubernetes-pods", instance="10.1.2.3:12345"}`

Exporters

Exports metrics from third-party systems as Prometheus metrics

- *kube-state-metrics* - listens to the Kubernetes API server and generates metrics about the state of the objects
- *node-exporter* - exports hardware and OS metrics exposed by *NIX kernels
- *stackdriver-exporter* - requests Stackdriver API for the metrics

Many more - <https://prometheus.io/docs/instrumenting/exporters/>

PromQL

- Prometheus provides a functional query language called PromQL

```
sum by (backend) (rate(
  traefik_backend_requests_total{
    protocol="http", code=~"2.."
  }[5m]
))
```

- <https://timber.io/blog/promql-for-humans/>

Content

Docker-compose setup

- Petstore
 - Micronaut based application - <https://micronaut.io>
 - Including Micrometer - <https://micrometer.io>
- Prometheus
- Grafana
- Traefik
- See <https://github.com/kiwigrd/prometheus-demo>

Conclusion and Outlook

Conclusion and Outlook

- Business metrics vs. technical metrics
- Sidecars
- Thanos

References

- [1] <https://puppet.com/resources/ebook/cio-guide-to-devops>
- <https://techbeacon.com/app-dev-testing/agile-devops-continuous-delivery-evolution-software-delivery>