Cloud Native meetup

Monitoring

Dušan Katona & Jakub Coufal

Who we are?

Dušan Katona

- Release mgmt, QA, Agile
- DevOps enthusiast
- Used docker since 0.7.6
- K8S migration

Jakub Coufal

- Software Engineer
- DevOps enthusiast
- Distributed systems enthusiast

Wandera

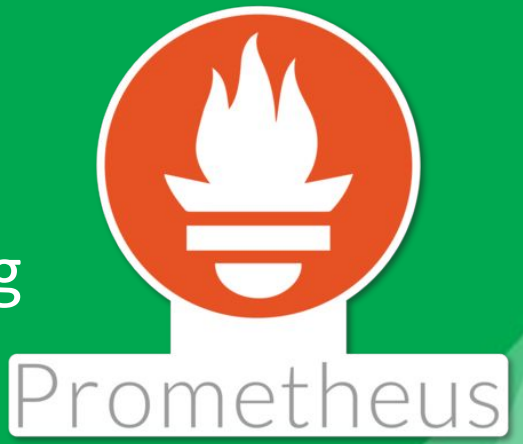
- Mobile security gateway solution
- Microservices backend (30 services, mostly in Java) - running in AWS
- More than 200 proxies around the world
- Apart from proxies, 90 nodes running our core and DBs

Monitoring in Wandera

- traditionally Graphite - push model
- services push metrics via Carbon relay
- quite a high cost of running the Graphite cluster ($8000/month)
- reliability problems
- dashboards in Grafana not so flexible due to single dimensional metrics

New Monitoring in Wandera

- With migration to K8S we looked at different monitoring
- That supports service discovery

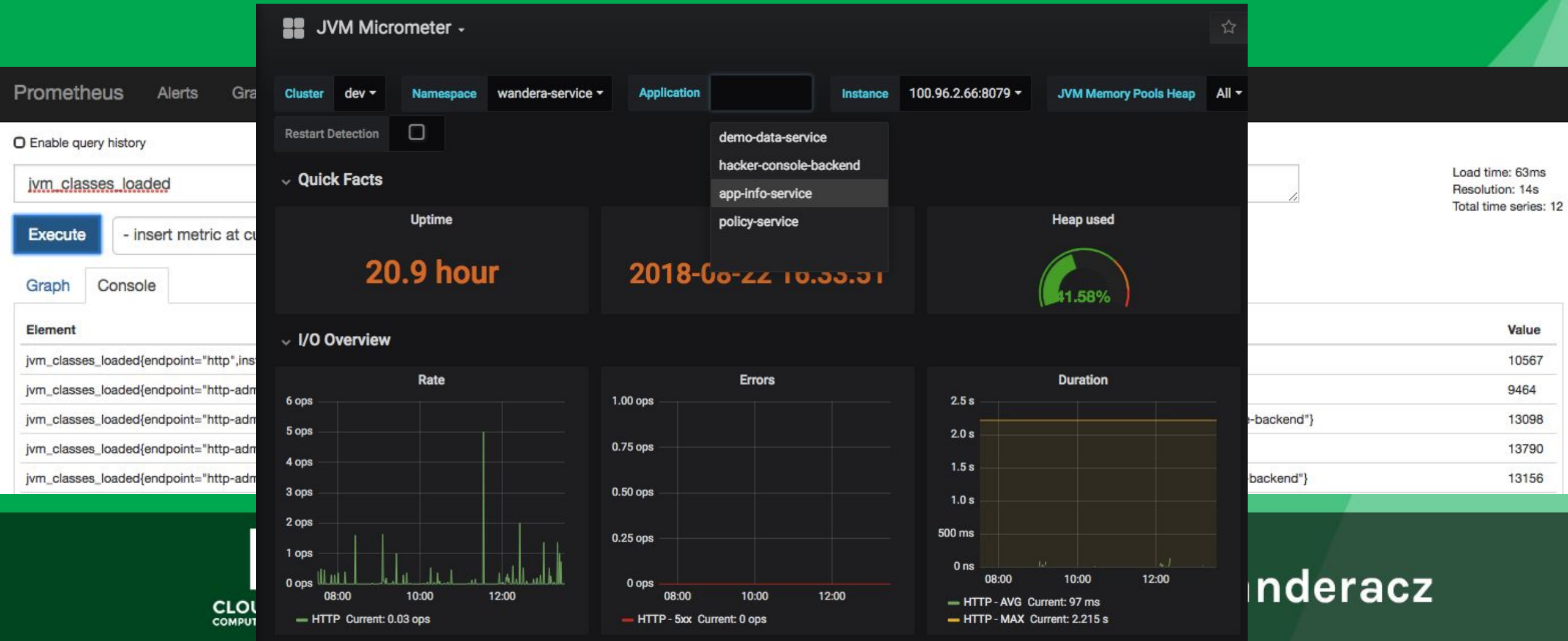


Prometheus

- de-facto standard for monitoring in K8S
- pull based - Prometheus scrapes endpoints
- paired with Alertmanager for delivering/grouping alerts
- more efficient storage
- multidimensional metrics identified by metric name + key=value

Why should you consider Prometheus?

- powerful querying and more flexible Grafana dashboards



Why should you consider Prometheus?

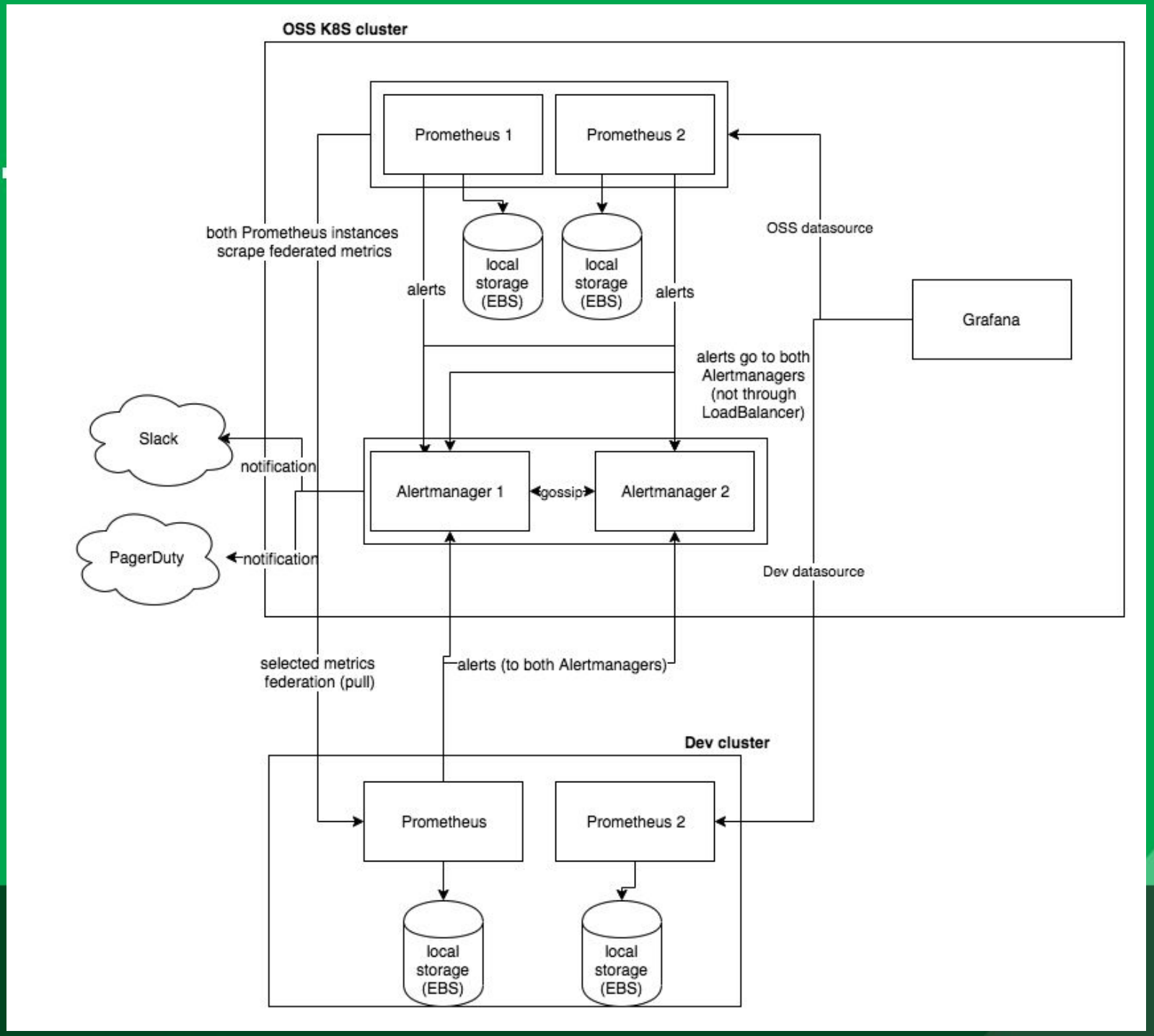
- developers can easily test metrics due to pull model
- documenting metrics is easy

```
# HELP prometheus_engine_queries The current number of queries being executed or waiting.
# TYPE prometheus_engine_queries gauge
prometheus_engine_queries 0
```

- works with service discovery in K8S - every new pod is scraped automatically
- you can monitor static infrastructure (DBs, our old infra) via static scrape config too or push metrics via push gateway

Monitoring Architecture

- **Prometheus in each environment** (dev, qa, stg, prod) which scrapes metrics from scrape targets inside the env
- One **global Prometheus** in OSS cluster with federation setup to other clusters
- **AlertManager** which deduplicates alerts, group them and route them to external notifying service
- **Grafana** for visualizing metrics


OSS K8S cluster
Prometheus 1
Prometheus 2
both Prometheus instances scrape federated metrics
local storage (EBS)
local storage (EBS)
alerts
alerts
OSS datasource
Grafana
alerts go to both Alertmanagers (not through LoadBalancer)
Slack
notification
Alertmanager 1
gossip
Alertmanager 2
PagerDuty
notification
Dev datasource
selected metrics federation (pull)
alerts (to both Alertmanagers)
Dev cluster
Prometheus
Prometheus 2
local storage (EBS)
local storage (EBS)

wanderacz

Caveats

- Alertmanager runs in 2 replicas and you have to send metrics to both AMs (not through the Load balancer)
- PD/Slack URI in plaintext in AM config
- Prometheus can be quite memory intensive (3 GBs per replica in our current setup)
- Setup federation and alerts so you know Prometheus is down or not sending alerts

Implementation in K8s

- prometheus-operator
  - K8s operator
  - CRDs - Prometheus, AlertManager, ServiceMonitor, PrometheusRule
  - Generates configuration
  - Custom configuration

```
apiVersion: monitoring.coreos.com/v1
kind: Prometheus
metadata:
  labels:
    app: "{{ template "meta.name" . }}"
    chart: {{ template "meta.chart" . }}
    release: {{ .Release.Name }}
    heritage: {{ .Release.Service }}
    component: prometheus
  name: prometheus
  namespace: monitoring
spec:
  baseImage: "{{ .Values.prometheus.image.repository }}"
  version: "{{ .Values.prometheus.image.tag }}"
```

```
"apiVersion": "apps/v1beta2",
"metadata": {"name": "prometheus-prometheus"...},
"spec": {
  "replicas": 2,
  "selector": {
    "matchLabels": {
      "app": "prometheus",
      "prometheus": "prometheus"
    }
  },
  "template": {
    "metadata": {
      "creationTimestamp": null,
      "labels": {
```

```
global:
  scrape_interval: 30s
  scrape_timeout: 10s
  evaluation_interval: 30s
  external_labels:
    cluster: krg01.dev.wandera.co.uk
    env: dev
    prometheus: monitoring/prometheus
    prometheus_replica: prometheus-prometheus-0
alerting:
  alert_relabel_configs:
  - separator: ;
    regex: prometheus_replica
    replacement: $1
    action: labeldrop
  alertmanagers:
  - static_configs:
    - targets:
      - alerts.krg01.oss.wandera.net
    scheme: https
    path_prefix: /0
    timeout: 10s
  - static_configs:
    - targets:
      - alerts.krg01.oss.wandera.net
    scheme: https
    path_prefix: /1
    timeout: 10s
rule_files:
- /etc/prometheus/config_out/rules/rules-0/*
scrape_configs:
```

Prometheus Alerts Graph Status Help

Enable query history

Expression (press Shift+Enter for newlines)

Execute - insert metric at cursor -

Graph Console

Element	Value
no data	

Remove Graph

Add Graph

```
{{ toYaml .Values.prometheus.resources | indent 4 }}
  retention: "{{ .Values.prometheus.retention }}"
{{- if .Values.prometheus.routePrefix }}
  routePrefix: "{{ .Values.prometheus.routePrefix }}"
{{- end }}
  serviceAccountName: prometheus
{{- if .Values.prometheus.storageSpec }}
  storage:
{{ toYaml .Values.prometheus.storageSpec | indent 4 }}
{{- end }}
  additionalScrapeConfigs:
```

```
    "securityContext": {
      "runAsUser": 1000,
      "runAsNonRoot": true,
      "fsGroup": 2000
    },
    "schedulerName": "default-scheduler"
  }
},
"volumeClaimTemplates": [
  {
    "metadata": {
```

```
    regex: (.*)
    target_label: endpoint
    replacement: http-admin
    action: replace
- job_name: monitoring/cert-manager/0
  scrape_interval: 30s
  scrape_timeout: 10s
  metrics_path: metrics
  scheme: http
  kubernetes_sd_configs:
  - api_server: null
    role: endpoints
    namespaces:
      names:
      - kube-system
  relabel_configs:
  - source_labels: [__meta_kubernetes_service_label_app]
    separator: ;
    regex: cert-manager
    replacement: $1
    action: keep
  - source_labels: [__meta_kubernetes_service_label_chart]
    separator: ;
    regex: cert-manager-0.1.0
    replacement: $1
```

Alertmanager Alerts Silences Status

New Silence

Status

Uptime: 2018-08-03T14:07:03.739072767Z

Mesh Status

Name: 0a:58:64:60:07:03

Nick Name: alertmanager-wandera-0

Peers:

- **Name:** 0a:58:64:60:07:03
  Nick Name: alertmanager-wandera-0
  UID: 8348617845605839000
- **Name:** 0a:58:64:60:02:03
  Nick Name: alertmanager-wandera-1
  UID: 7009501131611992000

Connections:

- **Address:** 100.96.2.3:6783
  Info: none 0a:58:64:60:02:03(alertmanager-wandera-1)
  State: established

Version Information

Branch: HEAD

BuildDate: 20180213-08:16:42

Implementation in K8

```yaml
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app: {{ template "meta.name" . }}
    chart: {{ template "meta.chart" . }}
    release: {{ .Release.Name }}
    heritage: {{ .Release.Service }}
```

m

s plain config

config-management (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Error
http://100.96.11.67:8080/actuator/prometheus	UP	endpoint="http" instance="100.98.11.67:8080" namespace="config" pod="config-manager-controller-6d55565b4c-9gddp" service="config-manager-controller"	9.666s ago	

```yaml
  namespaceSelector:
    matchNames:
      - config
  endpoints:
  - port: http
    interval: 30s
    path: /actuator/prometheus
```

```yaml
- job_name: monitoring/config-manager/0
  scrape_interval: 30s
  scrape_timeout: 10s
  metrics_path: /actuator/prometheus
  scheme: http
  kubernetes_sd_configs:
  - api_server: null
    role: endpoints
    namespaces:
      names:
      - config
  relabel_configs:
  - source_labels: [__meta_kubernetes_service_label_app]
    separator: ;
    regex: config-management
    replacement: $1
    action: keep
    separator: ;
    regex: http
    replacement: $1
    action: keep
  - source_labels: [__meta_kubernetes_namespace]
    separator: ;
    regex: (.*)
    target_label: namespace
    replacement: $1
    action: replace
  - source_labels: [__meta_kubernetes_pod_name]
    separator: ;
    regex: (.*)
    target_label: pod
    replacement: $1
    action: replace
  - source_labels: [__meta_kubernetes_service_name]
    separator: ;
    regex: (.*)
    target_label: service
```

CLOUD NATIVE COMPUTING FOUNDATION

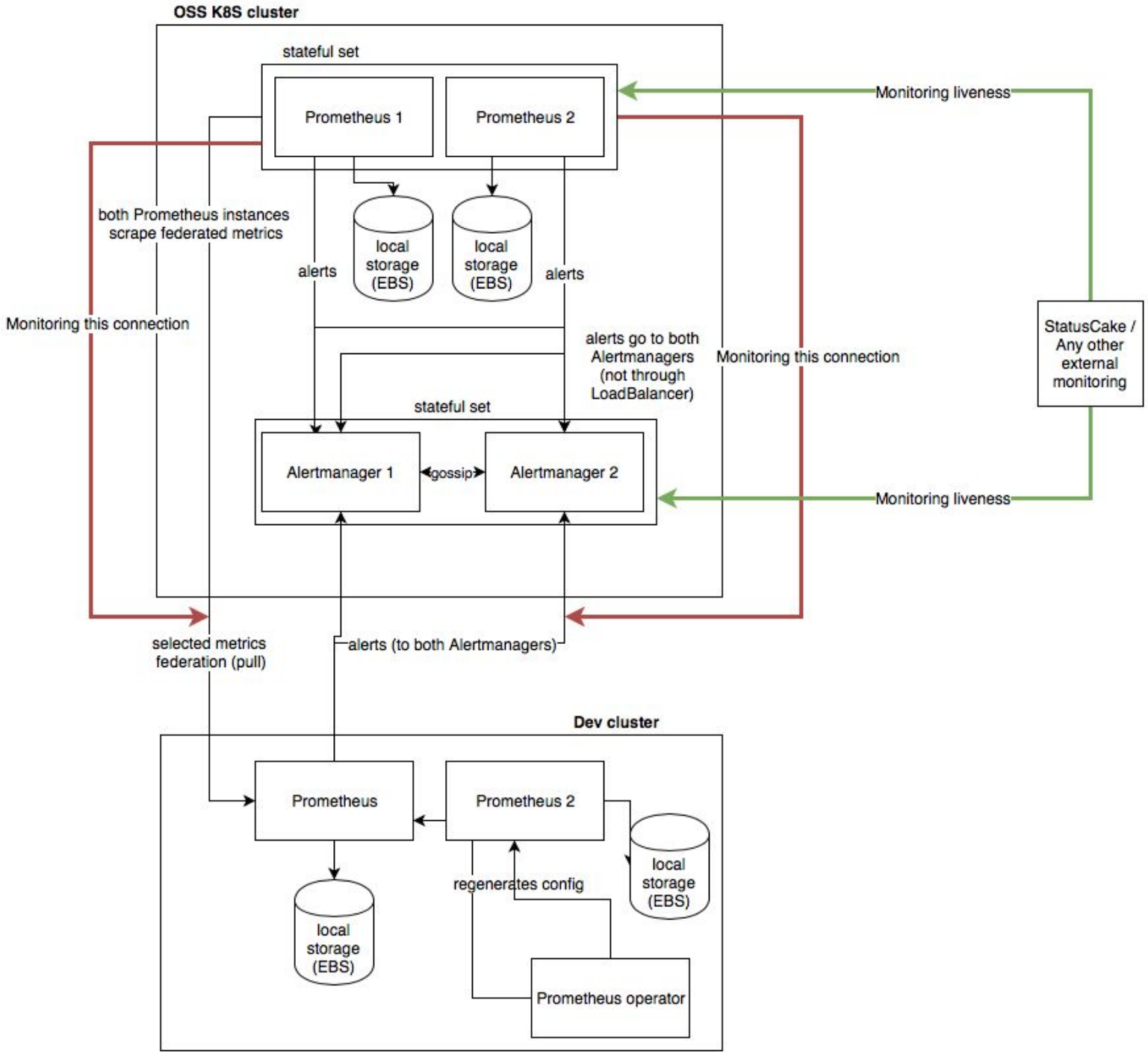
wander

wandera.c

deracz

Imple

- Sessio
- Ingres
  - Per
- Pod af
  - Ant
- Monit
  - noc
  - kub
  - etc
  - me



Key takeaways

- Prometheus is a modern de-facto standard for monitoring
- Great community and a lot of open source projects (that you sometimes need to contribute to)
- Don’t be afraid to use it even for static infrastructure

Q&A

Questions?

CLOUD NATIVE
COMPUTING FOUNDATION

wandera
wandera.com

wanderacz