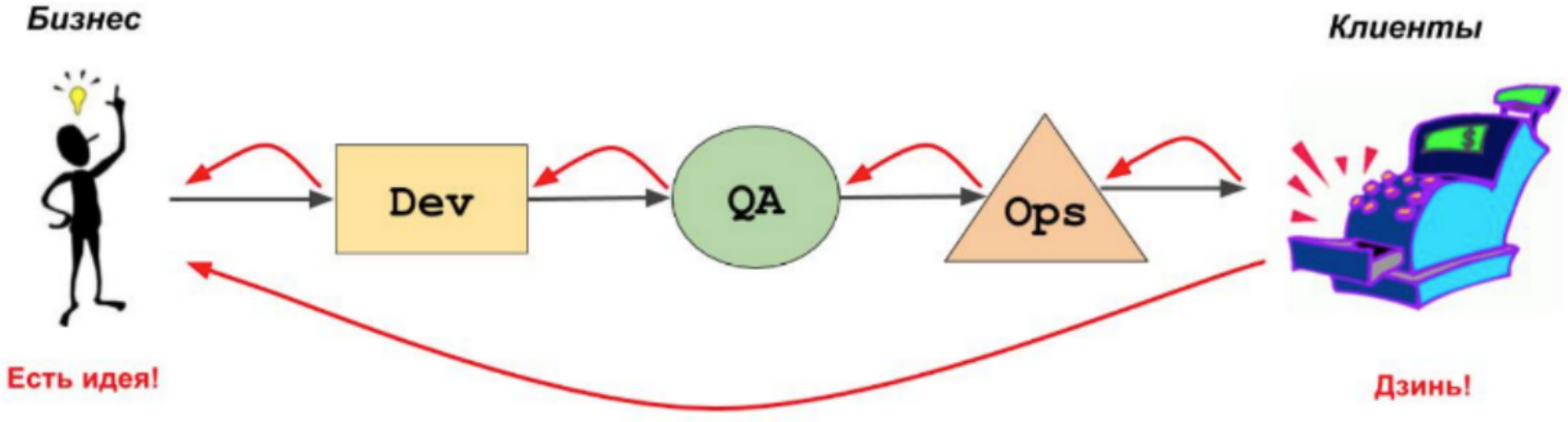
# Kubernetes. Мониторинг. Логирование

# Не забудь включить запись!

# План

- Мониторинг Kubernetes
- Логирование в Kubernetes
- Операторы для Kubernetes

# Второй путь DevOps



# Мониторинг

# Что отслеживать

- Работоспособность приложений
- Метрики приложений
- Метрики хостов
- Метрики pod’ов и контейнеров
- Метрики и работоспособность самого Kubernetes

# Что есть у Kubernetes

- probes
- cAdvisor
- metrics-server
- kube-state-metrics

# Probes

Периодические проверки pod'а на жизнеспособность.

Как узнать, что сервис "жив" и готов к работе?

- **ExecAction** - выполнить команду и ждать exit code 0
- **TCPSocketAction** - проверить, что TCP-порт открыт
- **HTTPGetAction** - отправить HTTP GET-запрос

# Liveness probes

- Приложение запущено и "живо"
- Если probe неуспешна - перезапуск контейнера*

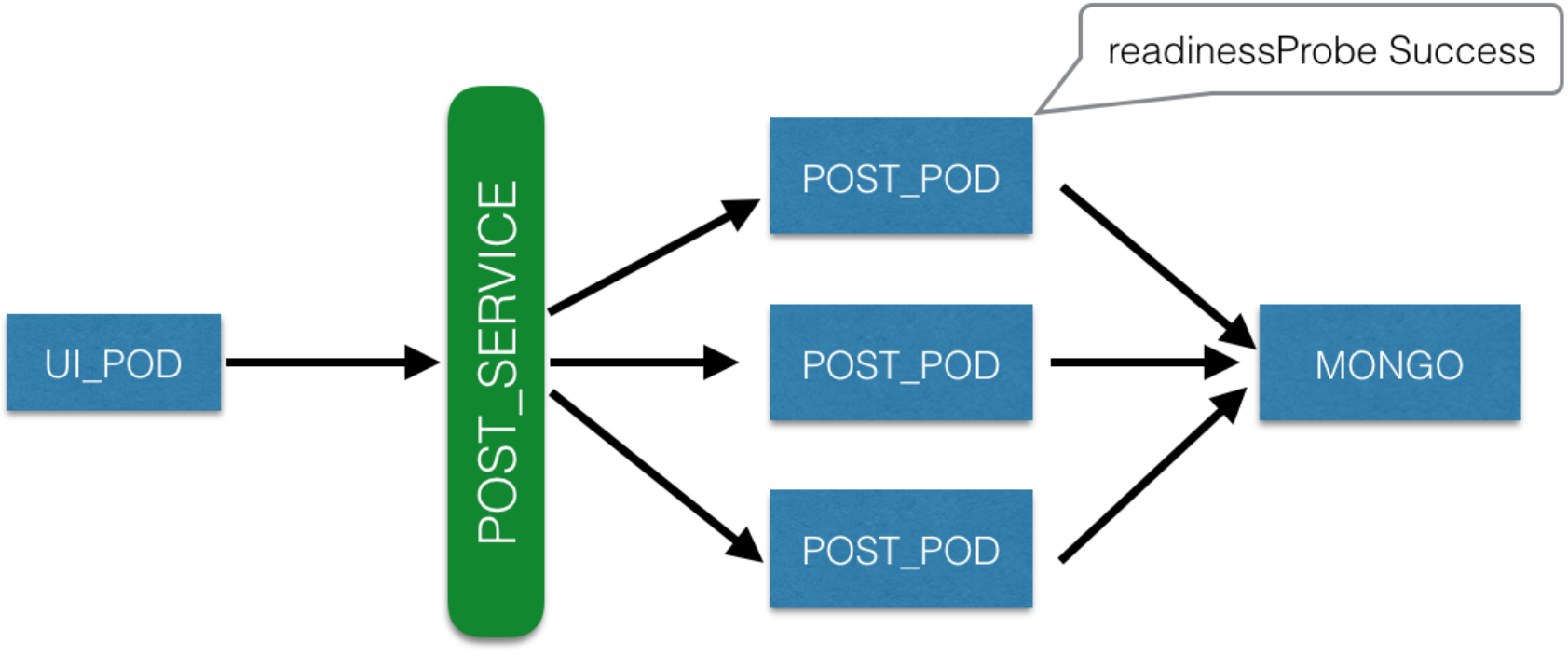
```
apiVersion: v1
kind: Pod
...
spec:
  containers:
  - name: liveness
    image: liveness
    livenessProbe:
      tcpSocket:
        port: 8080
      initialDelaySeconds: 5
      periodSeconds: 10
```

# Readiness probes

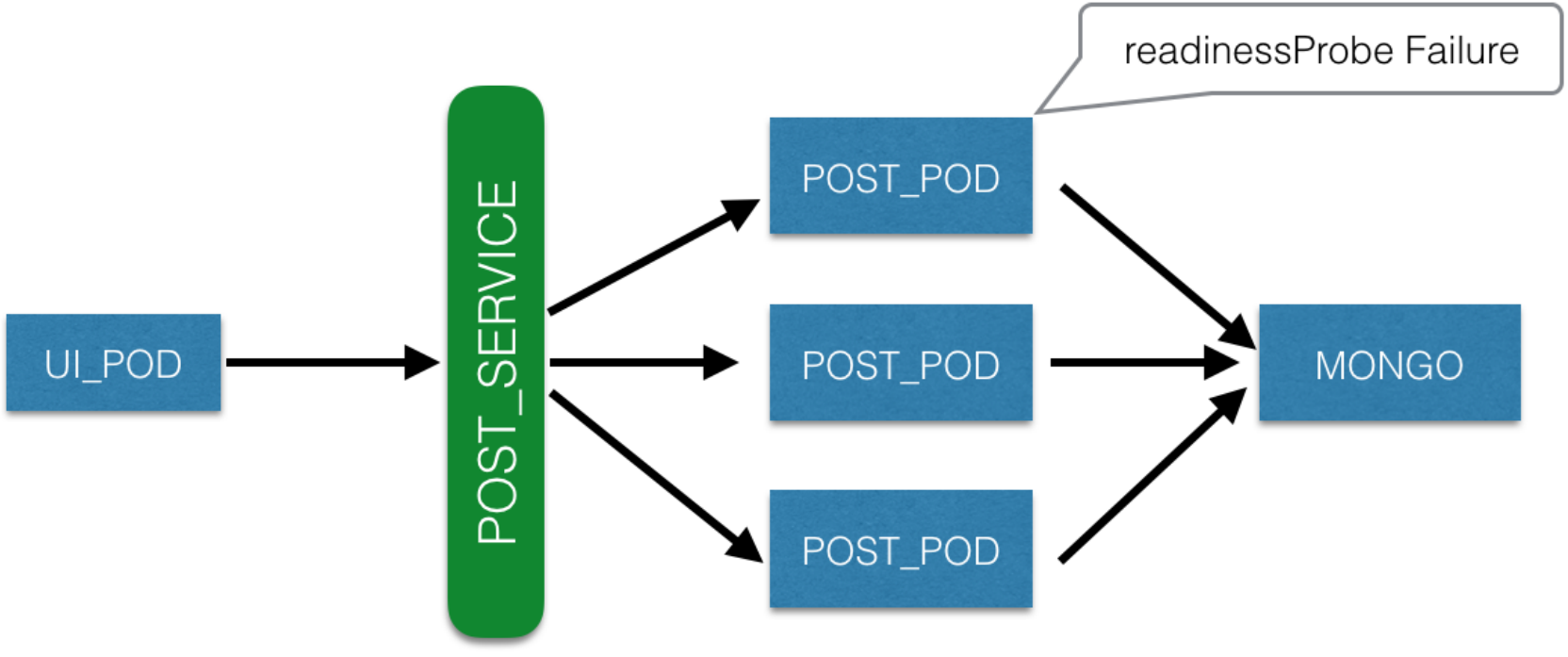
- Приложение готово обслуживать запросы
- Если probe неуспешна - удаление pod из балансировки

```
apiVersion: v1
kind: Pod
...
spec:
  containers:
  - name: readiness
    image: readiness
    readinessProbe:
      httpGet:
        path: /healthz
        port: 8080
      initialDelaySeconds: 3
      periodSeconds: 3
```

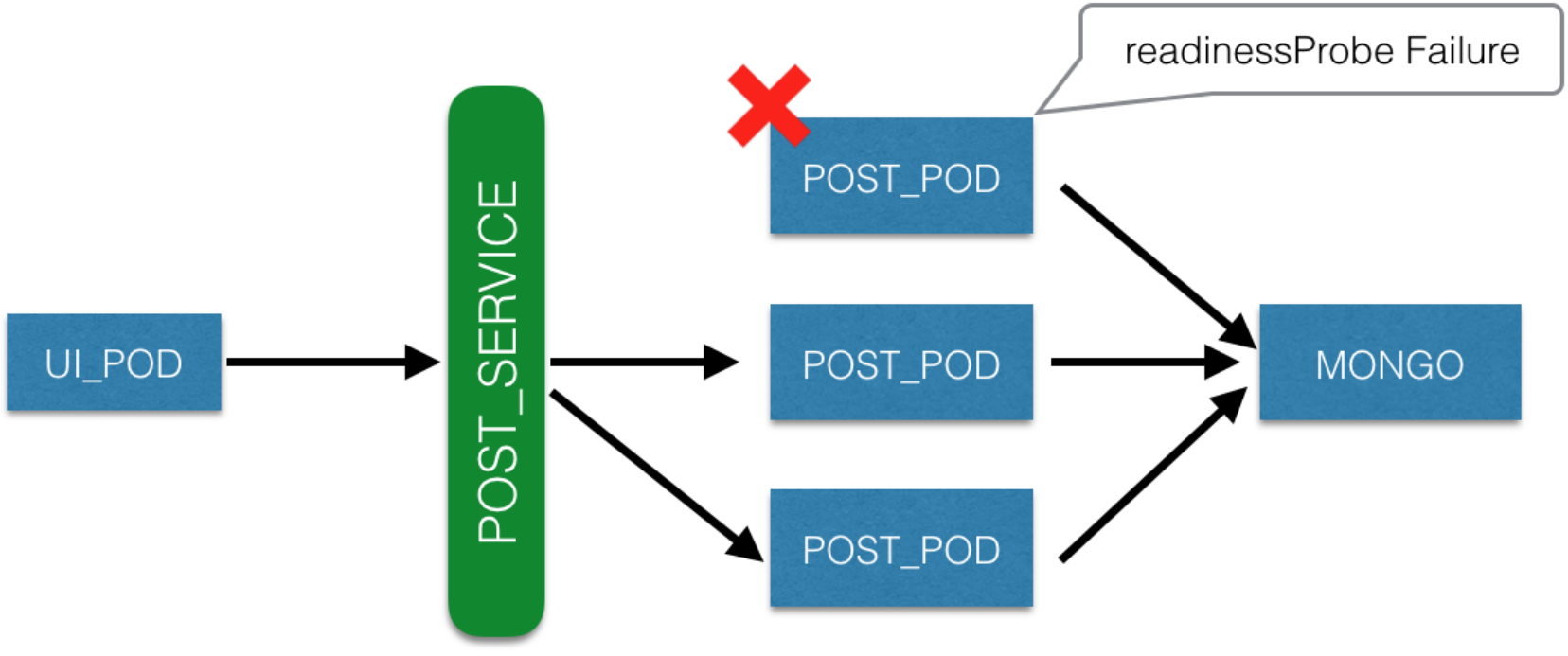
# Readiness probes: Good



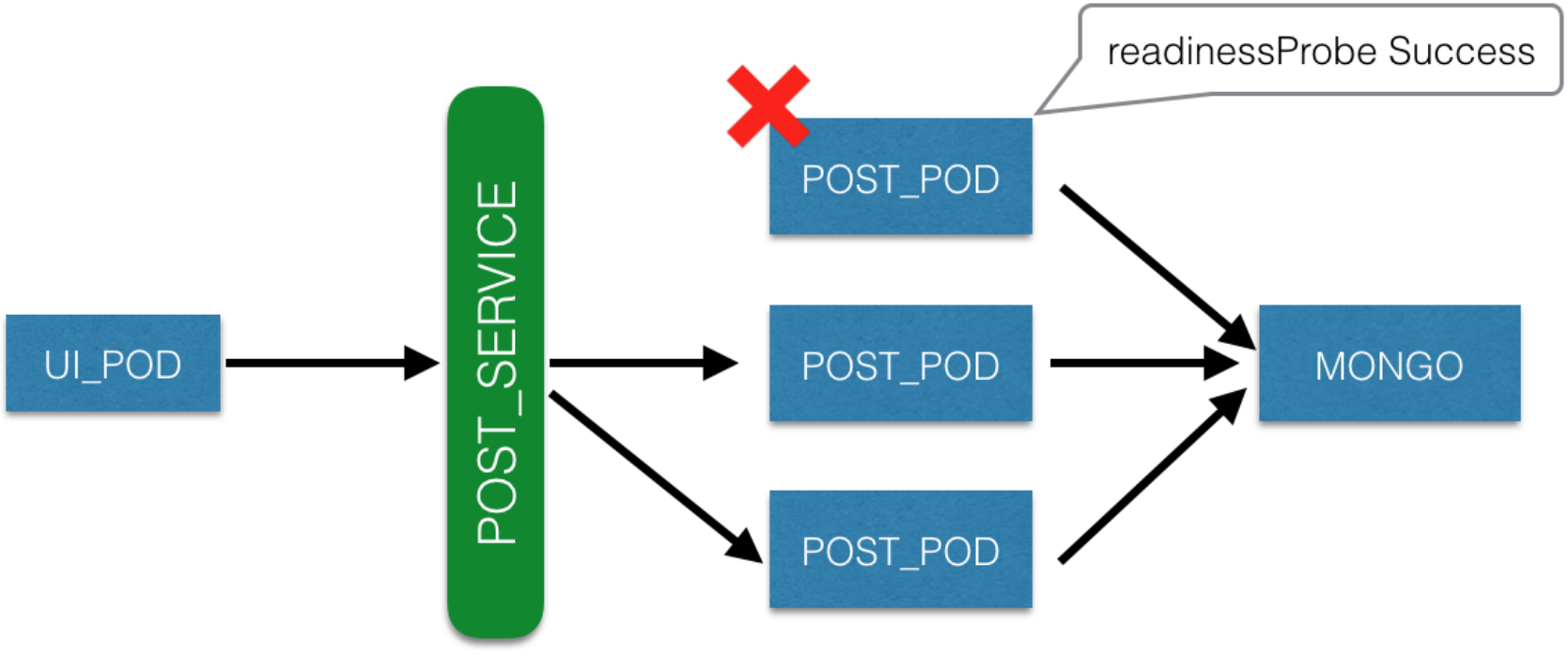
# Readiness probes: Failed



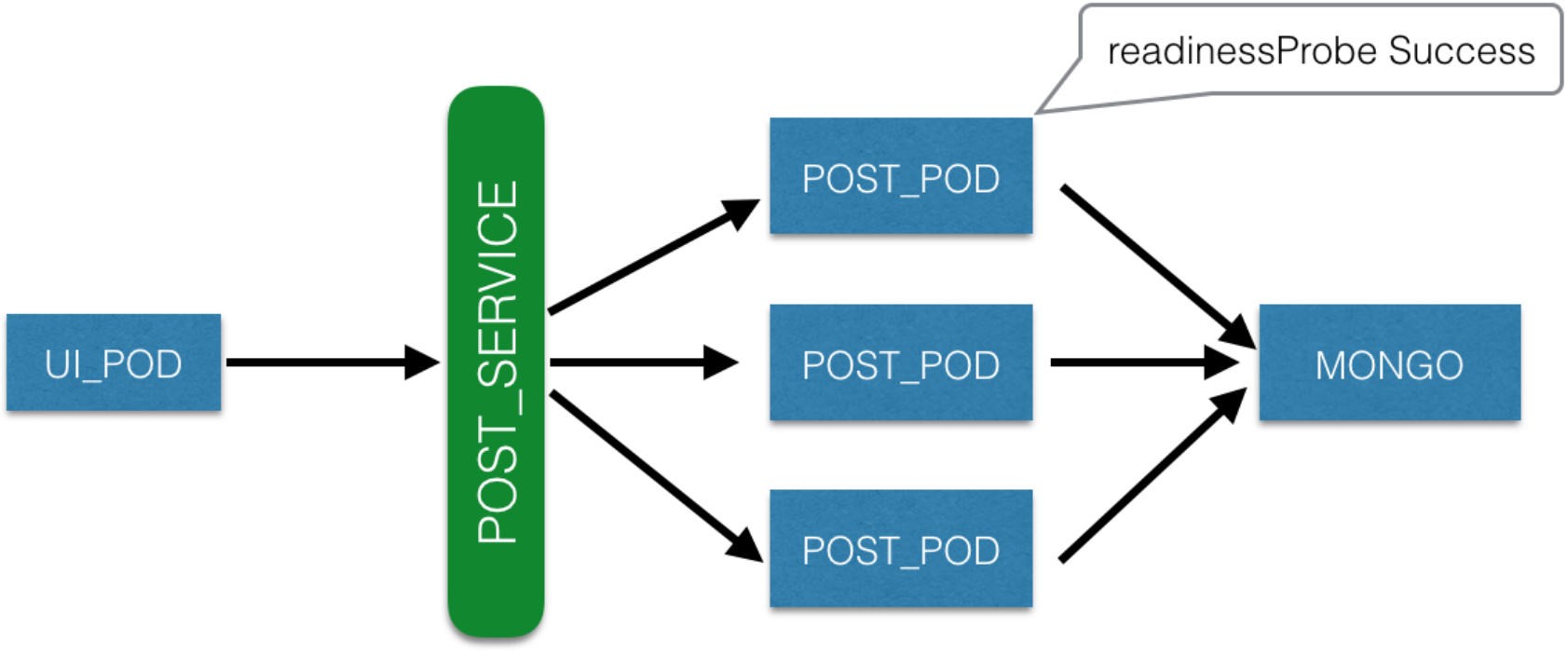
# Readiness probes: Disabled



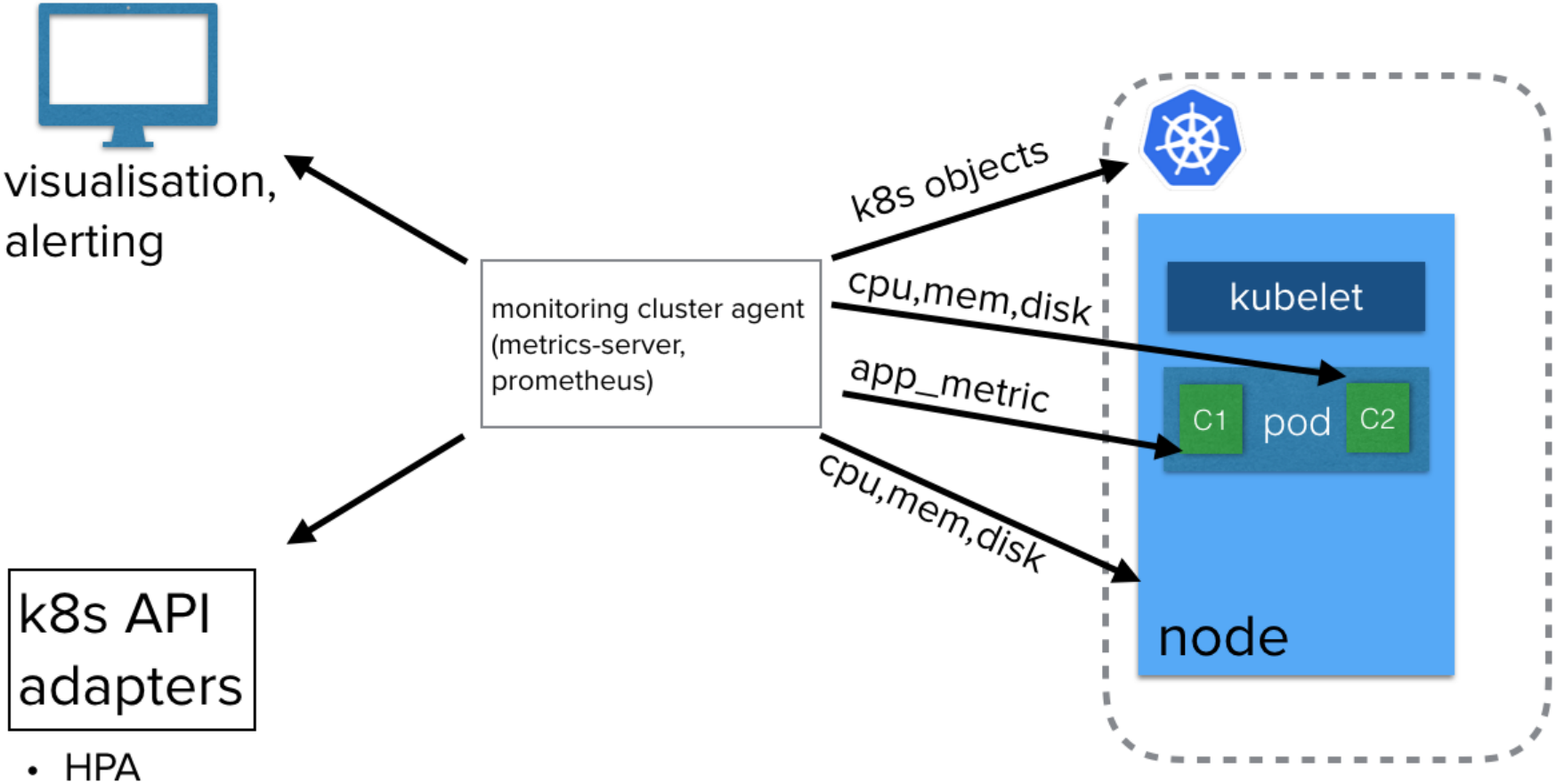
# Readiness probes: Recover



# Readiness probes: Good



# Monitoring pipeline




k8s API adapters

- HPA
- kubectl top

# cAdvisor

- Встроен в kubelet
- Получает информацию о контейнерах от Docker
- Работает как exporter для Prometheus

# cAdvisor

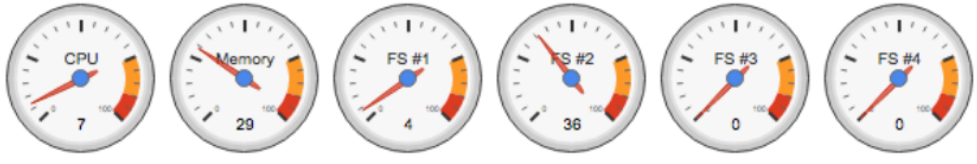
- CPU
- Memory
- Network
- Block I/O

Список предоставляемых метрик

# cAdvisor

## Deprecated feature for k8s

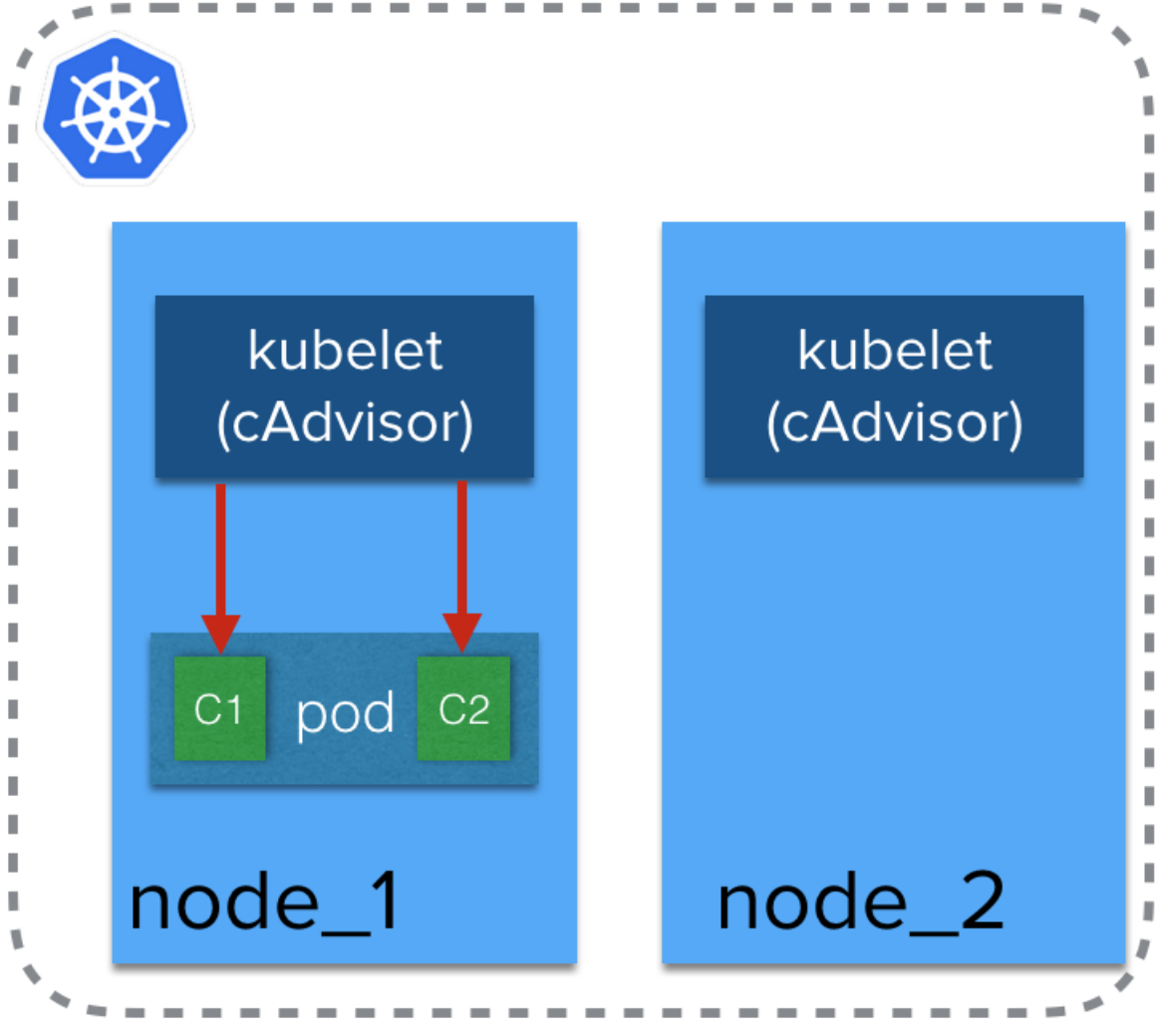
**Overview**




**Processes**

| User | PID | PPID | Start Time | CPU % | MEM % | RSS | Virtual Size | Status | Running Time | Command | Container |
|---|---|---|---|---|---|---|---|---|---|---|---|
| root | 1 212 | 1 | 17:28 | 2.90 | 1.40 | 107.89 MiB | 1.48 GiB | Ssl | 00:04:32 | kubelet | /system.slice/kubelet.ser |
| root | 6 431 | 6 413 | 18:53 | 2.80 | 2.80 | 213.98 MiB | 245.01 MiB | Ssl | 00:01:56 | prometheus | /kubepods/burstable/pod488 |
| root | 2 803 | 2 793 | 17:29 | 1.70 | 2.40 | 181.46 MiB | 556.54 MiB | Sl | 00:02:41 | fluentd | /kubepods/burstable/pod6a8 |
| root | 2 139 | 2 138 | 17:29 | 1.40 | 0.40 | 30.65 MiB | 43.39 MiB | Sl | 00:02:12 | calico-felix | /kubepods/burstable/pod6a8 |
| root | 1 160 | 1 | 17:28 | 1.00 | 0.70 | 59.06 MiB | 1.15 GiB | Ssl | 00:01:32 | dockerd | /system.slice/docker.ser |
| root | 340 | 1 | 17:28 | 0.10 | 0.90 | 74.50 MiB | 116.83 MiB | Ss | 00:00:18 | systemd-journal | /system.slice/systemd-jour |
| root | 1 238 | 1 | 17:28 | 0.10 | 0.30 | 23.44 MiB | 349.43 MiB | Ssl | 00:00:09 | node-problem-de | /system.slice/node-problem |
| root | 1 779 | 1 667 | 17:28 | 0.10 | 0.40 | 32.13 MiB | 303.63 MiB | Sl | 00:00:15 | kube-proxy | /kubepods/burstable/podaec |
| root | 1 | 0 | 17:28 | 0.00 | 0.00 | 7.39 MiB | 101.99 MiB | Ss | 00:00:02 | systemd | /init.s |

# cAdvisor



# metrics-server

## Inspired by Heapster (deprecated)

- Собирает метрики cAdvisor
- In-memory хранение последних значений метрик
- Отсутствует endpoint для Prometheus
- Встраивается в API для использования в других частях Kubernetes:
  - HPA
  - kubectl top
  - Dashboard (Heapster only)

# kubectl top

```
$ kubectl top nodes

NAME                          CPU(cores)   CPU%      MEMORY(bytes)   MEMORY%
gke-cluster-f9c66281-rgld     116m         12%       1030Mi          88%
gke-cluster-f9c66281-dbb2     72m          7%        845Mi           72%
```

```
$ kubectl top pods

NAME                                      CPU(cores)   MEMORY(bytes)
post-test-post-54b7cbcc69-pnbf7           4m           42Mi
post-test-post-54b7cbcc69-9md82           4m           41Mi
post-test-post-54b7cbcc69-mhj5r           4m           42Mi
```

# kube-state-metrics

Собирает из Kubernetes API информацию о логических объектах кластера:

- Deployment
- ReplicaSet
- StatefulSet
- Pod
- Job
- etc...

Работает как exporter для Prometheus

Список собираемых метрик

# Prometheus

- Развитие проекта началось в 2012
- Создан на основе Borgmon, бывшими работниками Google
- Open Source
- Написан на Go
- Whitebox, Pull система
- **Service Discovery**

# Prometheus

Источники метрик для Prometheus:

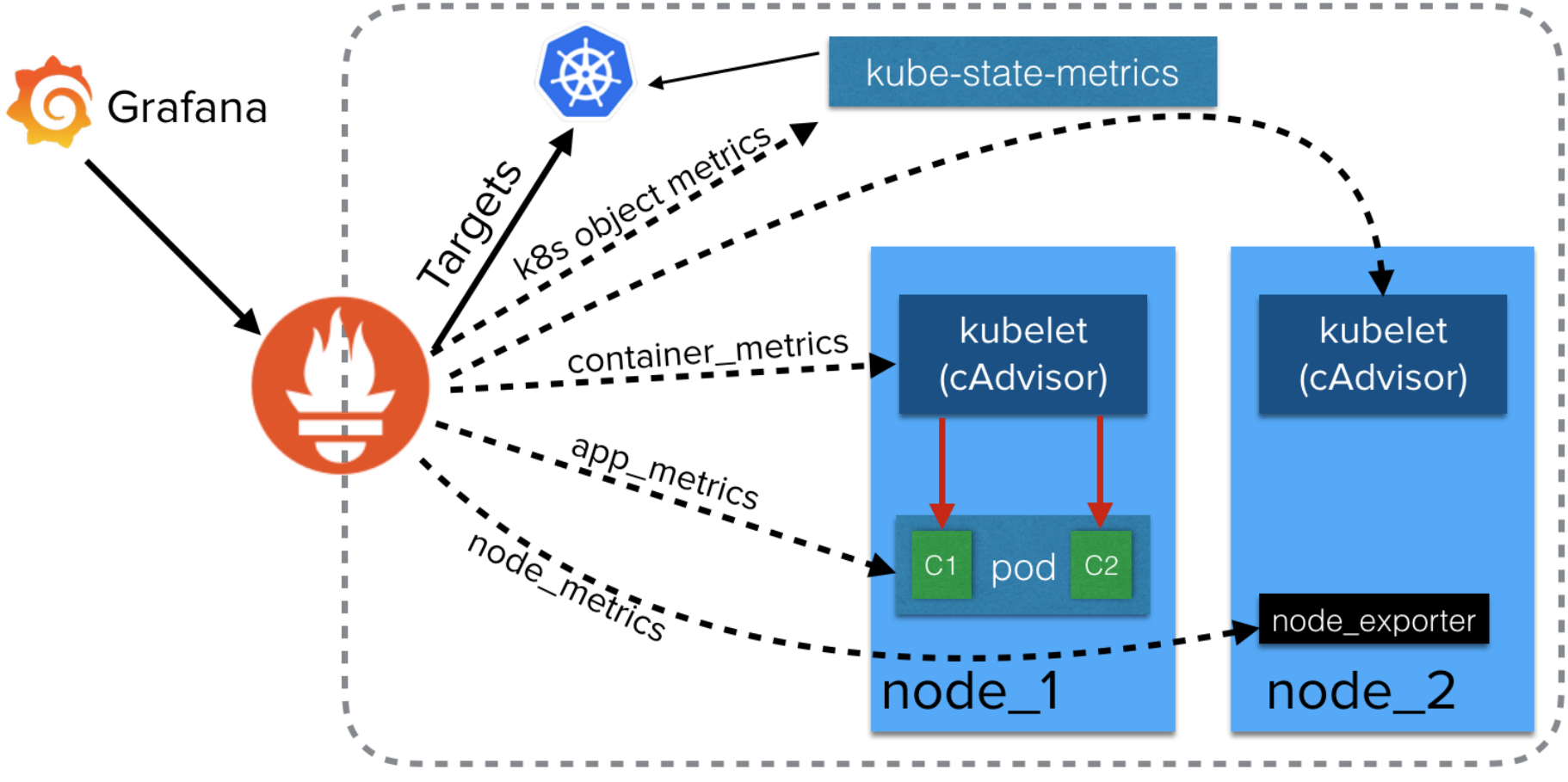
- Метрики kube-state-metrics
- Метрики Node Exporter
- Метрики приложений

# Prometheus

Источники метрик для Prometheus:

- Метрики etcd (**/metrics**)
- Метрики API servers (**/metrics**)
- Метрики cAdvisor (**/api/v1/nodes/node-X/proxy/metrics/cadvisor**)

# Prometheus


Grafana
kube-state-metrics
Targets
k8s object metrics
container_metrics
app_metrics
node_metrics
kubelet
(cAdvisor)
kubelet
(cAdvisor)
C1
pod
C2
node_exporter
node_1
node_2

# Prometheus

- Targets - источник для сбора метрик
- Jobs - группа источников с определенным набором правил

# Prometheus

```
scrape_configs:
  - job_name: 'post-endpoints'
    kubernetes_sd_configs:
      - role: endpoints
...

  - job_name: 'kubernetes-nodes'
    kubernetes_sd_configs:
      - role: nodes
...
```

# Prometheus

```
scrape_configs:
  - job_name: 'post-endpoints'
    kubernetes_sd_configs:
      - role: endpoints
    api_server: …      # адрес k8s API
    tls_config: …      # CA-сертификат k8s API
    bearer_token: …    # токен serviceAccount'а
    namespaces: …      # в каких namespace'ах искать
      names:
        - default
        - dev
```

# Prometheus

**Roles**:

- Node
- Endpoints
- Pod
- Service
- Ingress

# Prometheus

## Node

```
scrape_configs:
  - job_name: 'kubernetes-nodes'
    kubernetes_sd_configs:
      - role: node
```

**Targets**:

- node_1
- node_2

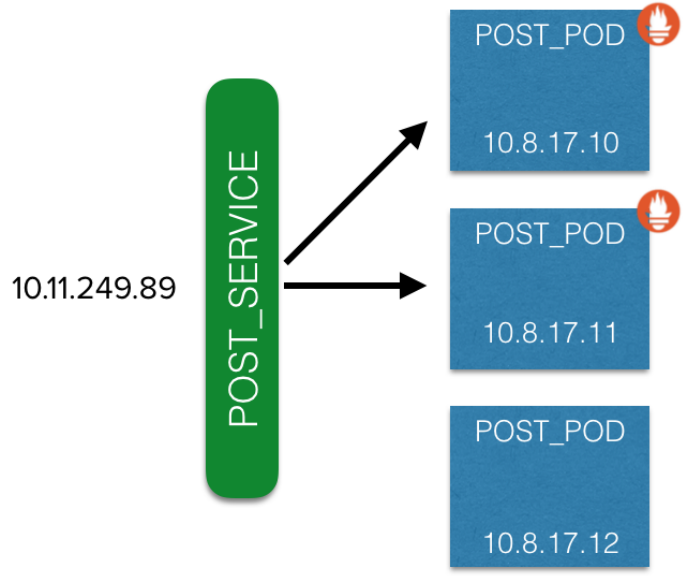
# Prometheus

## Endpoints

```
scrape_configs:
  - job_name: 'post-endpoints'
    kubernetes_sd_configs:
      - role: endpoints
```

**Targets**:

- 10.8.17.10
- 10.8.17.11



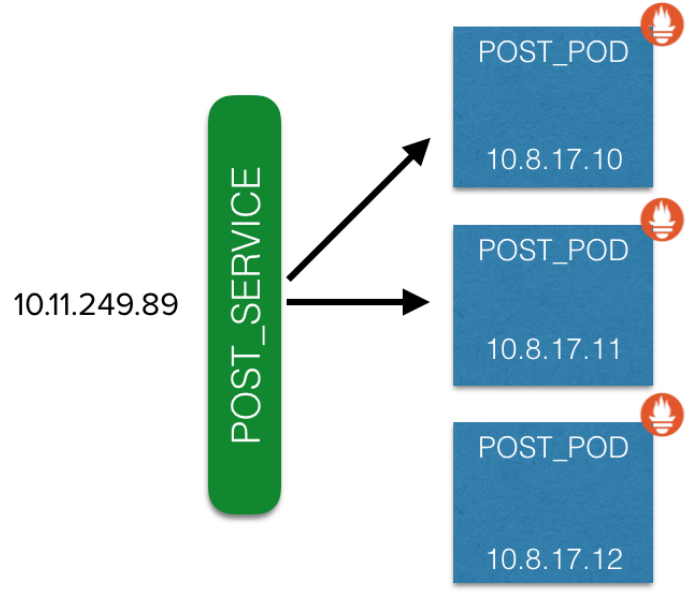
# Prometheus

## Pods

```
scrape_configs:
  - job_name: 'post-pods'
    kubernetes_sd_configs:
      - role: pod
```

**Targets**:

- 10.8.17.10
- 10.8.17.11
- 10.8.17.12



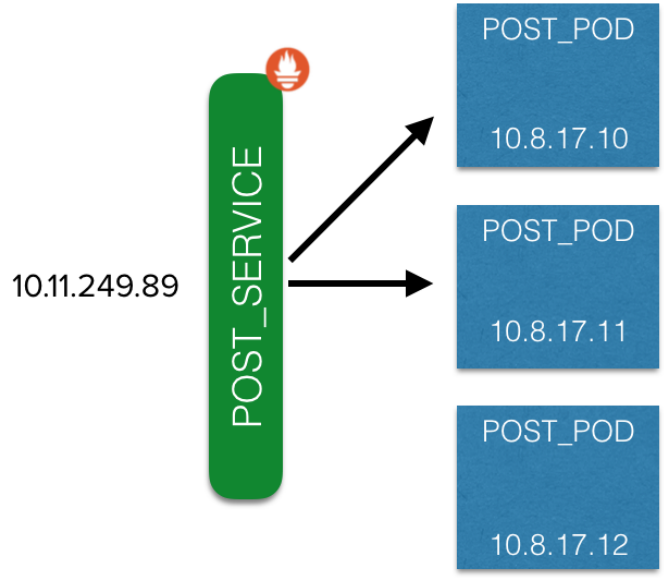
# Prometheus

## Service

```
scrape_configs:
  - job_name: 'post-service'
    kubernetes_sd_configs:
      - role: service
```

Targets:

- 10.11.249.89



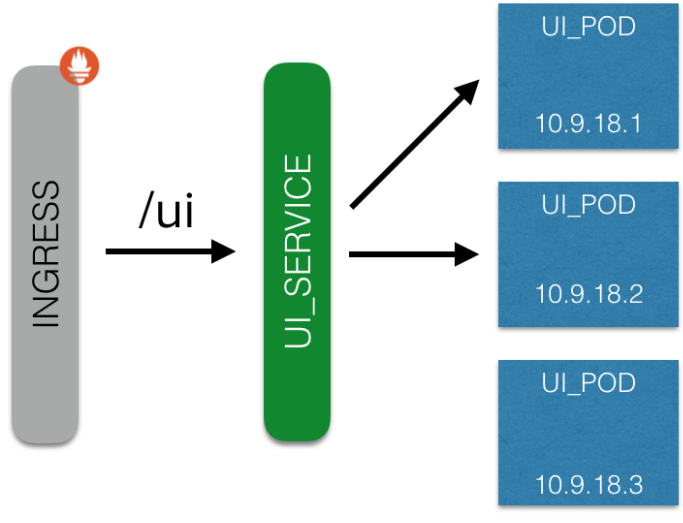
# Prometheus

## Ingress

```
scrape_configs:
  - job_name: 'ui-ingress'
    kubernetes_sd_configs:
      - role: ingress
```

**Targets**:

- ui_ingress/ui



## Meta labels (pod)

- __meta_kubernetes_pod_name
- __meta_kubernetes_pod_label_LABELNAME
- ...

# Prometheus

```
apiVersion: v1
kind: Service
metadata:
  name: post
  labels:
    app: reddit
    component: post
  annotations:
    prometheus.io/scrape: "false"
    prometheus.io/path: "/healthz"
spec:
```

# Relabeling

Фильтрация:

```
relabel_configs:
  - source_labels: [__meta_kubernetes_pod_label_component]
    action: keep
    regex: post
```

# Relabeling

Перенос меток:

```
relabel_configs:
  - source_labels: [__meta_kubernetes_pod_name]
    target_label: kubernetes_name
```

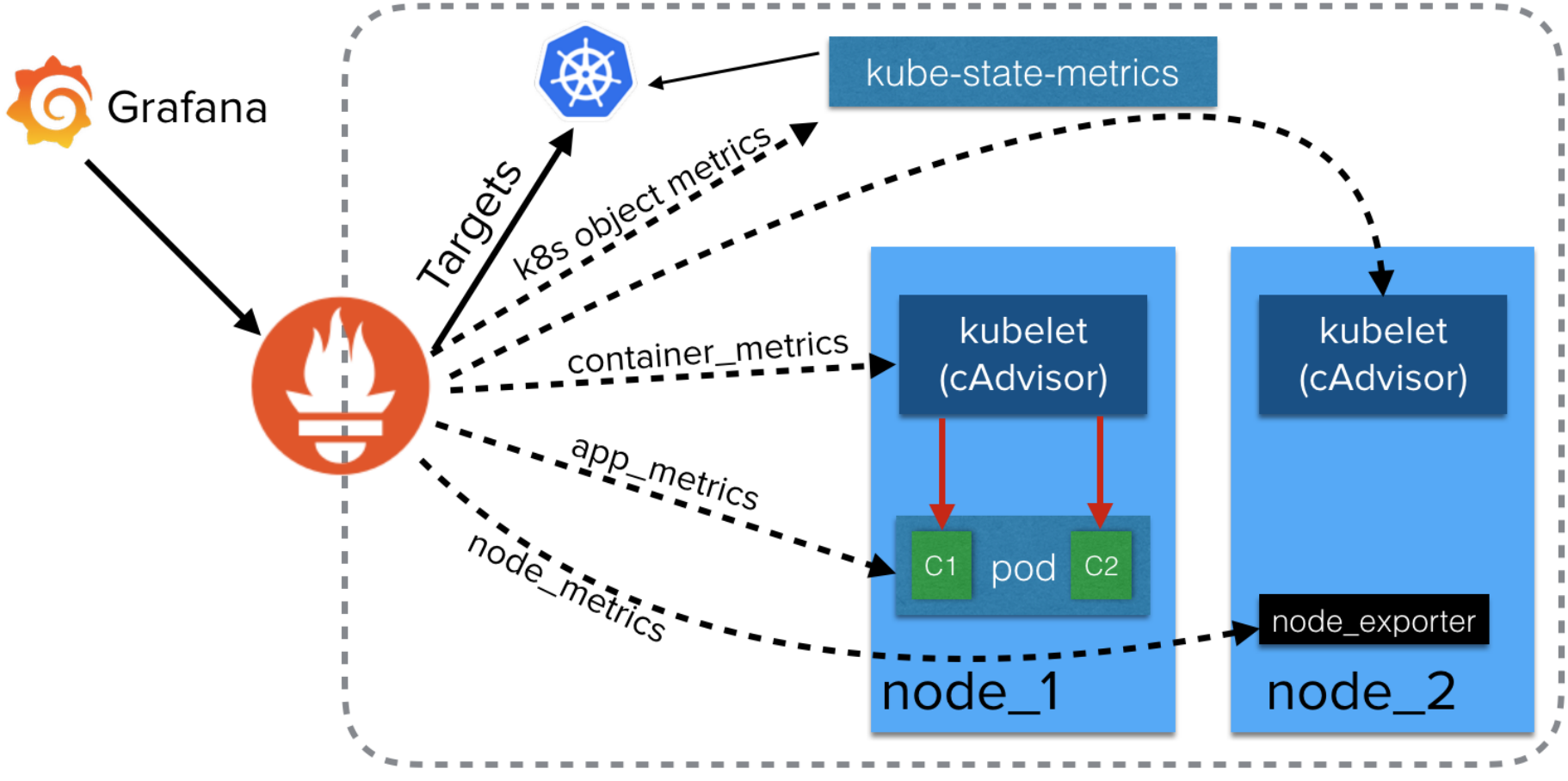
# Relabeling

```
relabel_configs:
  - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_scrape]
    action: keep
    regex: true
  - source_labels: [__meta_kubernetes_service_annotation_prometheus_io_path]
    action: replace
    target_label: __metrics_path__
    regex: (.+)
```

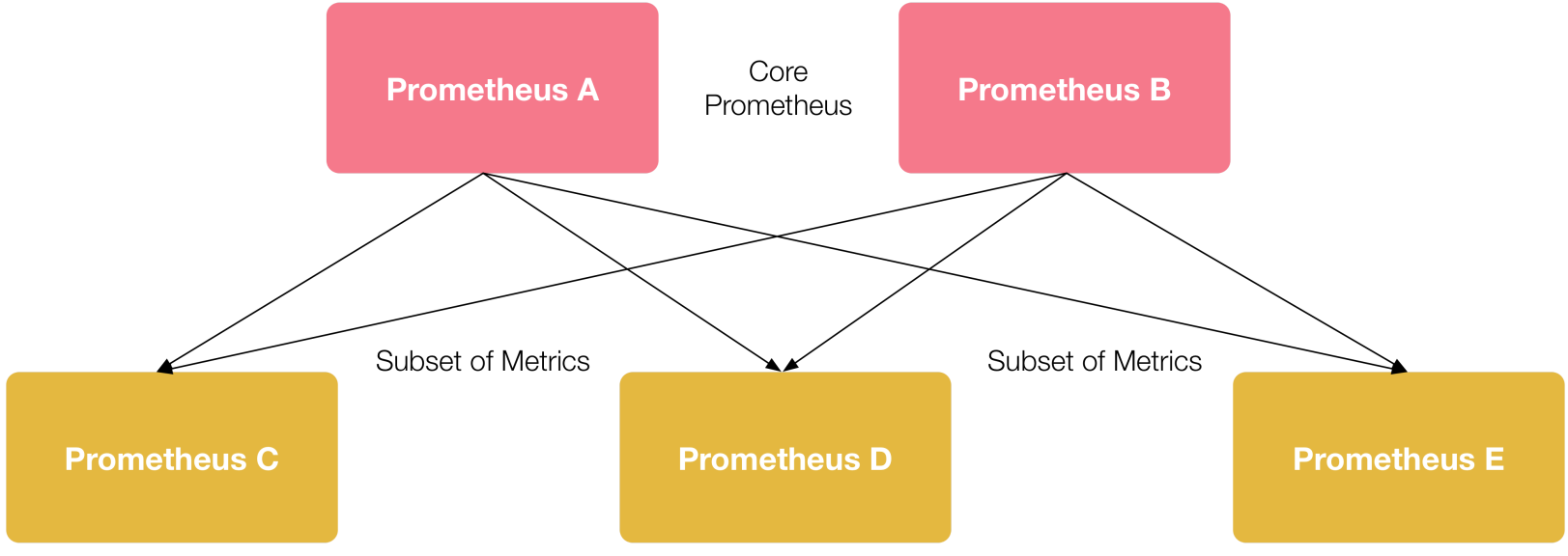
# Prometheus

## Статья про Service Discovery и Relabeling

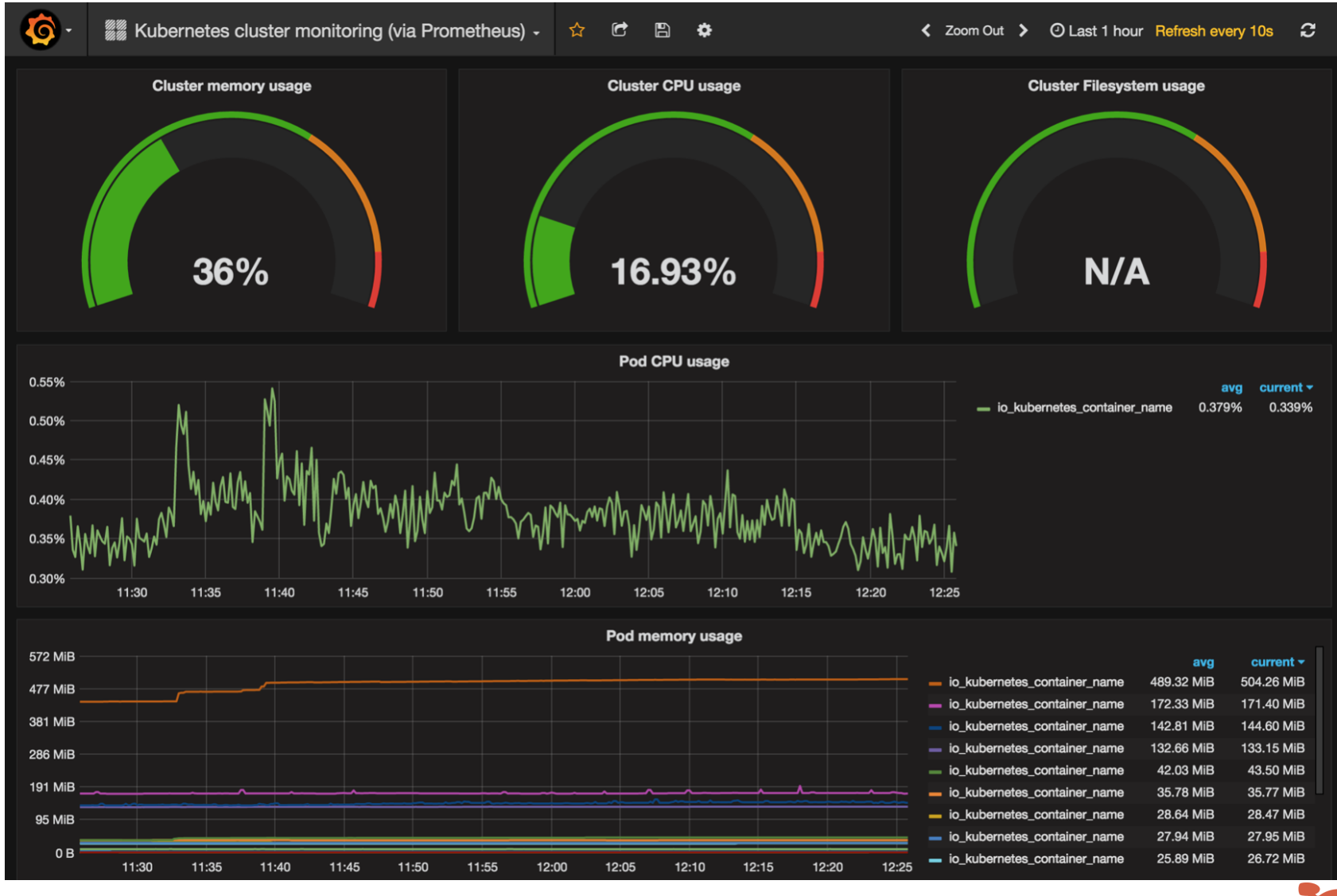
# Prometheus


Grafana
kube-state-metrics
Targets
k8s object metrics
container_metrics
app_metrics
node_metrics
kubelet
(cAdvisor)
kubelet
(cAdvisor)
C1
pod
C2
node_exporter
node_1
node_2

# Prometheus


Prometheus A
Core
Prometheus
Prometheus B
Subset of Metrics
Subset of Metrics
Prometheus C
Prometheus D
Prometheus E

# Prometheus


Kubernetes cluster monitoring (via Prometheus)
Zoom Out
Last 1 hour
Refresh every 10s
Cluster memory usage
36%
Cluster CPU usage
16.93%
Cluster Filesystem usage
N/A
Pod CPU usage
0.55%
0.50%
0.45%
0.40%
0.35%
0.30%
11:30
11:35
11:40
11:45
11:50
11:55
12:00
12:05
12:10
12:15
12:20
12:25
avg
current
io_kubernetes_container_name 0.379% 0.339%
Pod memory usage
572 MiB
477 MiB
381 MiB
286 MiB
191 MiB
95 MiB
0 B
avg
current
io_kubernetes_container_name 489.32 MiB 504.26 MiB
io_kubernetes_container_name 172.33 MiB 171.40 MiB
io_kubernetes_container_name 142.81 MiB 144.60 MiB
io_kubernetes_container_name 132.66 MiB 133.15 MiB
io_kubernetes_container_name 42.03 MiB 43.50 MiB
io_kubernetes_container_name 35.78 MiB 35.77 MiB
io_kubernetes_container_name 28.64 MiB 28.47 MiB
io_kubernetes_container_name 27.94 MiB 27.95 MiB
io_kubernetes_container_name 25.89 MiB 26.72 MiB

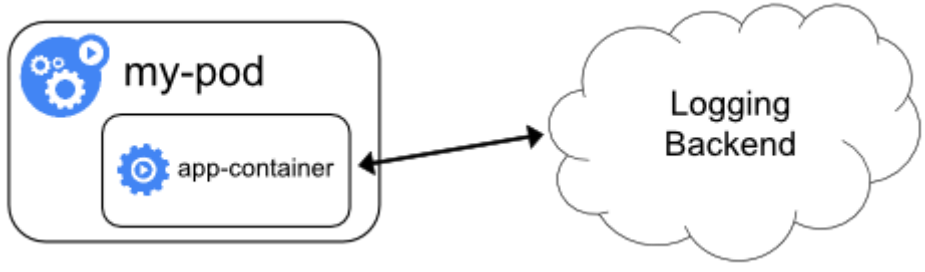
# Логирование

# kubectl logs

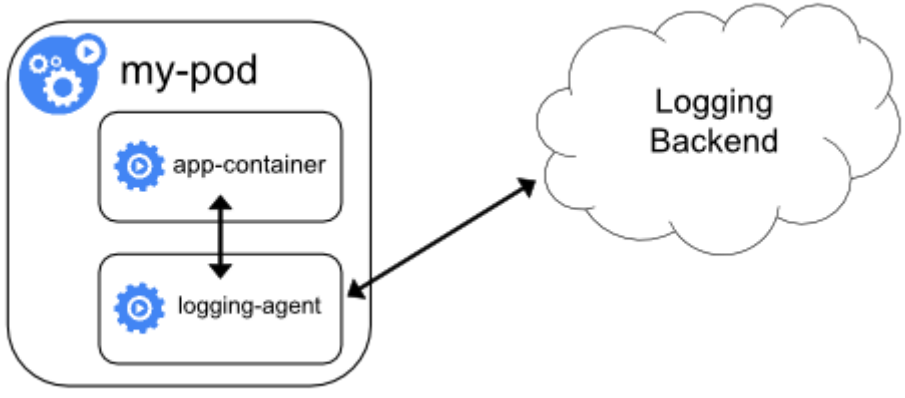
```
$ kubectl logs pod_name container_name --tail 10 -f

10.8.19.5 - - [04/Dec/2017 15:04:16] "GET /metrics HTTP/1.1" 200 -
10.8.19.5 - - [04/Dec/2017 15:05:16] "GET /metrics HTTP/1.1" 200 -
10.8.19.5 - - [04/Dec/2017 15:06:16] "GET /metrics HTTP/1.1" 200 -
10.8.19.5 - - [04/Dec/2017 15:07:16] "GET /metrics HTTP/1.1" 200 -
10.8.19.5 - - [04/Dec/2017 15:08:16] "GET /metrics HTTP/1.1" 200 -
10.8.19.5 - - [04/Dec/2017 15:09:16] "GET /metrics HTTP/1.1" 200 -
```

У Kubernetes нет встроенных механизмов для отправки логов (таких как logging drivers в Docker)

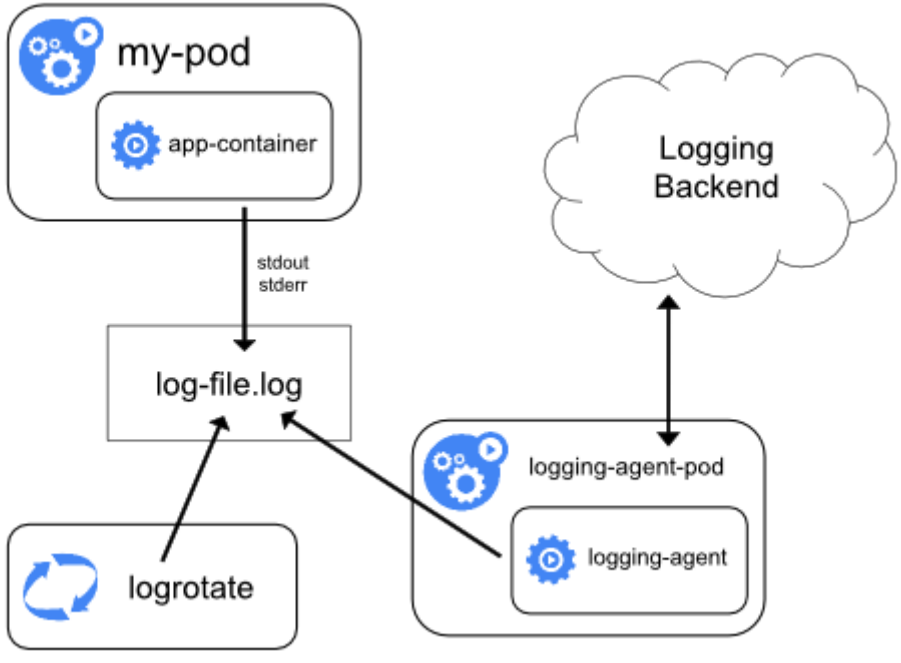
# Логирование: From app



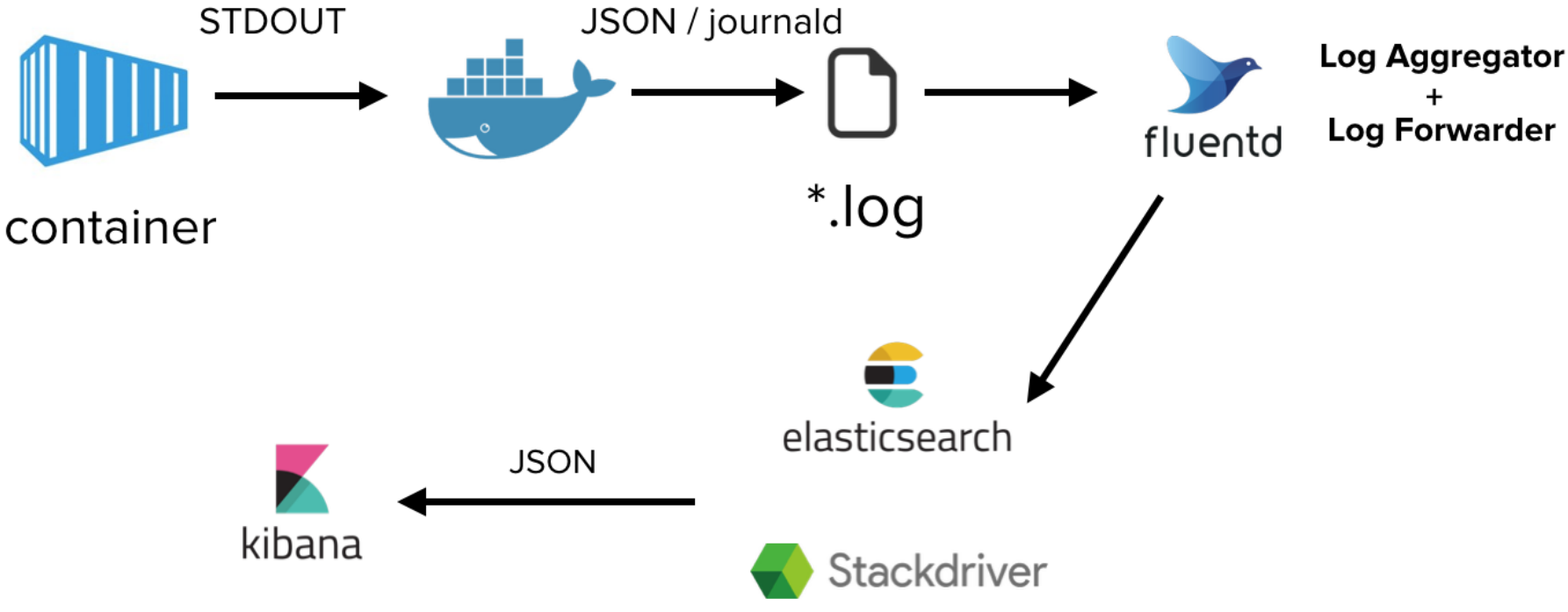
# Логирование: Sidecar



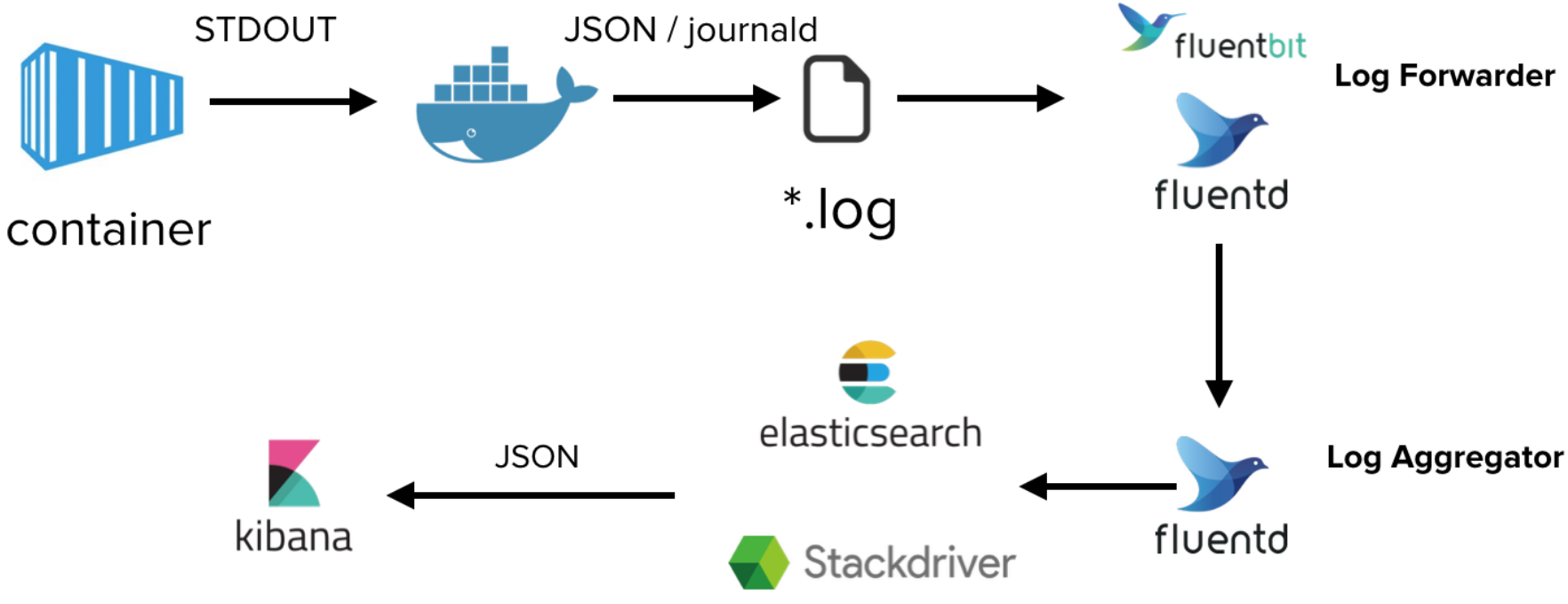
# Логирование: External pod



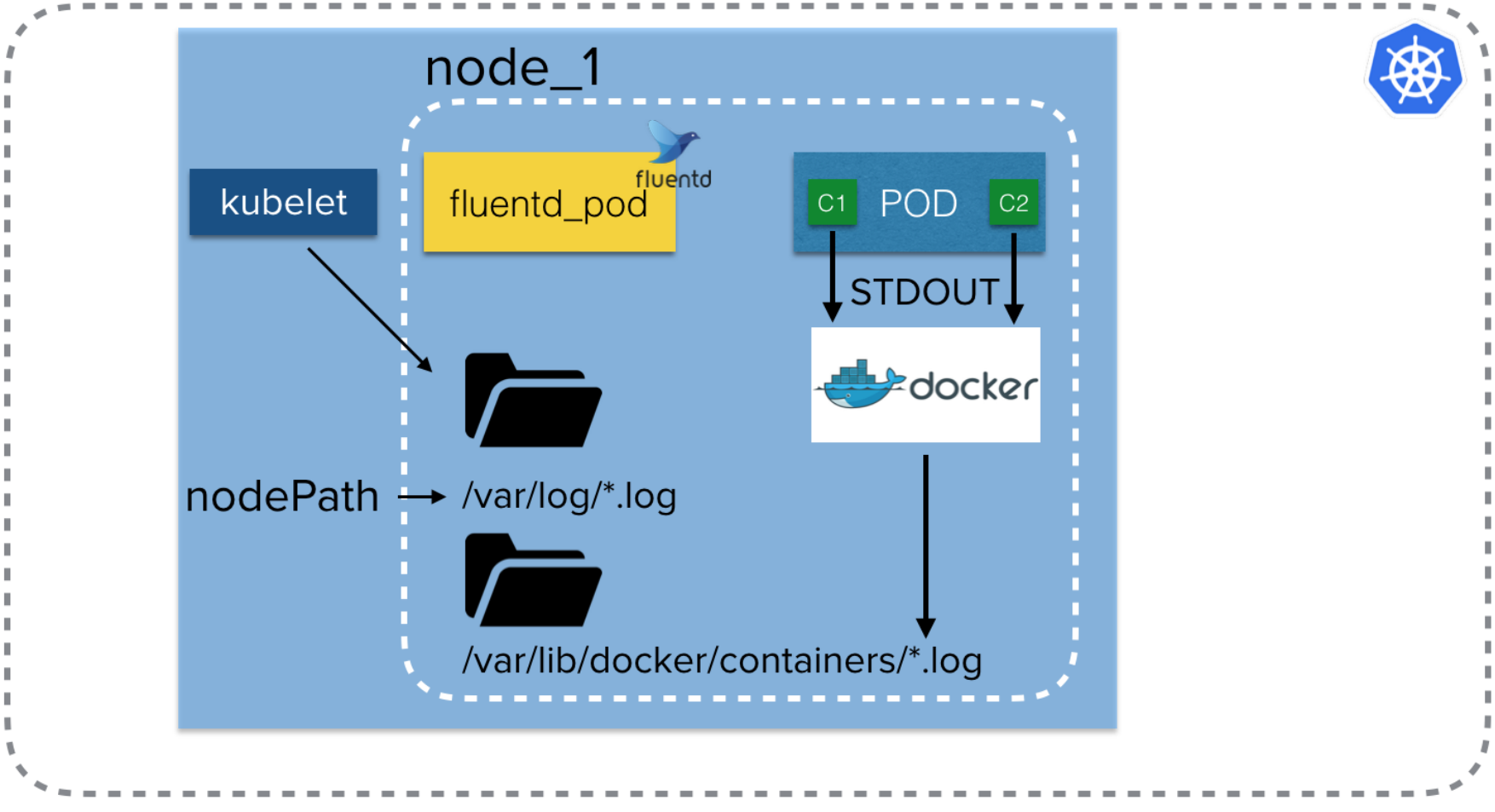
# Логирование



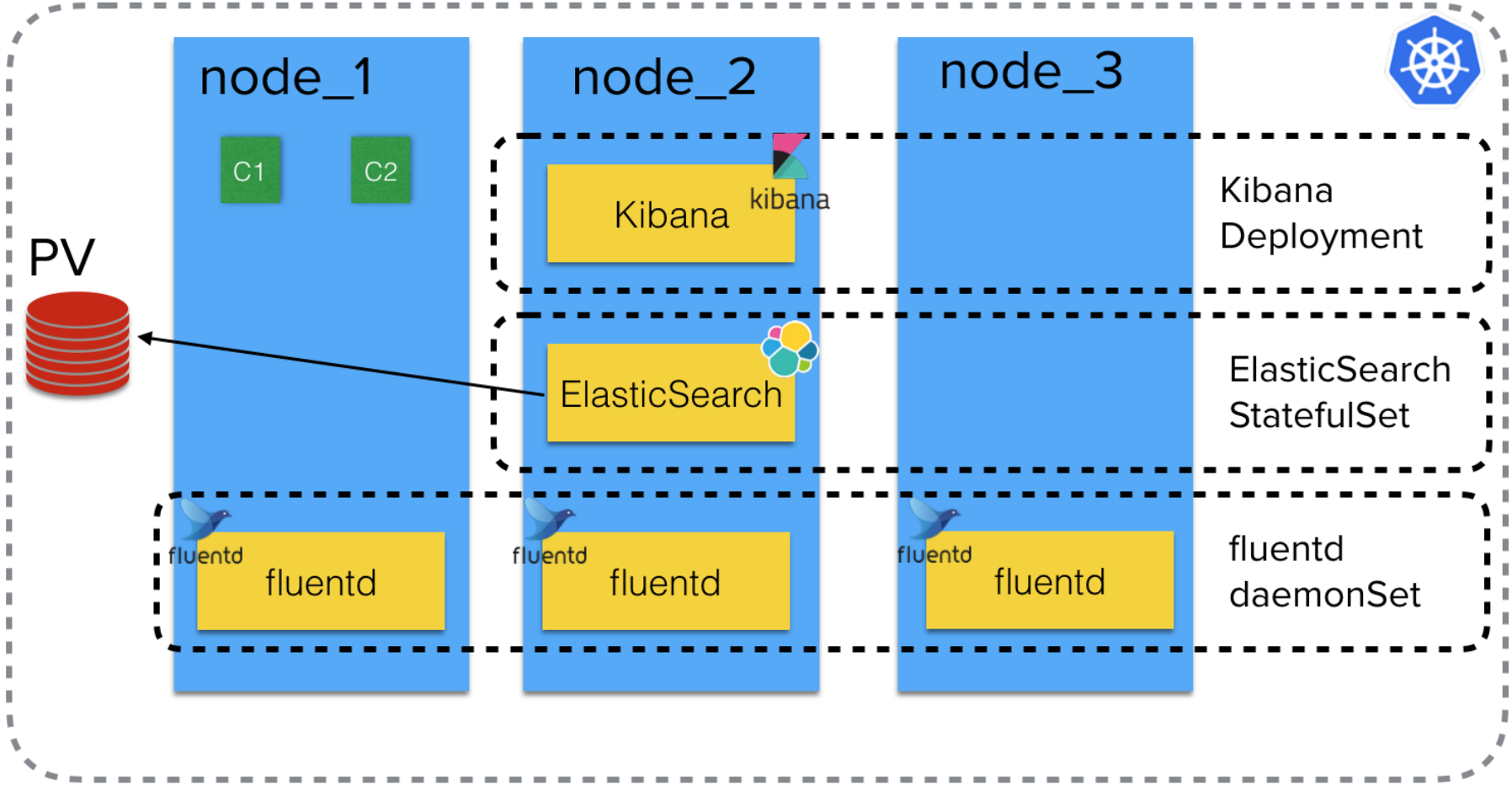
# Логирование



# Логирование



# Логирование



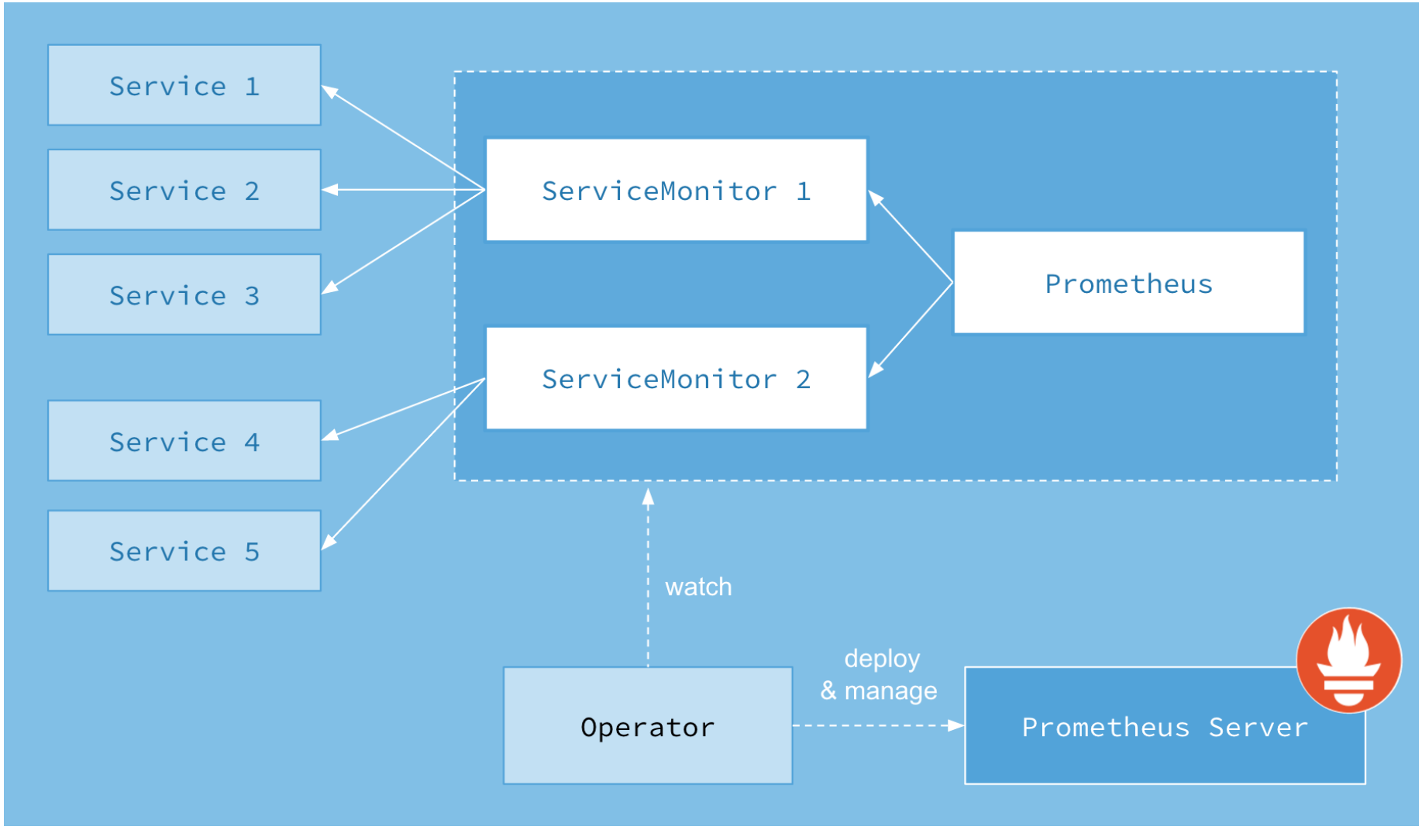
# Операторы


An Operator represents human operational knowledge in software, to reliably manage an application.

CoreOS

- etcd Operator
- Rook Operator
- Prometheus Operator
- Vault Operator

# Prometheus operator



# Prometheus operator

## Getting started

```
kubectl apply -f prometheus-operator.yml
```

```
$ kubectl get crd
NAME                                    AGE
alertmanagers.monitoring.coreos.com     39m
backendconfigs.cloud.google.com         1h
prometheuses.monitoring.coreos.com      39m
prometheusrules.monitoring.coreos.com   39m
servicemonitors.monitoring.coreos.com   39m
```

# Prometheus operator

prometheus.yml:

```
kind: Prometheus
metadata:
  name: prometheus
  labels:
    app: prometheus
spec:
  serviceAccountName: prometheus
  serviceMonitorSelector:
    matchLabels:
      prometheus: servicemonitor
  resources:
    requests:
      memory: 400Mi
  enableAdminAPI: false
```

# Prometheus operator

ui-servicemonitor.yml:

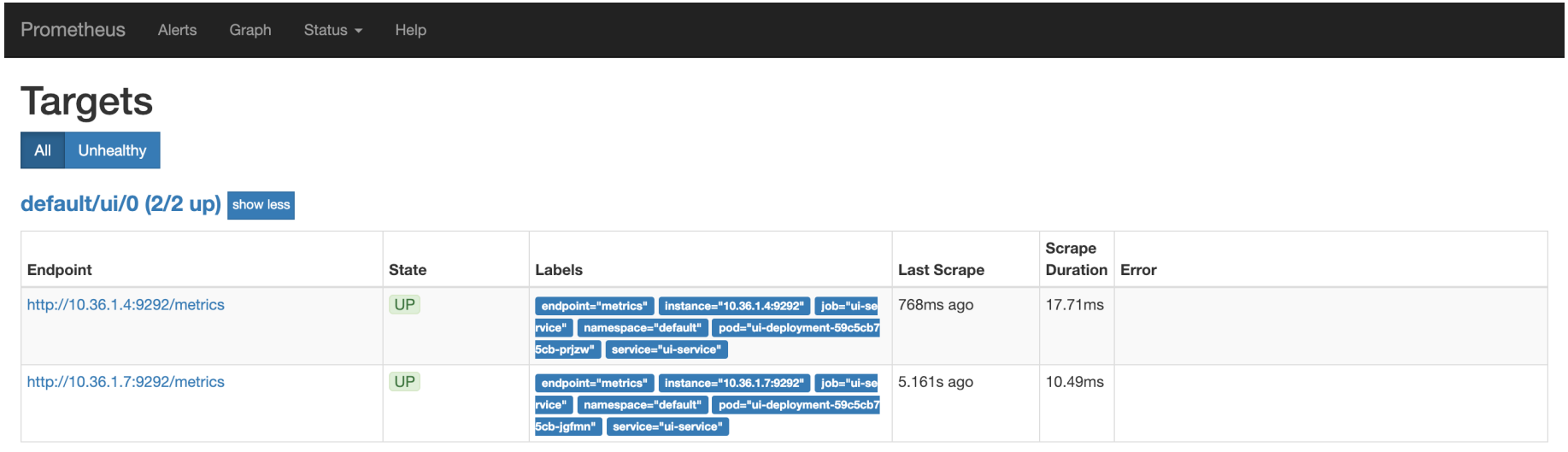
```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: ui
  labels:
    prometheus: servicemonitor
spec:
  selector:
    matchLabels:
      app: ui
  endpoints:
  - path: /metrics
    port: metrics
    interval: 10s
```

# Prometheus operator

ui-service.yml:

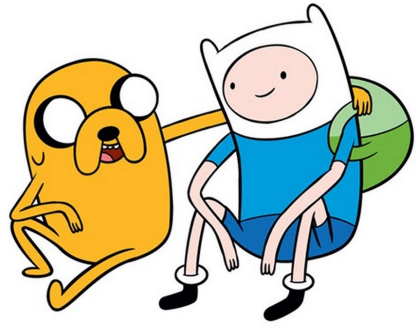
```
kind: Service
apiVersion: v1
metadata:
  name: ui-service
  labels:
    app: ui
spec:
  selector:
    app: ui
  ports:
  - name: metrics
    protocol: TCP
    port: 9292
    targetPort: 9292
```

# Prometheus operator



# Полезные ссылки

- Operating within Normal Parameters: Monitoring Kubernetes (SREcon 2019)
- Масштабирование приложения в Kubernetes на основе метрик из Prometheus
- Kube eagle
- Долгосрочное хранение метрик Prometheus



# Спасибо за внимание!

**Время для ваших вопросов!**