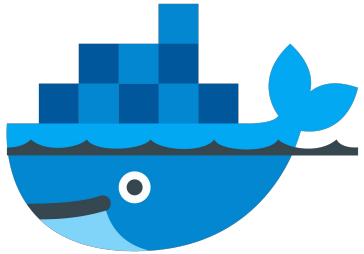
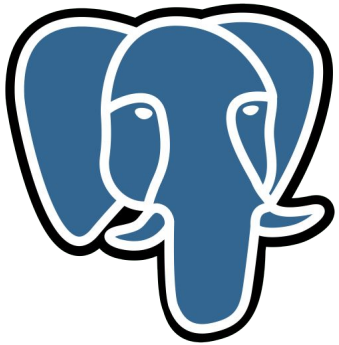
Счастливая жизнь с Kubernetes в продакшне

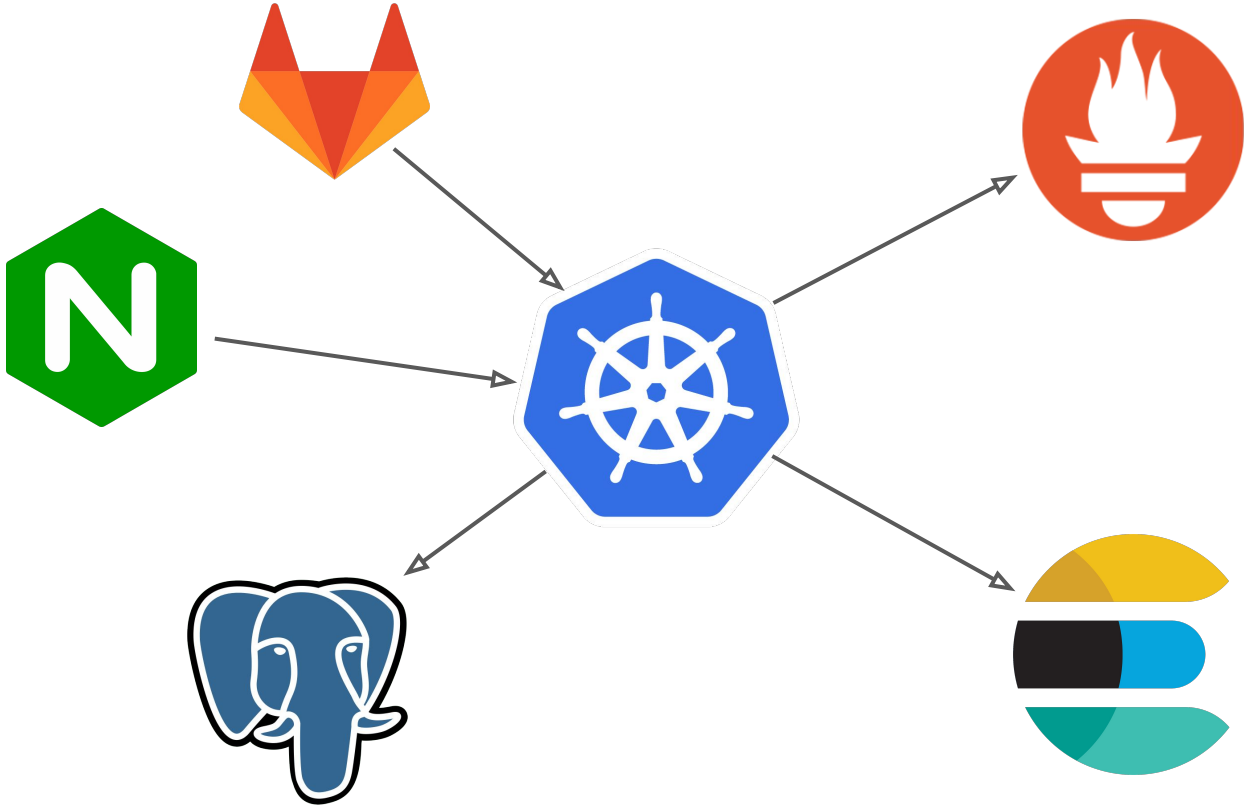
Дехтярёв Евгений

2GIS




Stack





Поговорим

Kubernetes

Мониторинг

Журналирование

Бэкенды

Роутеры

Доставка в k8s

Путь к Kubernetes

Проблема

Скорость доставки приложений

Задачи

Унификация инфраструктуры

Унификация доставки приложений

Путь к Kubernetes

Проблема

Скорость доставки приложений - **упаковка в контейнеры**

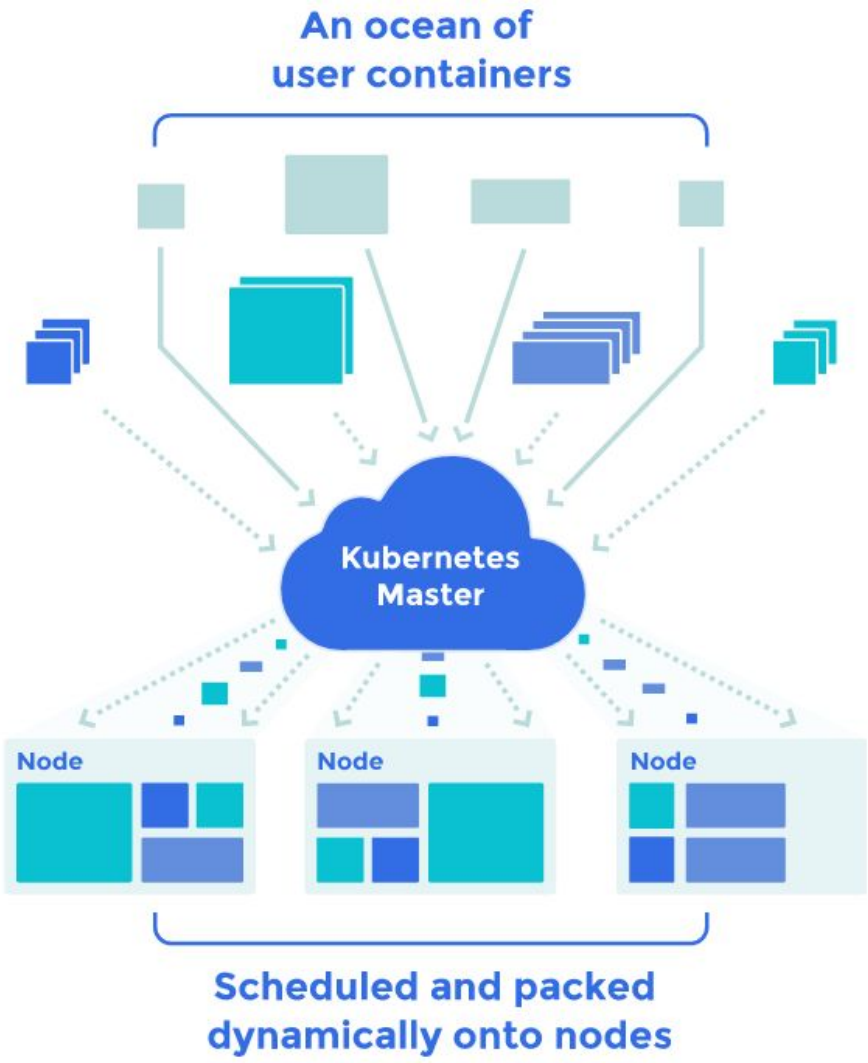
Задачи

Унификация инфраструктуры

Унификация доставки приложений

Один оркестратор

Kubernetes


An ocean of
user containers
Kubernetes
Master
Node
Node
Node
Scheduled and packed
dynamically onto nodes

Testing

Staging

Testing

Staging

Testing

Production **x4**

500+ приложений

500+ приложений

- API

500+ приложений

- API
- Gitlab

500+ приложений

- API
- Gitlab
- Jobs

500+ приложений

- API
- Gitlab
- Jobs
- m.2gis.ru

500+ приложений

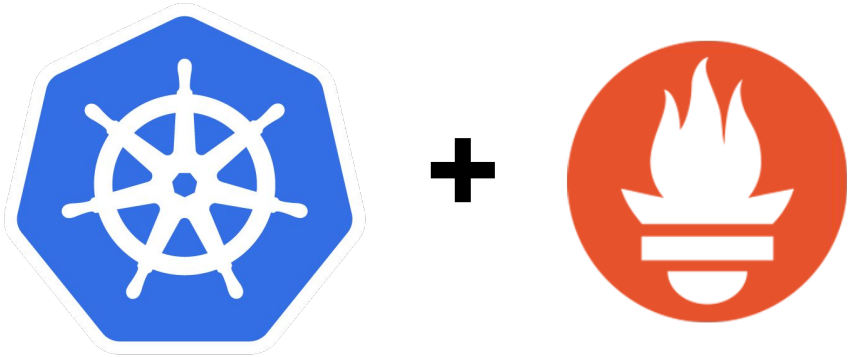
- API
- Gitlab
- Jobs
- m.2gis.ru
- Карта

500+ приложений

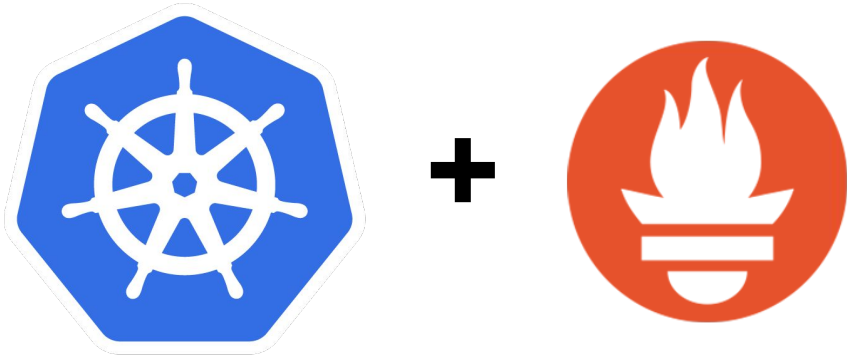
- API
- Gitlab
- Jobs
- m.2gis.ru
- Карта

15 - 20 kRPS

Мониторинг



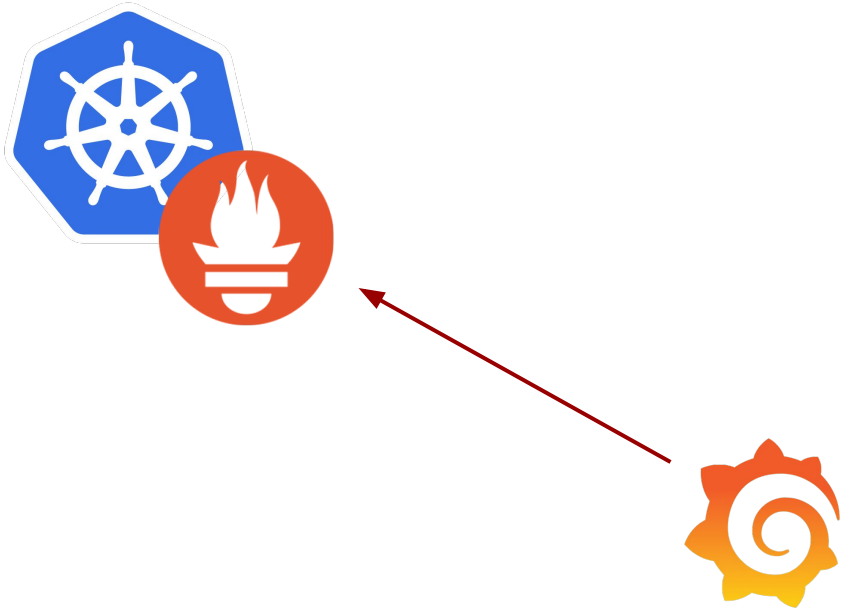
cAdvisor
Kube-state-metrics
Node exporter



cAdvisor
Kube-state-metrics
Node exporter

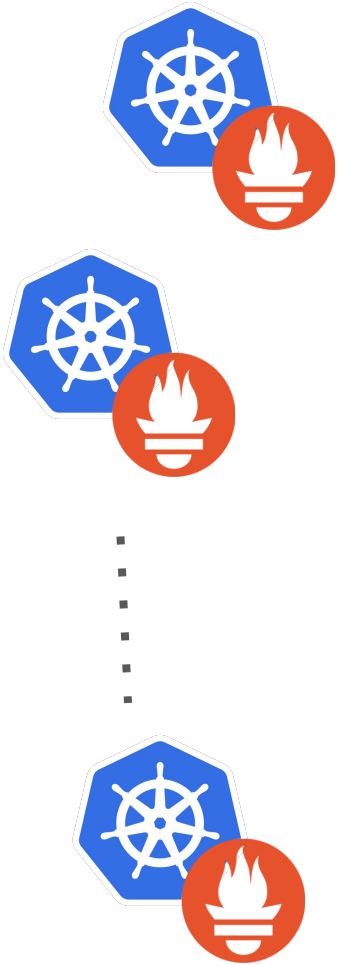
+

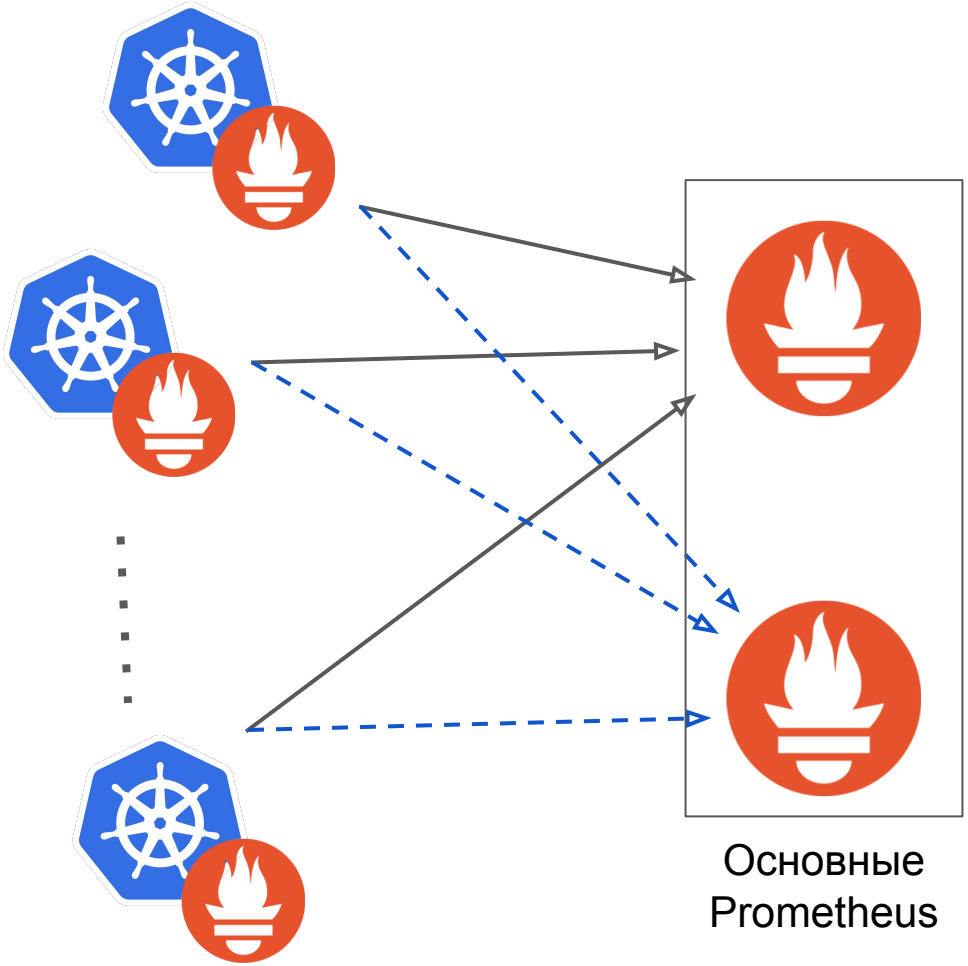
Service discovery
Alertmanager
Grafana

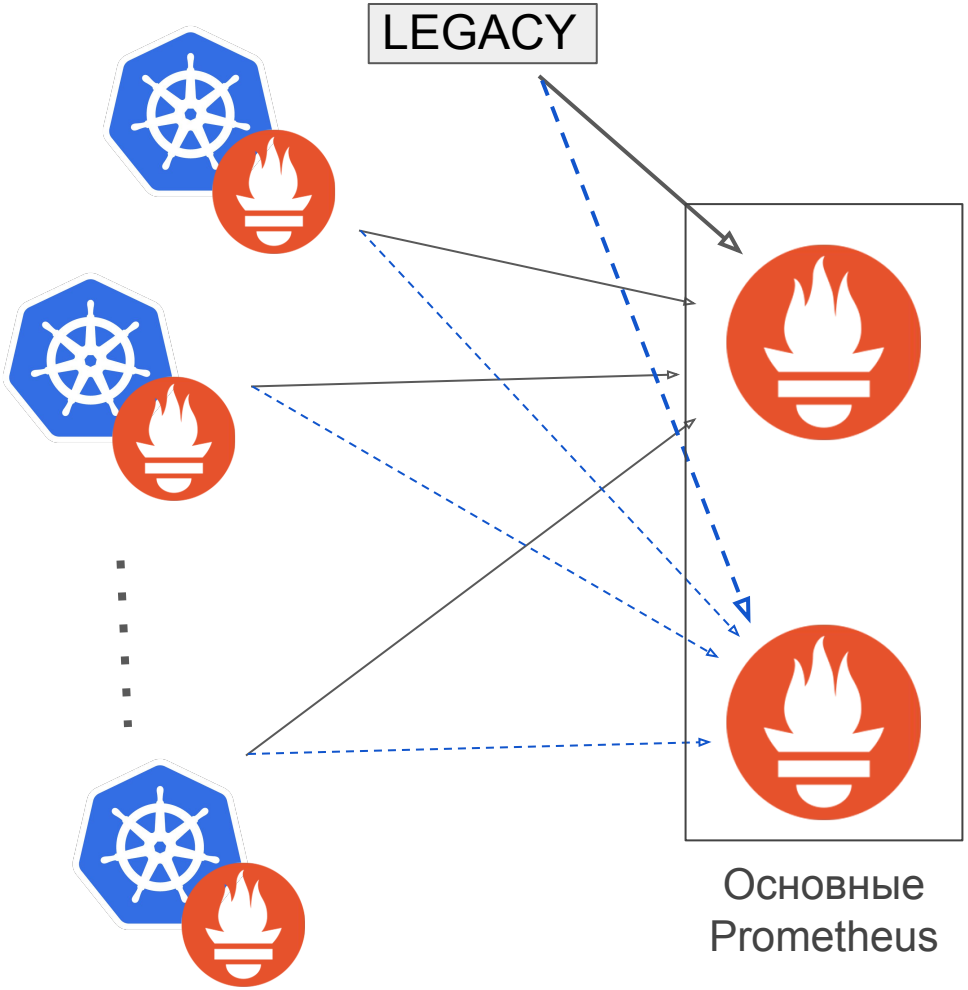


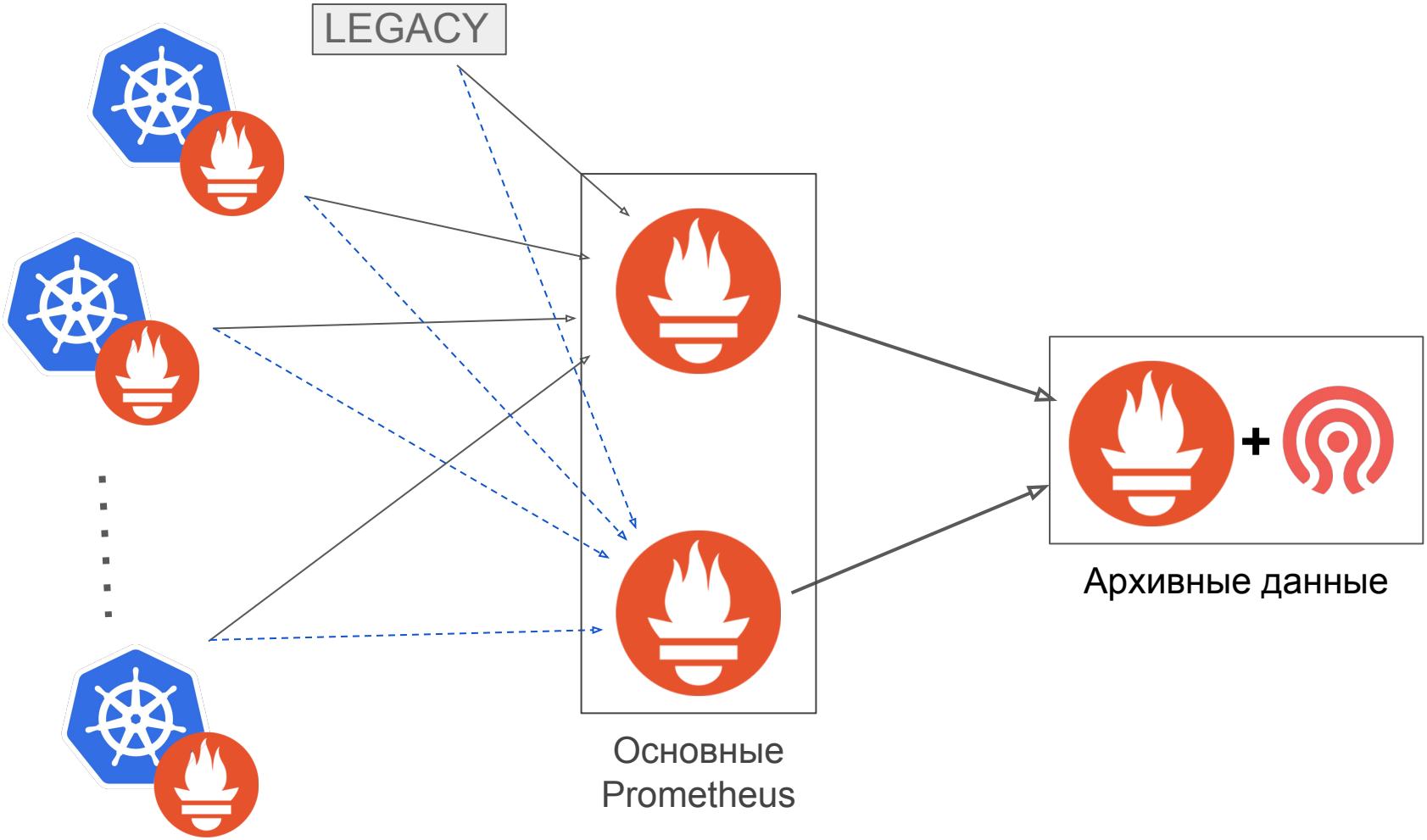


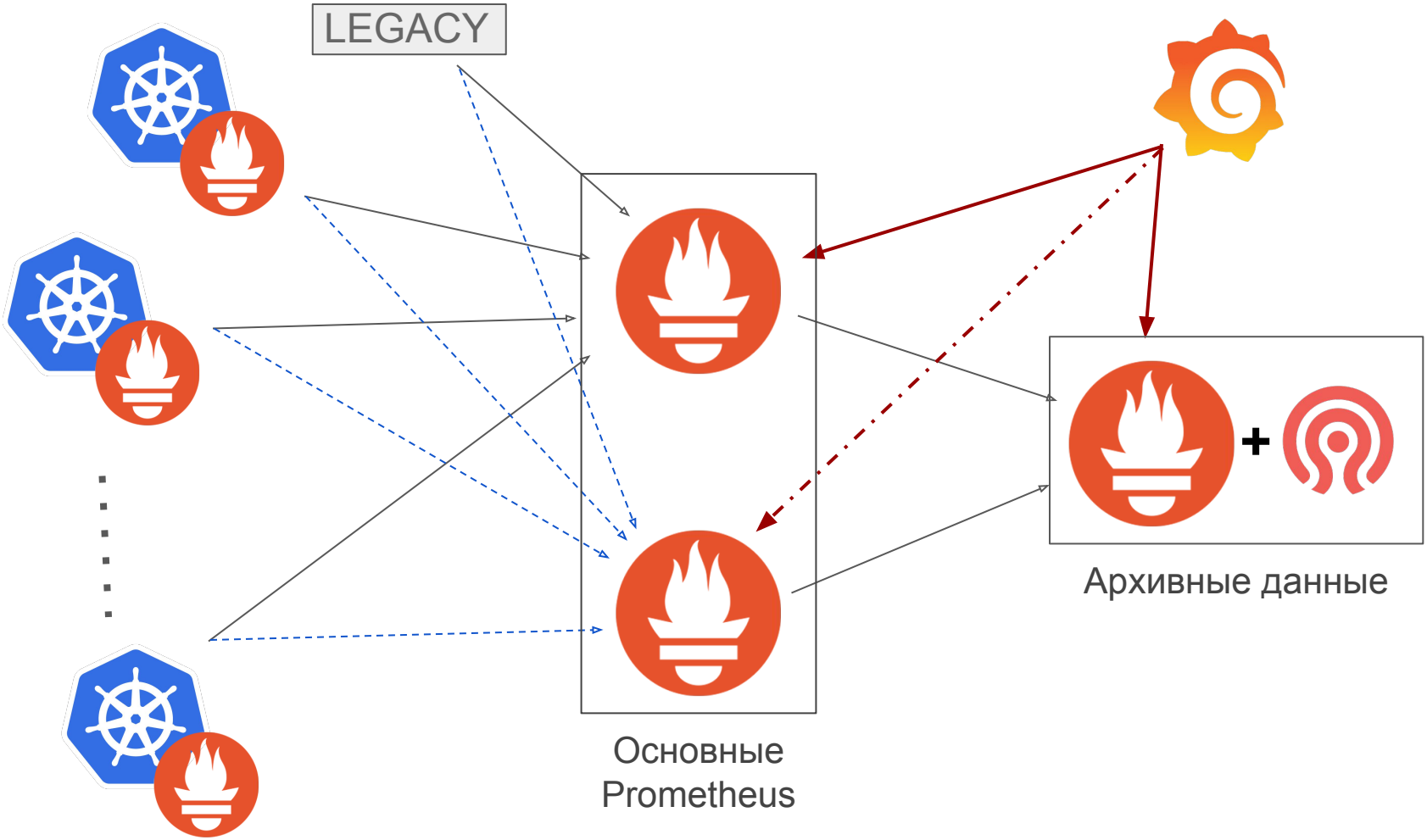


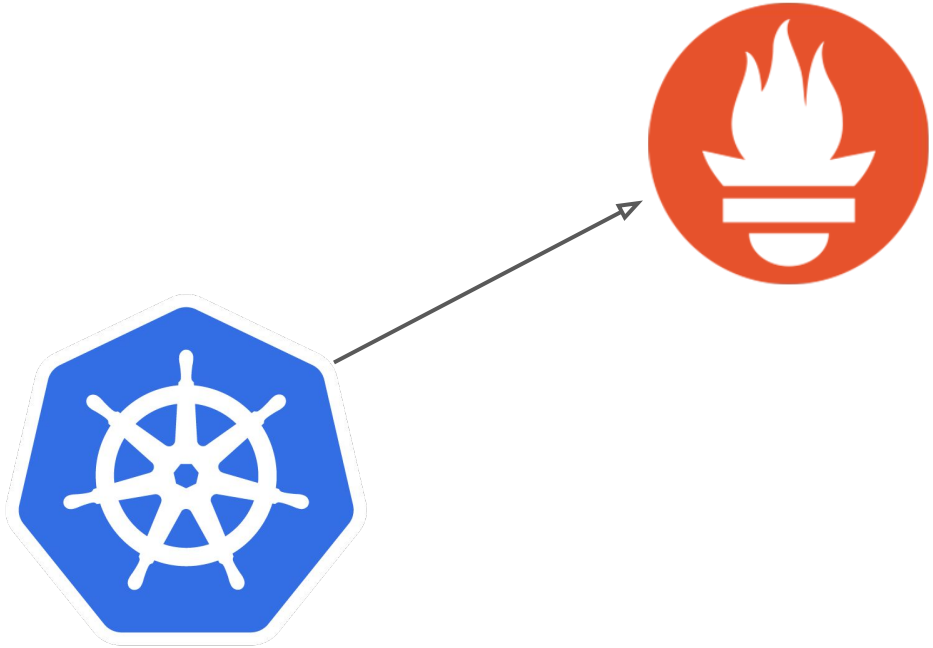




Основные
Prometheus

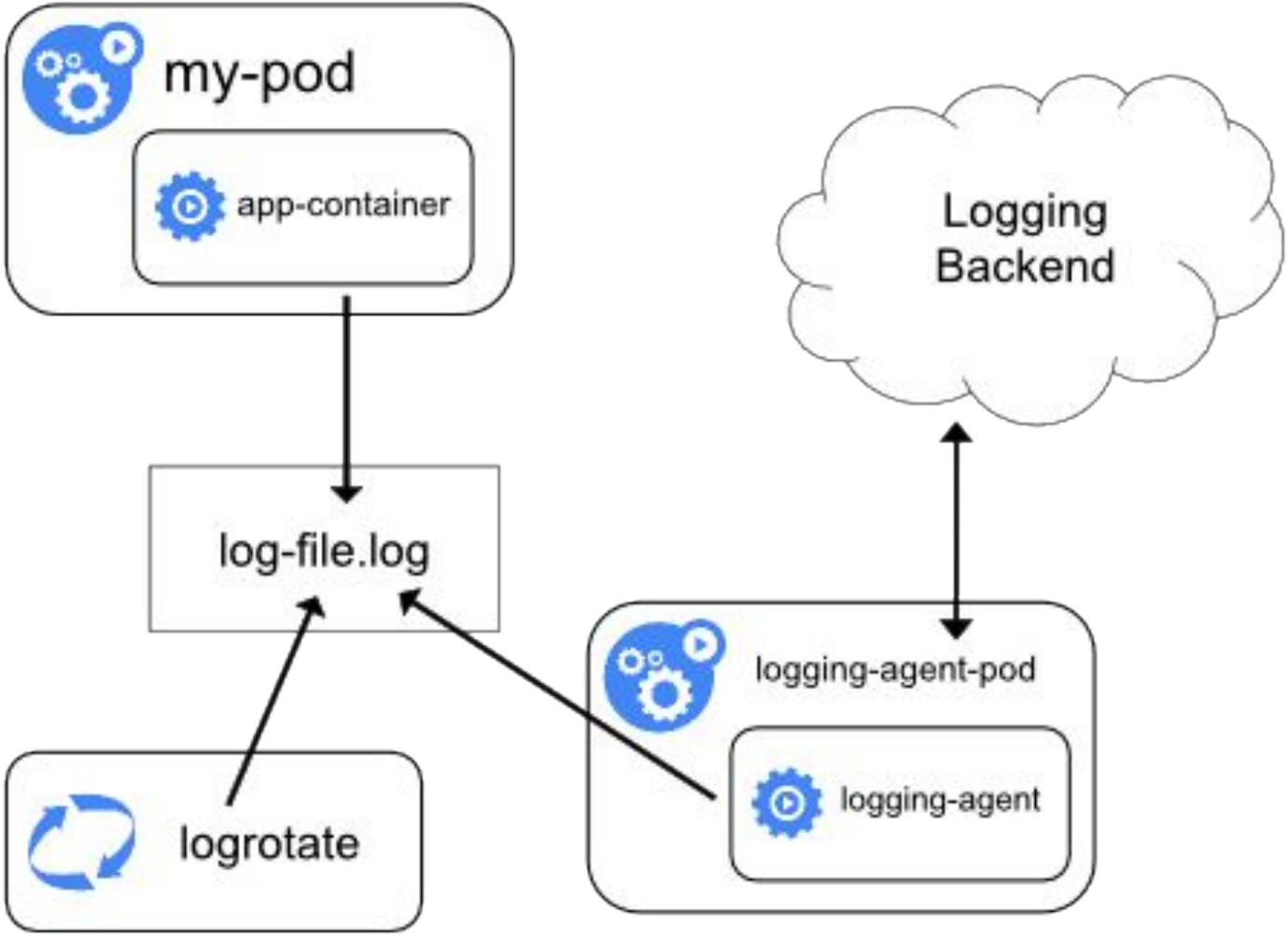

LEGACY
Основные
Prometheus


LEGACY
Основные
Prometheus
Архивные данные


LEGACY
Основные
Prometheus
Архивные данные



Журналирование


my-pod
app-container
Logging
Backend
log-file.log
logging-agent-pod
logging-agent
logrotate


log-file.log
Logging agent
Logging Backend

Fluent/Logstash
~ 1 CPU
~ 1 Gb RAM

Fluent/Logstash
~ 1 CPU
~ 1 Gb RAM

100 k8s node
~ 100 CPU
~ 100 Gb RAM

Fluent/Logstash
~ 1 CPU
~ 1 Gb RAM

100 k8s node
~ 100 CPU
~ 100 Gb RAM

2GIS Golang Logging Agent
~ 0.5 CPU
~ 0.1 GB RAM

Fluent/Logstash
~ 1 CPU
~ 1 Gb RAM

100 k8s node
~ 100 CPU
~ 100 Gb RAM

2GIS Golang Logging Agent
~ 0.5 CPU
~ 0.1 GB RAM

100 k8s node
~ 50 CPU
~ 10 Gb RAM

Fluent/Logstash
~ 1 CPU
~ 1 Gb RAM

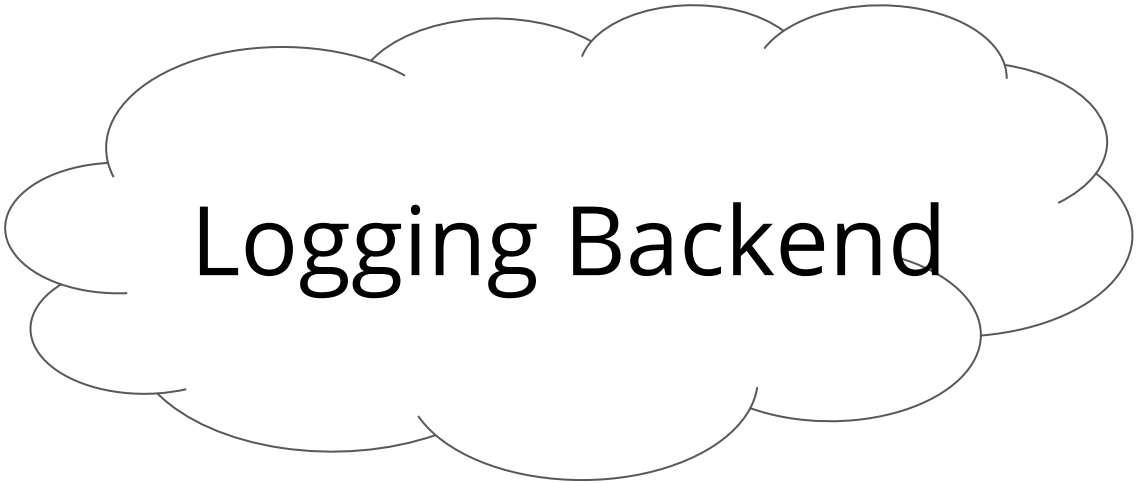
100 k8s node
~ 100 CPU
~ 100 Gb RAM

2GIS Golang Logging Agent
~ 0.5 CPU
~ 0.1 GB RAM

100 k8s node
~ 50 CPU
~ 10 Gb RAM

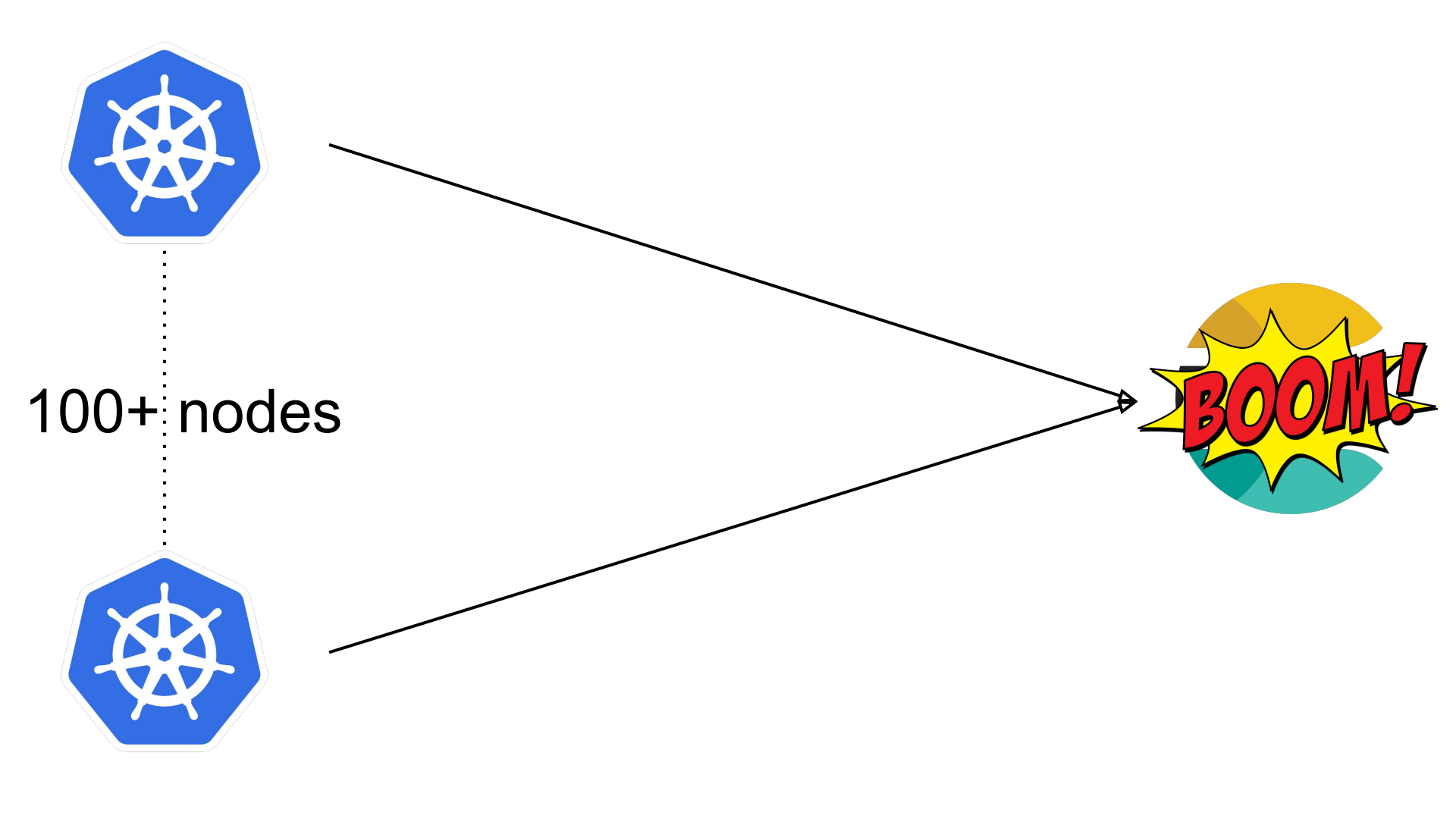
SLA-metrics

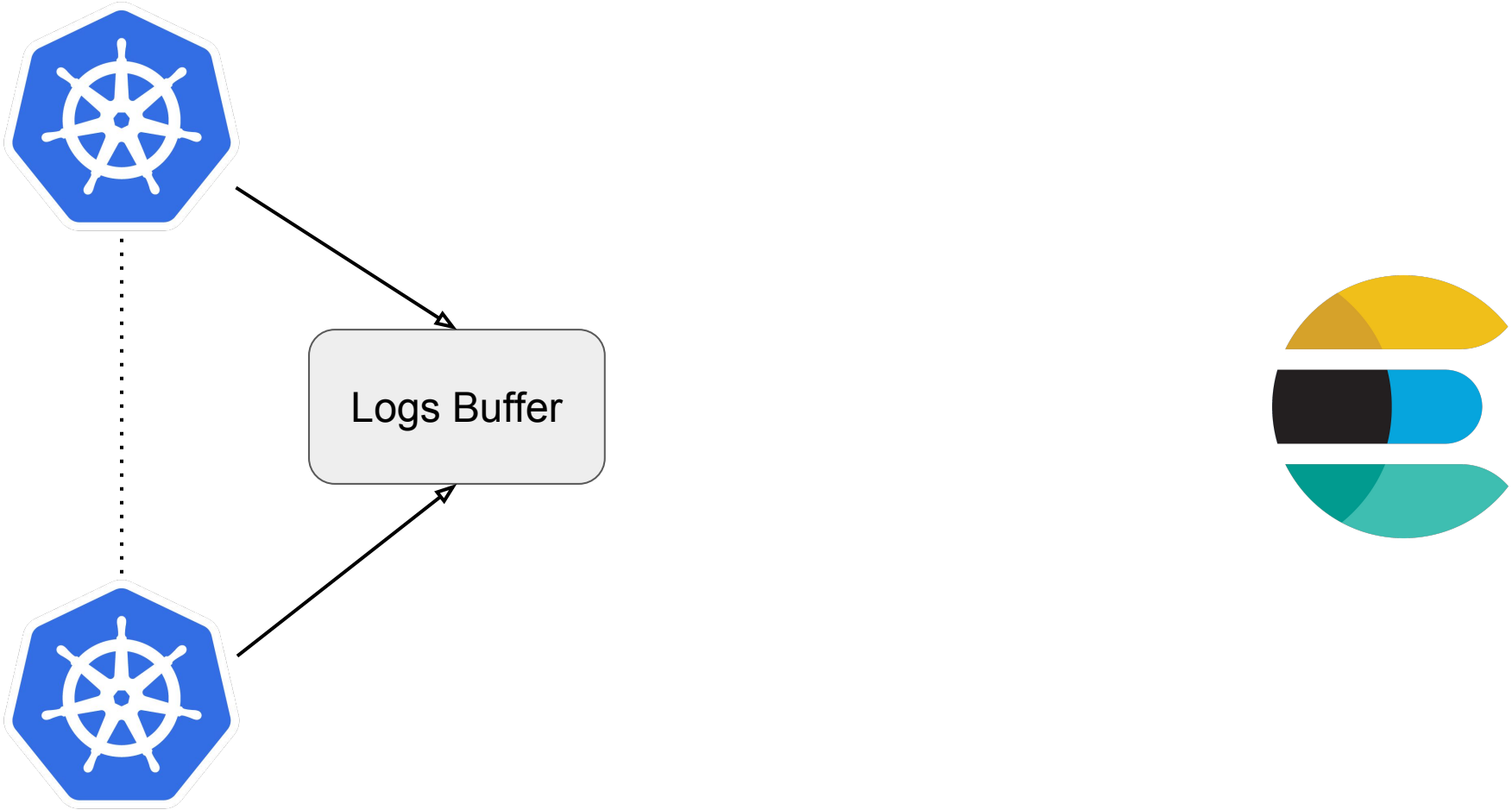
- Относительная доступность
- 95% время ответа метода

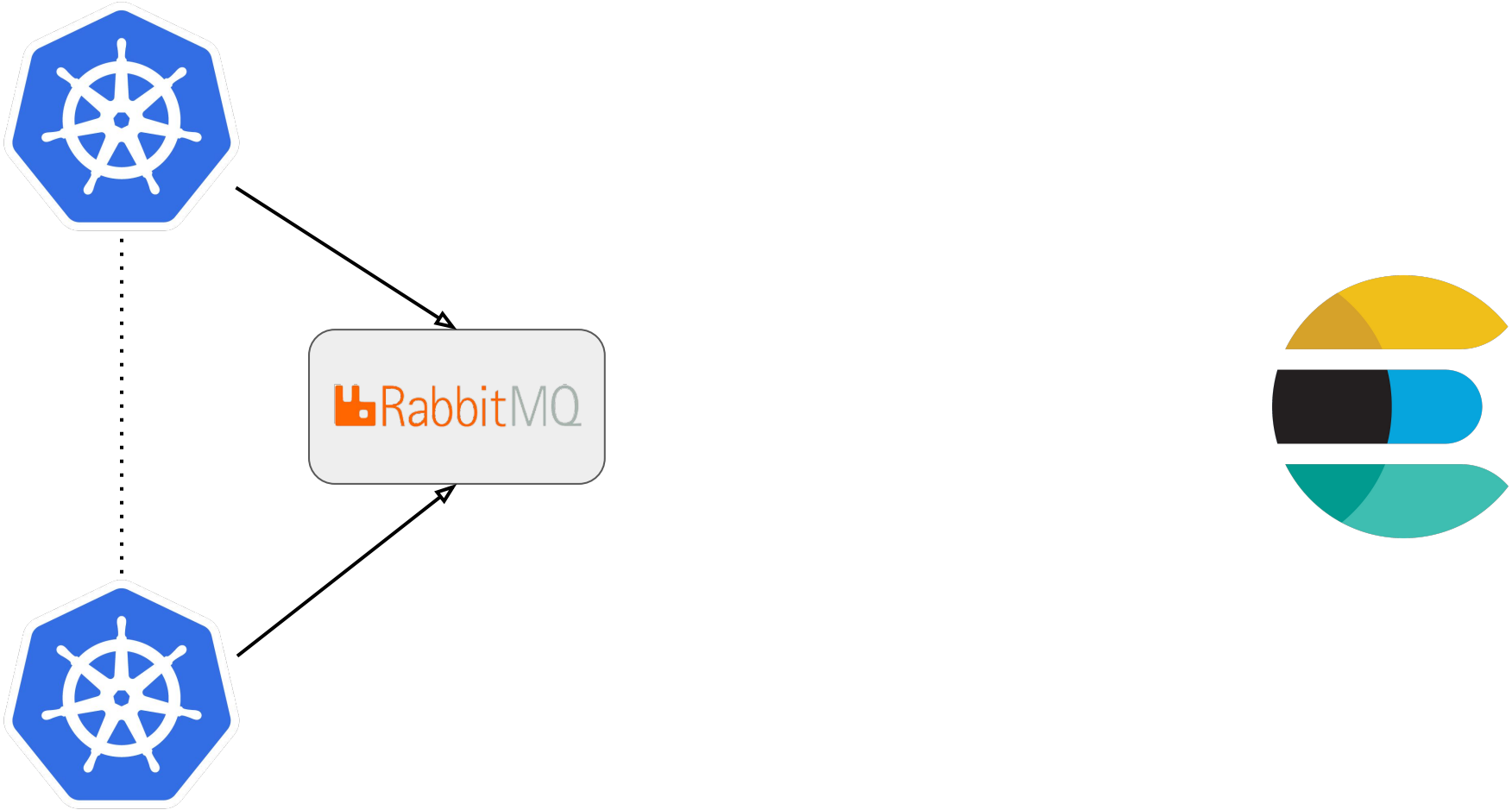

Logging Backend

100+ nodes

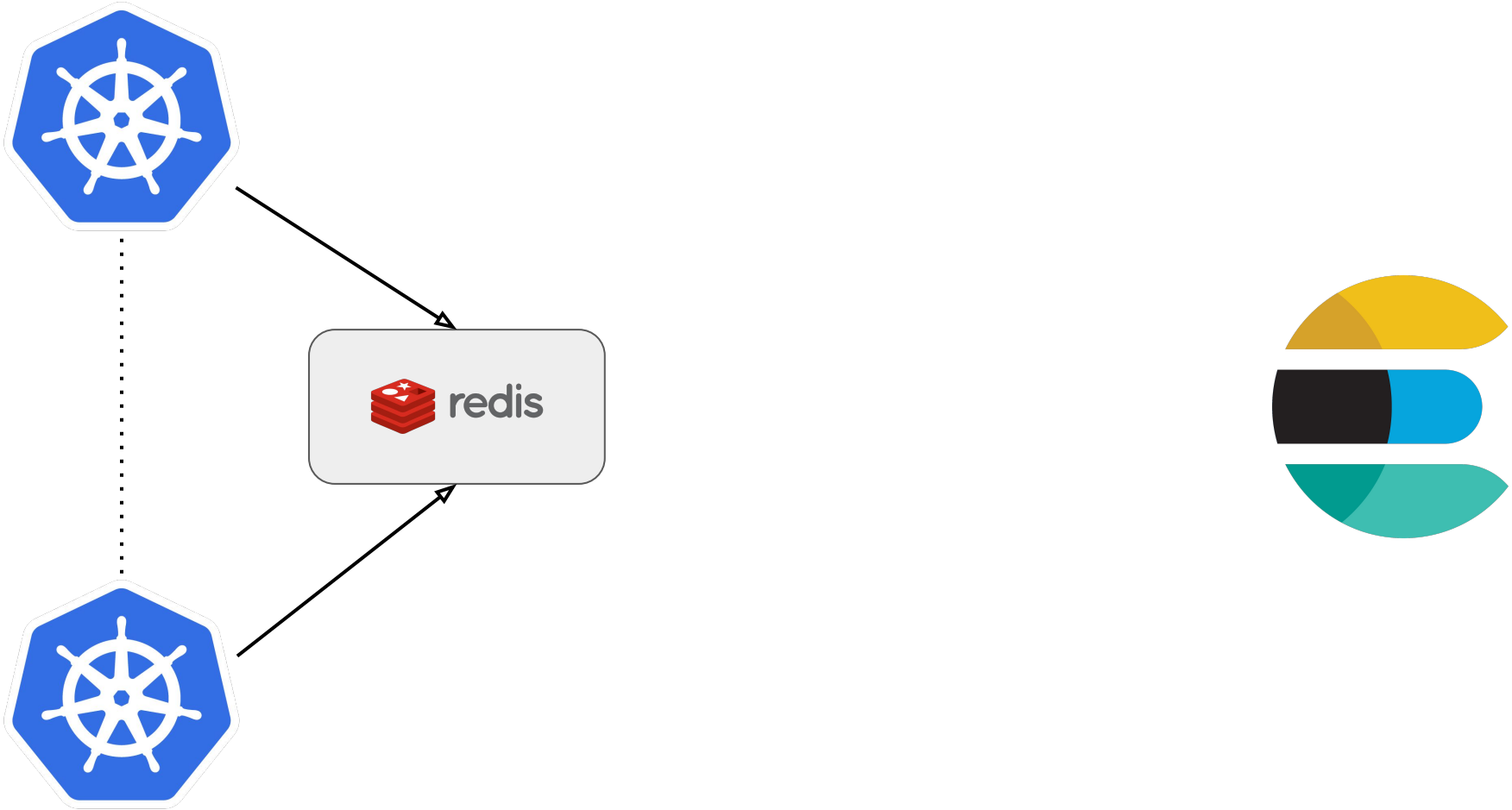
?

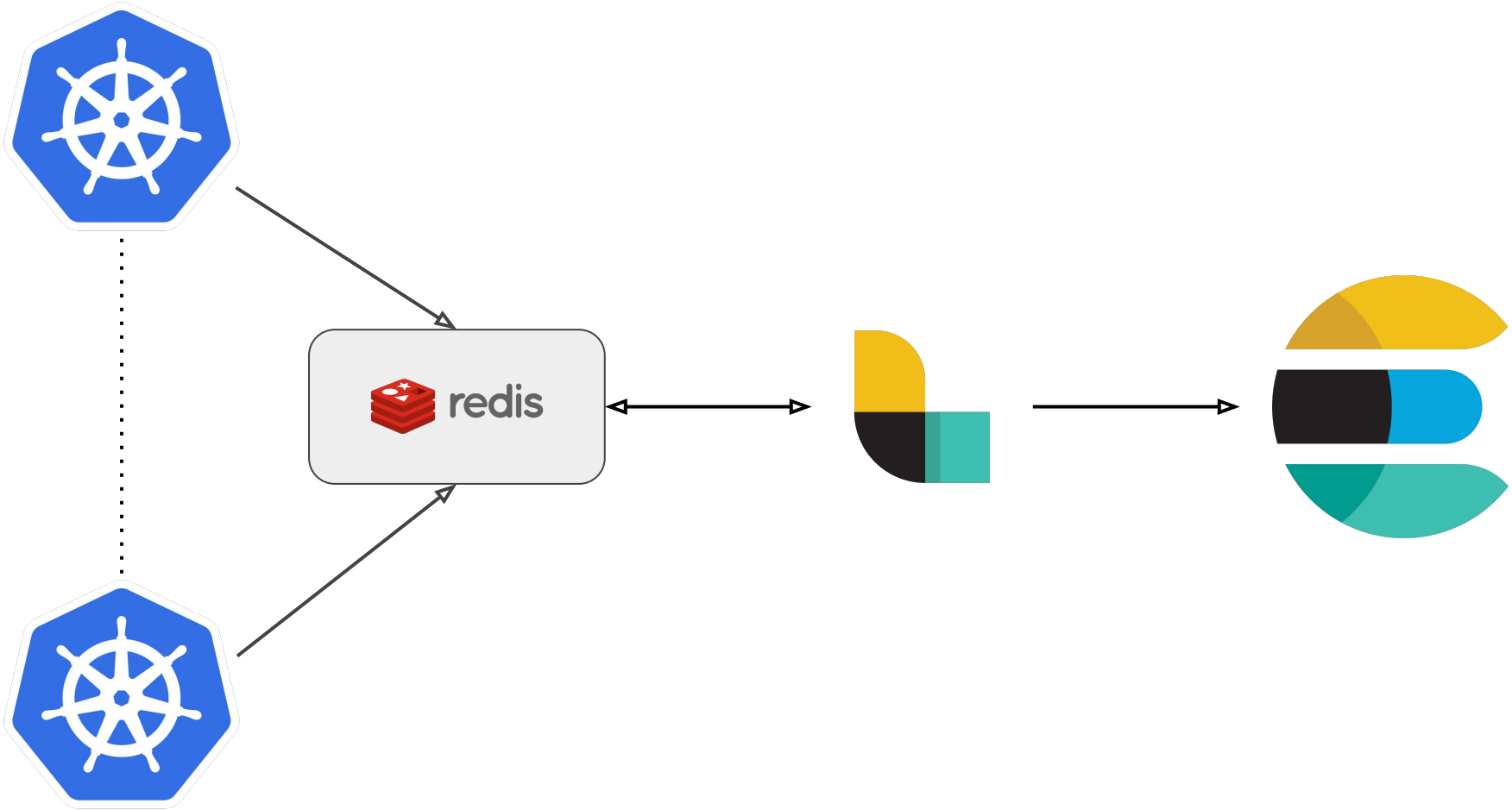

100+ nodes
BOOM!

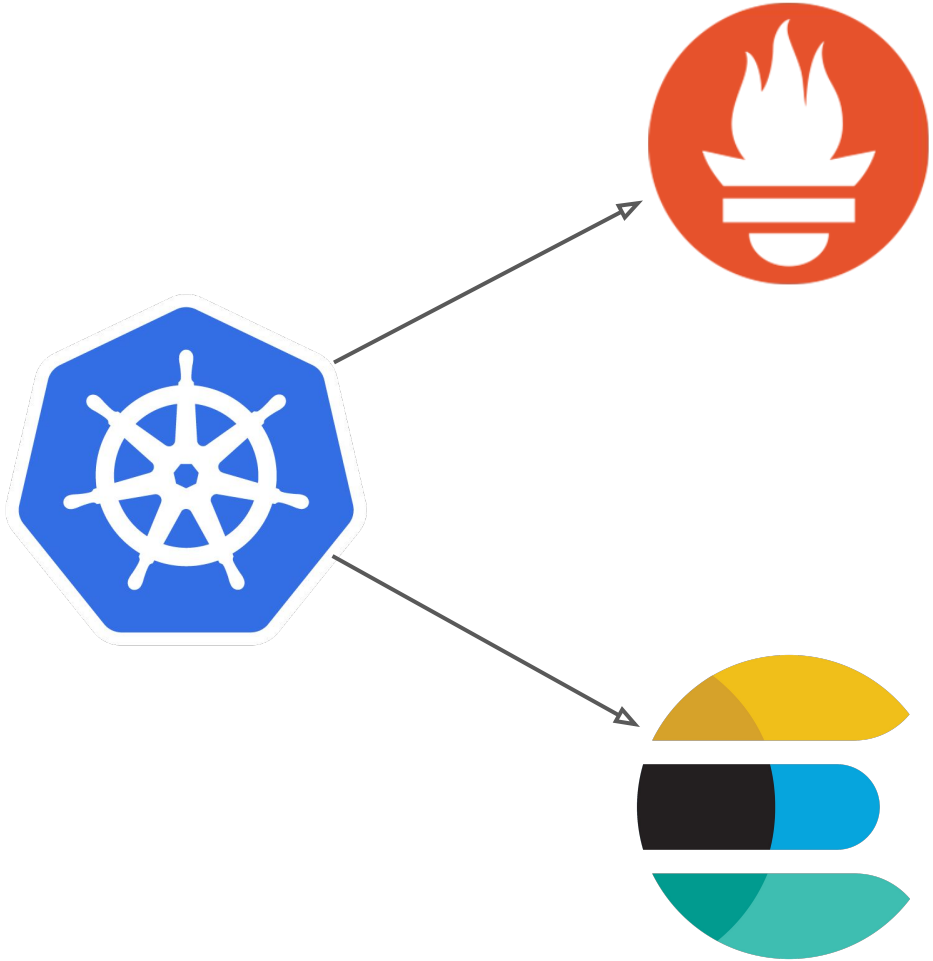

Logs Buffer


RabbitMQ


RabbitMQ
FAIL


redis


redis



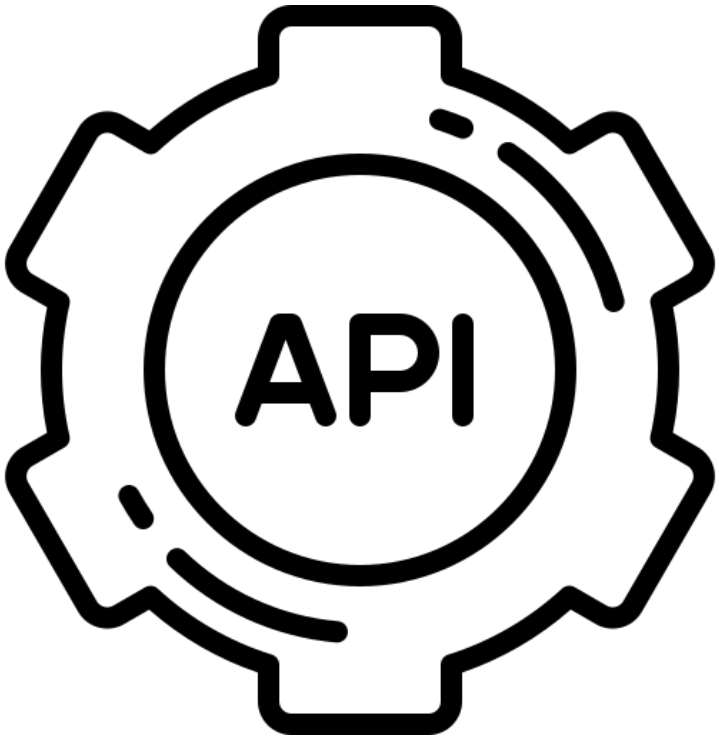
Бэкенды


JS

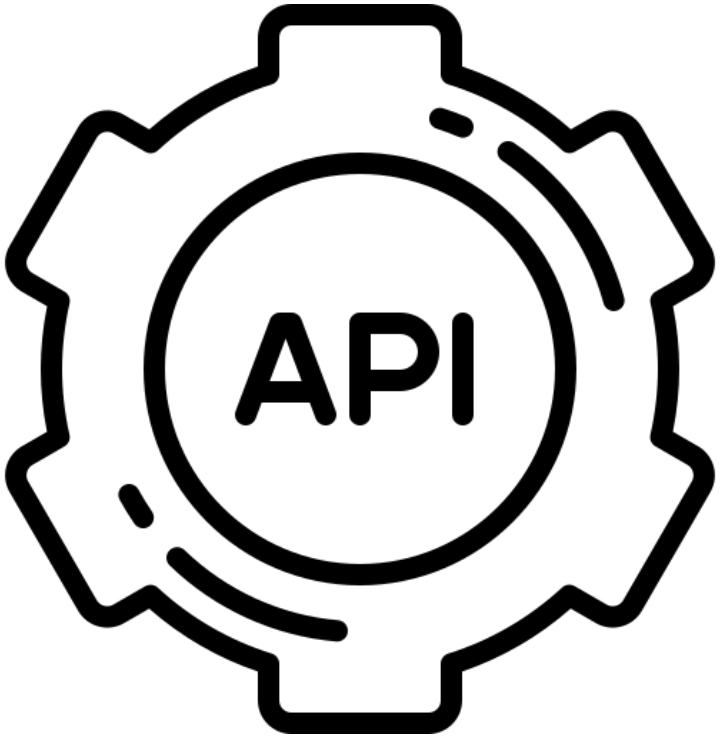
WWW

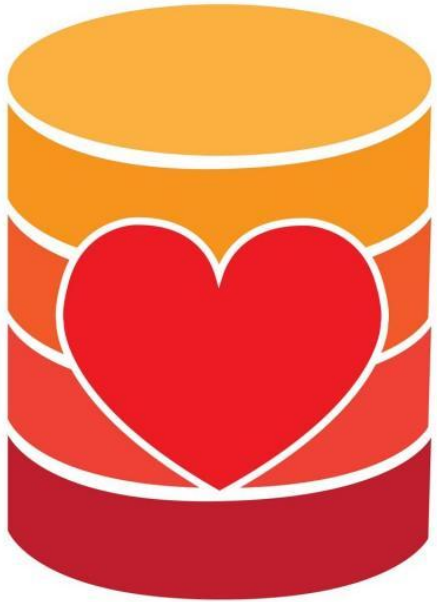

JOBS

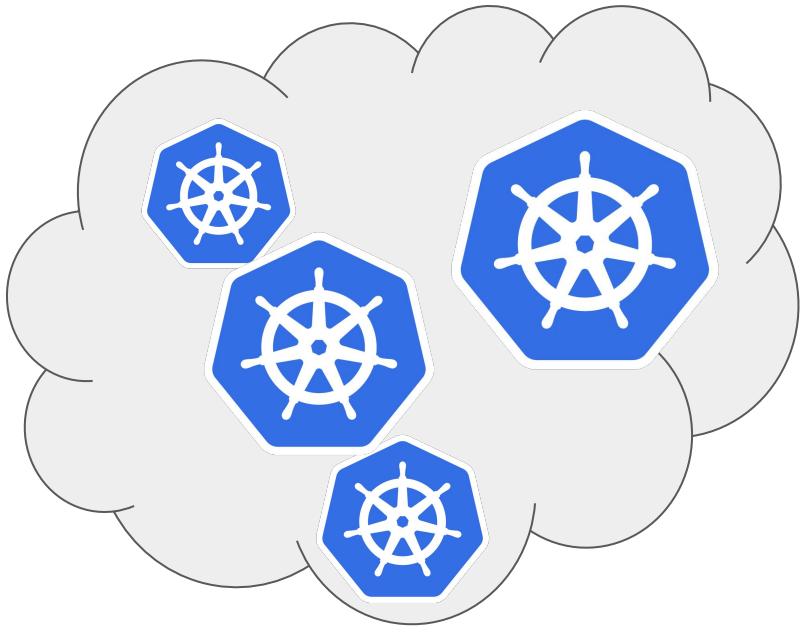
ML

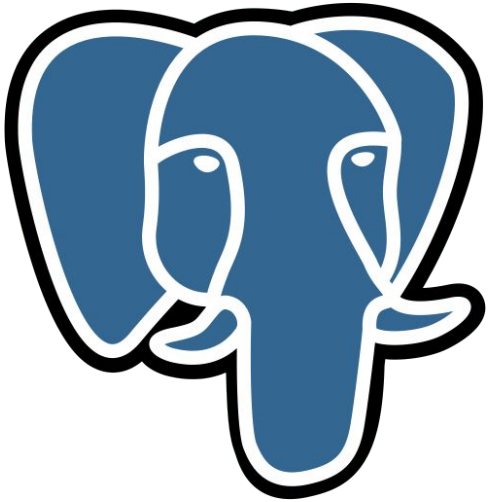

API

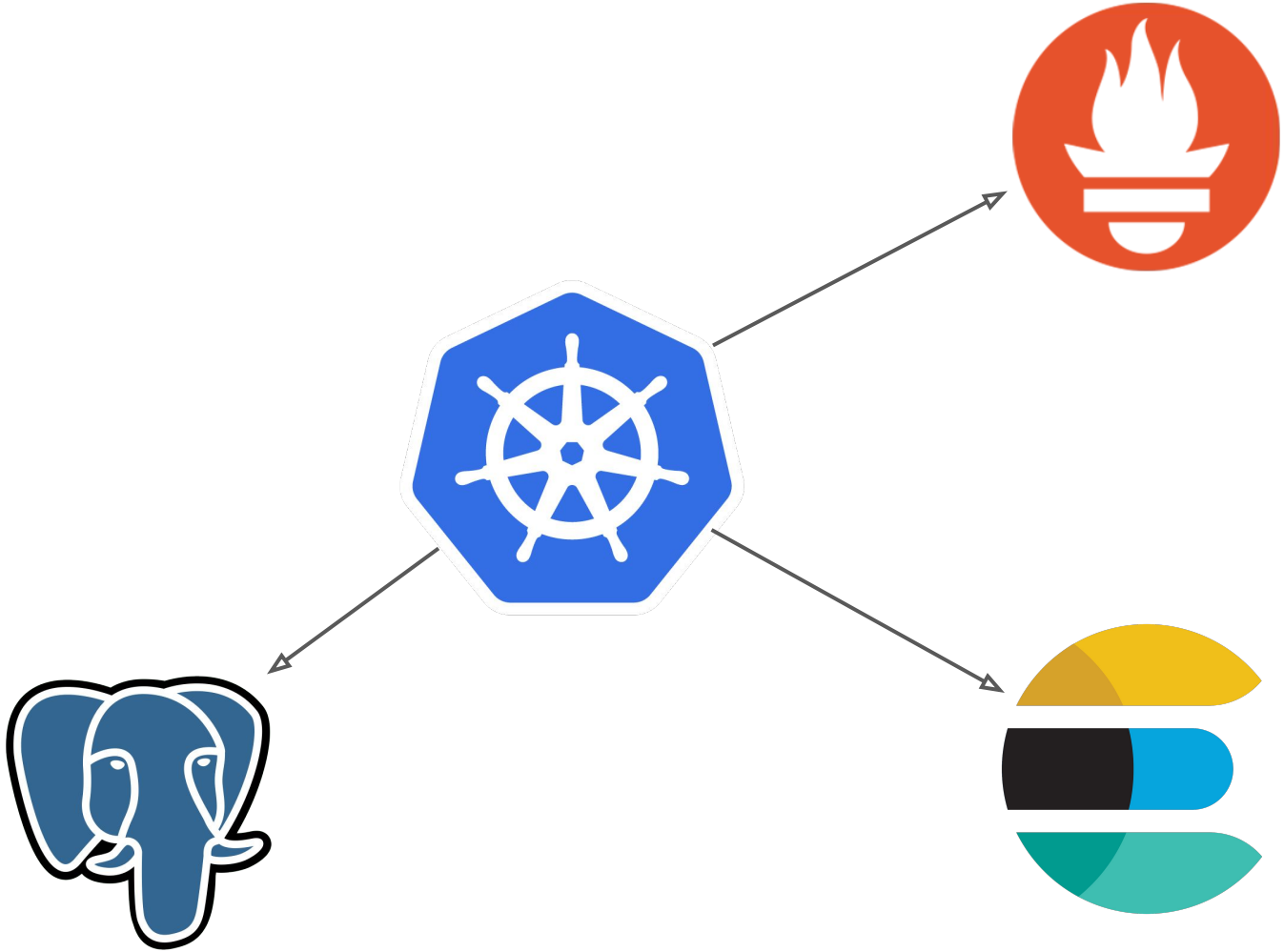
?


API

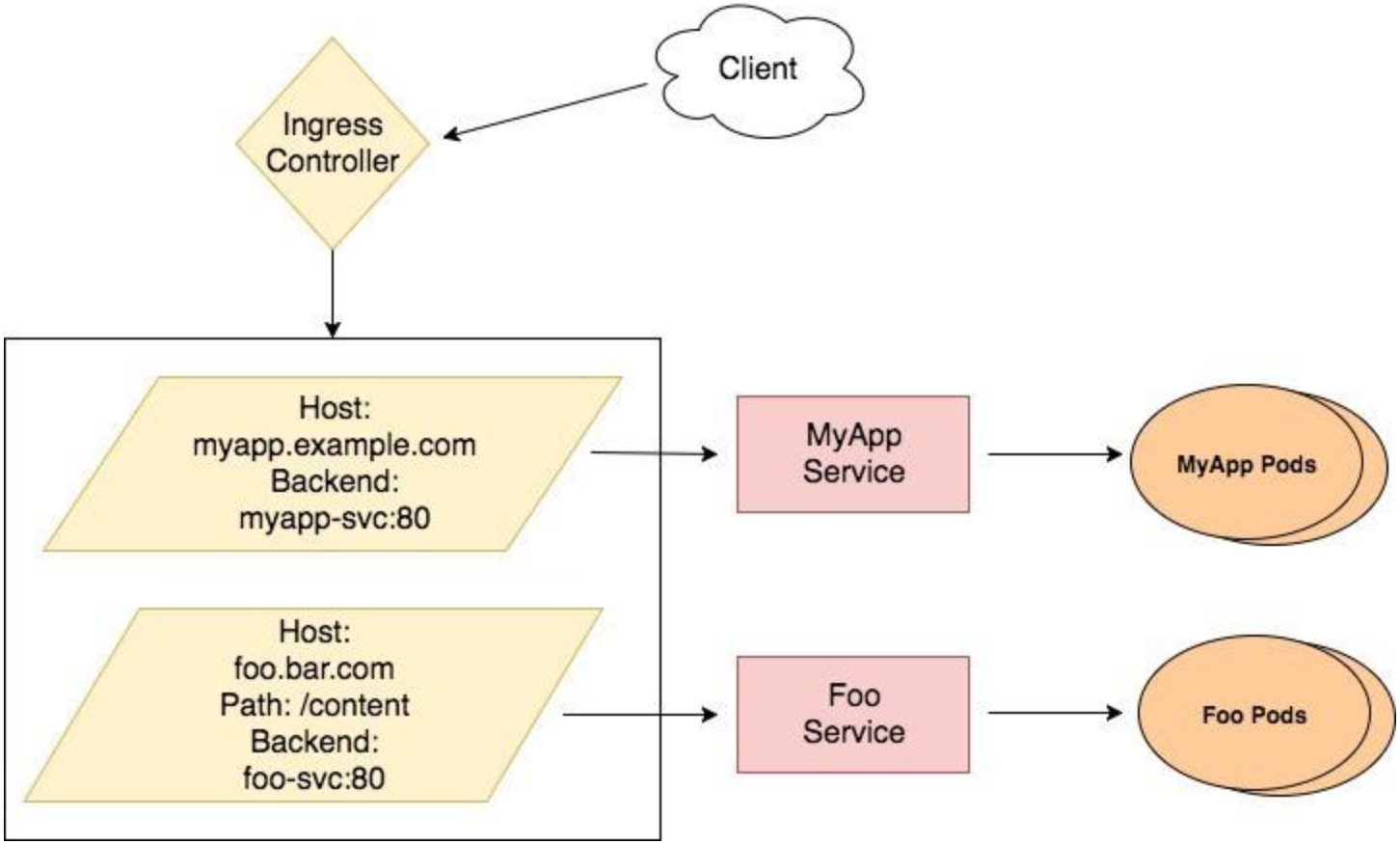


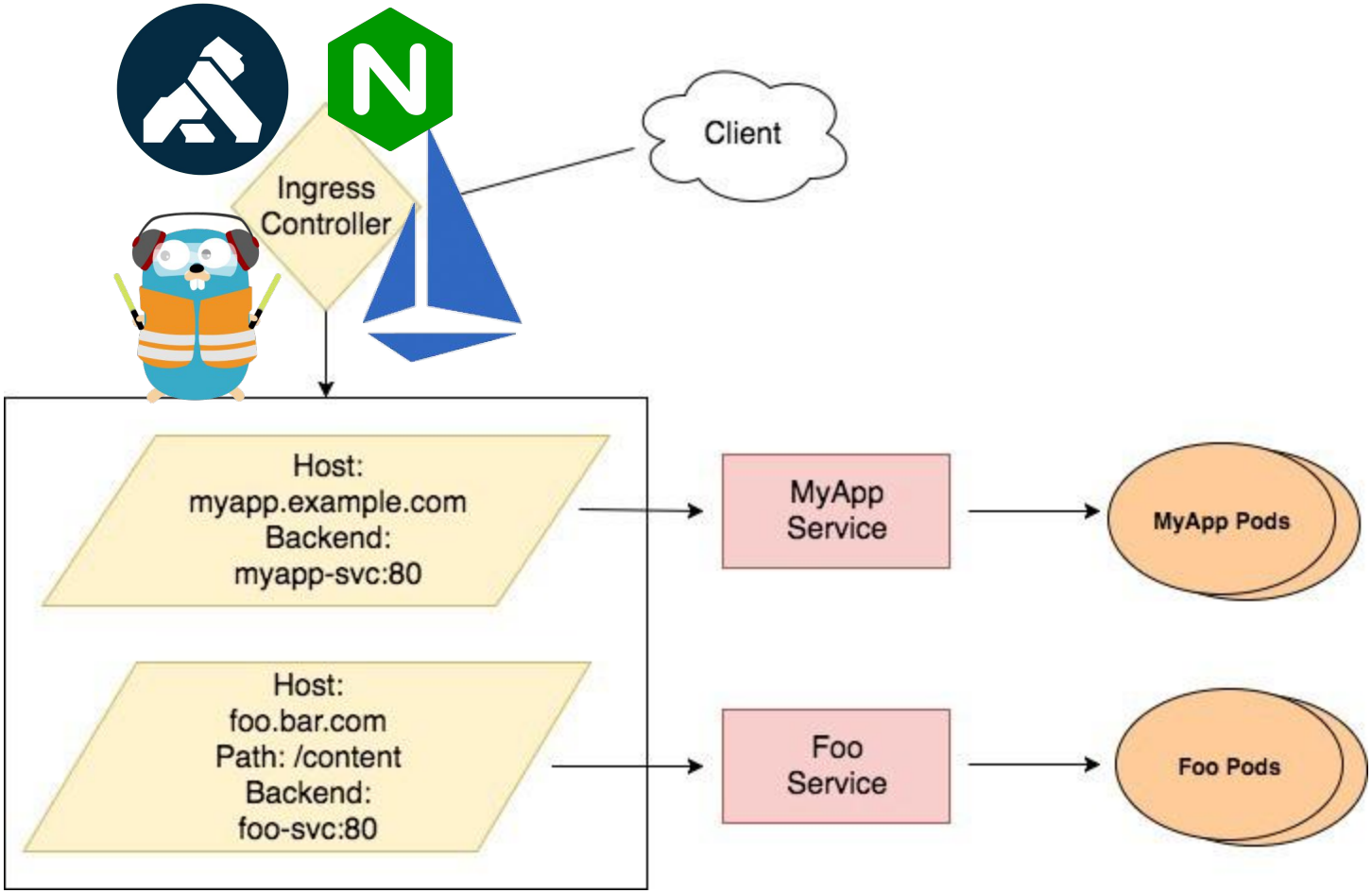


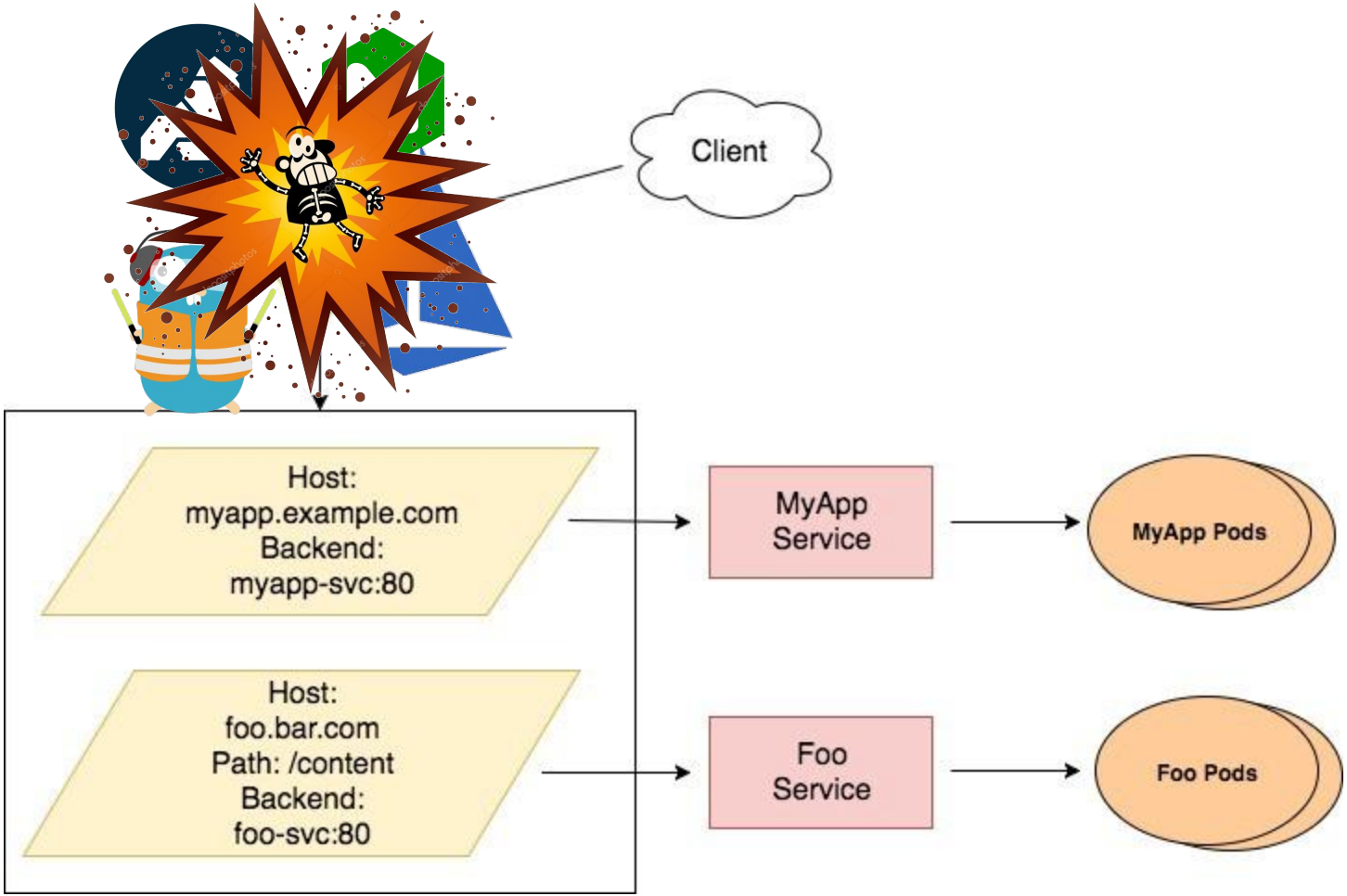


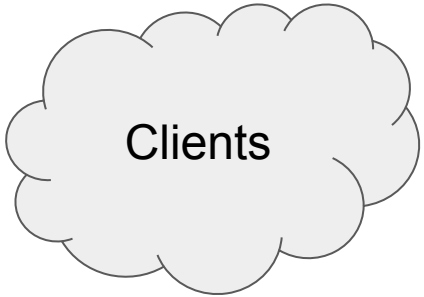


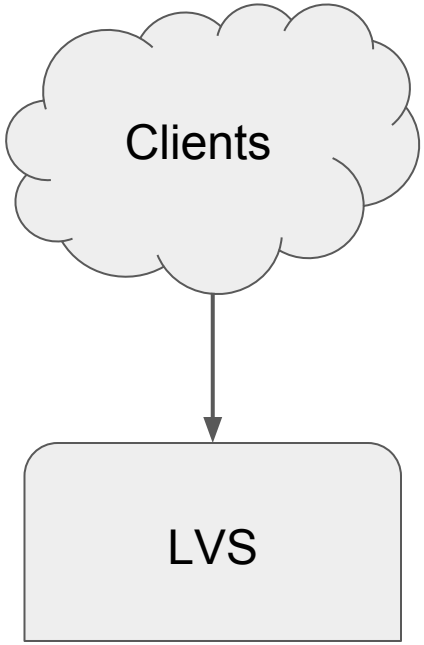
Входящий трафик

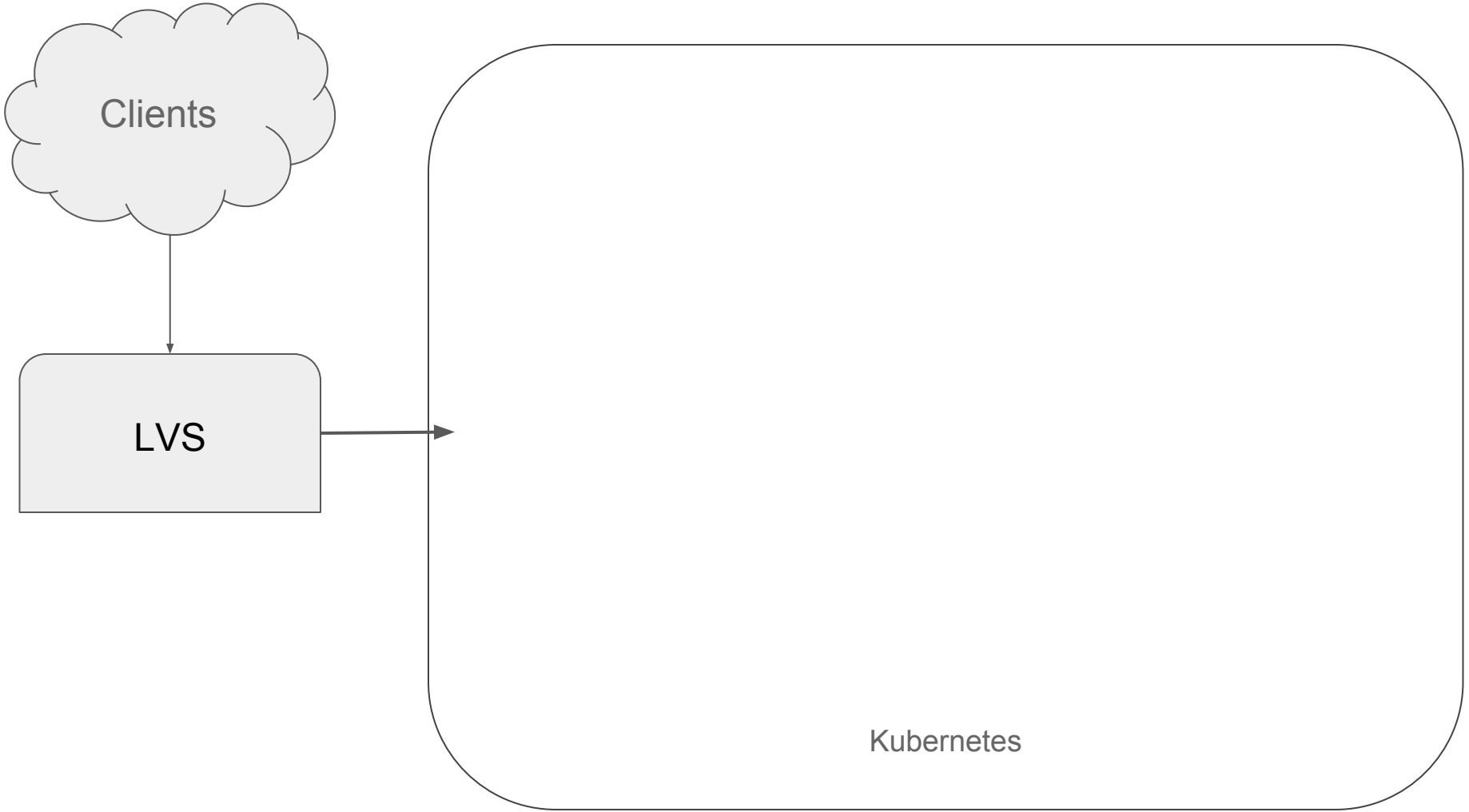

Client
Ingress Controller
Host:
myapp.example.com
Backend:
myapp-svc:80
MyApp Service
MyApp Pods
Host:
foo.bar.com
Path: /content
Backend:
foo-svc:80
Foo Service
Foo Pods

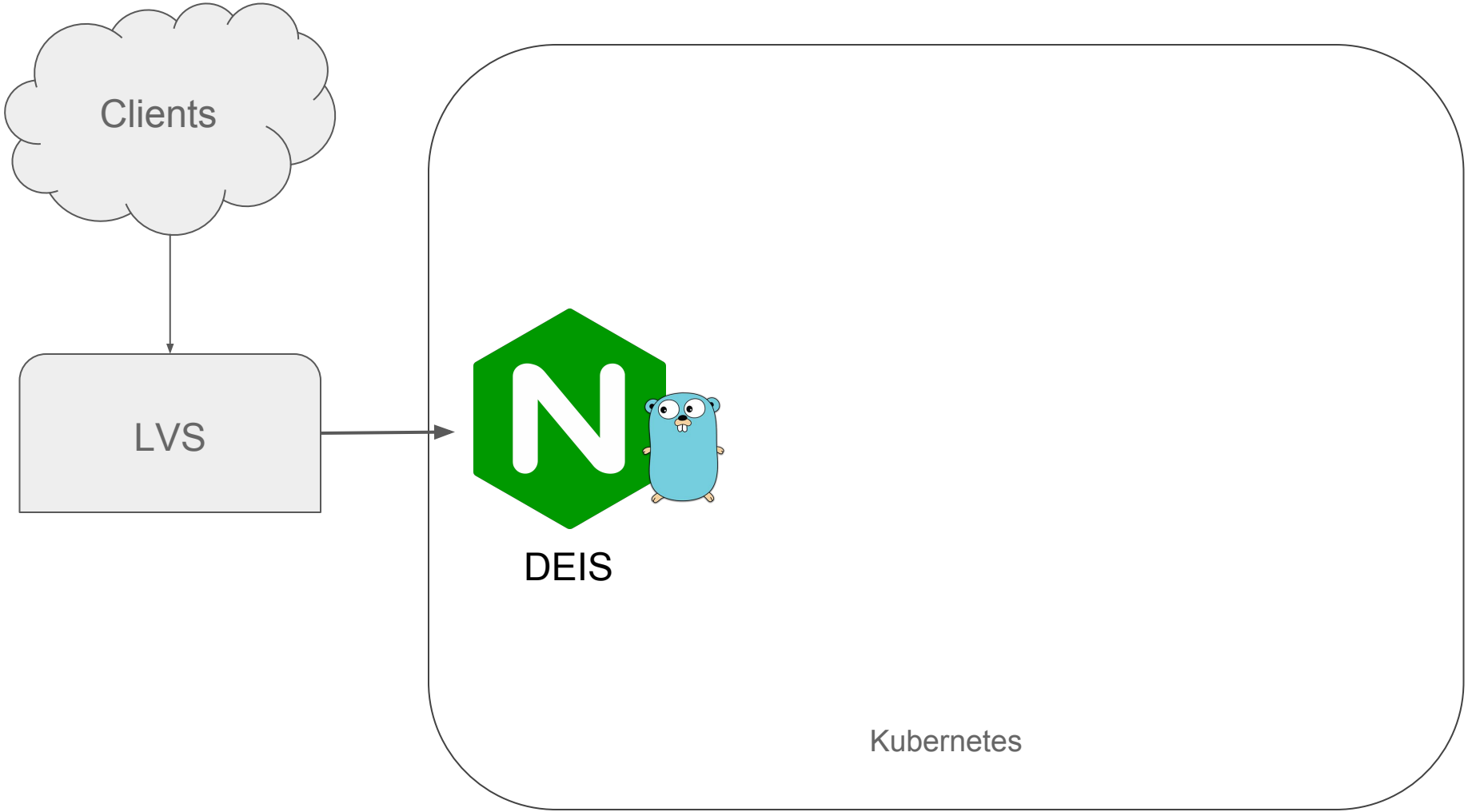

Client
Ingress Controller
Host:
myapp.example.com
Backend:
myapp-svc:80
MyApp Service
MyApp Pods
Host:
foo.bar.com
Path: /content
Backend:
foo-svc:80
Foo Service
Foo Pods

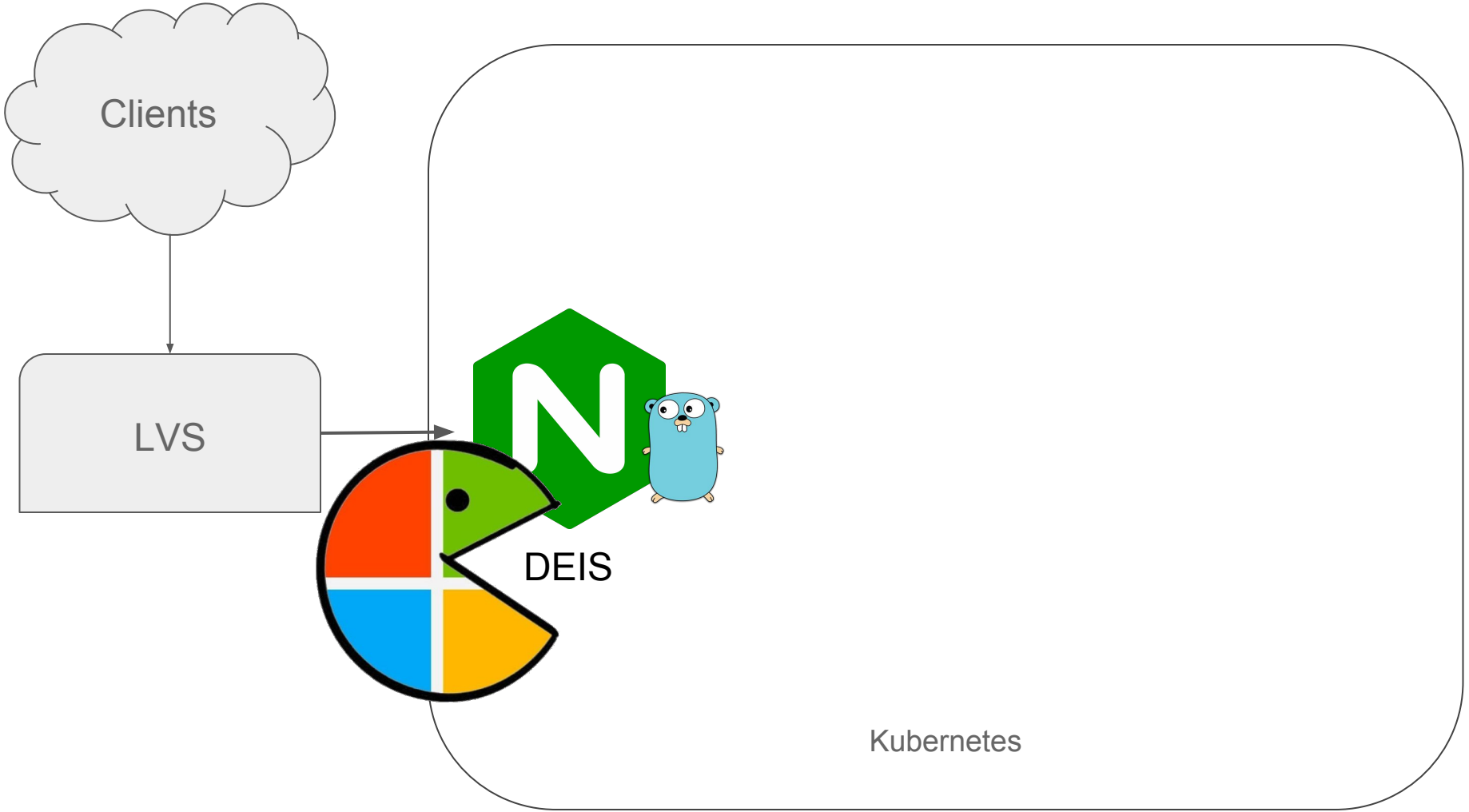

Client
Host:
myapp.example.com
Backend:
myapp-svc:80
MyApp
Service
MyApp Pods
Host:
foo.bar.com
Path: /content
Backend:
foo-svc:80
Foo
Service
Foo Pods

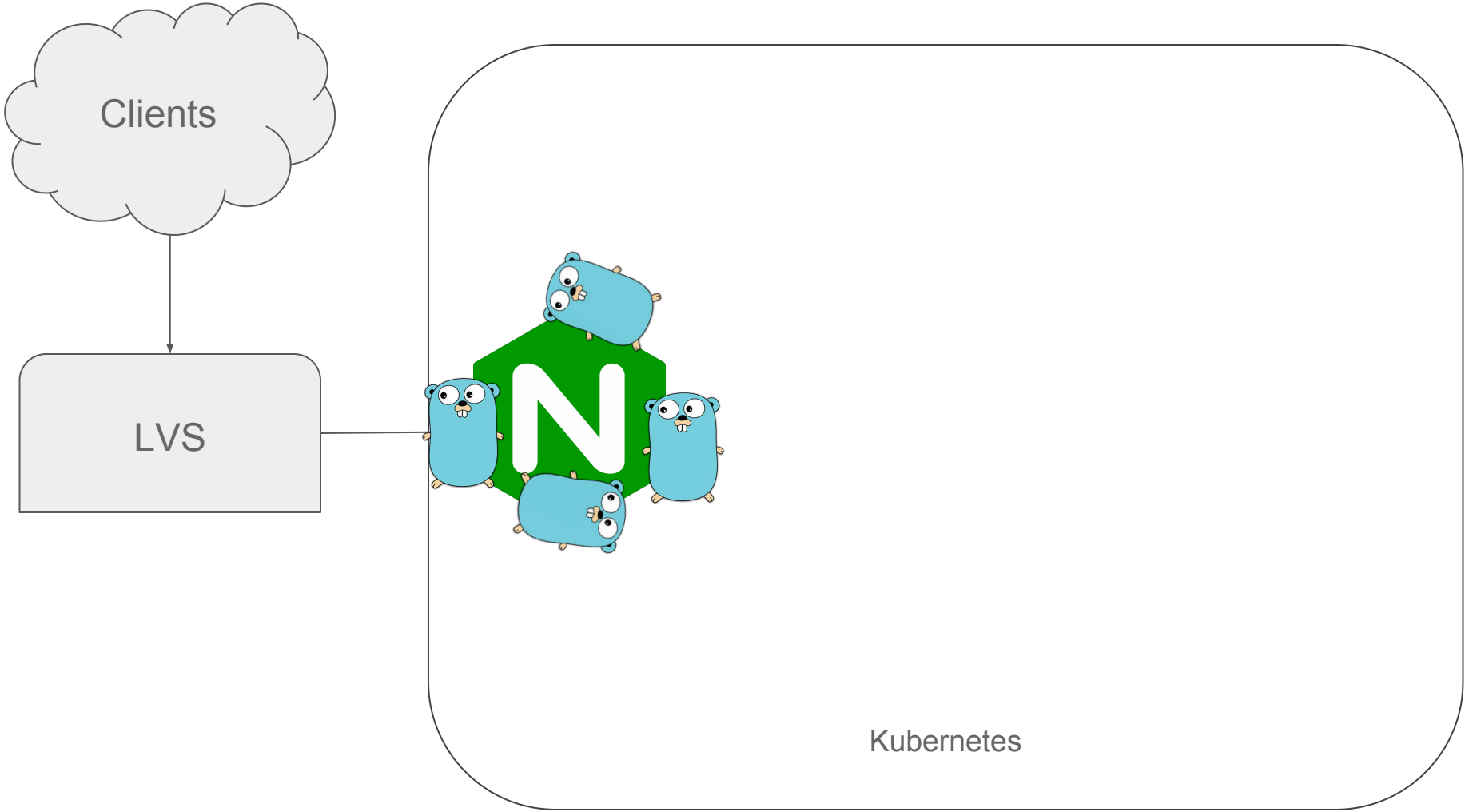

Clients

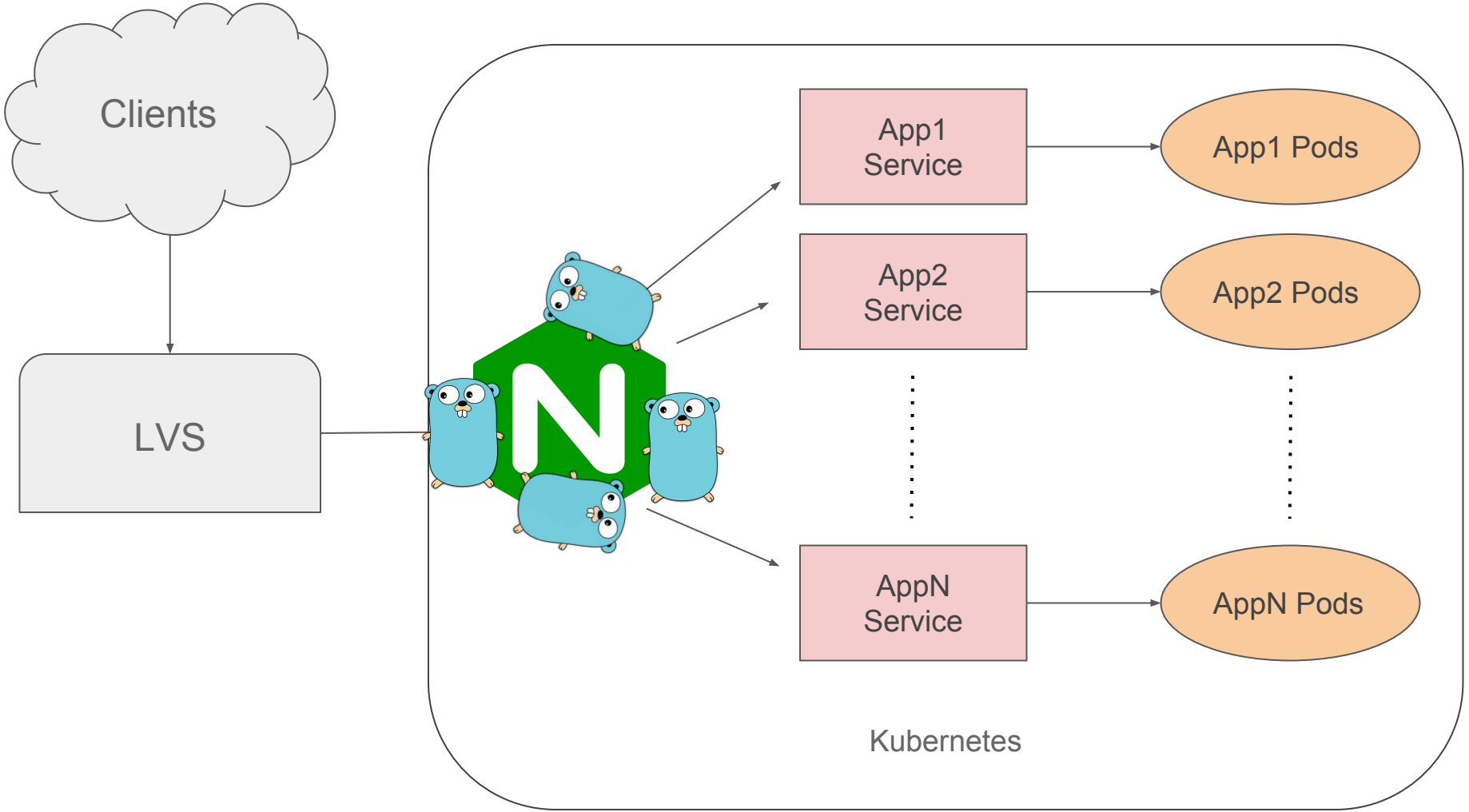

Clients
LVS

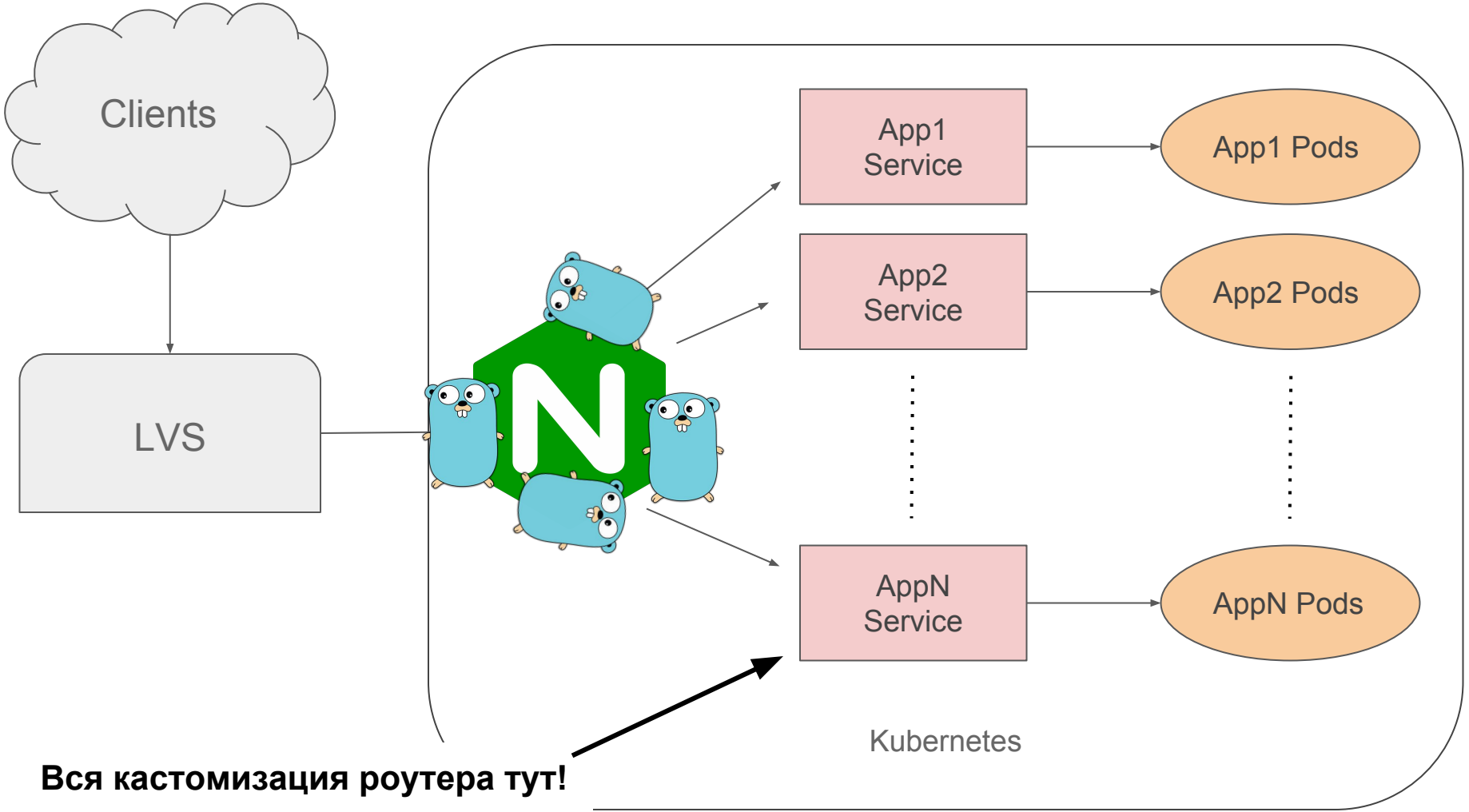

Clients
LVS
Kubernetes

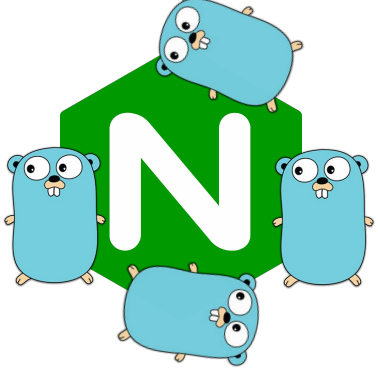

Clients
LVS
DEIS
Kubernetes


Clients
LVS
DEIS
Kubernetes


Clients
LVS
Kubernetes

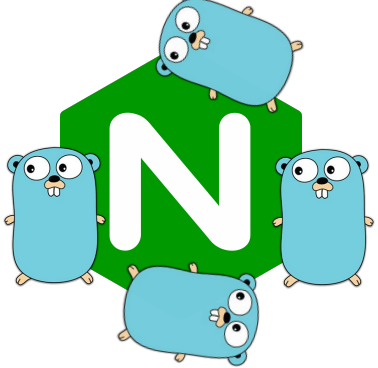

Clients
LVS
App1
Service
App2
Service
AppN
Service
App1 Pods
App2 Pods
AppN Pods
Kubernetes


Clients
LVS
App1
Service
App1 Pods
App2
Service
App2 Pods
AppN
Service
AppN Pods
Kubernetes
Вся кастомизация роутера тут!



Certificates

```
router.deis.io/domains: my-site.2gis.ru
router.deis.io/certificates: my-site.2gis.ru:any-2gis-ru
```

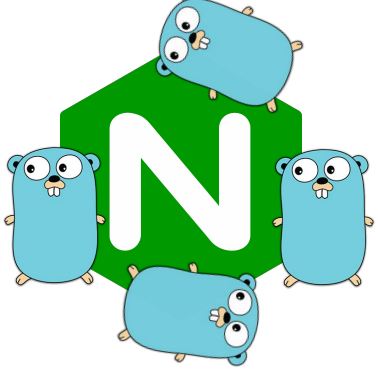


Certificates

```
router.deis.io/domains: my-site.2gis.ru
router.deis.io/certificates: my-site.2gis.ru:any-2gis-ru
```

Rate limit

```
router.deis.io/limitReqServer: 300,50
```



Certificates

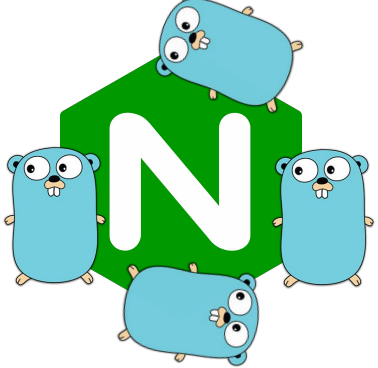
```
router.deis.io/domains: my-site.2gis.ru
router.deis.io/certificates: my-site.2gis.ru:any-2gis-ru
```

Rate limit

```
router.deis.io/limitReqServer: 300,50
```

Black / White lists

```
router.deis.io/blacklist: 123.123.123.123,124.124.124.124
```



Certificates

```
router.deis.io/domains: my-site.2gis.ru
router.deis.io/certificates: my-site.2gis.ru:any-2gis-ru
```

Rate limit

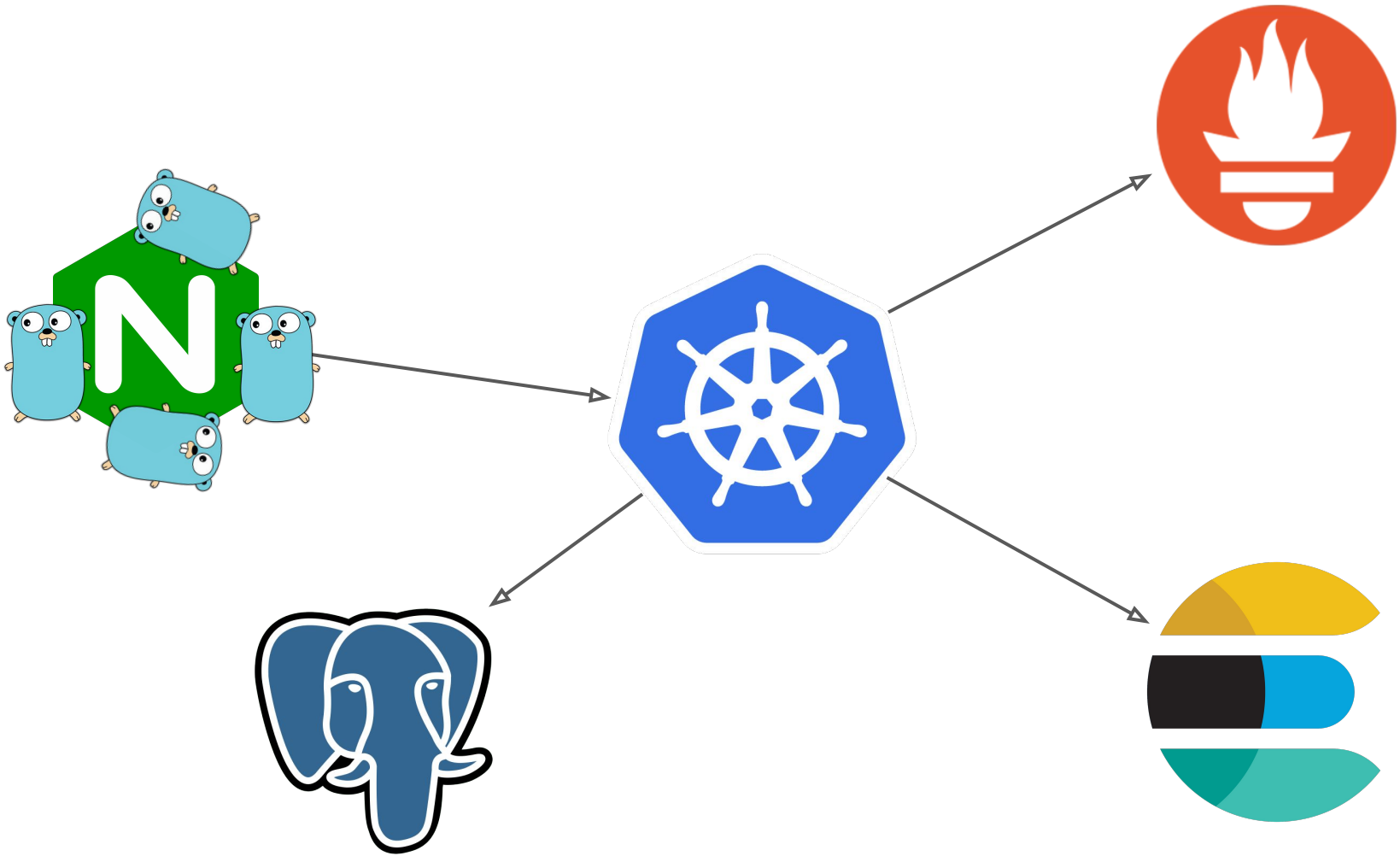
```
router.deis.io/limitReqServer: 300,50
```

Black / White lists

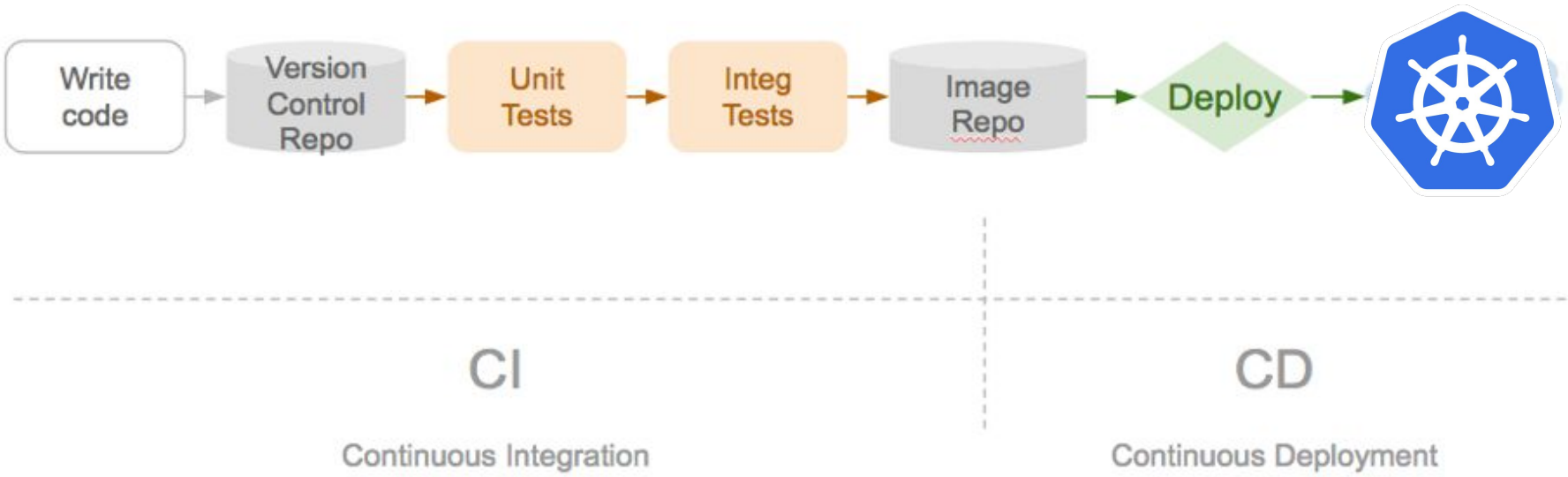
```
router.deis.io/blacklist: 123.123.123.123,124.124.124.124
```

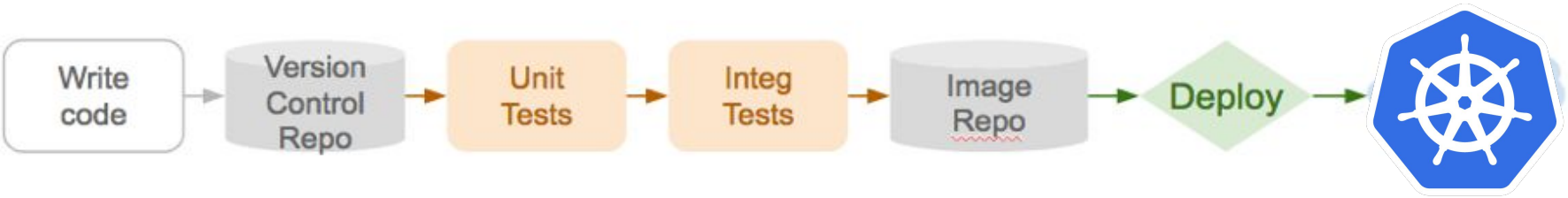
Request-ID

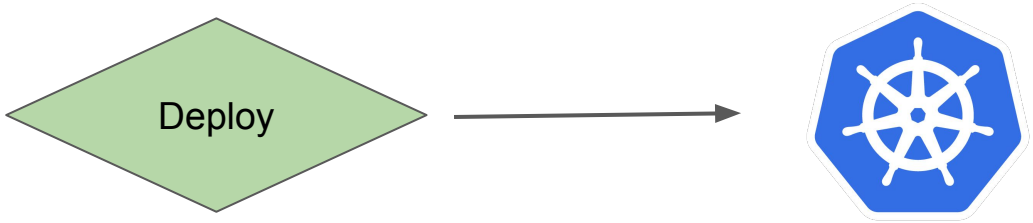
```
$ curl -s test.2gis.ru -I  | grep -i request
X-Request-Id: 5fd3c1051b56bc774908d4563f2d400b
```



CD для k8s


Write code
Version Control Repo
Unit Tests
Integ Tests
Image Repo
Deploy
CI
Continuous Integration
CD
Continuous Deployment


Write code
Version Control Repo
Unit Tests
Integ Tests
Image Repo
Deploy



k8s-handle

k8s-handle is a command line tool that facilitates continuous delivery for Kubernetes applications.

k8s-handle

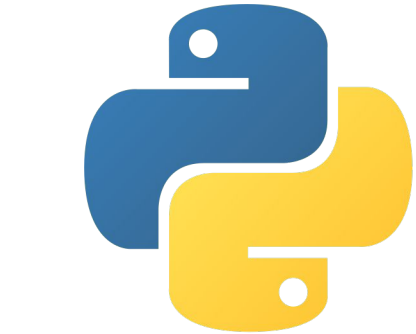
Написан на Python

k8s-handle

Написан на Python

K8s ресурсы:

YAML + Jinja2



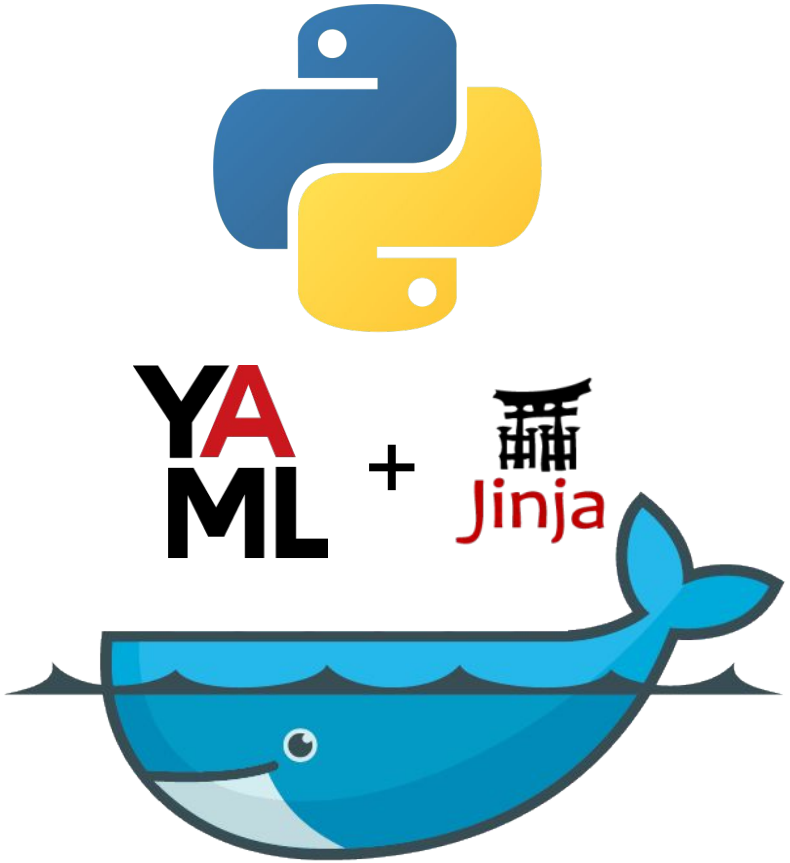
k8s-handle

Написан на Python

K8s ресурсы:

YAML + Jinja2

Поставка в docker images



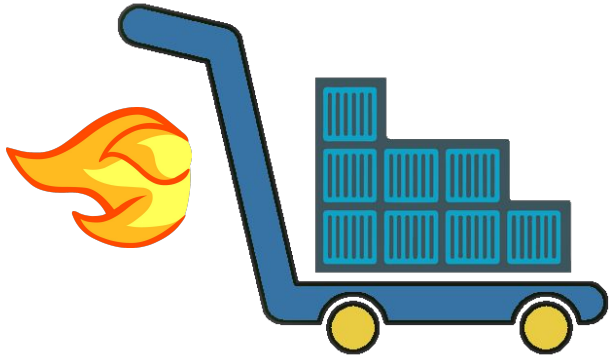
k8s-handle

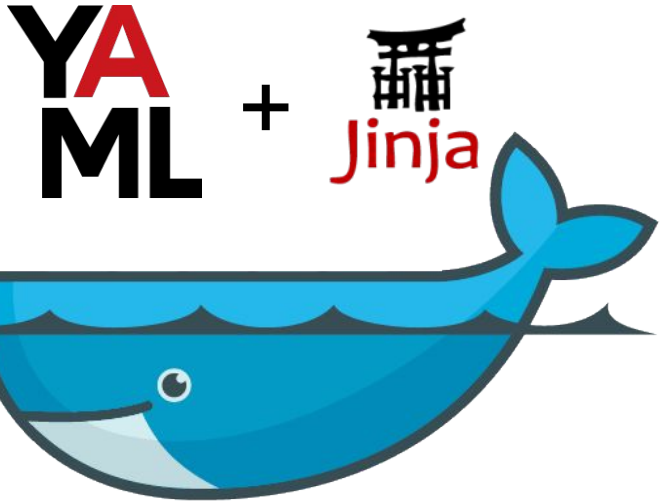
Написан на Python

K8s ресурсы:

YAML + Jinja2

Поставка в docker images





k8s-handle-example

```
$ tree
.
├── config.yaml
└── templates
    ├── deployment.yaml.j2
    └── service.yaml.j2
```

k8s-handle-example

```
$ cat config.yaml
---
common:
  app_name: k8s-handle-example
  app_port: 80

  replicas_count: 1
  image_path: nginx
  image_version: 1.13-alpine

staging:
  templates:
  - template: deployment.yaml.j2
  - template: service.yaml.j2
```

k8s-handle-example

```
$ cat templates/deployment.yaml.j2
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ app_name }}
spec:
  replicas: {{ replicas_count }}
  ...
  template:
    spec:
      containers:
      - name: {{ app_name }}
        image: {{ image_path }}:{{ image_version }}
        ports:
        - containerPort: {{ app_port }}
```

k8s-handle-example

```
$ cat templates/service.yaml.j2
apiVersion: v1
kind: Service
metadata:
  name: {{ app_name }}
spec:
  type: NodePort
  ports:
    - name: http
    port: 80
    targetPort: {{ app_port }}
  selector:
    app: {{ app_name }}
```

k8s-handle-example

```
$ docker run \
            -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
            2gis/k8s-handle \
            k8s-handle deploy -s staging --use-kubeconfig
```

k8s-handle-example

```
$ docker run \
          -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
          2gis/k8s-handle \
          k8s-handle deploy -s staging --use-kubeconfig
```

k8s-handle-example

```
$ docker run \
            -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
            2gis/k8s-handle \
            k8s-handle deploy -s staging --use-kubeconfig
```

k8s-handle-example

```
$ docker run \
            -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
            2gis/k8s-handle \
            k8s-handle deploy -s staging --use-kubeconfig
```

k8s-handle-example

```
$ docker run \
             -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
             2gis/k8s-handle \
             k8s-handle deploy -s staging --use-kubeconfig

INFO:templating:Trying to generate file from template "deployment.yaml.j2" in "/tmp/k8s-handle"
INFO:templating:File "/tmp/k8s-handle/deployment.yaml" successfully generated
INFO:templating:Trying to generate file from template "secret.yaml.j2" in "/tmp/k8s-handle"
INFO:templating:File "/tmp/k8s-handle/service.yaml" successfully generated
```

k8s-handle-example

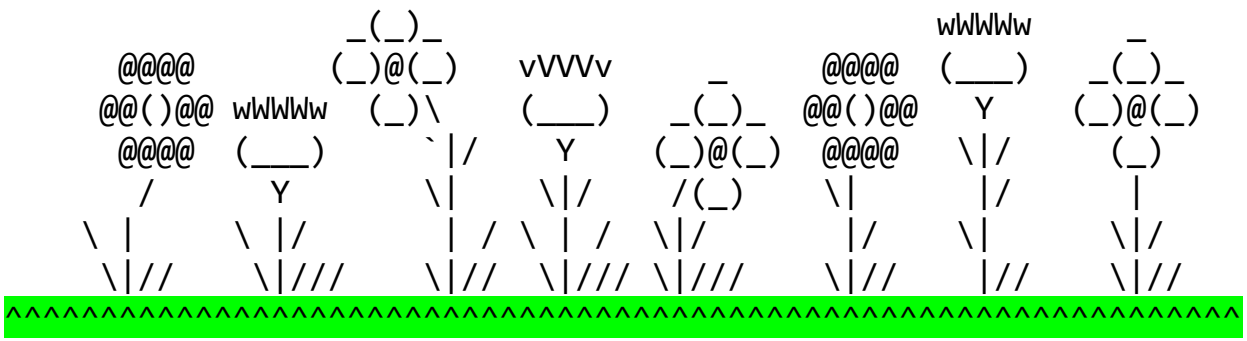
```
$ docker run \
             -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
             2gis/k8s-handle \
             k8s-handle deploy -s staging --use-kubeconfig

INFO:templating:Trying to generate file from template "deployment.yaml.j2" in "/tmp/k8s-handle"
INFO:templating:File "/tmp/k8s-handle/deployment.yaml" successfully generated
INFO:templating:Trying to generate file from template "secret.yaml.j2" in "/tmp/k8s-handle"
INFO:templating:File "/tmp/k8s-handle/service.yaml" successfully generated
INFO:k8s.resource:Deployment "k8s-handle-example" does not exist, create it
INFO:k8s.resource:Service "k8s-handle-example" does not exist, create it
```

k8s-handle-example

```
$ docker run \
            -v $(pwd):/tmp/ -v "$HOME/.kube:/root/.kube" \
            2gis/k8s-handle \
            k8s-handle deploy -s staging --use-kubeconfig

INFO:templating:Trying to generate file from template "deployment.yaml.j2" in "/tmp/k8s-handle"
INFO:templating:File "/tmp/k8s-handle/deployment.yaml" successfully generated
INFO:templating:Trying to generate file from template "service.yaml.j2" in "/tmp/k8s-handle"
INFO:templating:File "/tmp/k8s-handle/service.yaml" successfully generated
INFO:k8s.resource:Deployment "k8s-handle-example" does not exist, create it
INFO:k8s.resource:Service "k8s-handle-example" does not exist, create it
```

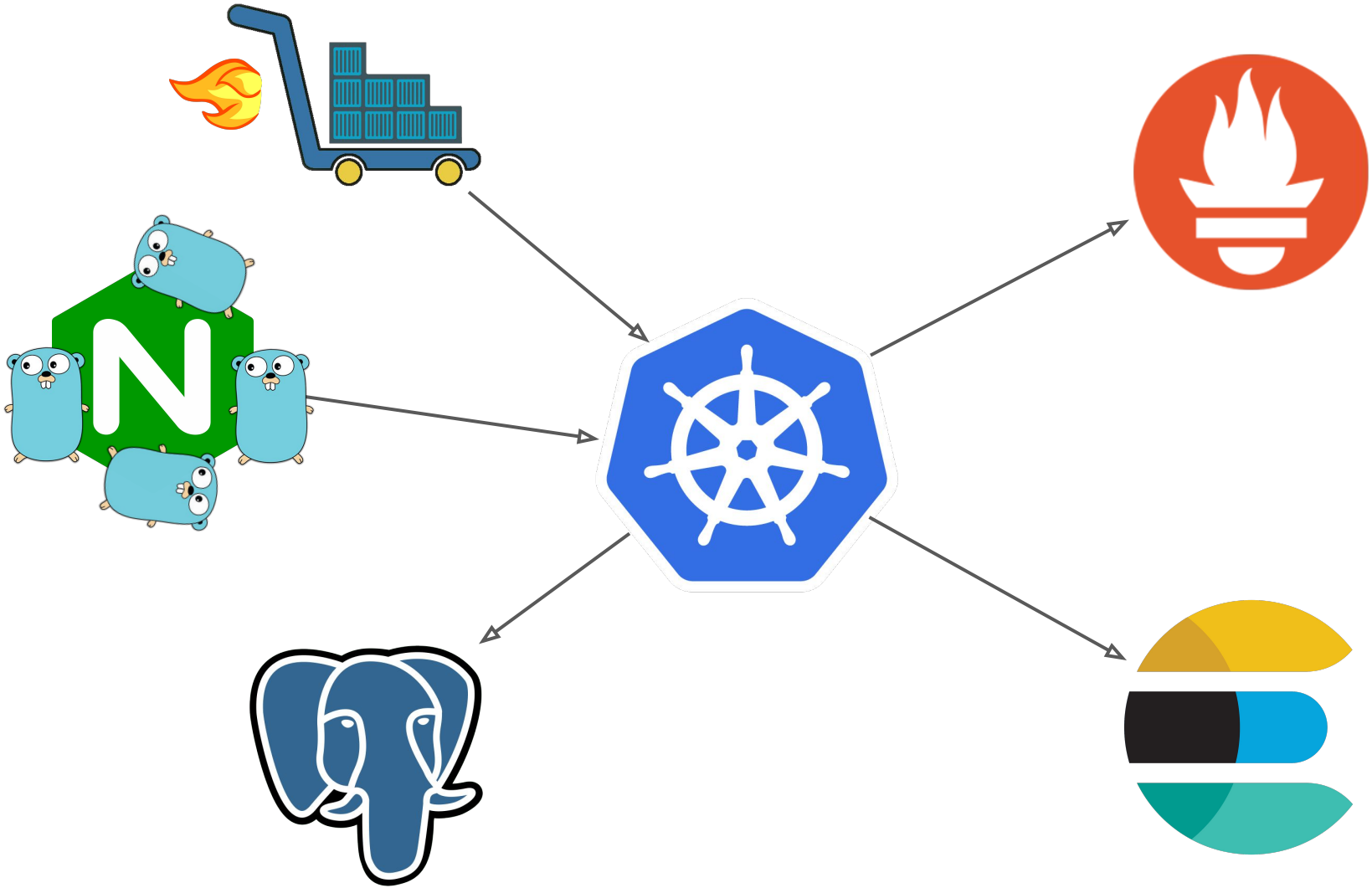


```
$ kubectl get deploy k8s-handle-example
NAME                 DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
k8s-handle-example   1        1        1           1          11s


$ kubectl get service k8s-handle-example
NAME                 TYPE       CLUSTER-IP      EXTERNAL-IP  PORT(S)        AGE
k8s-handle-example   NodePort   10.100.235.27   <none>       80:31524/TCP   12s


$ kubectl get pods -l app=k8s-handle-example
NAME                                    READY   STATUS    RESTARTS   AGE
k8s-handle-example-59bdc9bb6b-248r4     1/1     Running   0          17s


$ kubectl get pods k8s-handle-example-59bdc9bb6b-248r4
-o=custom-columns=NAME:.spec.containers[0].name,IMAGE:.spec.containers[0].image
NAME                 IMAGE
k8s-handle-example   nginx:1.13-alpine
```



Выводы

Kubernetes в вакууме — не решает проблем

Экосистема — обязательна

Удобная экосистема — залог успеха

Спасибо!

Вопросы?

Дехтярёв Евгений
e.dekhtyarev@2gis.ru
github.com/dekhtyarev/devoops-readme

2GIS