

СЛЁРМ

+



Southbridge

# Устройство кластера, сеть, отказоустойчивость

# План

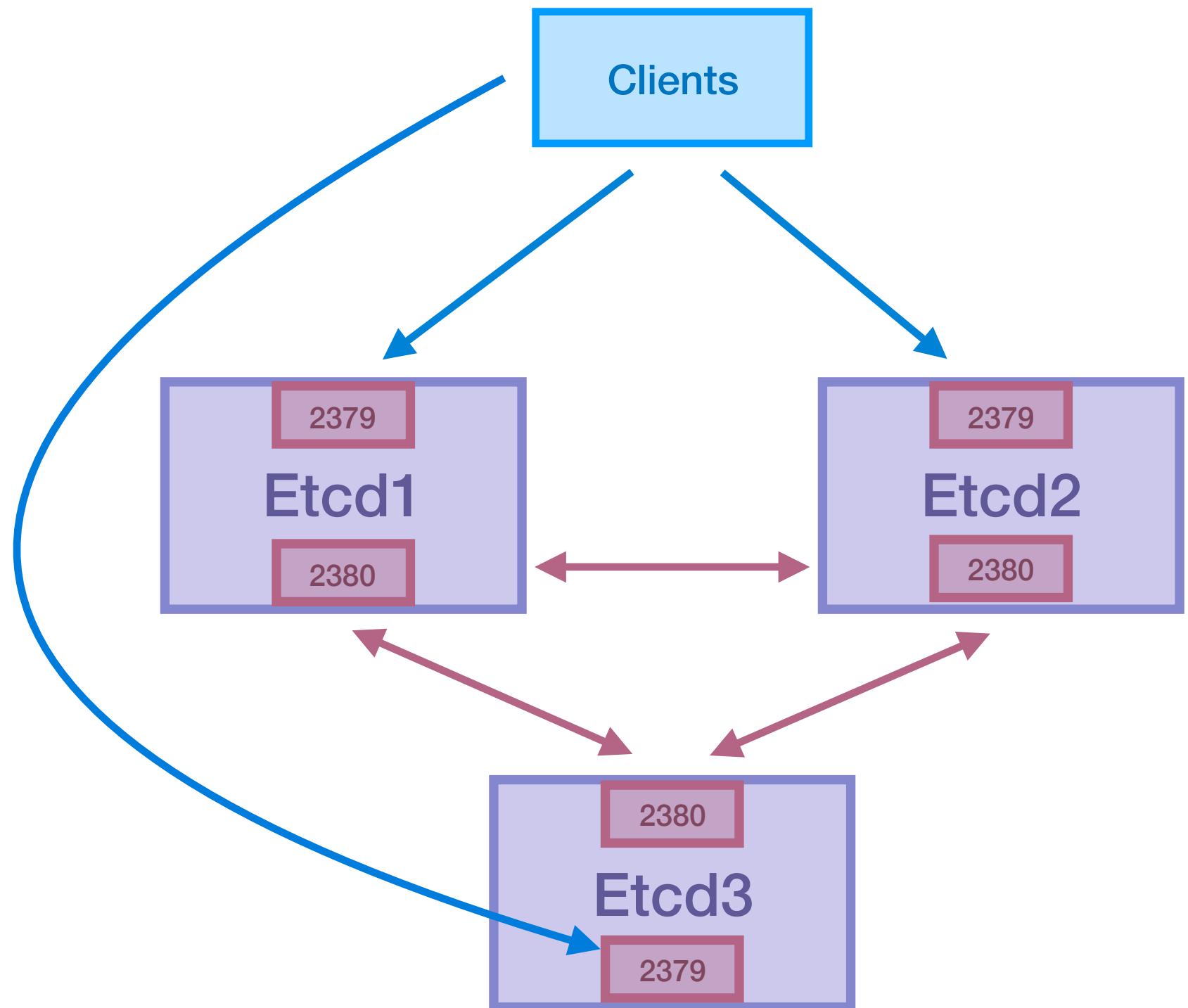
- Компоненты кластера k8s и их взаимодействие
- Сети в Kubernetes
- Отказоустойчивость кластера

# Компоненты кластера

- Etcd
- API server
- Controller-manager
- Scheduler
- Kubelet
- Kube-proxy

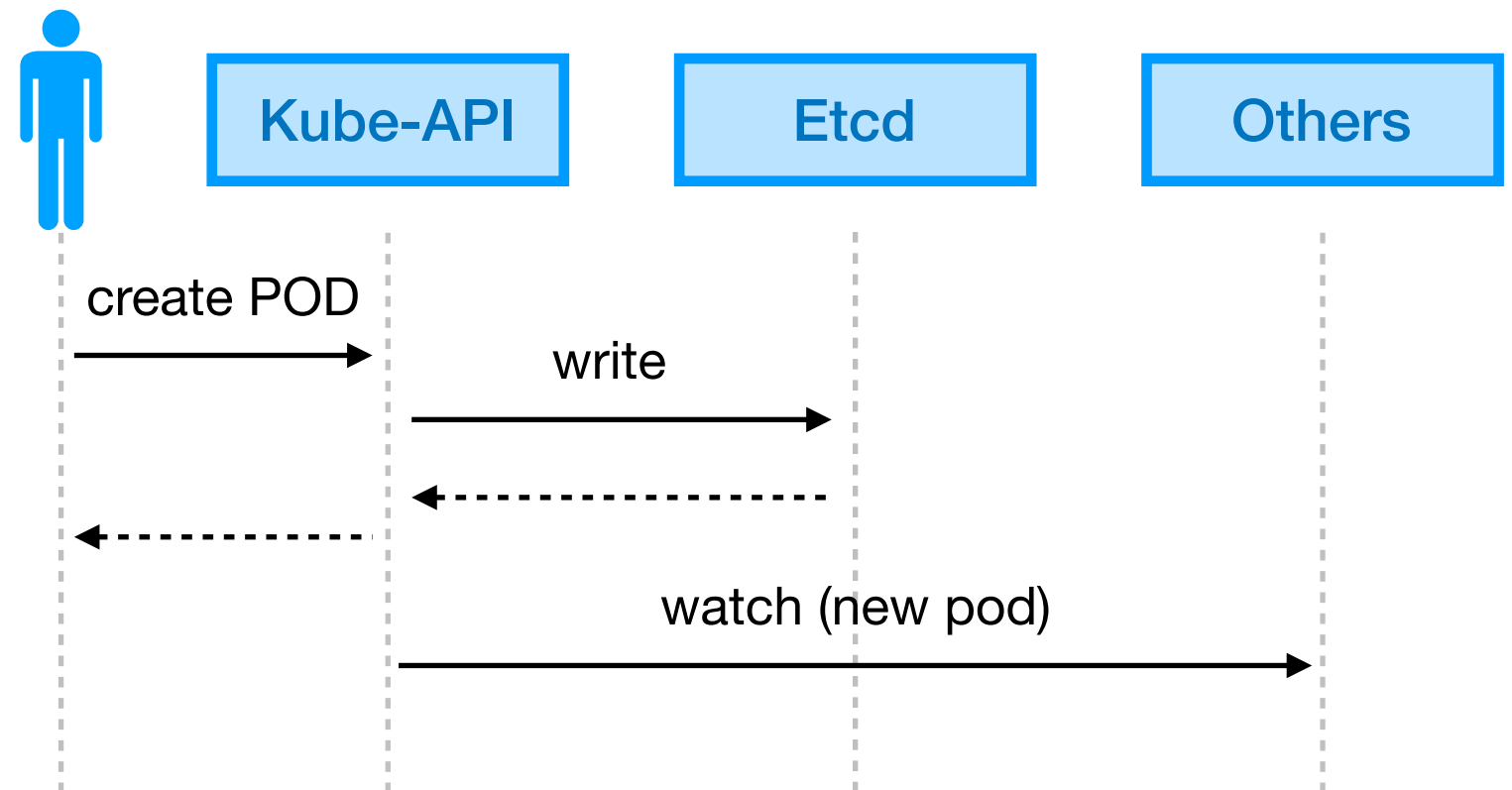
# Etcd

- Хранит всю информацию о кластере
- `etcdctl`
- v2/v3
- Требует быстрых ДИСКОВ



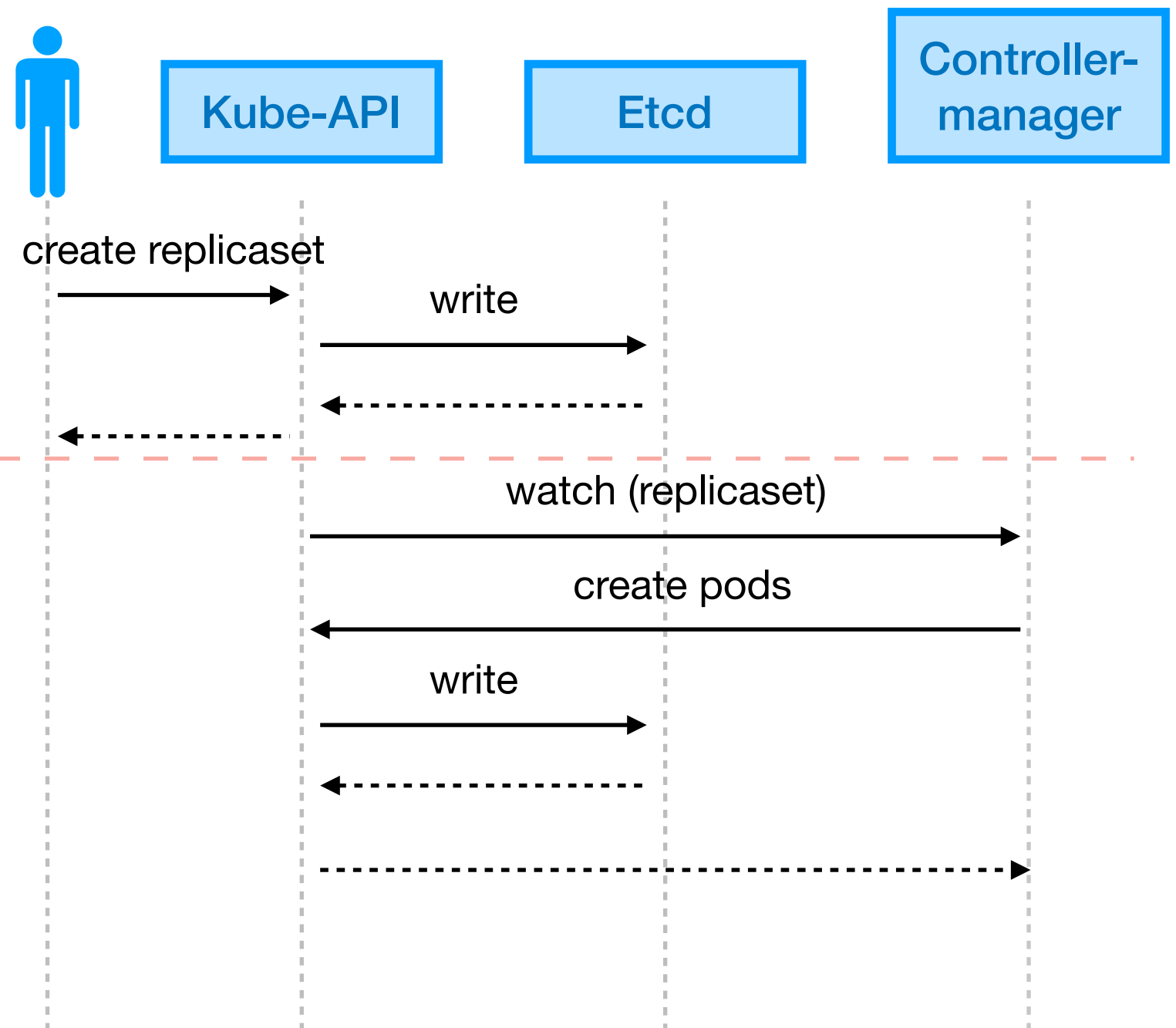
# API server

- Центральный компонент
- Общается с Etcd
- REST
- Authentication and authorization



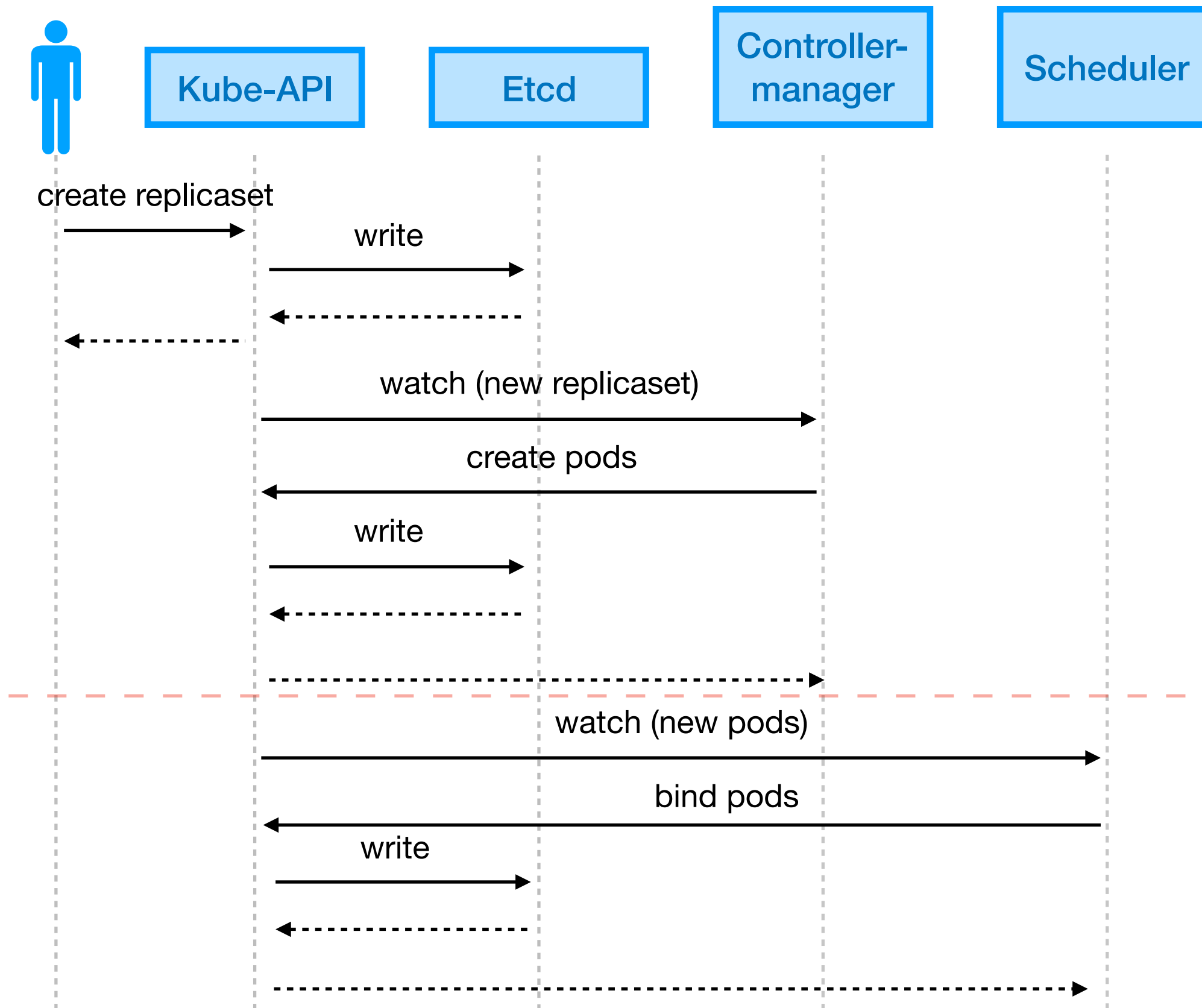
# Controller-manager

- Набор контроллеров
  - Node controller
  - Replication controller
  - Endpoints controller
- GC



# Scheduler

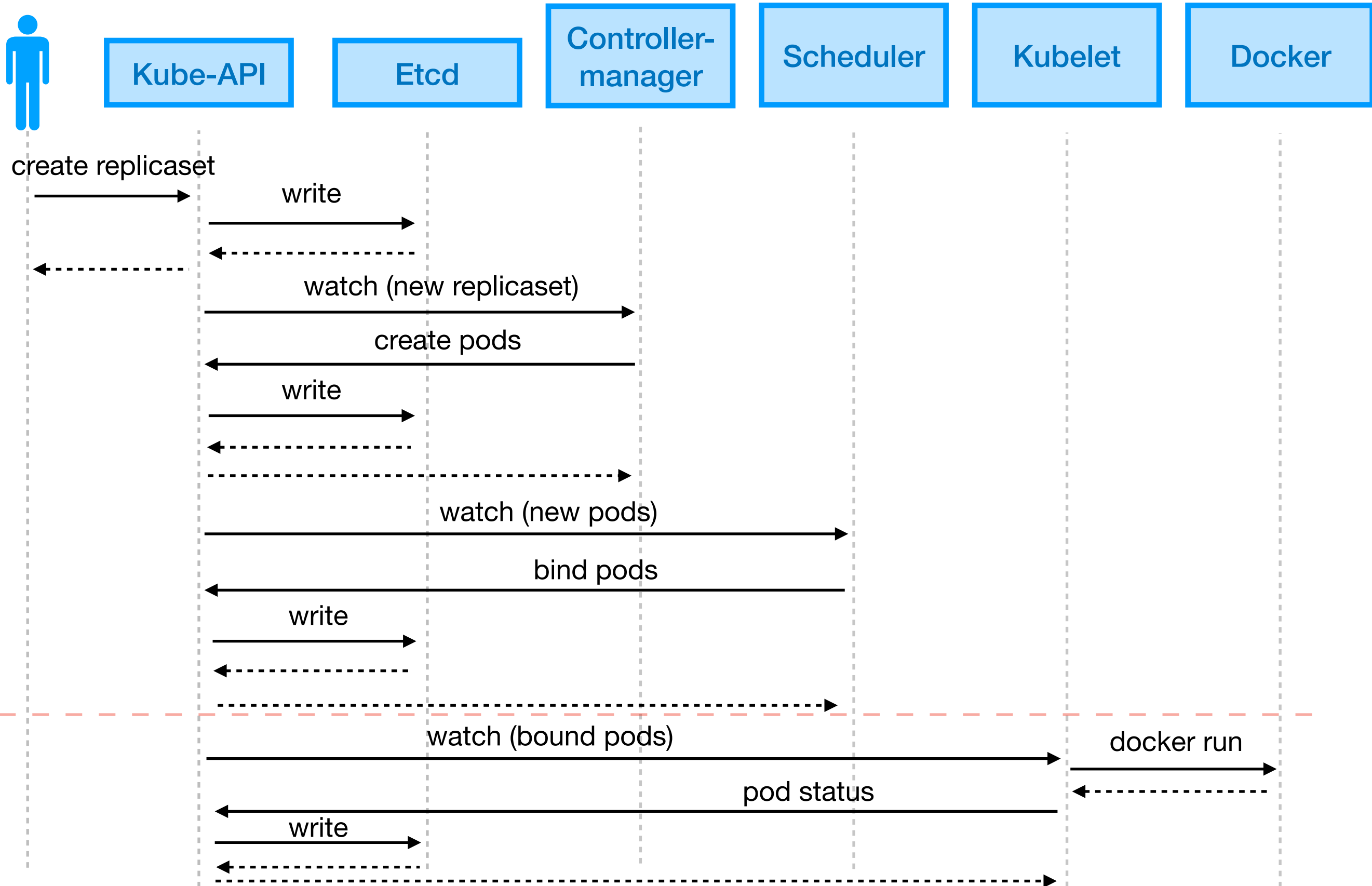
- Назначает PODы на ноды
  - QoS
  - Affinity / anti-affinity
  - Requested resources





# Kubelet

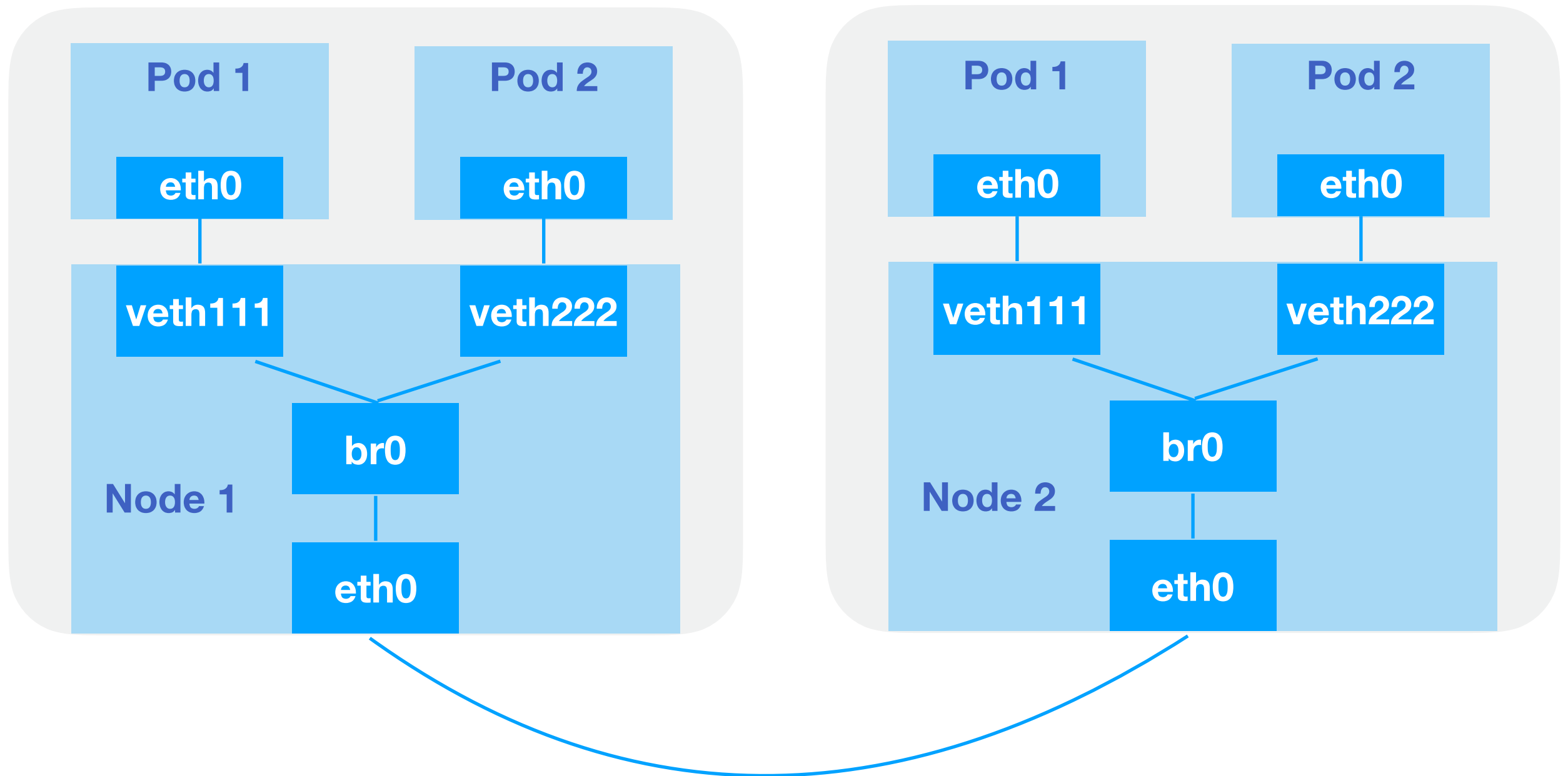
- Работает на каждой ноде
- Единственный компонент работающий не в Docker
- Отдает команды Docker daemon
- Создает PODы



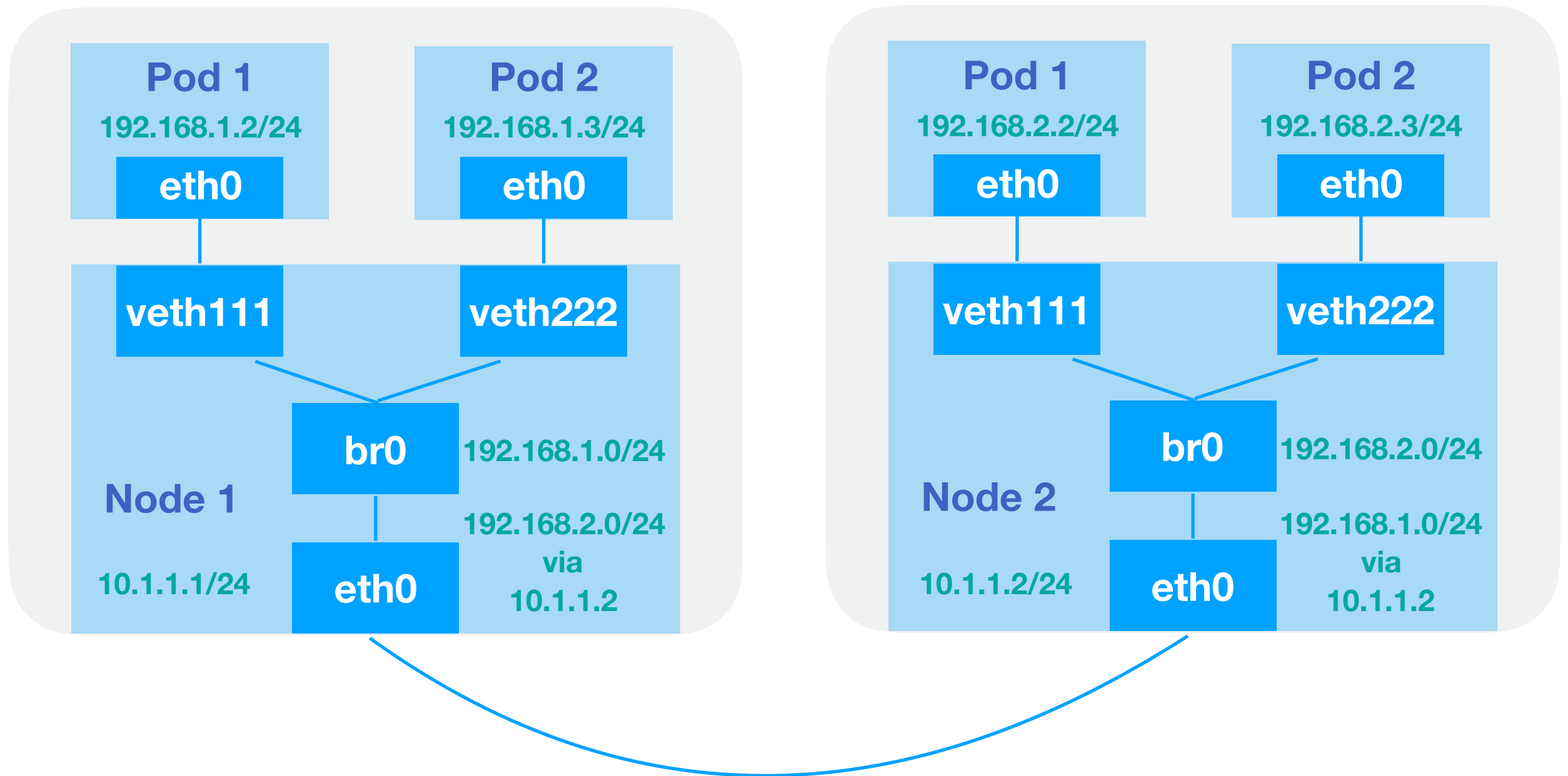
# Kube-proxy

- Управляет сетевыми правилами на нодах
- Фактически реализует Service

# Network



# Flannel [host-gw]



# Service

**-A KUBE-SERVICES**

**-d 1.1.1.1/32**

**-p tcp**

**-m comment --comment "mynamespace/myservice:http cluster IP"**

**-m tcp --dport 80**

**-j KUBE-SVC-UT6A43GJFBEDBO3V**

# Service

**-A KUBE-SERVICES**

**-d 1.1.1.1/32**

**-p tcp**

**-m comment --comment "mynamespace/myservice:http cluster IP"**

**-m tcp --dport 80**

**-j KUBE-SVC-UT6A43GJFBEDBO3V**

**-A KUBE-SVC-UT6A43GJFBEDBO3V**

**-m comment --comment "mynamespace/myservice:http"**

**-m statistic**

**--mode random --probability 0.500000000000**

**-j KUBE-SEP-MMYWB6DZJI4RZ5CQ**

**-A KUBE-SVC-UT6A43GJFBEDBO3V**

**-m comment --comment "mynamespace/myservice:http"**

**-j KUBE-SEP-J33LX377GA3DLDWM**

# Service

**-A KUBE-SVC-UT6A43GJFBEDBO3V**

**-m comment --comment "mynamespace/myservice:http"**

**-m statistic**

**--mode random --probability 0.500000000000**

**-j KUBE-SEP-MMYWB6DZJI4RZ5CQ**

**-A KUBE-SEP-MMYWB6DZJI4RZ5CQ**

**-p tcp**

**-m comment --comment "mynamespace/myservice:http"**

**-m tcp**

**-j DNAT**

**--to-destination 10.102.0.93:80**



# Service

```
-A KUBE-SVC-UT6A43GJFBEDBO3V  
  -m comment --comment "mynamespace/myservice:http"  
-j KUBE-SEP-J33LX377GA3DLDWM
```

```
-A KUBE-SEP-J33LX377GA3DLDWM  
  -p tcp  
  -m comment --comment "mynamespace/myservice:http"  
  -m tcp  
-j DNAT  
  --to-destination 10.102.3.49:80
```

# Service

```
$ kubectl get po --namespace=mynamespace -o wide
```

pod-1	1/1	Running	0	6h	10.102.3.49
pod-2	1/1	Running	0	6h	10.102.0.93

# Service

- Статический IP
- DNS имя в kube-dns на этот IP
- Правила iptables для роутинга

# Ingress

```
$ kubectl exec --namespace kube-system nginx-ingress-controller-2536117660-1qmz3  
cat /etc/nginx/nginx.conf
```

```
upstream mynamespace-myservice-80 {
```

```
    # Load balance algorithm; empty for round robin, which is the default  
    least_conn;
```

```
    keepalive 32;
```

```
    server 10.102.3.49:80 max_fails=0 fail_timeout=0;
```

```
    server 10.102.0.93:80 max_fails=0 fail_timeout=0;
```

```
}
```

# Ingress

```
$ kubectl exec --namespace kube-system nginx-ingress-controller-2536117660-1qmz3  
cat /etc/nginx/nginx.conf
```

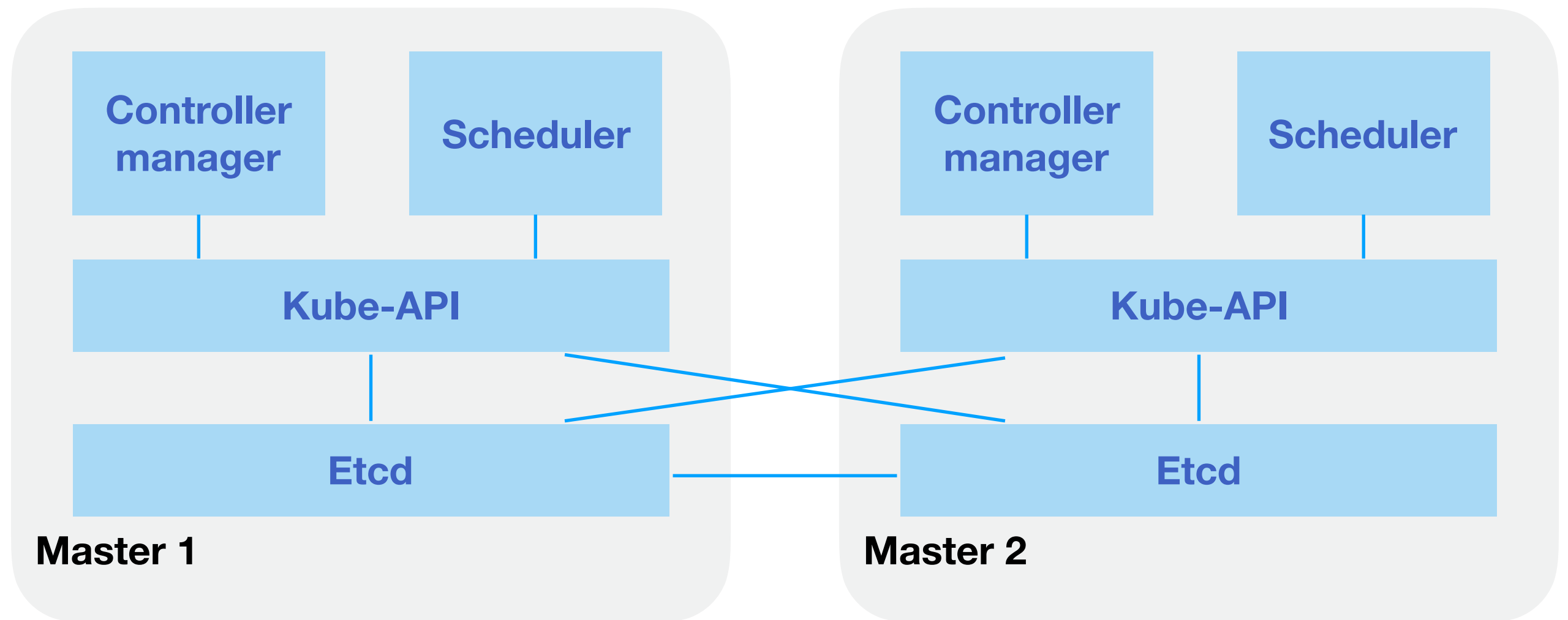
```
server {  
    server_name my.host.foo ;  
    listen 80;  
    location / {  
        set $proxy_upstream_name "mynamespace-myservice-80";  
        proxy_pass http://mynamespace-myservice-80;  
    }  
}
```

# Ingress

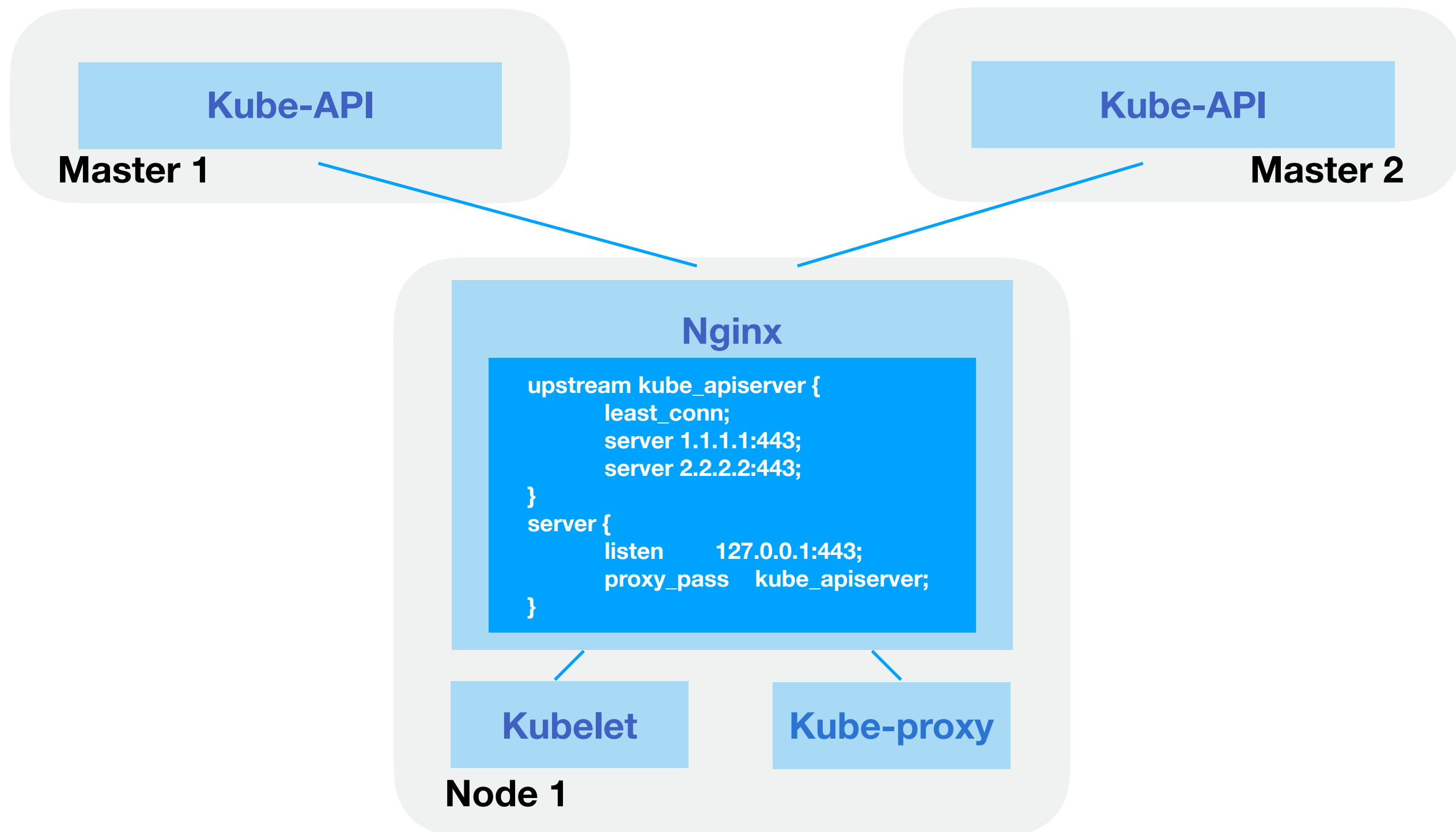
```
$ kubectl exec --namespace kube-system nginx-ingress-controller-2536117660-1qmz3  
cat /etc/nginx/nginx.conf
```

```
log_format upstreaminfo '$the_real_ip - [$the_real_ip] - $remote_user [$time_local]  
"$request" $status $body_bytes_sent "$http_referer" "$http_user_agent"  
$request_length $request_time [$proxy_upstream_name] $upstream_addr  
$upstream_response_length $upstream_response_time $upstream_status';
```

# Отказоустойчивый сетяп мастеров



# Взаимодействие НОДЫ -> мастера





СЛЁРМ

+



Southbridge

# Перерыв