Автомасштабирование в kubernetes

План

1) Когда нужно масштабирование

2) Архитектура масштабируемого приложения

3) Docker и K8s

4) Масштабирование приложения в K8s

5) Масштабирование K8s в облаке

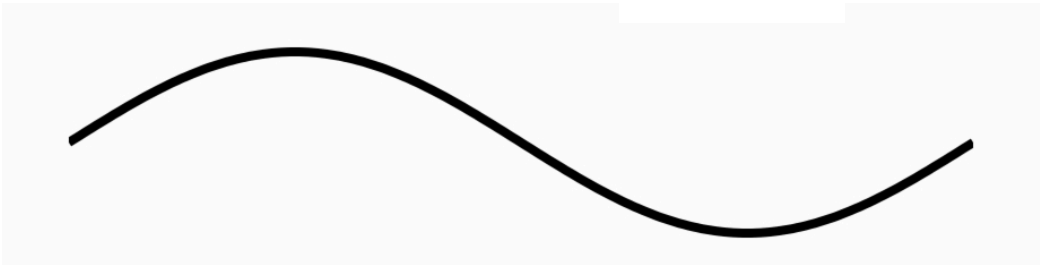
6) Нюансы масштабирования

Автомасштабирование в kubernetes

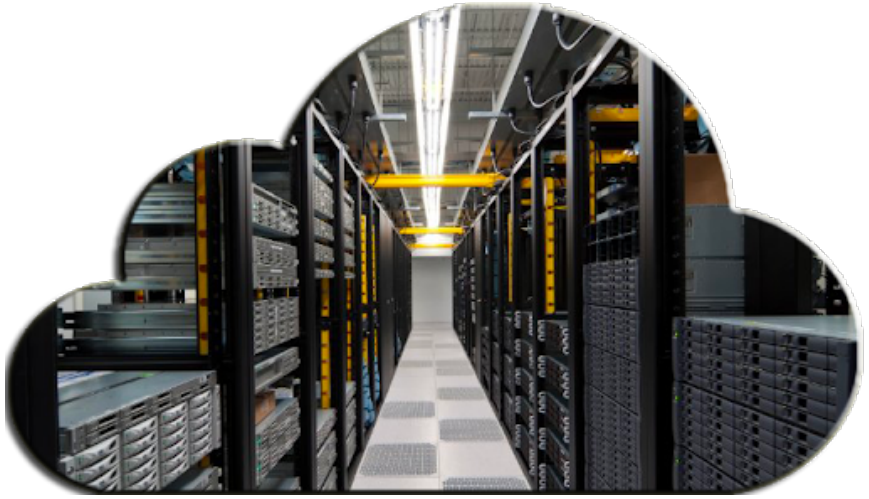
1) Когда нам нужно масштабирование

Стартап

Периодическая нагрузка



Инфраструктура



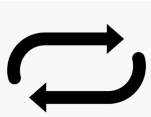
Требования к инфраструктуре

1) Возможность получить огромное (желательно бесконечное) количество ресурсов (ядра, оперативка)

2) Возможность получить эти ресурсы очень быстро

3) Возможность вернуть неиспользуемые ресурсы

4) Возможность распределить приложение в различных аппаратных блоках для отказоустойчивости



Автомасштабирование в kubernetes

---

2) Архитектура приложения

Пример архитектуры

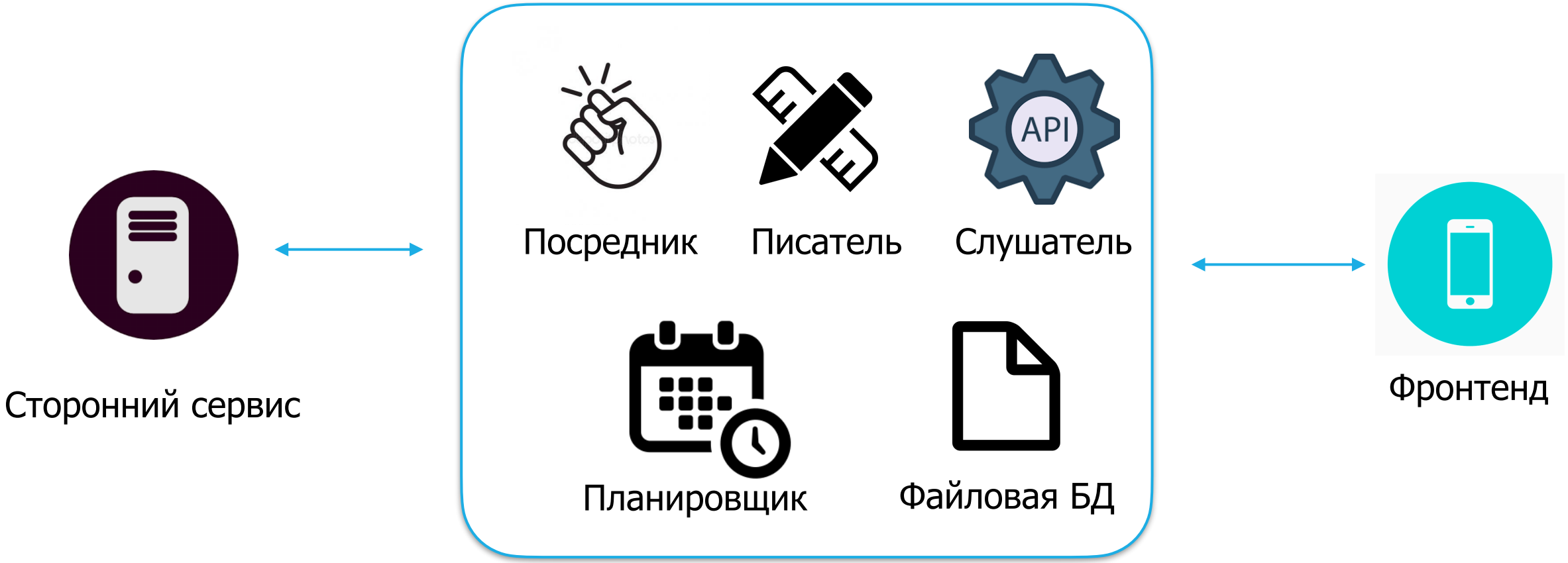
Сторонний сервис

Бэкенд

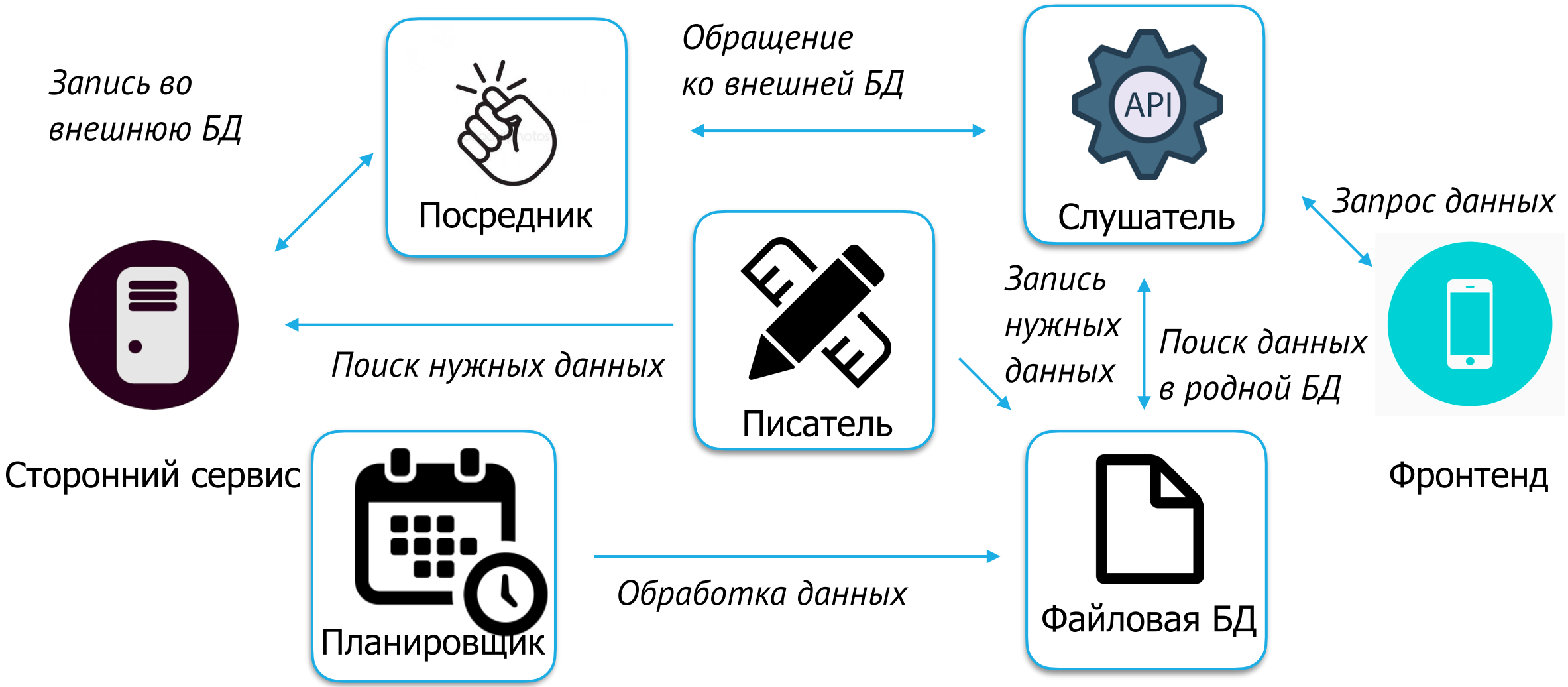
Фронтенд

Хранилище Сервер Имя:порт

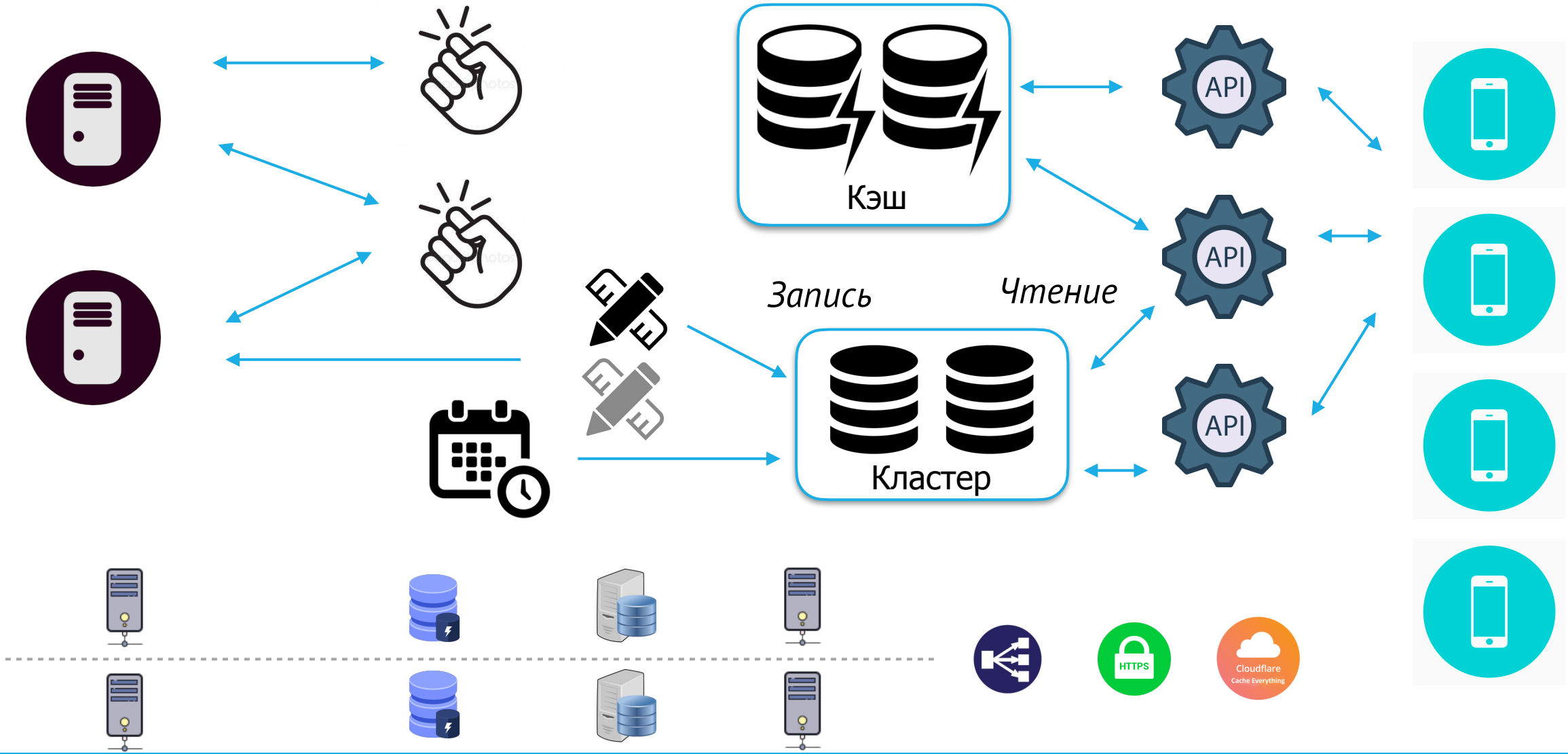
Логические блоки



Монолит в микросервисы



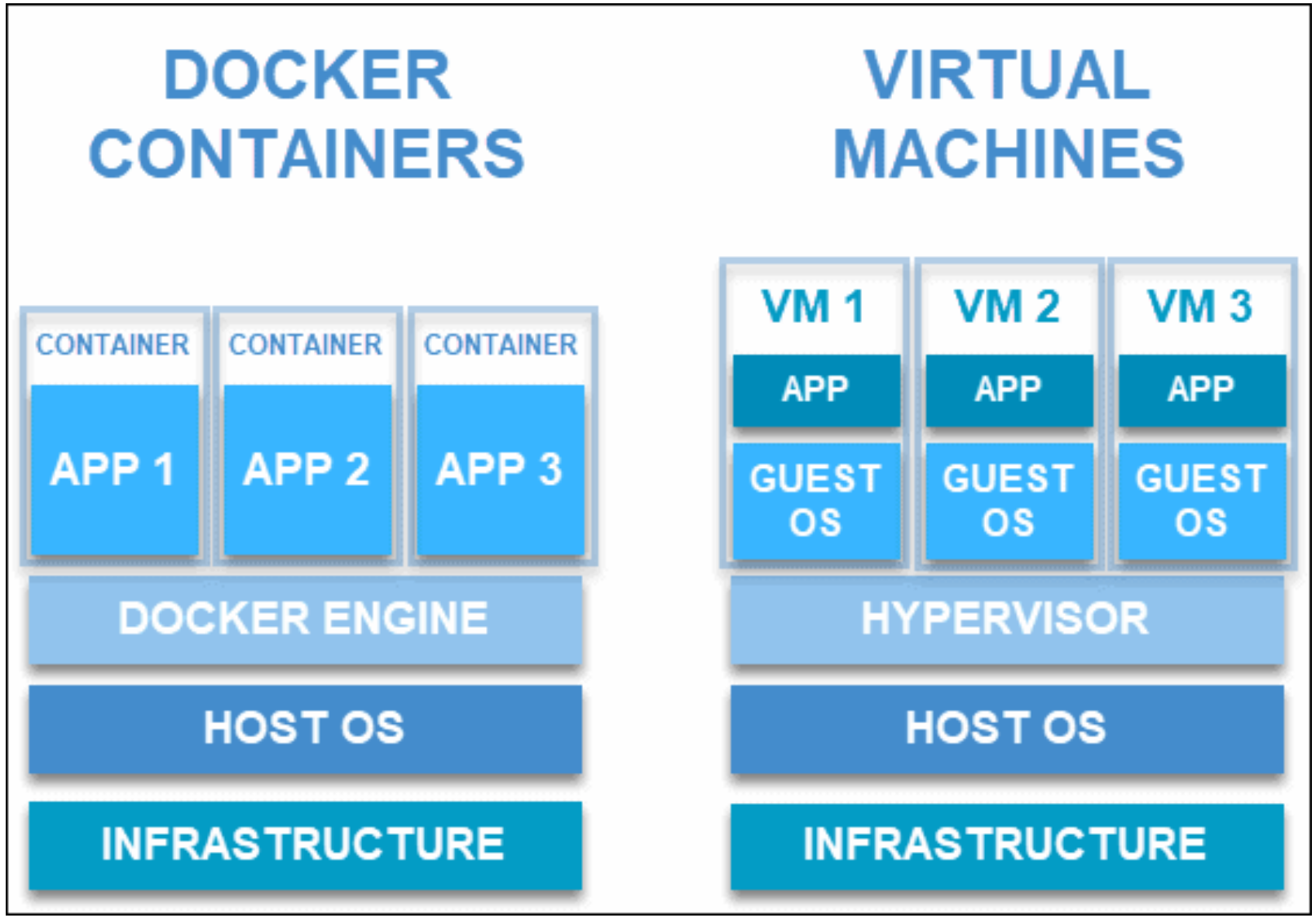
Пример готовой архитектуры



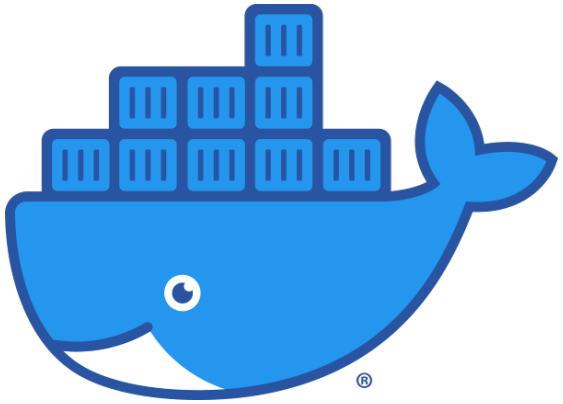
Автомасштабирование в kubernetes

3) Docker и K8s

Docker


DOCKER CONTAINERS
CONTAINER
APP 1
CONTAINER
APP 2
CONTAINER
APP 3
DOCKER ENGINE
HOST OS
INFRASTRUCTURE
VIRTUAL MACHINES
VM 1
APP
GUEST OS
VM 2
APP
GUEST OS
VM 3
APP
GUEST OS
HYPERVISOR
HOST OS
INFRASTRUCTURE

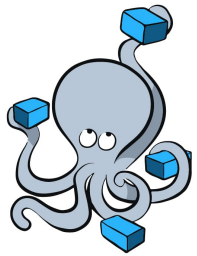
Docker



- минимизированное окружение;
- изоляция процессов;
- разделение ресурсов;
- прозрачные простые конфигурации;
- набор базовых сред;
- быстрое развертывание;
- слои;

- некоторое снижение производительности;
- возможные проблемы при работе stateful приложений;
- требования к архитектуре;

Оркестрация Docker



docker Compose

kubernetes

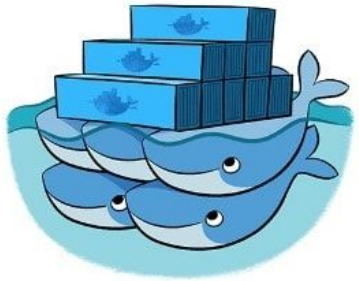
AWS Fargate

Amazon ECS

Amazon EKS

DigitalOcean

Docker Swarm



RED HAT® OPENSHIFT

Google Kubernetes Engine

Azure Kubernetes Service (AKS)

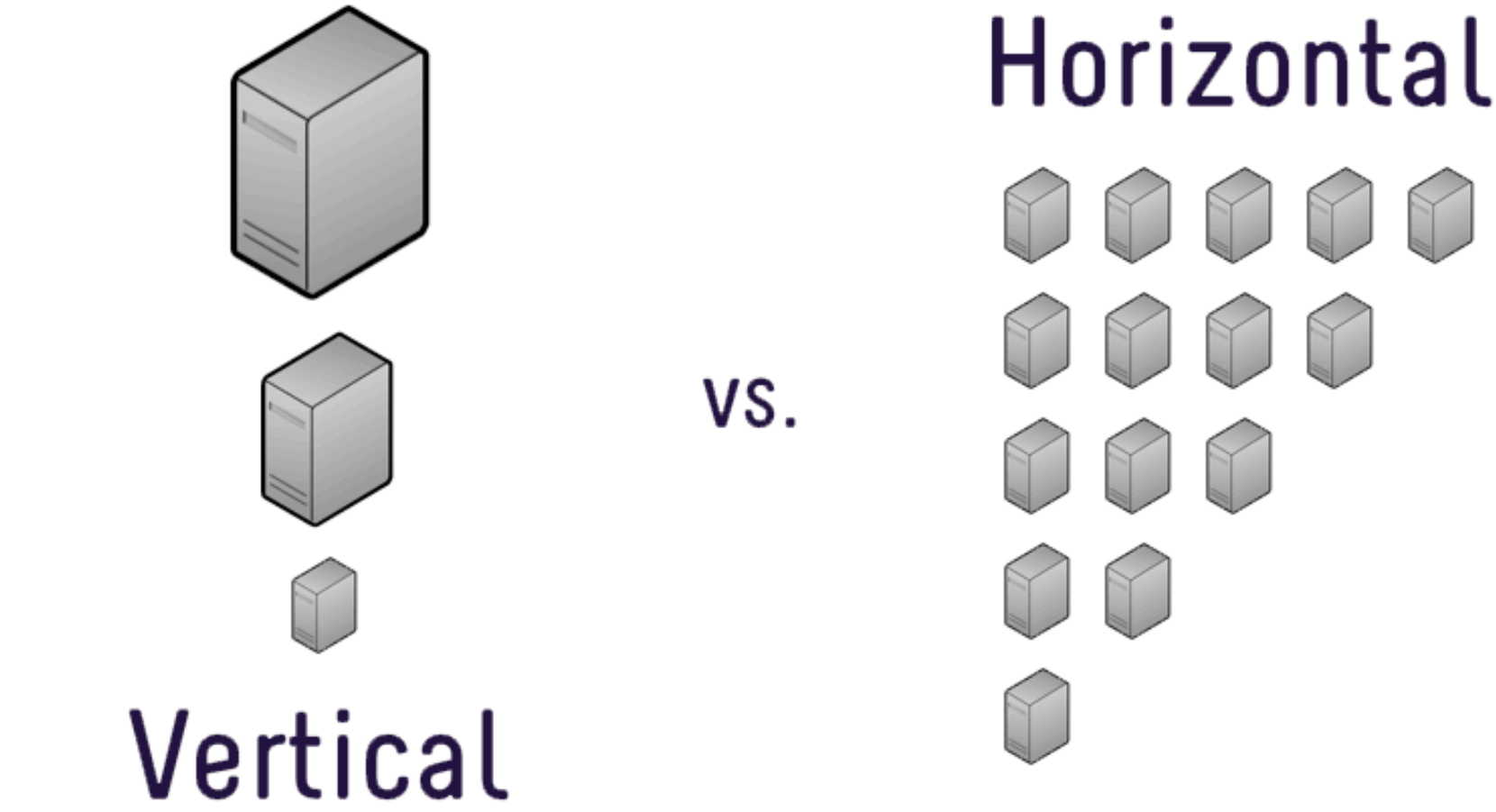
Yandex Cloud

Автомасштабирование в kubernetes

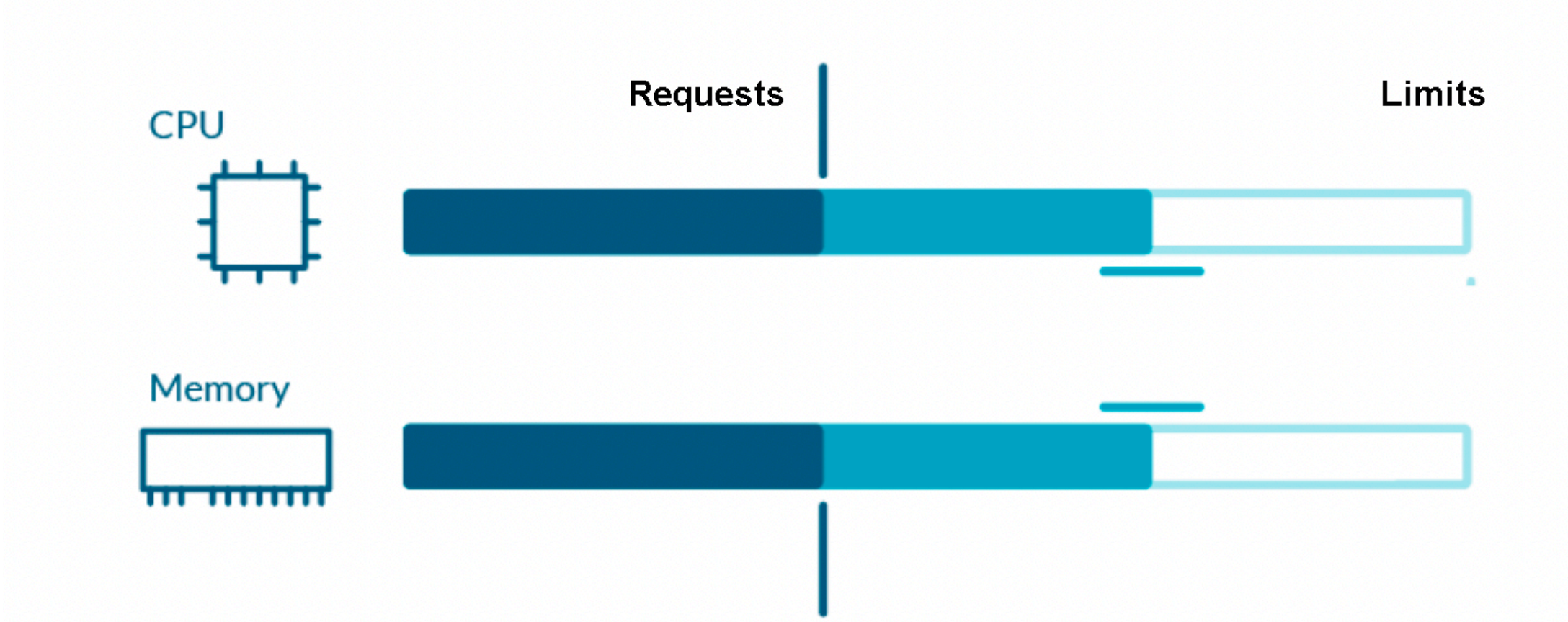
---

4) Масштабирование приложений в k8s

Типы масштабирования

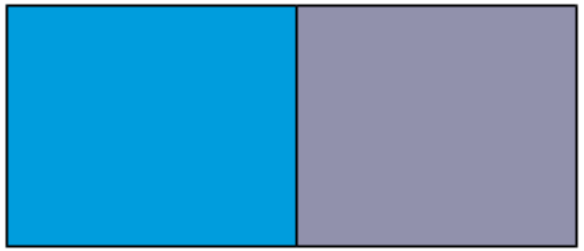


Вертикальное масштабирование



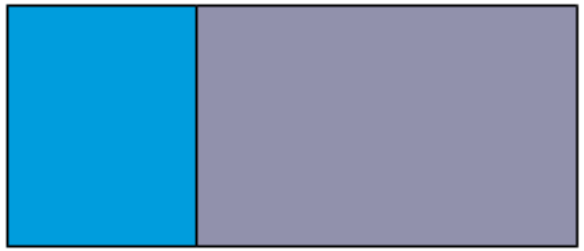
K8s QoS

Guaranteed



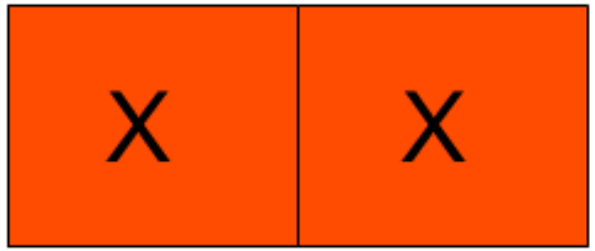
Requests and limits are equal

Burstable



Requests and limits are not equal

Best Effort



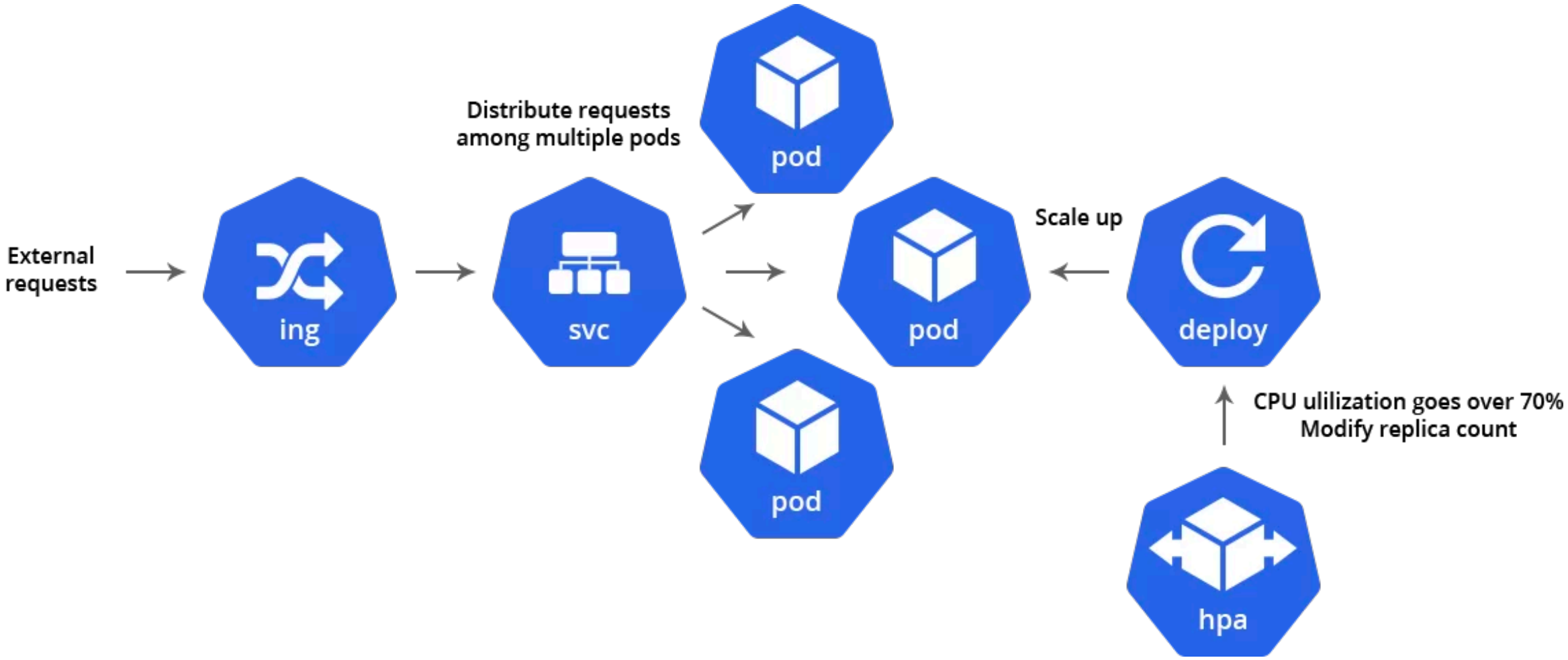
No requests and limits have been set

evictions

Пример указания ресурсов

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: db
    image: mysql
    env:
    - name: MYSQL_ROOT_PASSWORD
      value: "password"
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

Горизонтальное масштабирование



Пример указания hpa

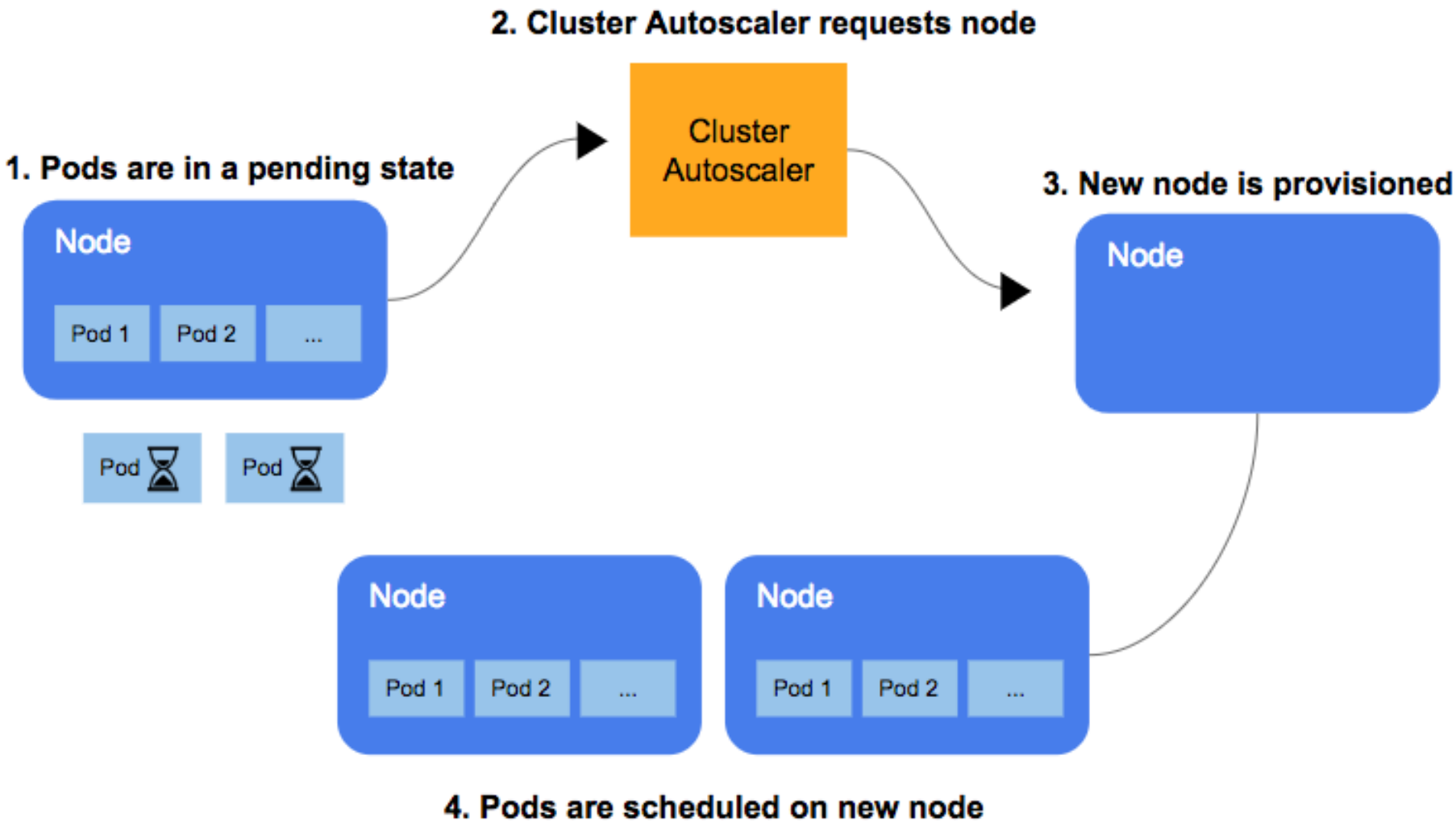
```yaml
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: php-apache
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-apache
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

Автомасштабирование в kubernetes

---

5) Масштабирование k8s в облаке

Cluster autoscaler



Cluster autoscaler

```
cluster-autoscaler:
  cloudProvider: aws
  awsRegion: us-east-1

  extraArgs:
    skip-nodes-with-local-storage: false
    skip-nodes-with-system-pods: false
    scan-interval: 10s

  extraArgs:
    scale-down-utilization-threshold: 0.85

  podAnnotations:
    iam.amazonaws.com/role: arn:aws:iam::1111111111111:role/k8s-cluster-autoscaler

  serviceMonitor:
    enabled: false
    interval: "10s"
```

AWS spot

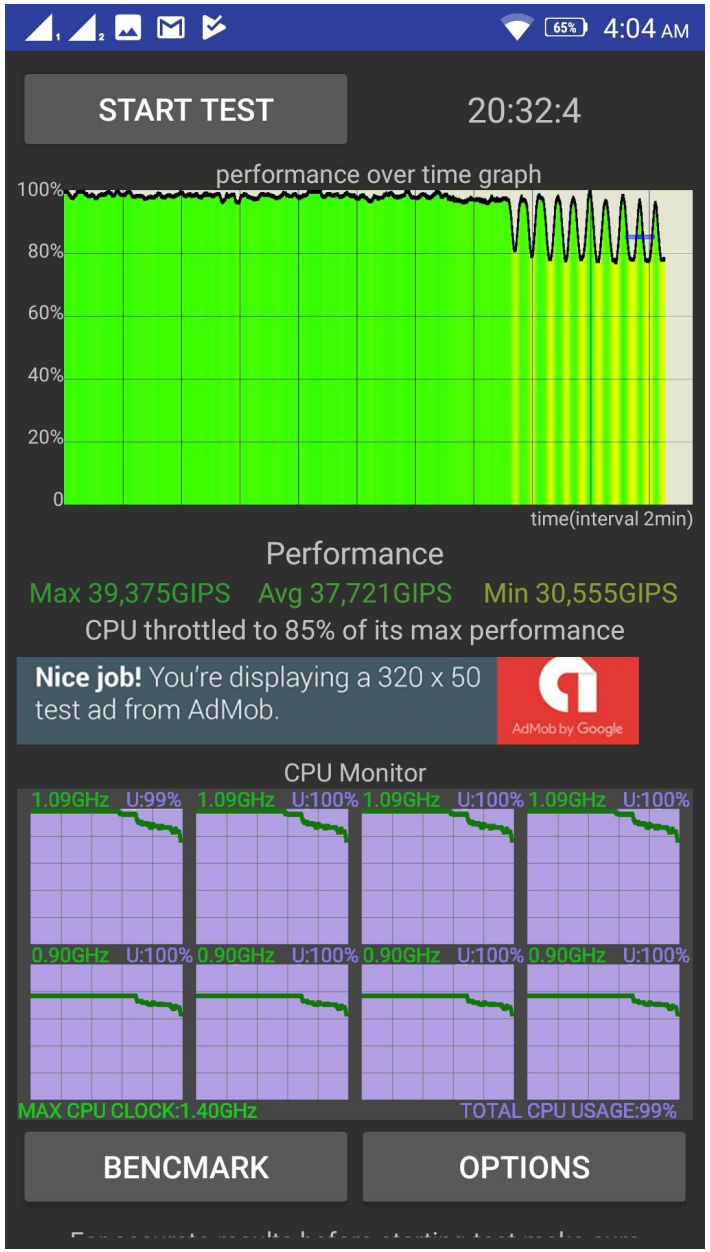

Spot
Instance
Regular EC2
Instance
C3
Optimized
EC2 Instance

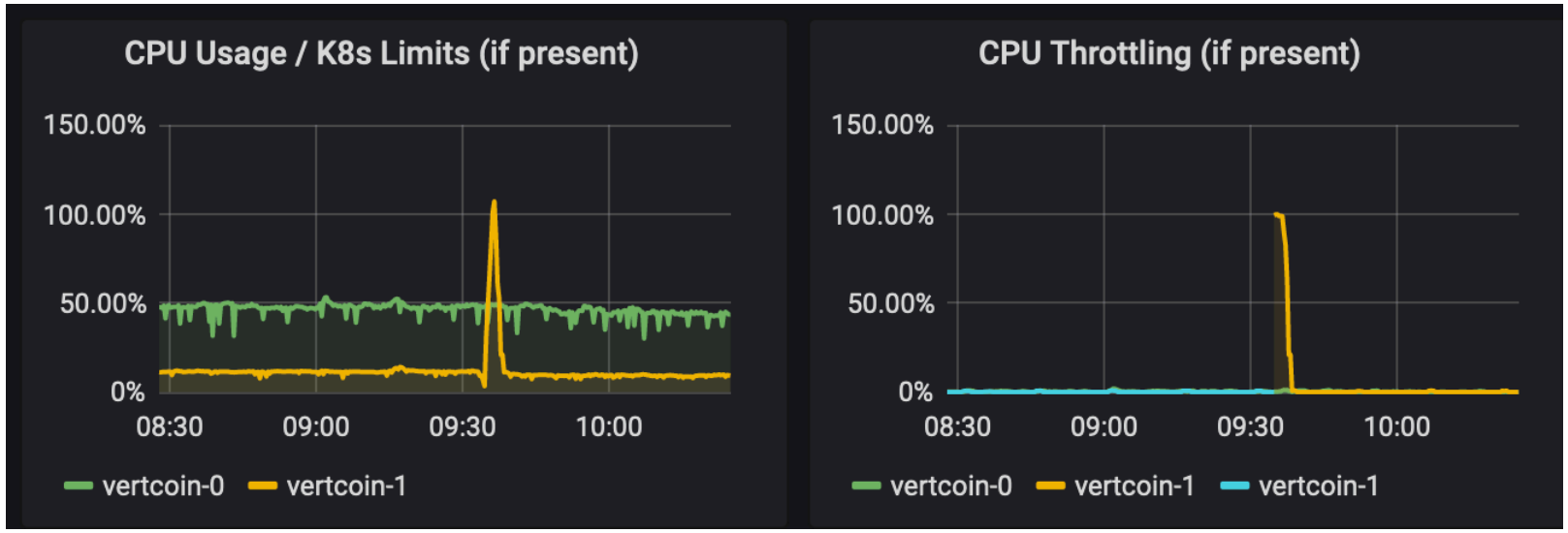
Автомасштабирование в kubernetes

---

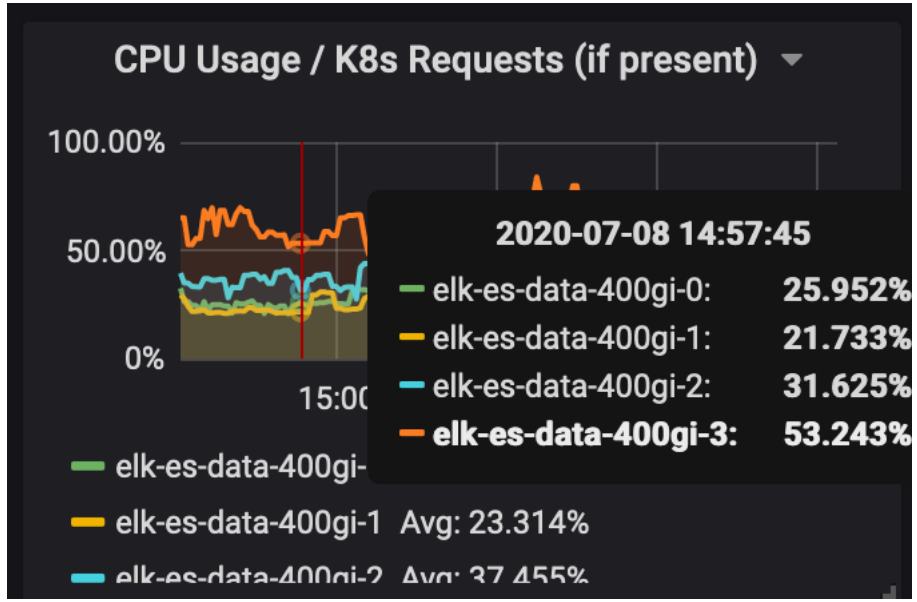
6) Нюансы масштабирования

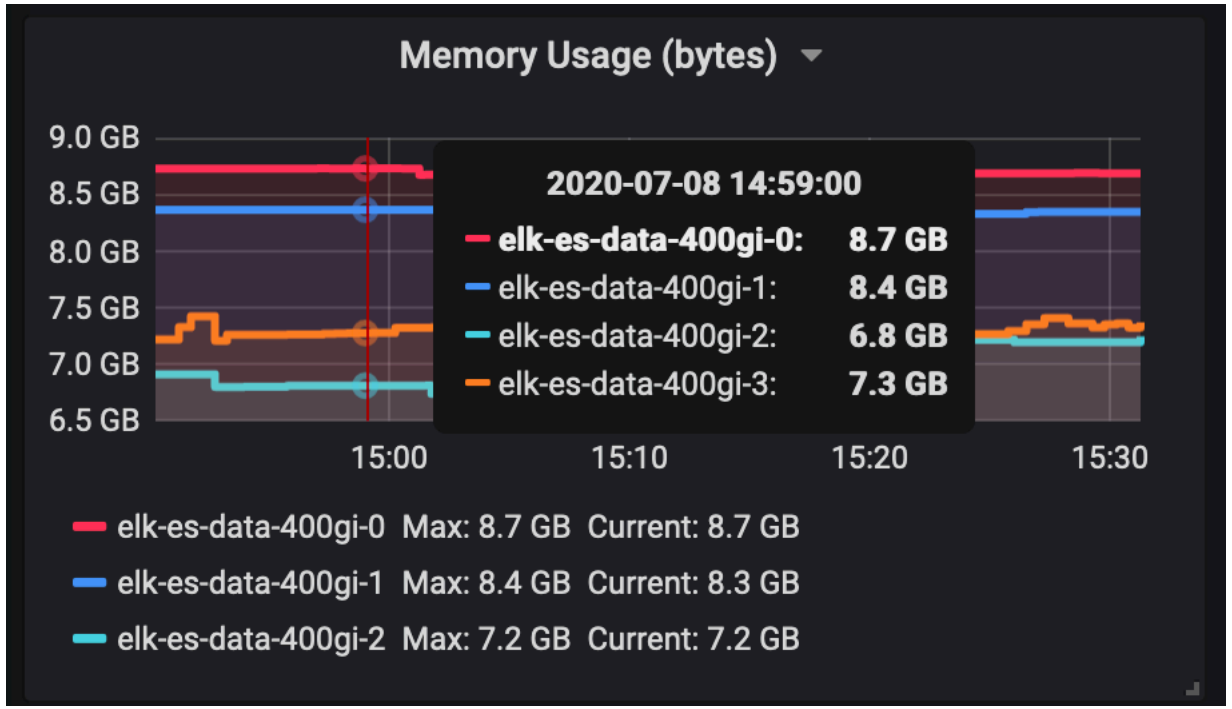
CPU Throttling



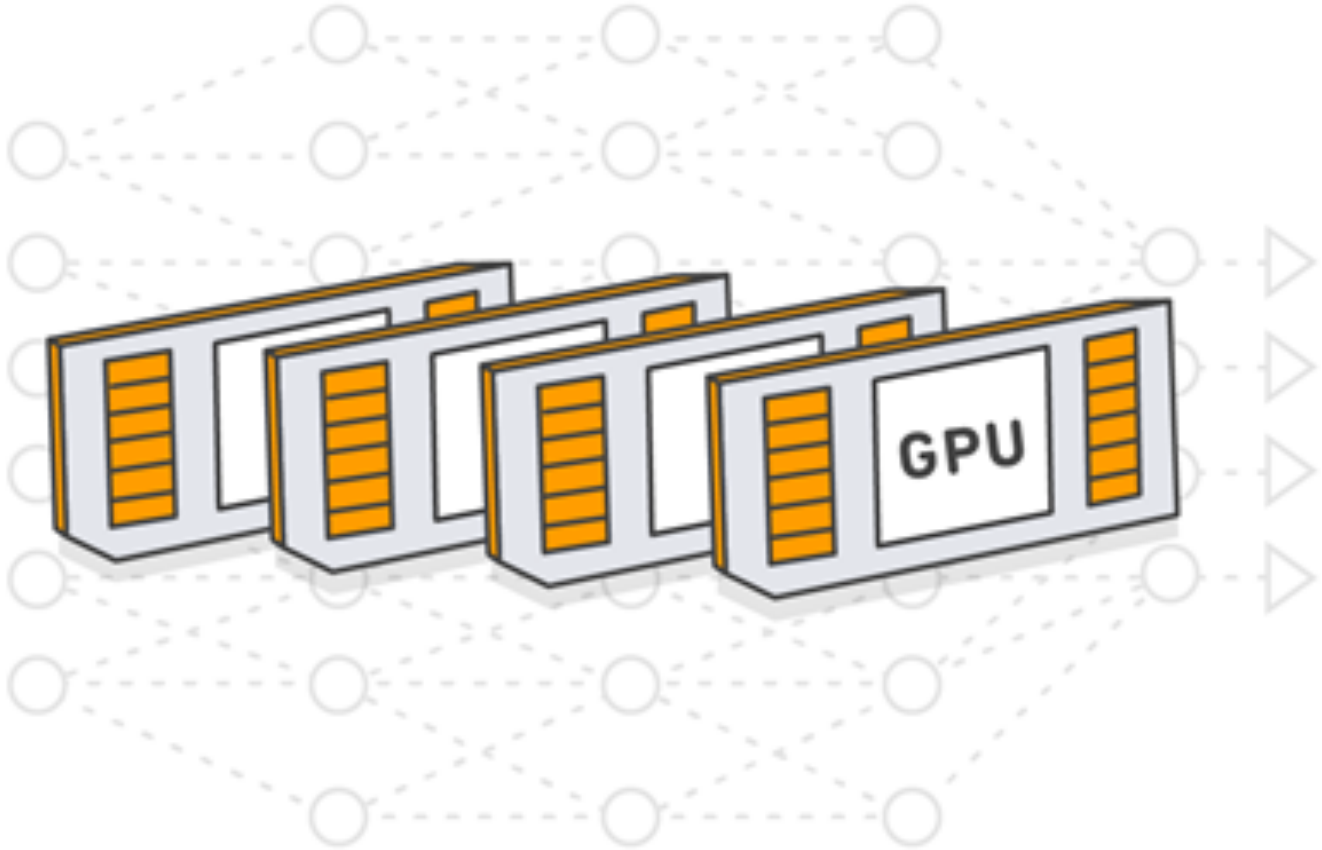


Неравномерная нагрузка

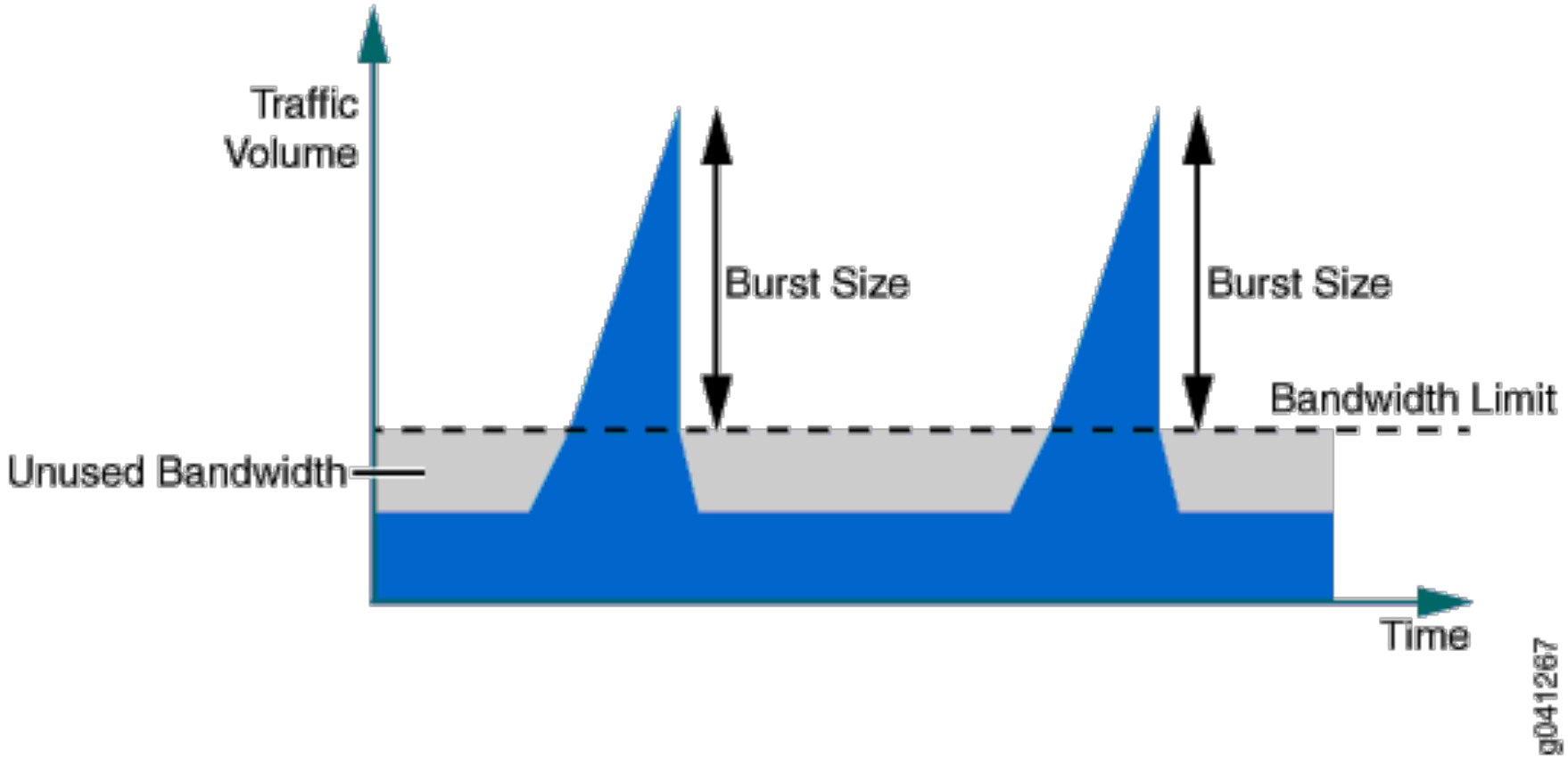




GPU



Burst


Traffic
Volume
Burst Size
Burst Size
Bandwidth Limit
Unused Bandwidth
Time
g041267


"That's all Folks!"