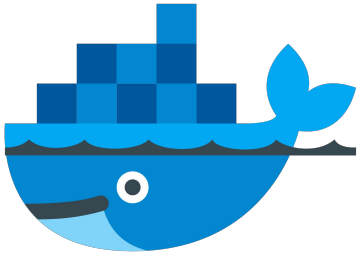
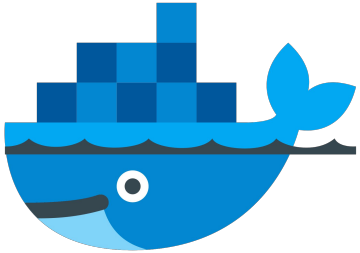
Stateful в k8s, которого мы боимся

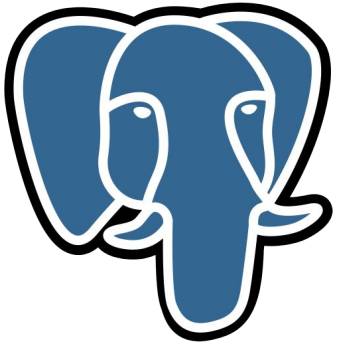
Женя Дехтярёв

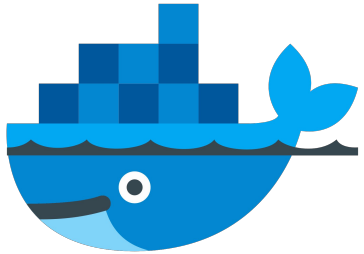
Про команду



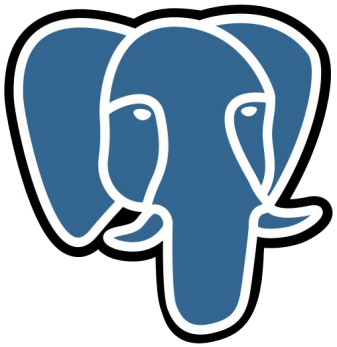
Про команду

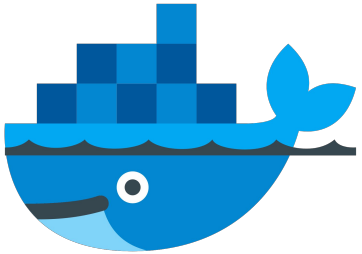




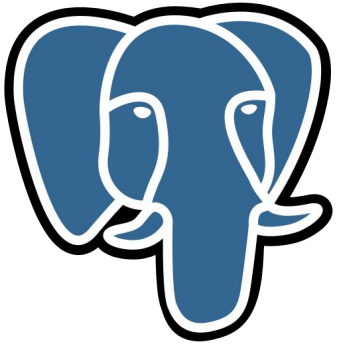


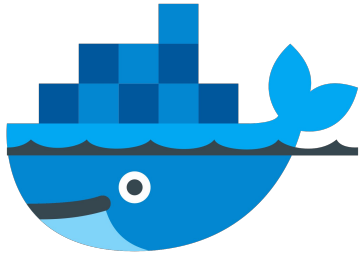

Stack



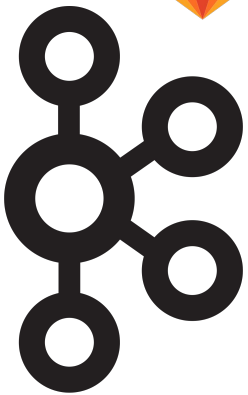


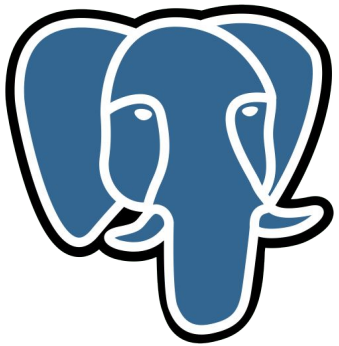

Stack






Stack





2ГИС — это

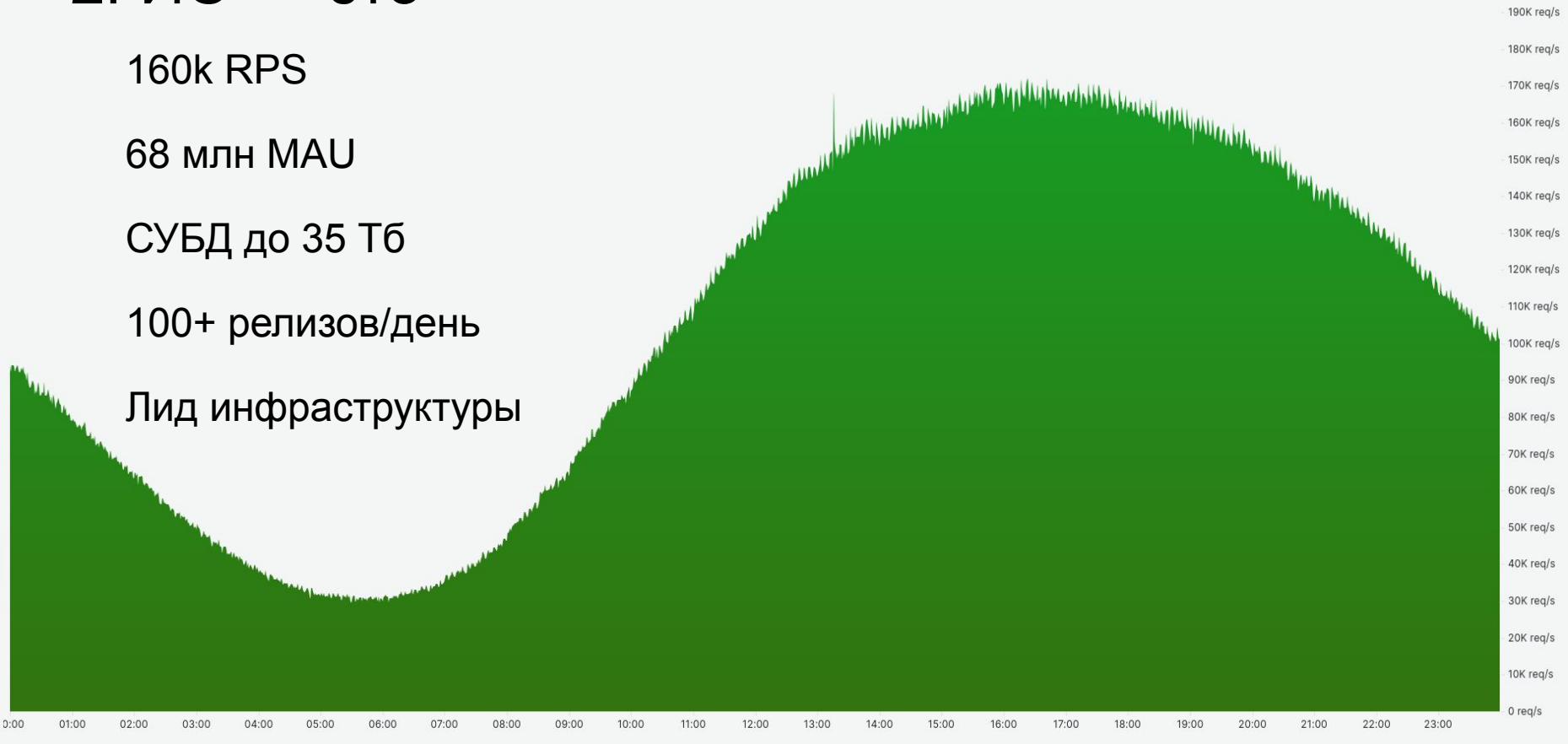
160k RPS

68 млн MAU

СУБД до 35 Тб

100+ релизов/день

Лид инфраструктуры



О чём поговорим

- Почему Stateful в k8s?
- Локальные и сетевые PV в k8s
- Особенности Stateful для PV
- Интересные случаи использования PV

Stateful vs Stateless

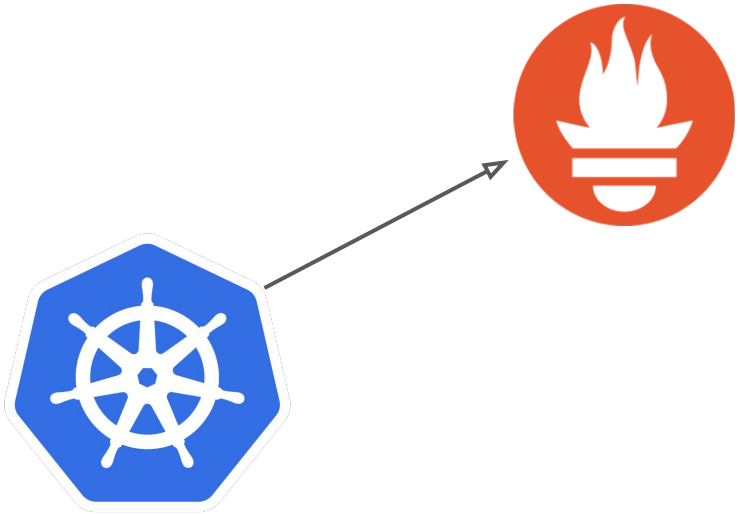
- **Stateless** — не храним данные, всегда чистый лист
  - Пример: REST API

Stateful vs Stateless

- **Stateless** — не храним данные, всегда чистый лист
  - Пример: REST API
- **Stateful** — храним данные, ответ зависит от данных
  - Пример: СУБД

Почему мы хотим stateful в k8s?

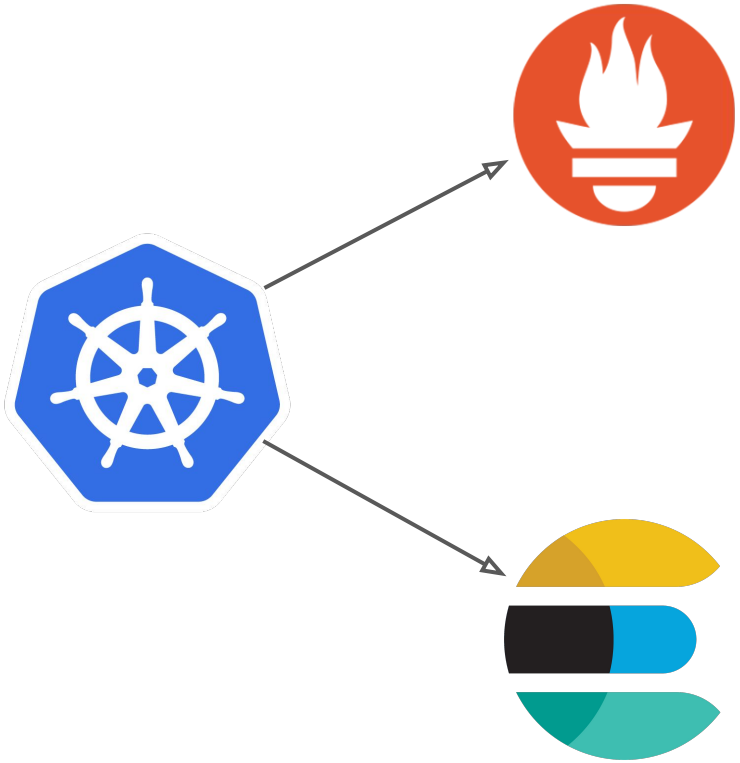
Платформа k8s — это окружение, которое позволяет экономить время на операционку



Экономия

Мониторинг

Метрики и алерты сразу из коробки

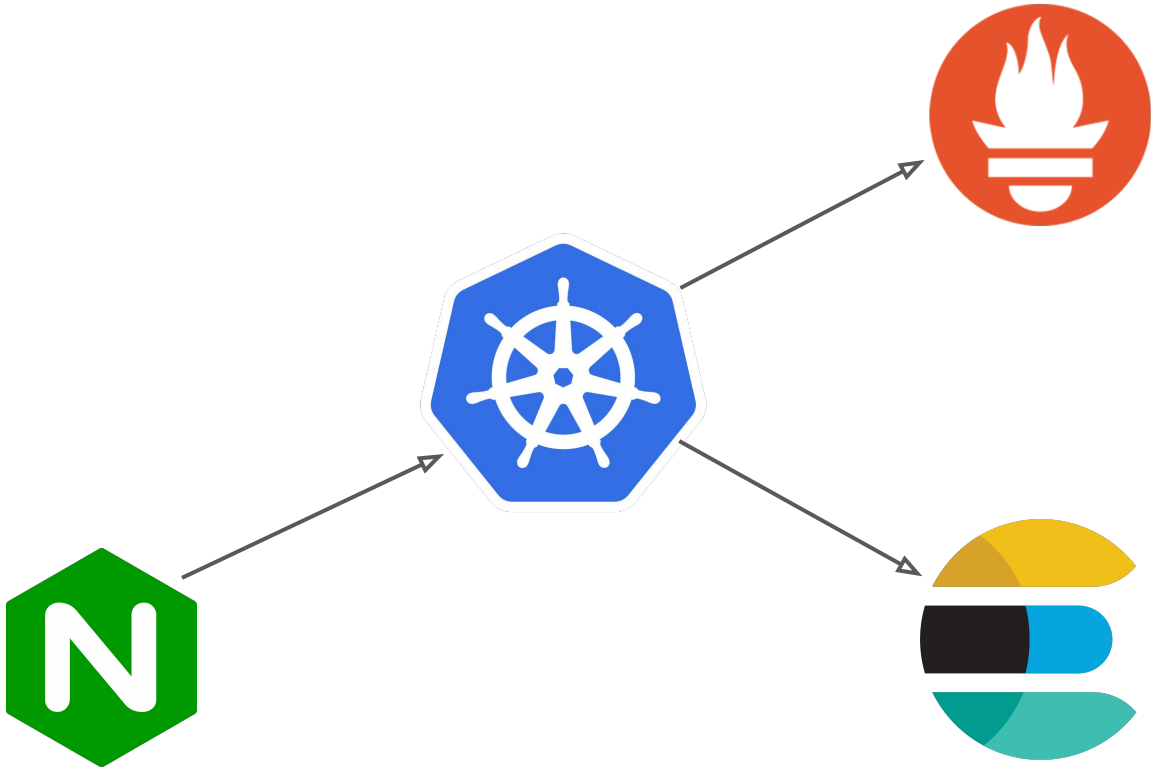


Экономия

Мониторинг

Логирование

Просто отправь в `/dev/stdout`



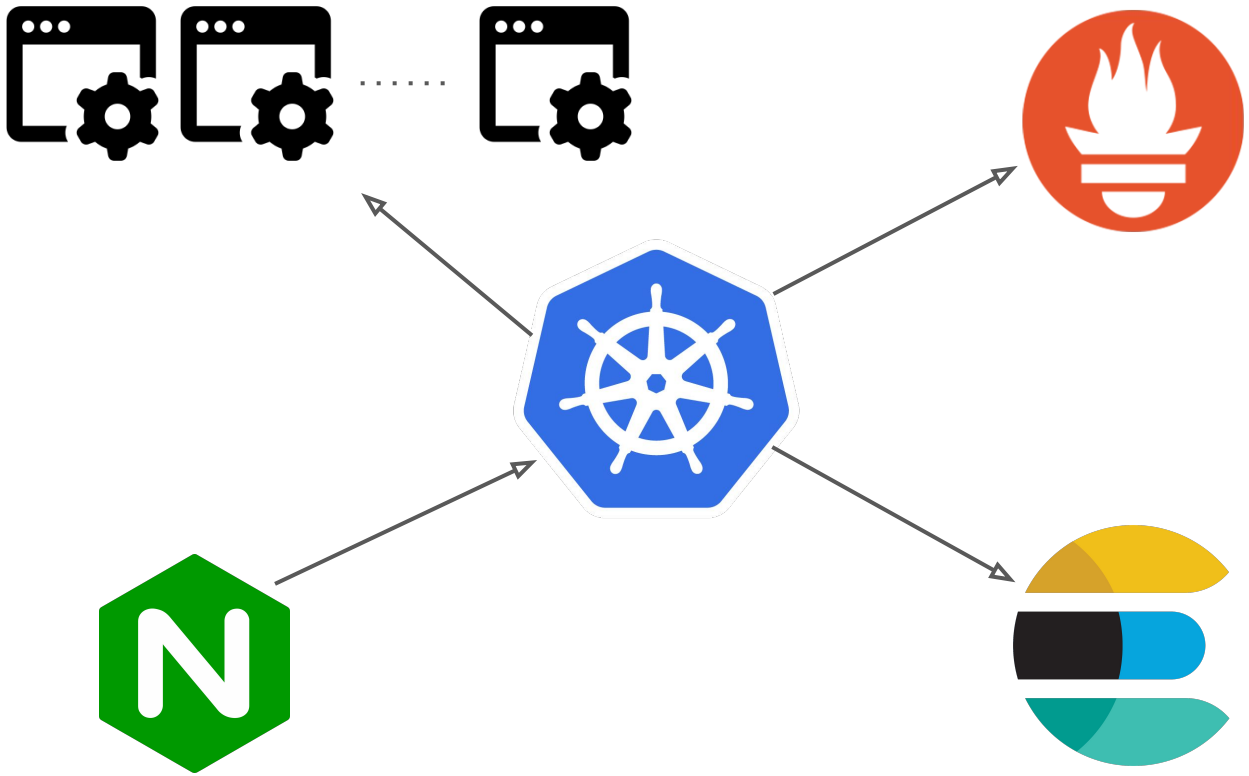
Экономия

Мониторинг

Логирование

Маршрутизация

Только настрой ingress



Экономия

Мониторинг

Логирование

Маршрутизация

Масштабирование

Масштабирование в 1 команду

Stateful-приложения

Локальные диски

- Cassandra
  - 5 Тб
- Kafka
  - 15 Тб
- Elasticsearch
  - 35 Тб
- PGSQL
  - Prod
- Clickhouse
- Mongo

Сетевые диски

- PGSQL
  - Dev-окружения (Spilo)
- Redis

Stateful-приложения

Локальные диски

- **Cassandra**
  - 5 Тб
- Kafka
  - 15 Тб
- **Elasticsearch**
  - 35 Тб
- PGSQL
  - Prod
- Clickhouse
- Mongo

Сетевые диски

- **PGSQL**
  - Dev-окружения (Spilo)
- Redis

Локальные Volume

Локальные Volume. Когда применять?

- Нужна максимальная производительность(600 нс ответ)

Локальные Volume. Когда применять?

- Нужна максимальная производительность(600 нс ответ)
- Нужны большие объёмы(4-8 Тб SSD — это ОК)

Локальные Volume. Когда применять?

- Нужна максимальная производительность(600 нс ответ)
- Нужны большие объёмы(4-8 Тб SSD — это ОК)
- Приложение умеет в самостоятельное восстановление
  - Потерять 1 LocalPV — это норма!

Локальные Volume. Пример

Elasticsearch, Cassandra

- Минимальные задержки диска

Локальные Volume. Пример

Elasticsearch, Cassandra

- Минимальные задержки диска
- Недоступность POD в несколько часов не приведёт ни к чему

Локальные Volume. Пример

Elasticsearch, Cassandra

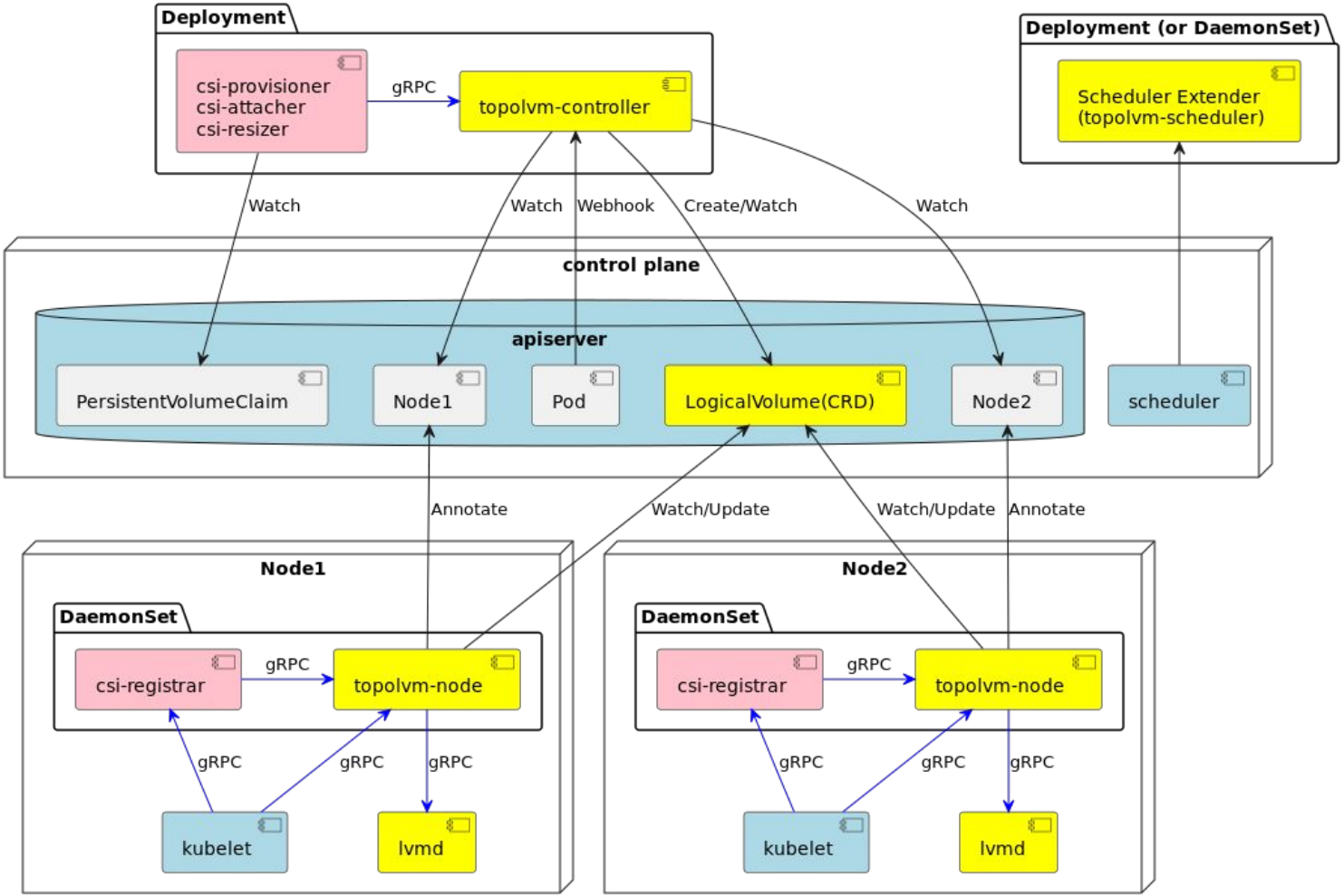
- Минимальные задержки диска
- Недоступность POD в несколько часов не приведёт ни к чему
- Самостоятельное восстановление реплик

Локальные Volume. Пример

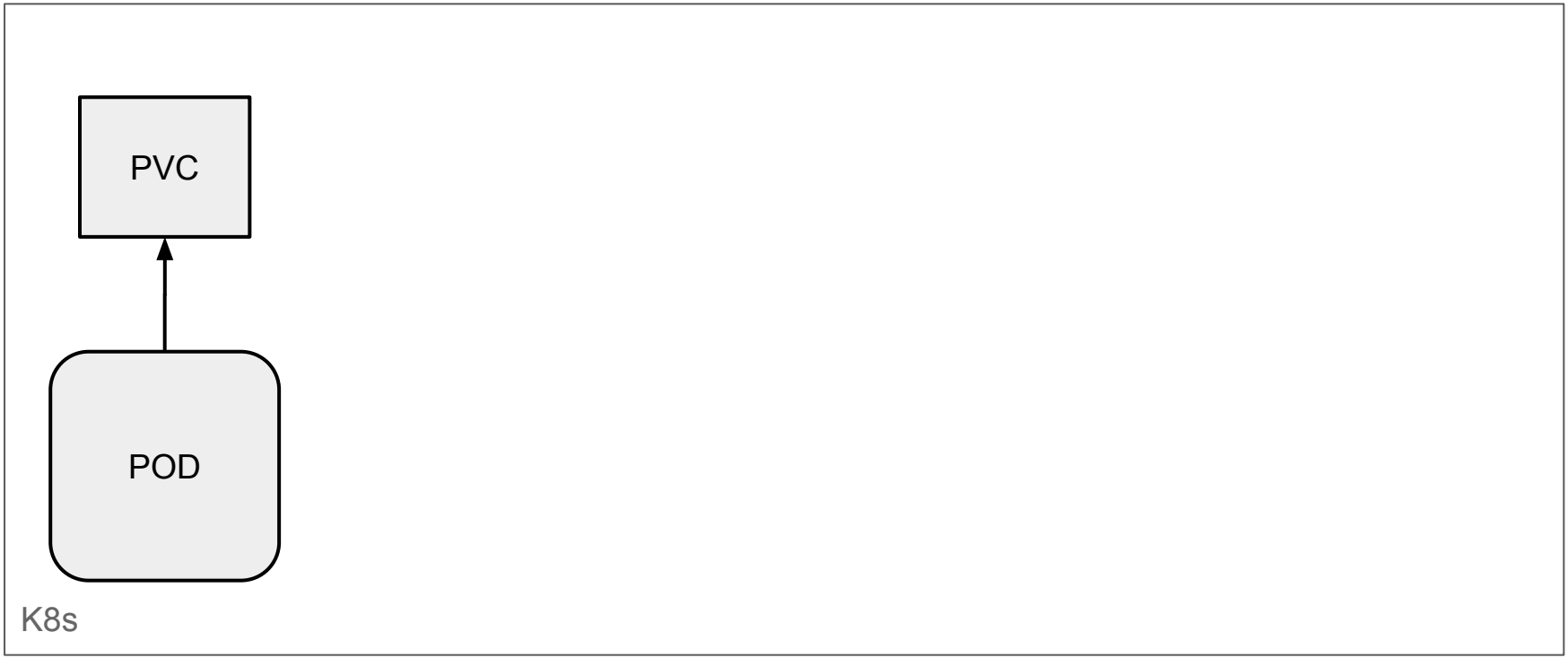
Elasticsearch, Cassandra

- Минимальные задержки диска
- Недоступность POD в несколько часов не приведёт ни к чему
- Самостоятельное восстановление реплик
- Самостоятельное шардирование

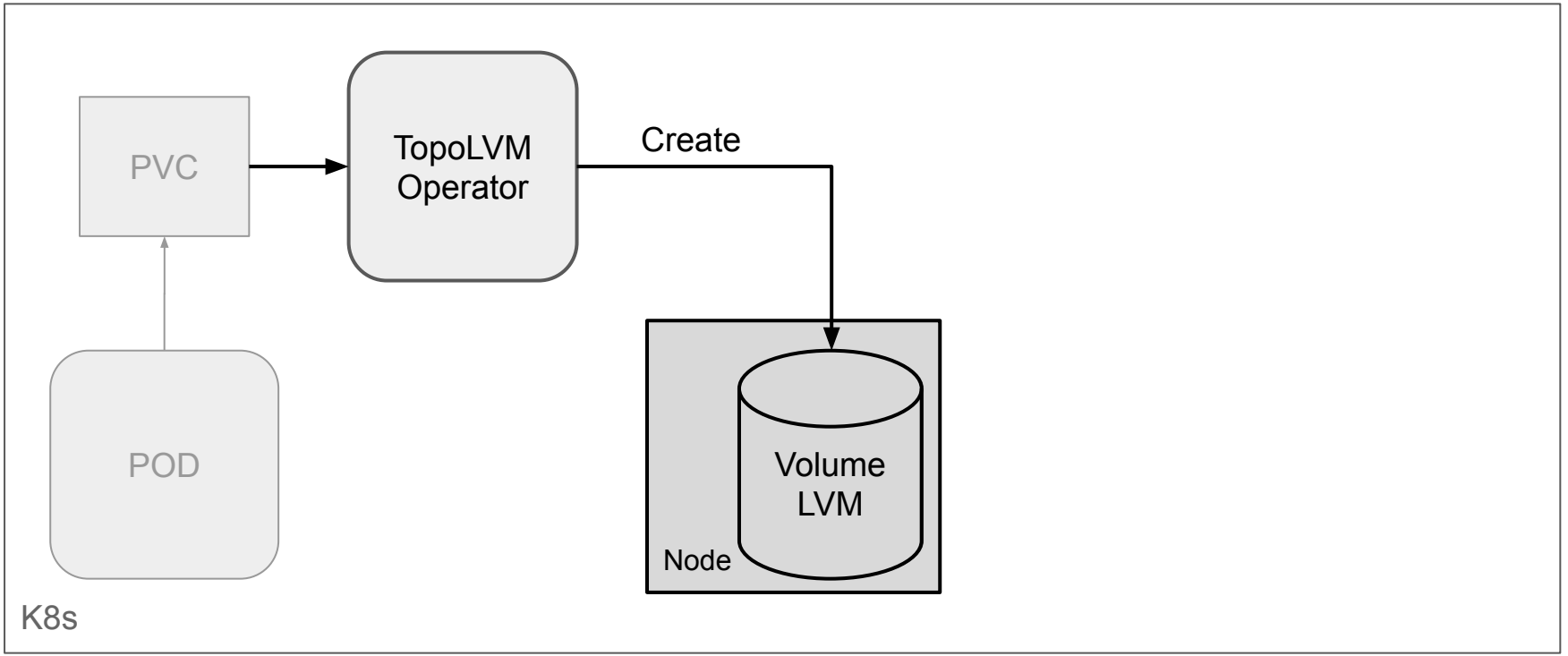
TOPOLVM


Deployment
csi-provisioner
csi-attacher
csi-resizer
gRPC
topolvm-controller
Deployment (or DaemonSet)
Scheduler Extender
(topolvm-scheduler)
Watch
Watch
Webhook
Create/Watch
Watch
control plane
apiserver
PersistentVolumeClaim
Node1
Pod
LogicalVolume(CRD)
Node2
scheduler
Annotate
Watch/Update
Watch/Update
Annotate
Node1
DaemonSet
csi-registrar
gRPC
topolvm-node
gRPC
gRPC
gRPC
kubelet
lvmd
Node2
DaemonSet
csi-registrar
gRPC
topolvm-node
gRPC
gRPC
gRPC
kubelet
lvmd

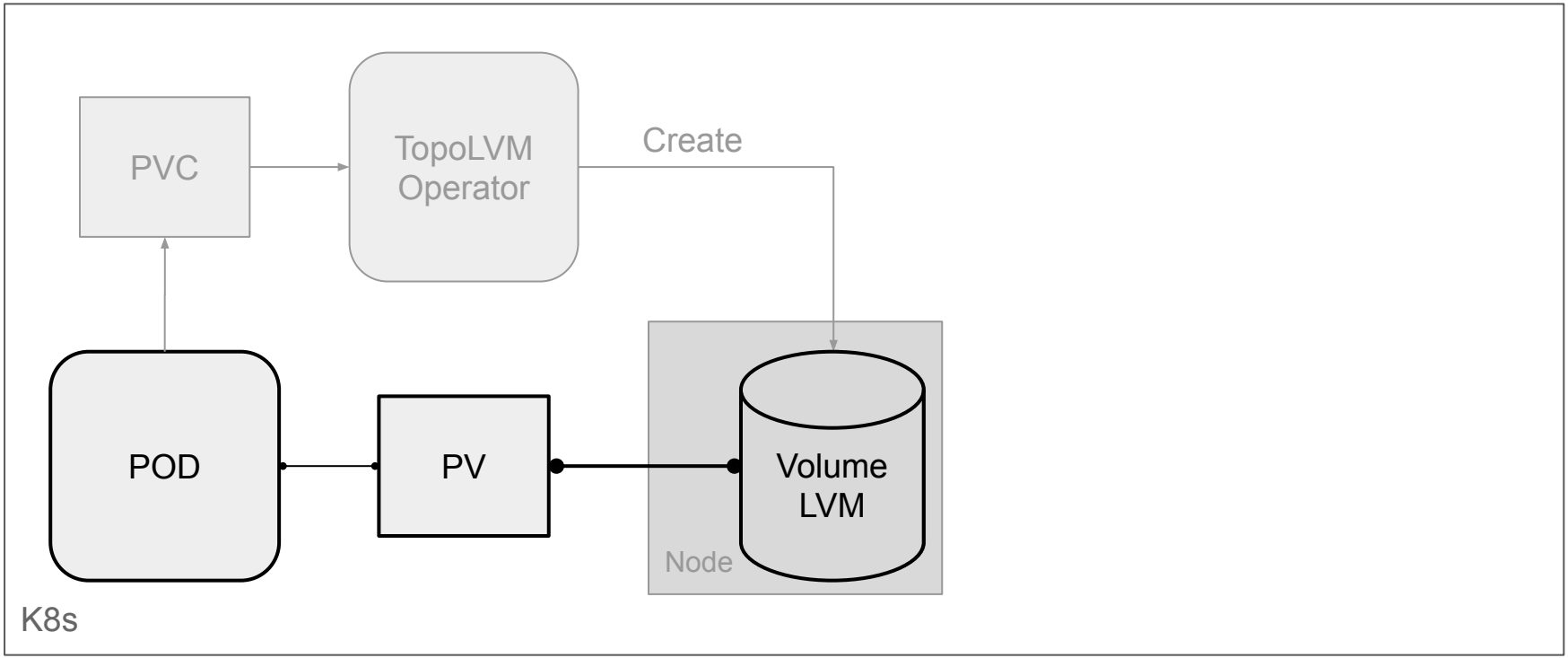
github.com/topolvm/topolvm



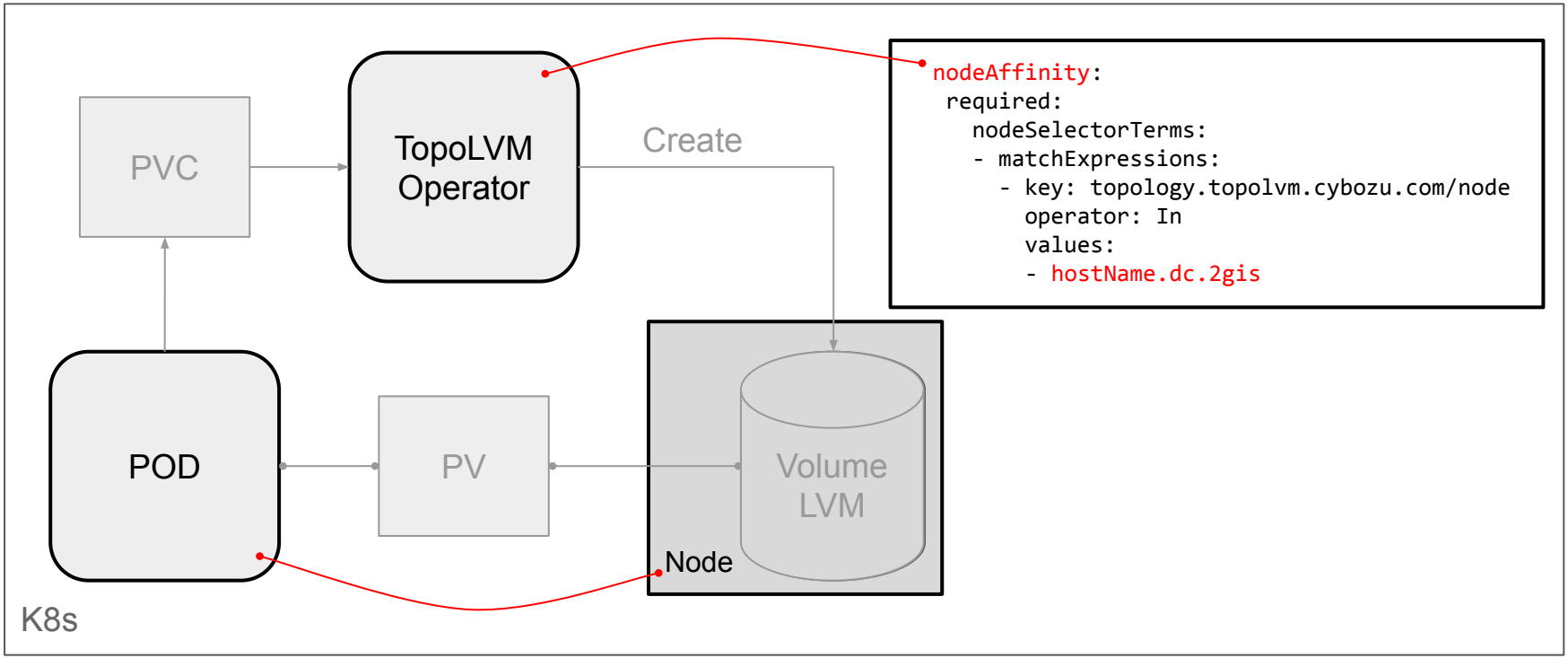
github.com/topolvm/topolvm



github.com/topolvm/topolvm



github.com/topolvm/topolvm



github.com/topolvm/topolvm 

- Плюсы
  - Динамический провижининг
    - *в пределах сервера
  - Легкое расширение
  - Метрики
  - Быстрый

github.com/topolvm/topolvm

- Плюсы
  - Динамический провижининг
    - *в пределах сервера
  - Легкое расширение
  - Метрики
  - Быстрый
- Минусы
  - Нужен оператор
  - LVM на нодах k8s

Сетевые Volume

Сетевые Volume. Когда применять?

- Dev/Stage, небольшие объёмы

Сетевые Volume. Когда применять?

- Dev/Stage, небольшие объёмы
- Нужна максимальная надёжность данных(Prod)

Сетевые Volume. Когда применять?

- Dev/Stage, небольшие объёмы
- Нужна максимальная надёжность данных(Prod)
- Не важно где запуститься, главное запуститься
  - Например CronJob с диском

Сетевые Volume. Пример

PGSQL

- 2 реплики (Master-Slave)

Сетевые Volume. Пример

PGSQL

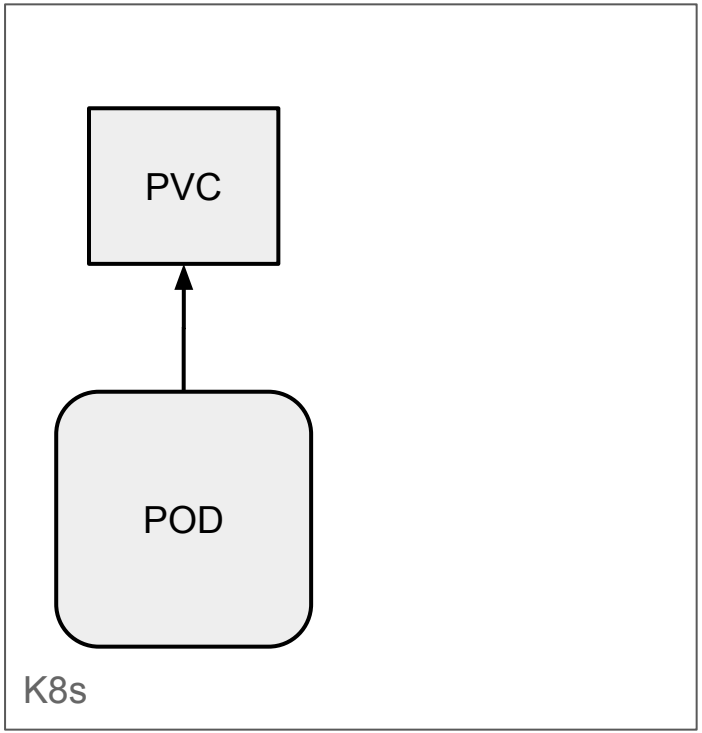
- 2 реплики (Master-Slave)
- Перезапуск POD = минимальный простой Slave

Сетевые Volume. Пример

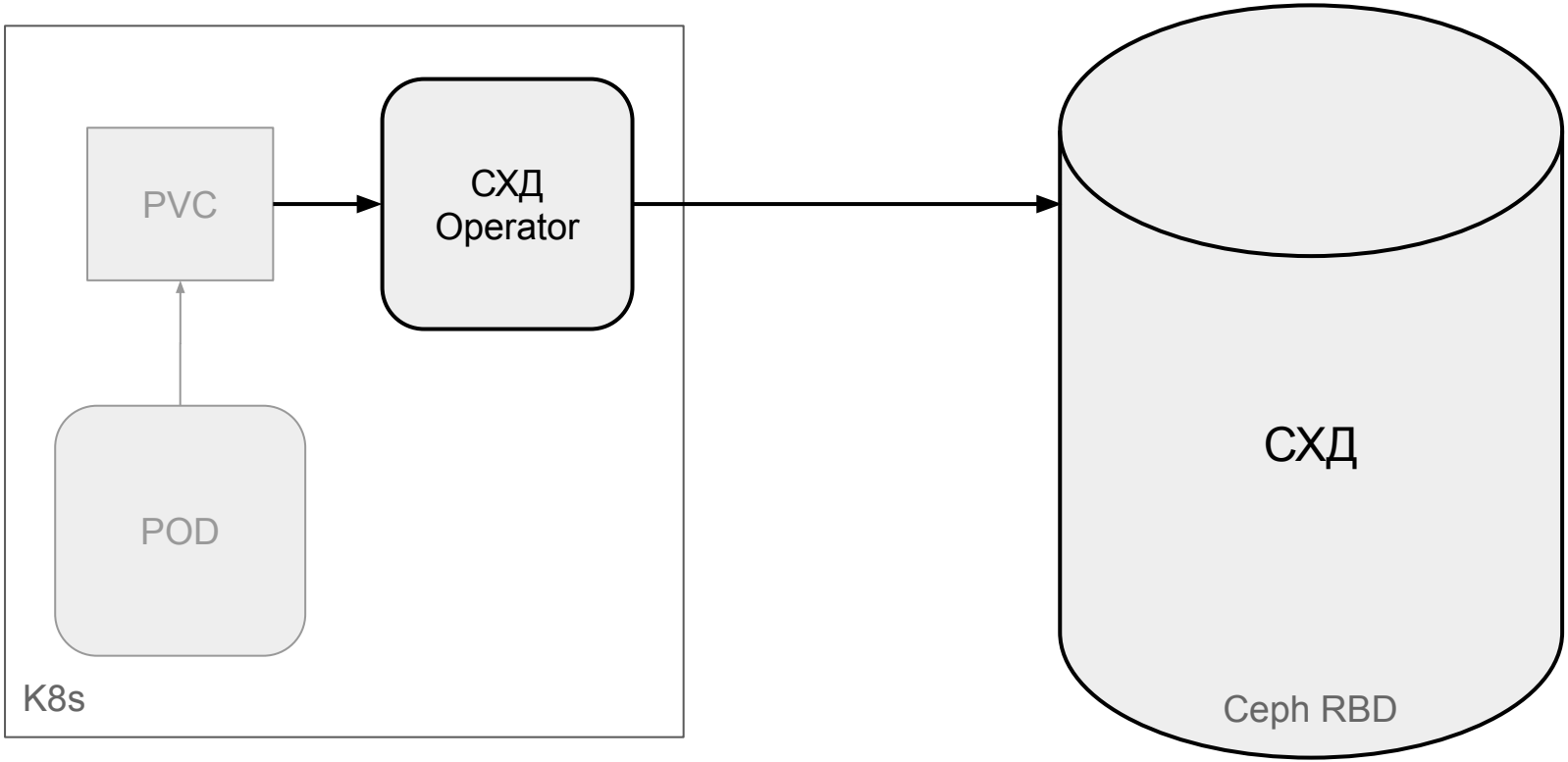
PGSQL

- 2 реплики (Master-Slave)
- Перезапуск POD = минимальный простой Slave
- WAL хватает, репликация не разваливалась

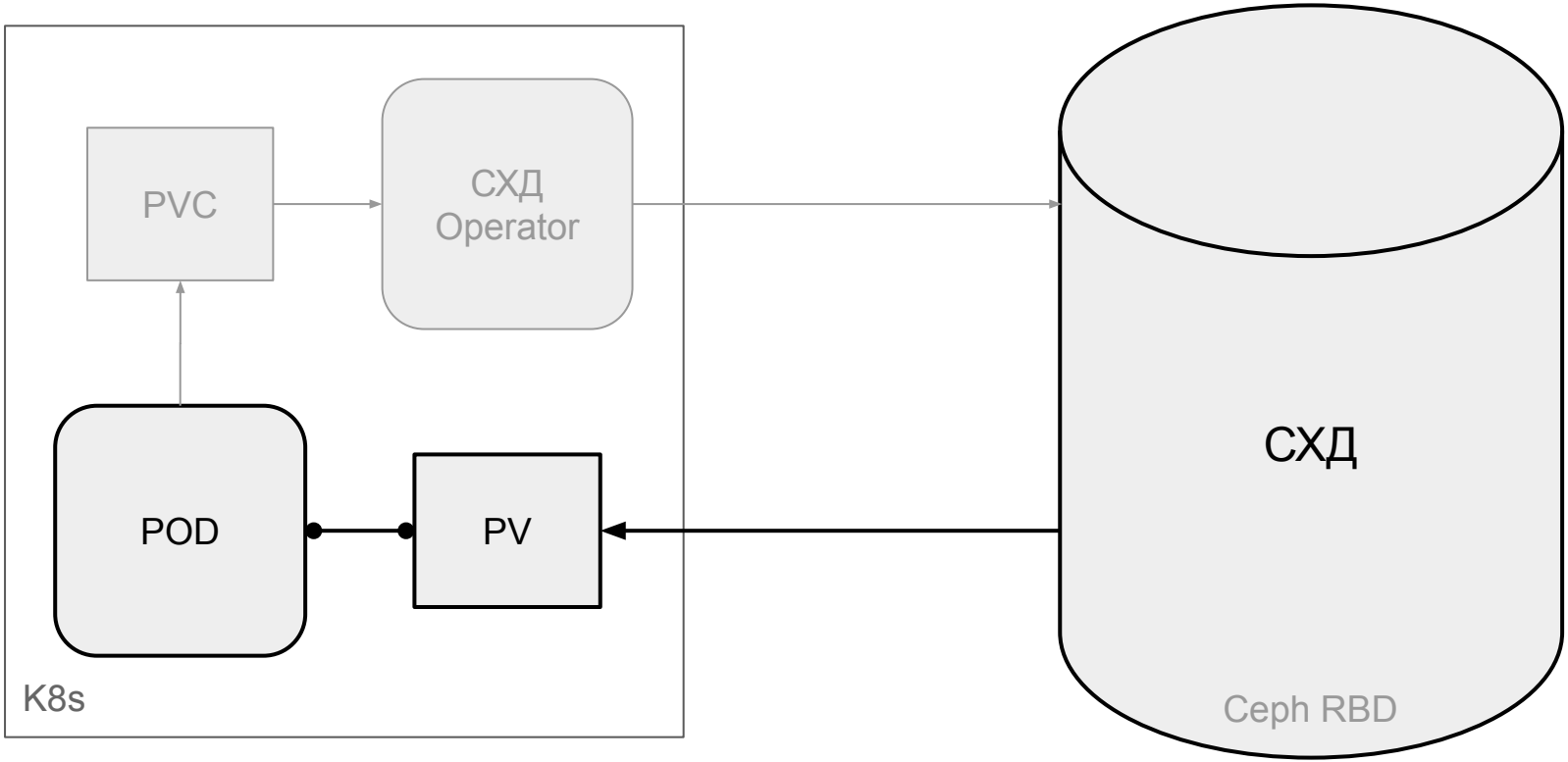
Сетевые Volume



Сетевые Volume



Сетевые Volume



Сетевые Volume. RBD Ceph

- Плюсы
  - Динамический провижининг
  - Простое расширение
  - Быстрое восстановление

Сетевые Volume. RBD Ceph

- Плюсы
  - Динамический провижининг
  - Простое расширение
  - Быстрое восстановление
- Минусы
  - Скорость зависит от:
    - скорости сети
    - дисков под CEPH
  - Деградация СХД ⟶ увеличение времени ответа

Особенности Stateful для PV

Локальные Volume в коммуналке

- PodAntiAffinity

Локальные Volume в коммуналке

- PodAntiAffinity
- PodPriority

Локальные Volume в коммуналке

- PodAntiAffinity
- PodPriority
- PodDisruptionBudget

Локальные Volume в коммуналке

- PodAntiAffinity
- PodPriority
- PodDisruptionBudget
- 2+ POD на 1 K8s Node — iowait

Локальные Volume в коммуналке

- PodAntiAffinity
- PodPriority
- PodDisruptionBudget
- 2+ POD на 1 K8s Node — iowait
- Не квотировать объём PV

Локальные Volume в коммуналке

- PodAntiAffinity
- PodPriority
- PodDisruptionBudget
- 2+ POD на 1 K8s Node — iowait
- Не квотировать объём PV
- Stateful, replicas 3+

Локальные Volume в коммуналке

- PodAntiAffinity
- PodPriority
- PodDisruptionBudget
- 2+ POD на 1 K8s Node — iowait
- Не квотировать объём PV
- Stateful, replicas 3+
- Бэкап!

Сетевые Volume в коммуналке

- PodAntiAffinity
  - *не обязательно

Сетевые Volume в коммуналке

- PodAntiAffinity
  - *не обязательно
- PodDisruptionBudget

Сетевые Volume в коммуналке

- PodAntiAffinity
  - *не обязательно
- PodDisruptionBudget
- Много операций с данными == iowait;

Сетевые Volume в коммуналке

- PodAntiAffinity
  - *не обязательно
- PodDisruptionBudget
- Много операций с данными == iowait;
- Обязательное квотирование и ограничения

Сетевые Volume в коммуналке

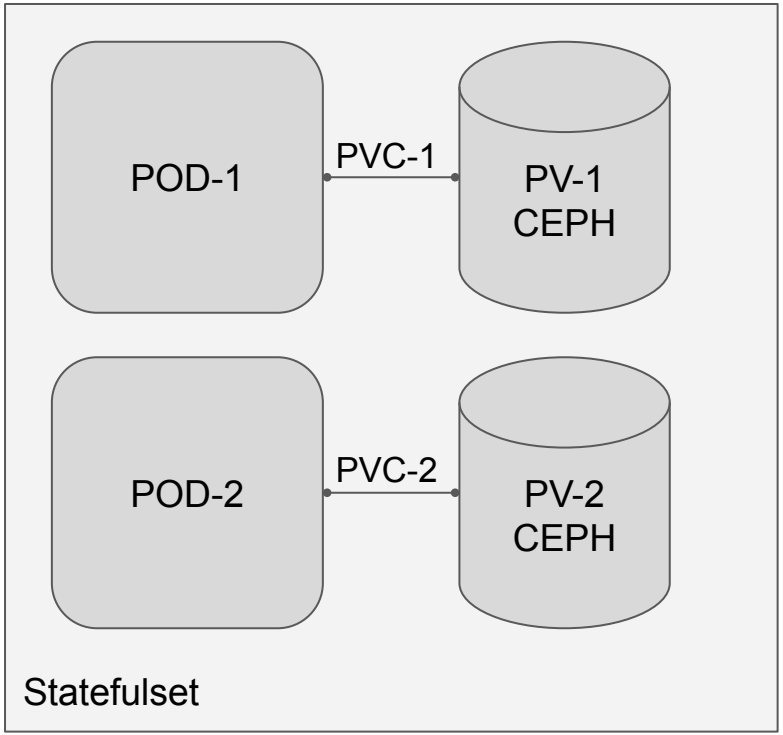
- PodAntiAffinity
  - *не обязательно
- PodDisruptionBudget
- Много операций с данными == iowait;
- Обязательное квотирование и ограничения
- Stateful, replicas 3+

Сетевые Volume в коммуналке

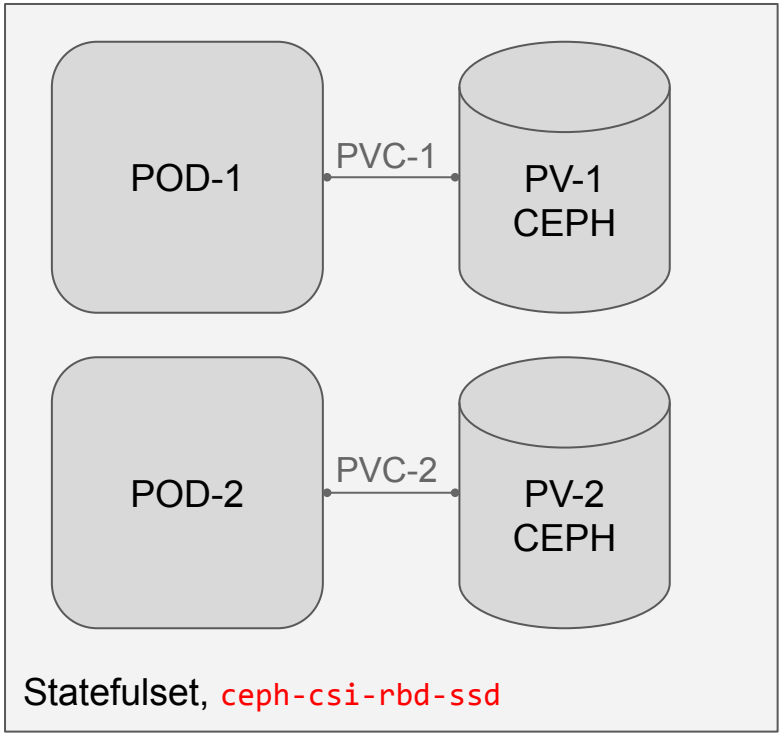
- PodAntiAffinity
  - *не обязательно
- PodDisruptionBudget
- Много операций с данными == iowait;
- Обязательное квотирование и ограничения
- Stateful, replicas 3+
- Бэкап!

Разные интересные штуки с практическими примерами

Миграция с одного на другой тип PV

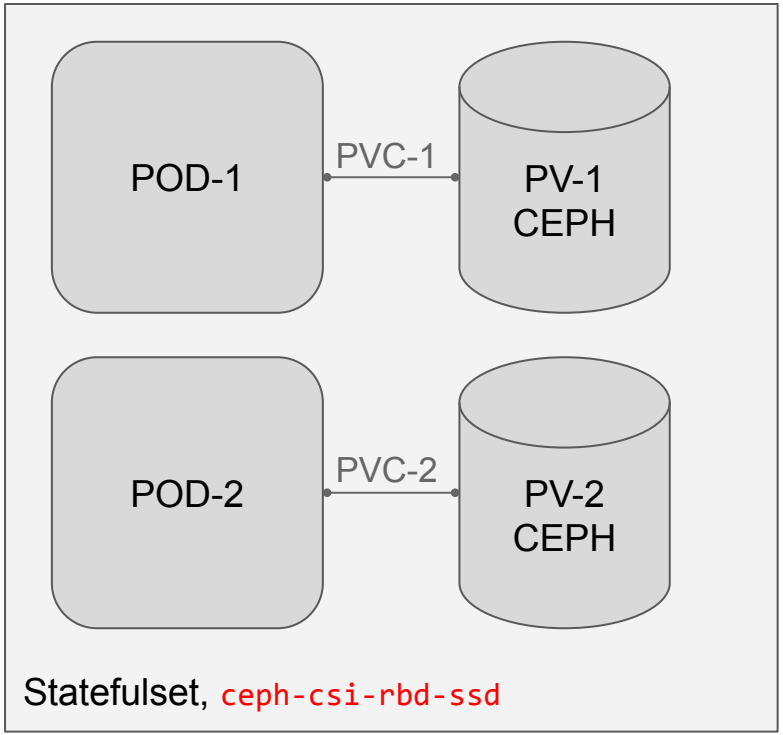


Миграция с одного на другой тип PV




```
kind: StatefulSet

spec:

  volumeClaimTemplates(PVC):

    spec:

      storageClassName: ceph-csi-rbd-ssd
```

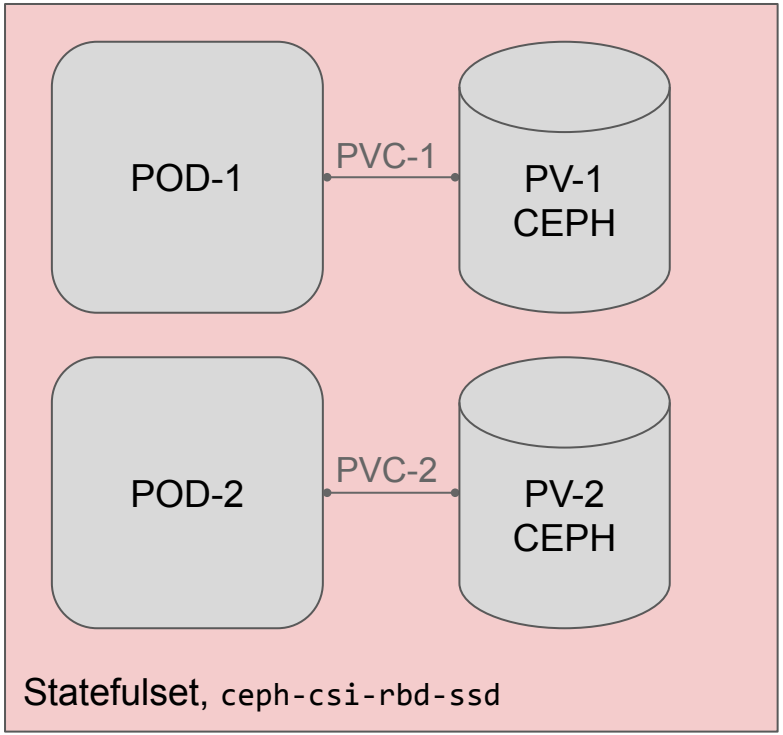
Миграция с одного на другой тип PV




```
kind: StatefulSet

spec:

  volumeClaimTemplates(PVC):

    spec:

      storageClassName: ceph-csi-rbd-ssd
```

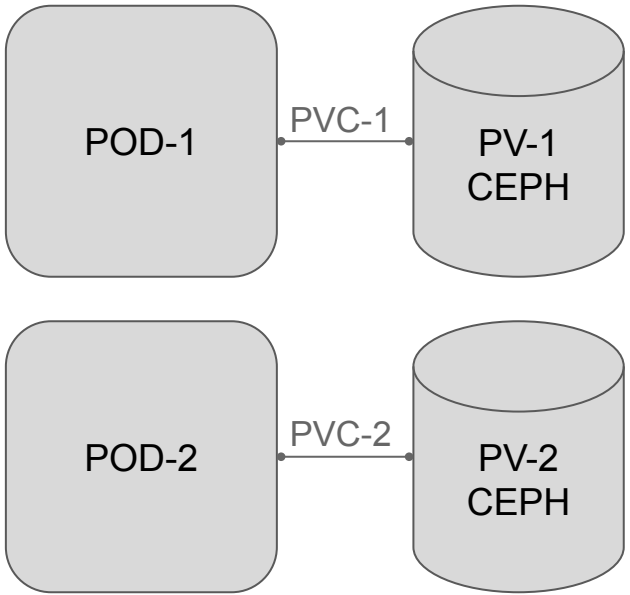
k8s не даёт изменить volumeClaimTemplates

Миграция с одного на другой тип PV



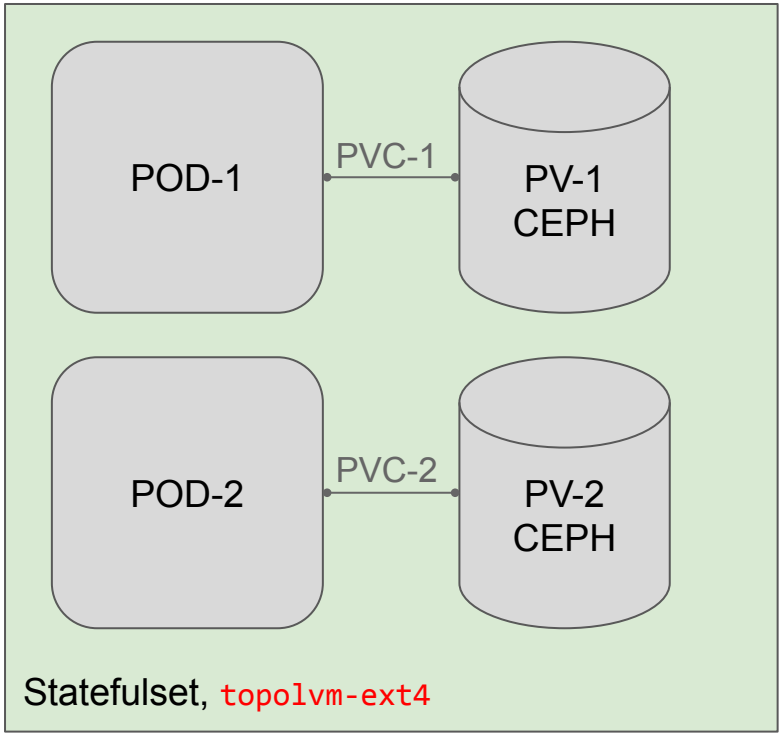

```
$ kubectl delete sts/name --cascade=orphan
```

Миграция с одного на другой тип PV



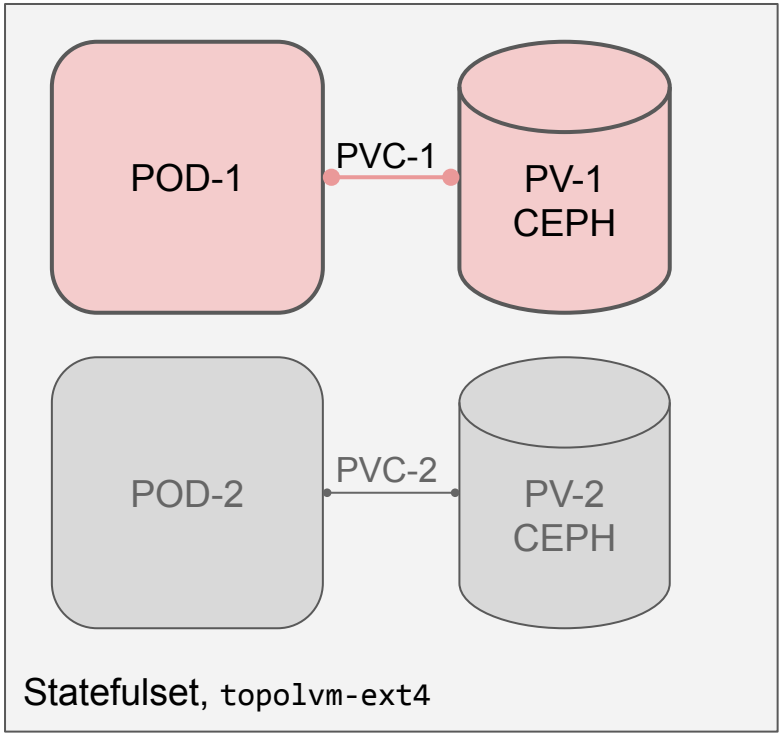

```
$ kubectl delete sts/name --cascade=orphan
```

Миграция с одного на другой тип PV



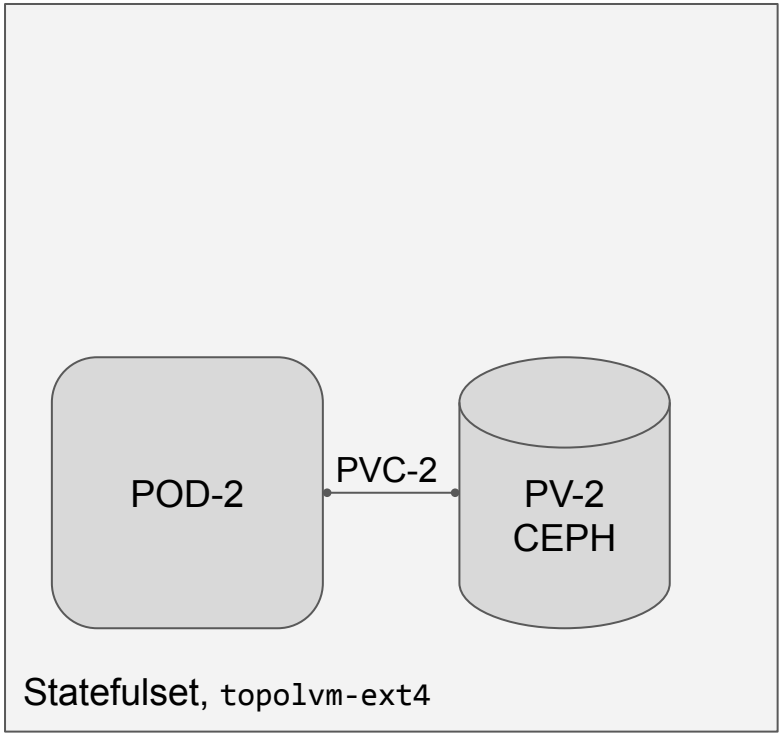

```
$ kubectl create -f sts_new_storage

---

kind: StatefulSet

spec:

  volumeClaimTemplates(PVC):

    spec:

      storageClassName: topolvm-ext4
```

Миграция с одного на другой тип PV



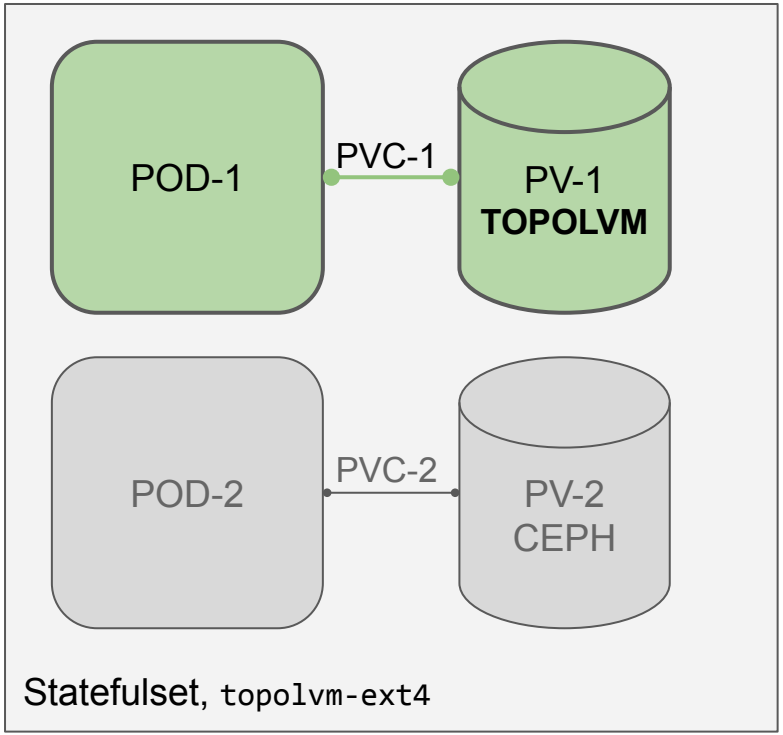

```
$ kubectl delete POD-1

$ kubectl delete PVC-1
```

Миграция с одного на другой тип PV



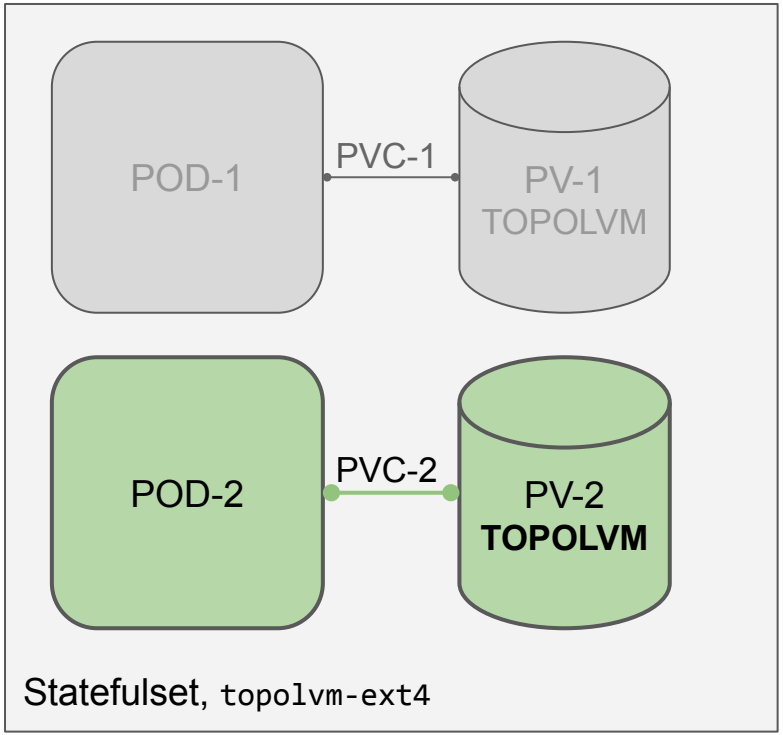
K8s пересоздаёт PV и POD

Миграция с одного на другой тип PV



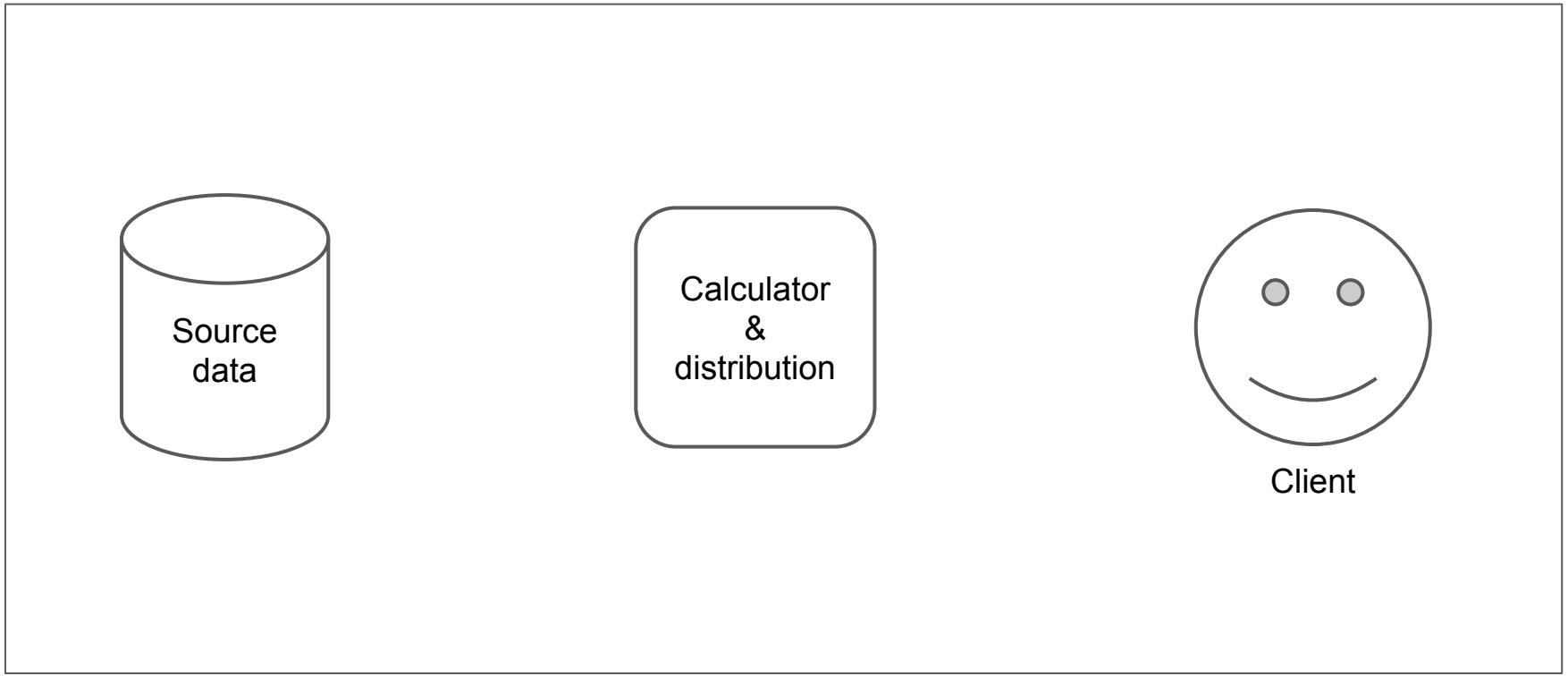
Повторяем с другими POD..

Миграция с одного на другой тип PV

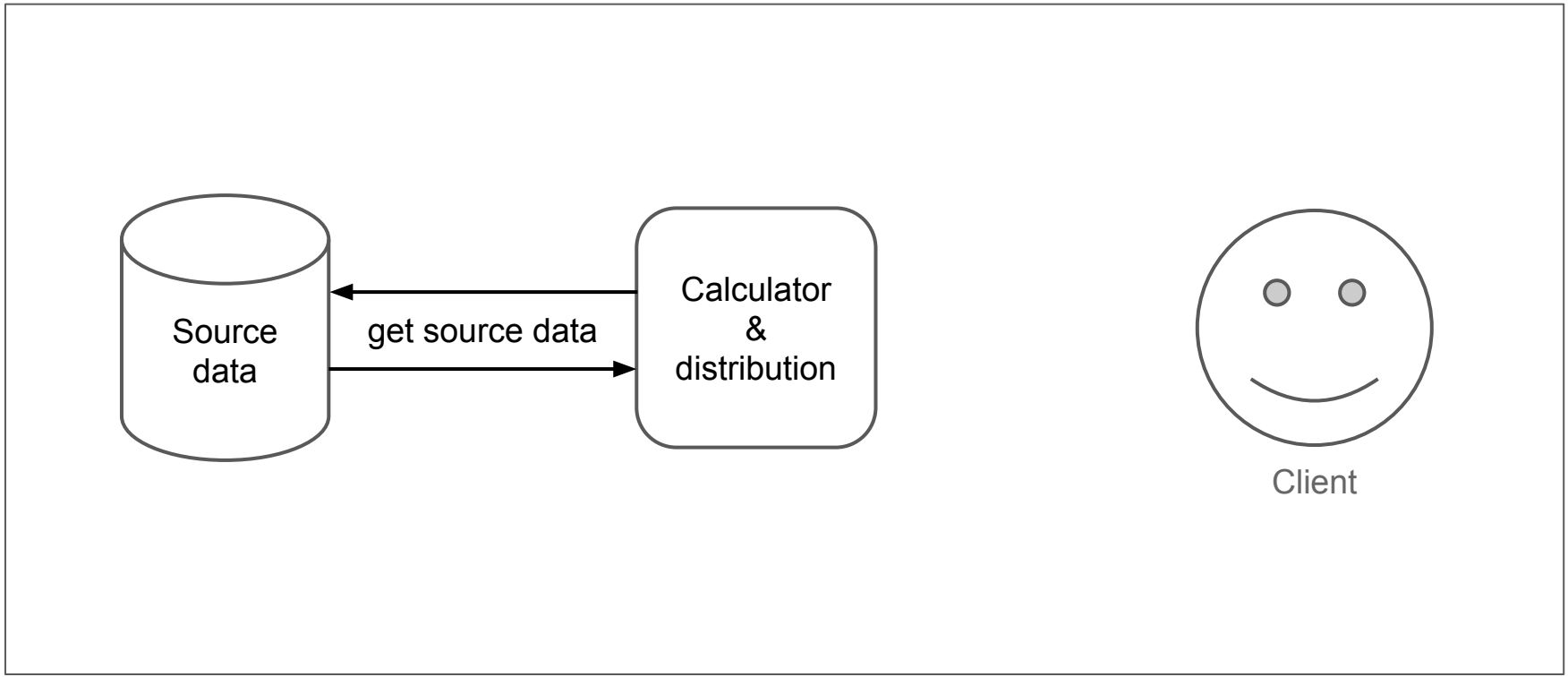


Profit!

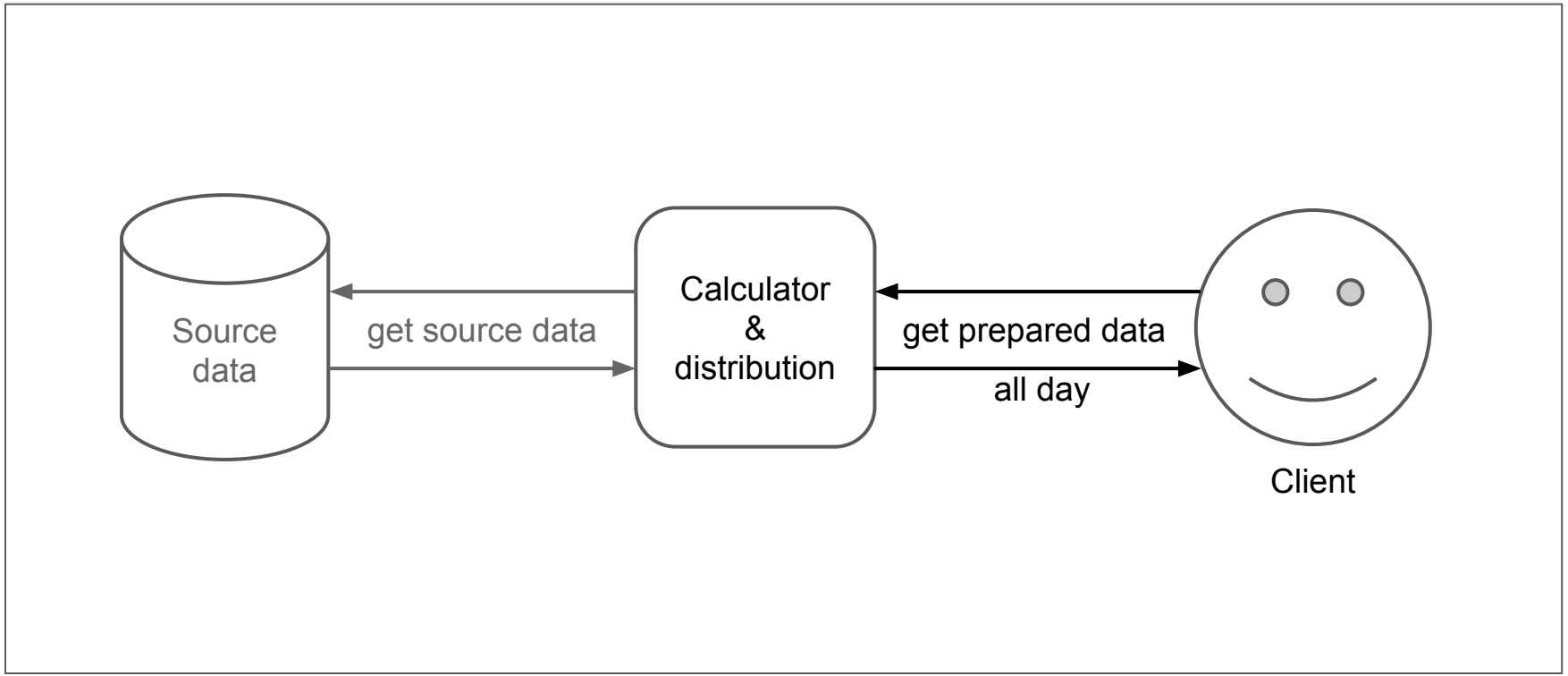
Sidecar и Volumes



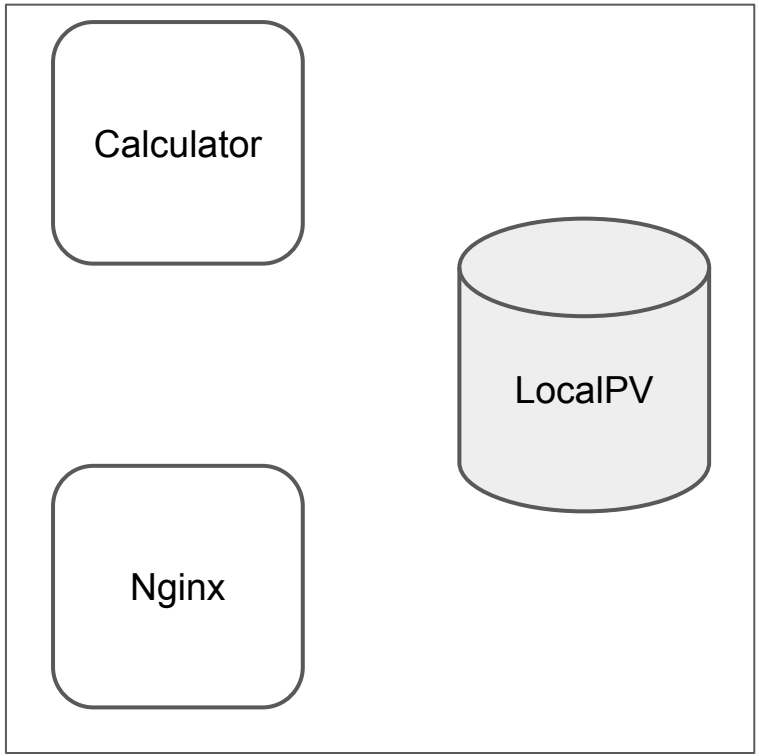
Sidecar и Volumes



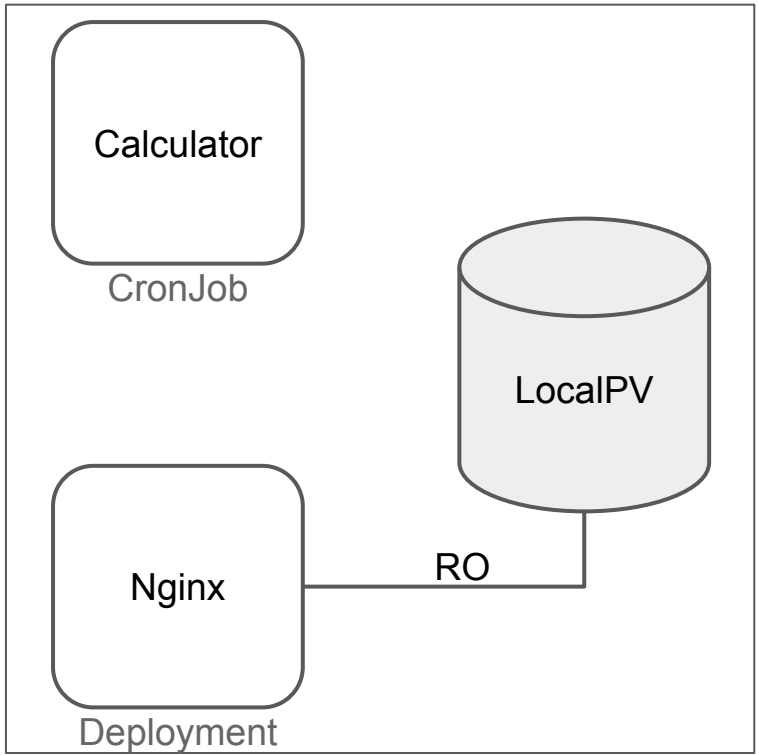
Sidecar и Volumes



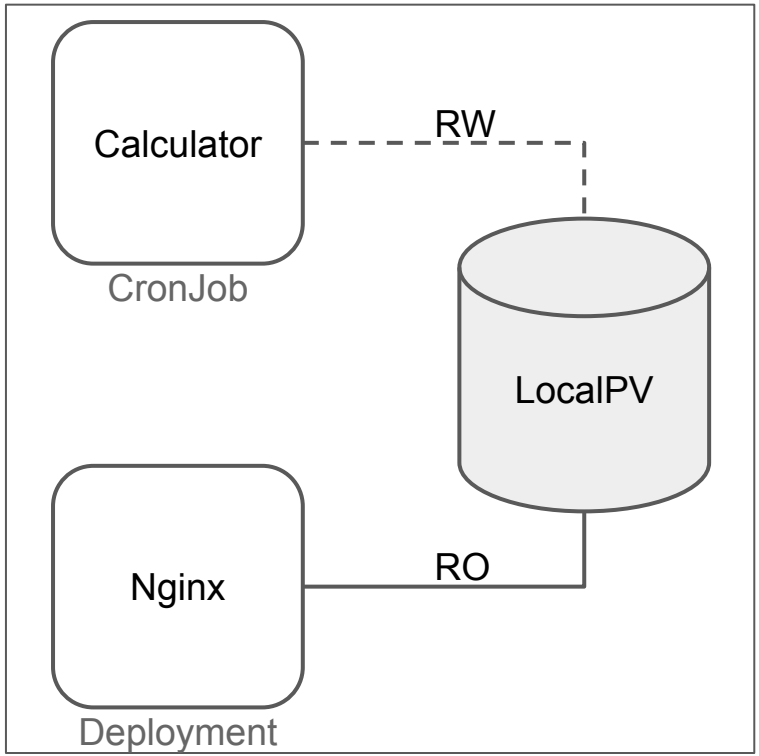
Sidecar и Volumes



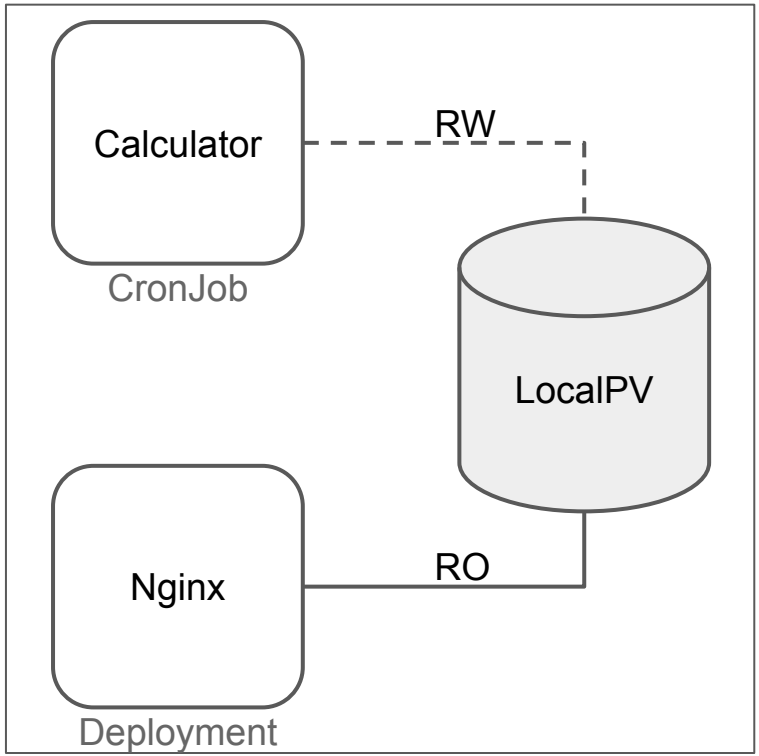
Sidecar и Volumes



Sidecar и Volumes

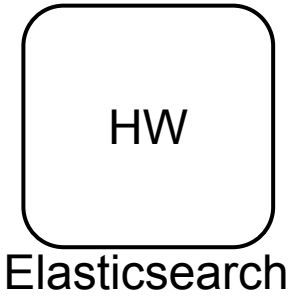


Sidecar и Volumes

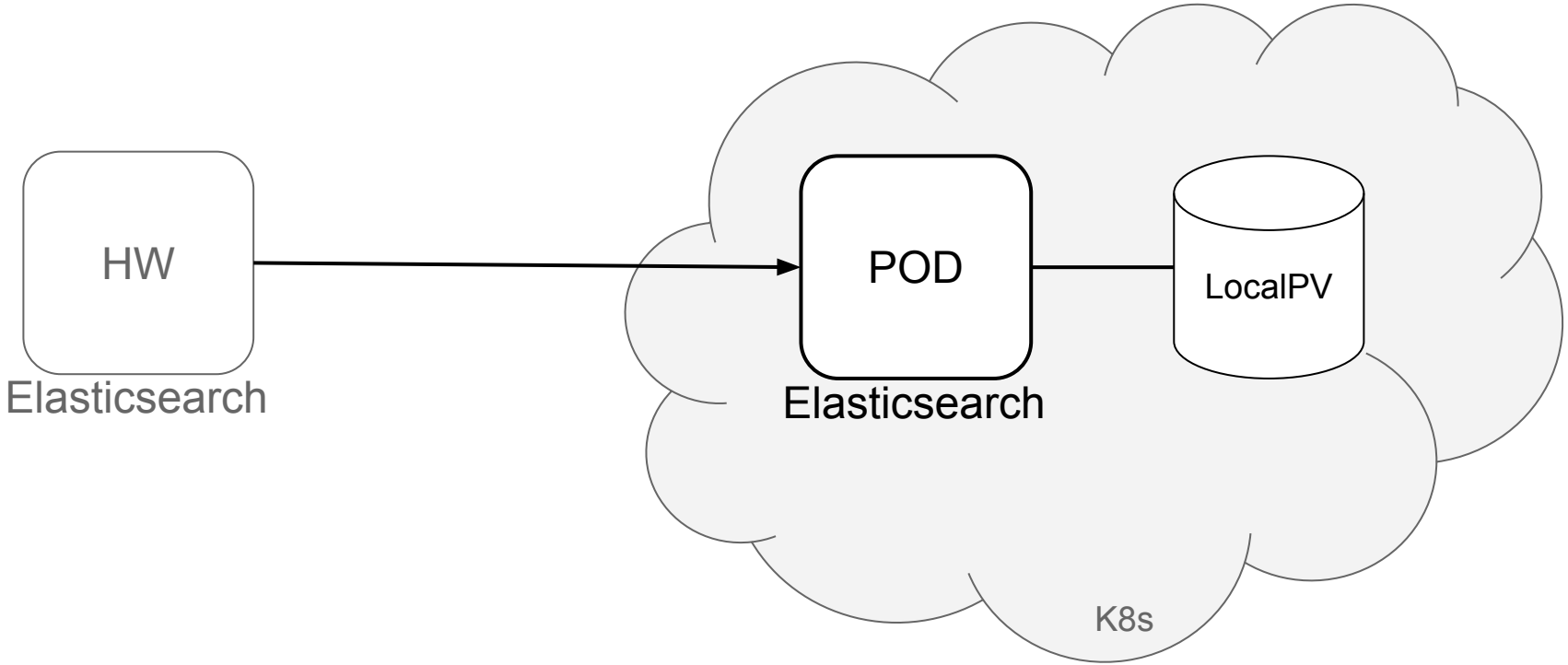



```
metadata:
 name: calculator
…
      affinity:
       podAffinity:
         requiredDuringSchedulingIgnoredDuringExecution:
         - labelSelector:
             matchExpressions:
             - key: app.kubernetes.io/instance
               operator: In
               values:
               - nginx
           topologyKey: kubernetes.io/hostname
```

Limits.memory, stateful & iowait

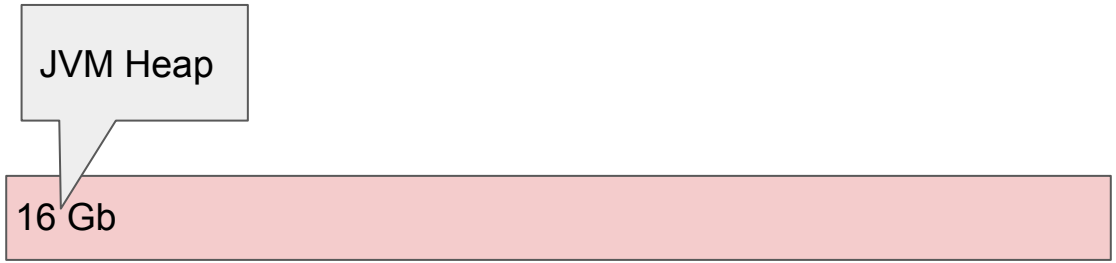


Limits.memory, stateful & iowait



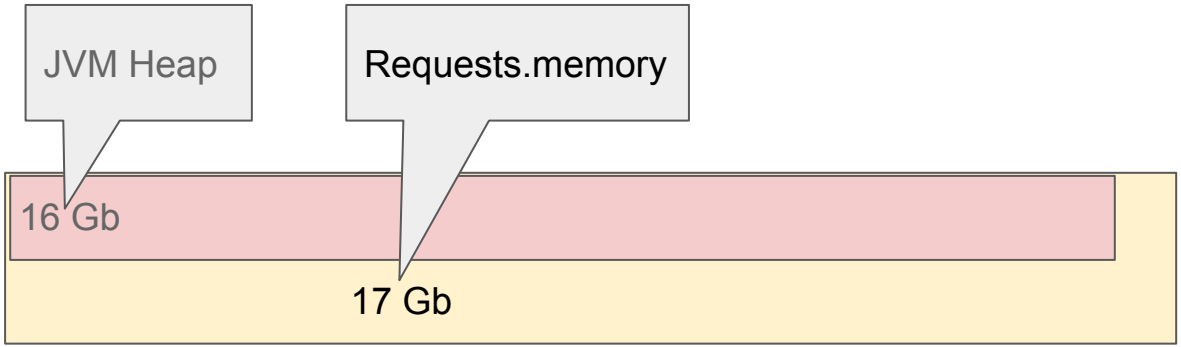
Limits.memory, stateful & iowait

Elasticsearch. JAVA



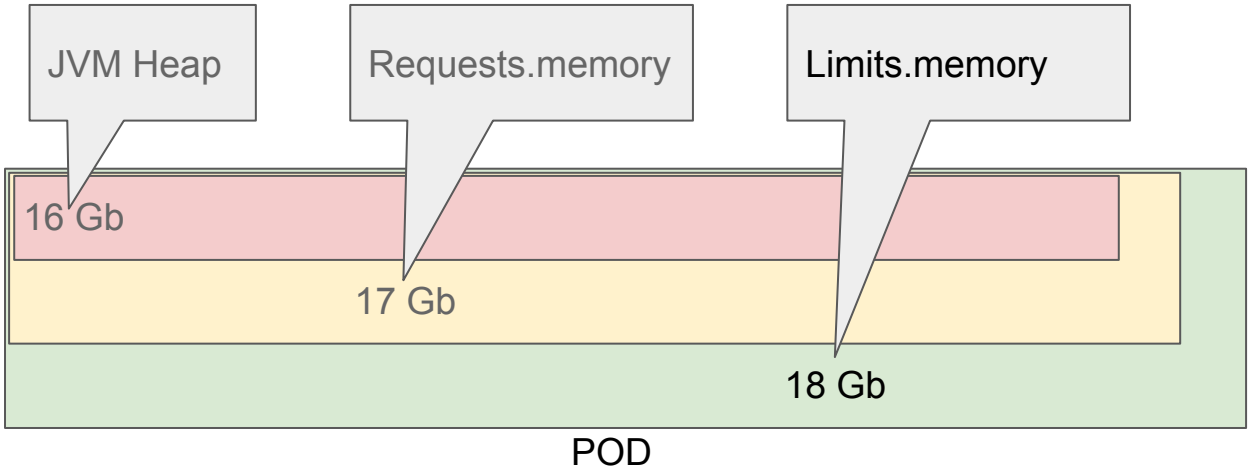
Limits.memory, stateful & iowait

Elasticsearch. JAVA



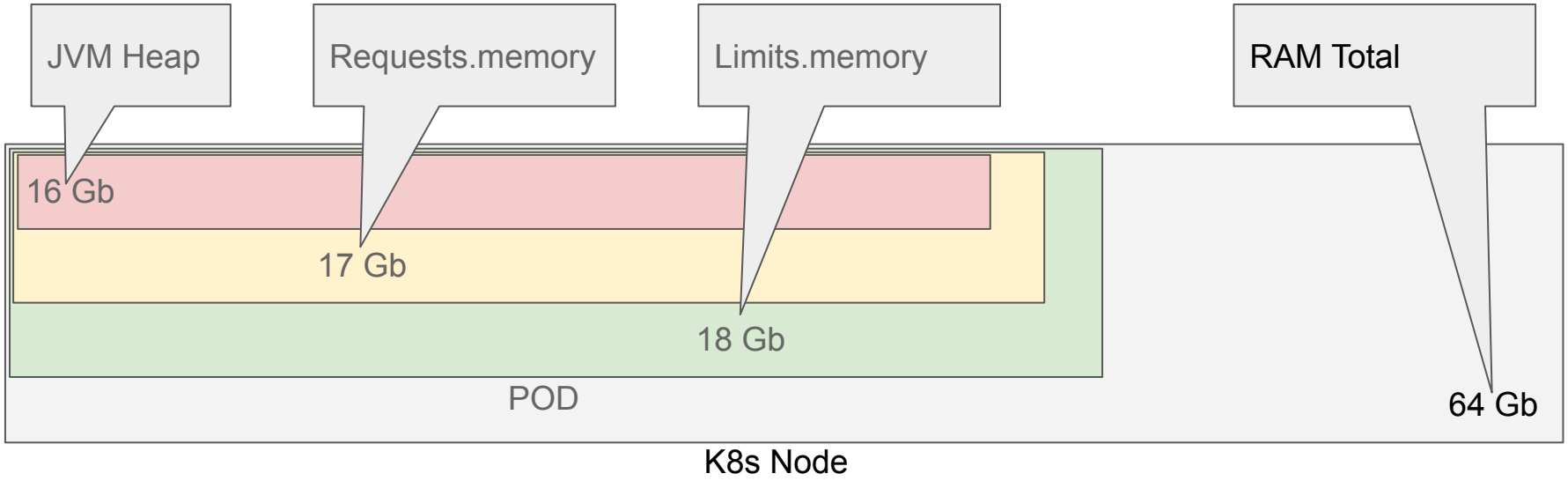
Limits.memory, stateful & iowait

Elasticsearch. JAVA



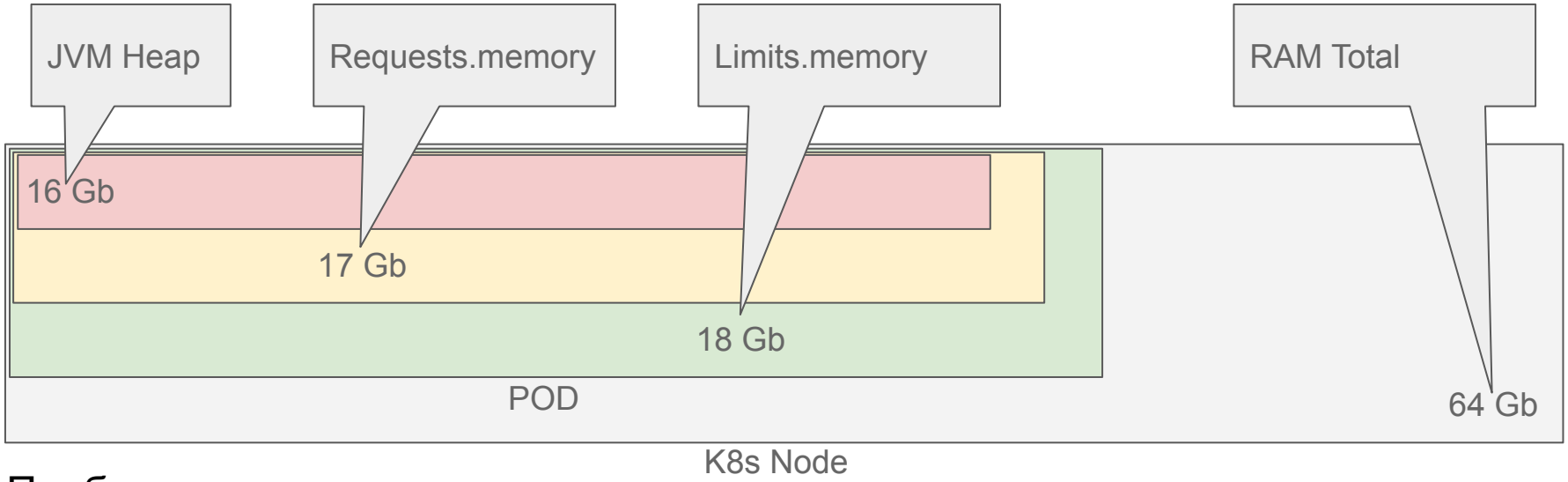
Limits.memory, stateful & iowait

Elasticsearch. JAVA



Limits.memory, stateful & iowait

Elasticsearch. JAVA

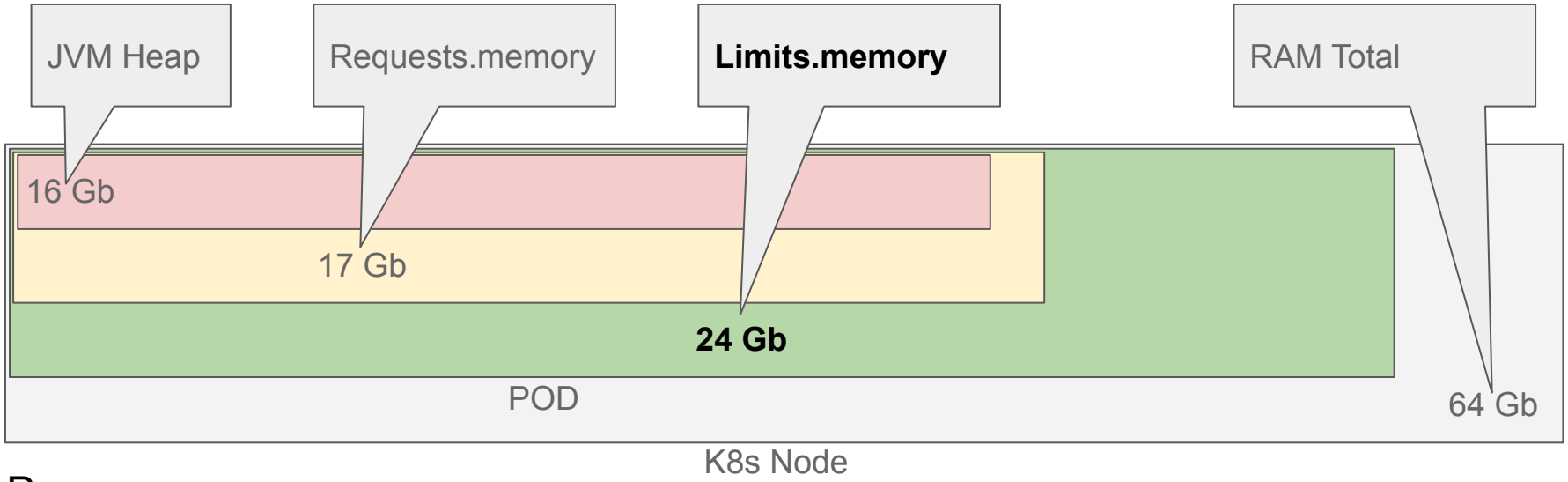


Проблема

- Активная работа с диском
- Сильный IOwait >15%

Limits.memory, stateful & iowait

Elasticsearch. JAVA



Решение

- Расширяем limits.memory для PageCache!
- IOwait проходит

Generic ephemeral volumes

- Сетевое или локальное хранилище
- Хранилище заданного размера
- Kubernetes сам выбирает куда зашедулить
- Типовые операции с томом(snapshot, resize)
- Можно наполнить начальными данными

Generic ephemeral volumes

Требования:

- Гарантия наличия места

Generic ephemeral volumes

Требования:

- Гарантия наличия места
- Стартовать в любой ситуации

Generic ephemeral volumes

Требования:

- Гарантия наличия места
- Стартовать в любой ситуации
- Быстрый диск - значит **локальный**

Generic ephemeral volumes. Пример

- Cronjob/Job
  - Расчётный сервис, большой объём данных

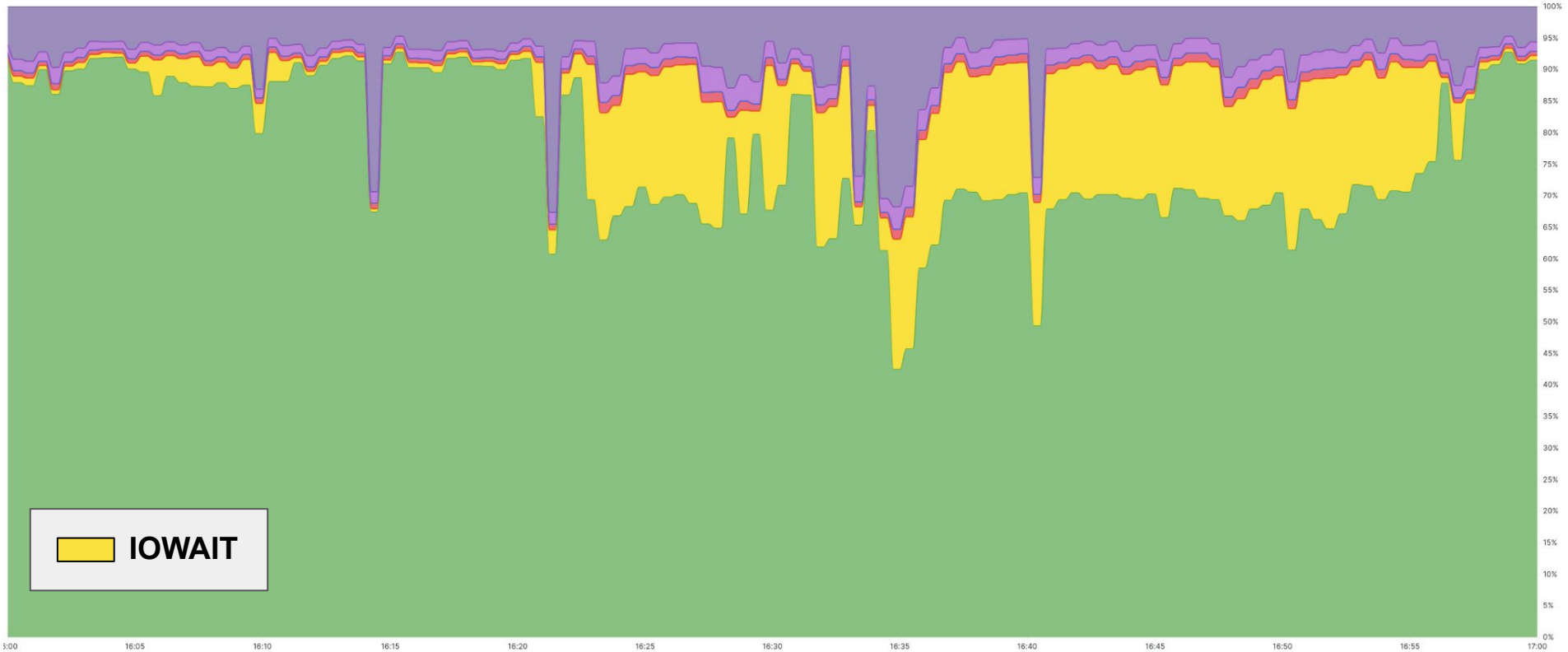
Generic ephemeral volumes. Пример

- Cronjob/Job
  - Расчётный сервис, большой объём данных
- Gitlab Runner
  - Кэш сборок

Generic ephemeral volumes. Пример

- Cronjob/Job
  - Расчётный сервис, большой объём данных
- Gitlab Runner
  - Кэш сборок
- Deployment
  - Скачать много данных на старте, но можно их терять

Ограничения IOPS



Ограничения IOPS. Как у нас?

- Подрывались 1 раз

Ограничения IOPS. Как у нас?

- Подрывались 1 раз
  - 2023 - SSD | 2024+ - NVMe

Metric	NVMe SSD	SATA SSD
IOPS	600,000+	100,000+
Latency (average)	80 microseconds	120 microseconds
Sequential Read Speed	3,500+ MB/s	550+ MB/s
Sequential Write Speed	3,000+ MB/s	520+ MB/s
Random Read Speed	400,000+ IOPS	40,000+ IOPS
Random Write Speed	400,000+ IOPS	80,000+ IOPS

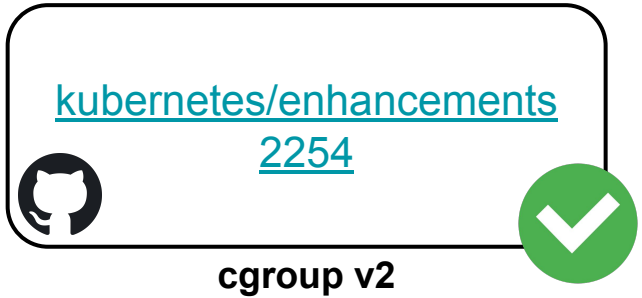
Ограничения IOPS. Как у нас?

- Подрывались 1 раз
  - 2023 - SSD | 2024+ - NVMe
- Ничего не делаем,

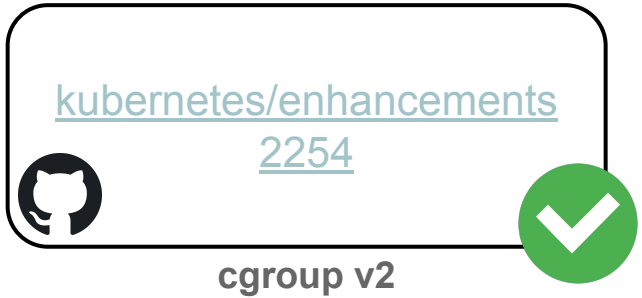
  костыль = 6+ месяцев

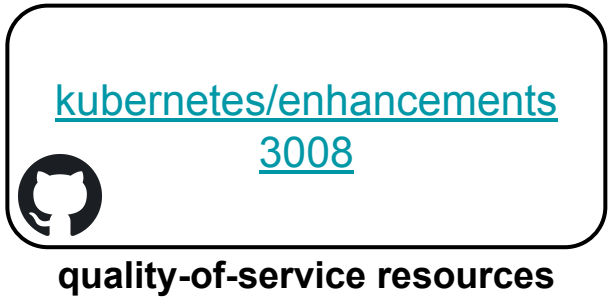
Metric	NVMe SSD	SATA SSD
IOPS	600,000+	100,000+
Latency (average)	80 microseconds	120 microseconds
Sequential Read Speed	3,500+ MB/s	550+ MB/s
Sequential Write Speed	3,000+ MB/s	520+ MB/s
Random Read Speed	400,000+ IOPS	40,000+ IOPS
Random Write Speed	400,000+ IOPS	80,000+ IOPS

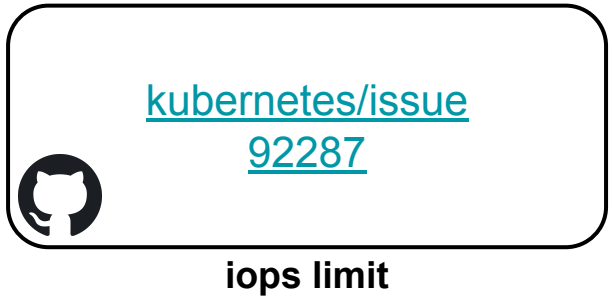
Ограничения IOPS. А что сообщество?



Ограничения IOPS. А что сообщество?







Плавный переход к результатам

Итоги: какие PV у нас используются

Список storageClass с которыми мы работаем:

- **topolvm-ext4** —локальное, стандартное, на SSD
- **topolvm-hdd-ext4** — локальное, медленное, большое, на HDD

Итоги: какие PV у нас используются

Список storageClass с которыми мы работаем:

- **topolvm-ext4** —локальное, стандартное, на SSD
- **topolvm-hdd-ext4** — локальное, медленное, большое, на HDD
- **ceph-csi-rbd** — сетевое, медленное, большое
- **ceph-csi-rbd-ssd** — сетевое, быстрое, маленькое

Итоги: какие PV у нас используются

Список storageClass с которыми мы работаем:

- **topolvm-ext4** —локальное, стандартное, на SSD
- **topolvm-hdd-ext4** — локальное, медленное, большое, на HDD
- **ceph-csi-rbd** — сетевое, медленное, большое
- **ceph-csi-rbd-ssd** — сетевое, быстрое, маленькое
- **nfs-<namespace>** — сетевое, с бэком на Netapp, ReadWriteMany

Итоги: правильный stateful

- Думать о кворуме

Итоги: правильный stateful

- Думать о кворуме
- Выпадение 1 POD — это норма
  - Настроить защиту!

Итоги: правильный stateful

- Думать о кворуме
- Выпадение 1 POD — это норма
  - Настроить защиту!
- Больше типов PV — не насилуем инфраструктуру

Итоги: правильный stateful

- Думать о кворуме
- Выпадение 1 POD — это норма
  - Настроить защиту!
- Больше типов PV — не насилуем инфраструктуру
- Можем советовать крутой оператор LocalPV
  - github.com/topolvm/topolvm 

Итоги: правильный stateful

- Думать о кворуме
- Выпадение 1 POD — это норма
  - Настроить защиту!
- Больше типов PV — не насилуем инфраструктуру
- Можем советовать крутой оператор LocalPV
  - github.com/topolvm/topolvm 
- Предоставляем разные типы PV и больше не поднимаем VM для Stateful

Итоги: правильный stateful

- Думать о кворуме
- Выпадение 1 POD — это норма
  - Настроить защиту!
- Больше типов PV — не насилуем инфраструктуру
- Можем советовать крутой оператор LocalPV
  - github.com/topolvm/topolvm 
- Предоставляем разные типы PV и больше не поднимаем VM для Stateful
- Бэкап!

Спасибо!

Вопросы?

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