avito.tech Москва 2024

Особенности адаптации классических СУБД к платформе DBaaS/k8s

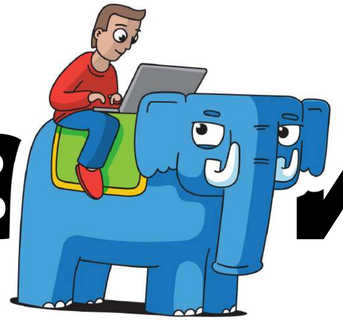
Андрей Лярский Engineer@SQL/DBA

Андрей Лярский

Engineer в DBA/SQL

PostgreSQL DBA.
Работаю в Авито в команде SQL@DBA с 2021 года.
Стек команды: PostgreSQL, CockroachDB, Golang.
Переобувался из Linux Admin → DBA, теперь DBA → SRE.
tg: @oxumorron

Как мы адаптировали PostgreSQL к платформе DBaaS/k8s



PostgreSQL является одной из наиболее распространённых «классических» СУБД в Авито

PostgreSQL в Avito

> 2k

Баз

Без учёта временных баз для тестов

> 100ТВ

Суммарный объём баз

Базы живут на локальных SSD

> 300

Физических машин

Машины разных поколений, могут быть существенные отличия по ресурсам

~ 4-5

Инженеров в команде

Инженеры тоже разных поколений

Ожидания от баз

1 2 3 4

Ожидания от базВ общем виде

High Availability & Fault Tolerance

Ожидания от базВ общем виде

- High Availability & Fault Tolerance
- Durability

Ожидания от базВ общем виде

- High Availability & Fault Tolerance
- Durability
- Scalability

Ожидания от базВ общем виде

- High Availability & Fault Tolerance
- Observability
- Durability
- Scalability

Ожидания от базPostgreSQL из коробки

High Availability & Fault Tolerance
Native: Streaming Replication

Ожидания от базPostgreSQL из коробки

- **High Availability & Fault Tolerance**
  Native: Streaming Replication

- **Durability**Native: Backup + WAL-archive
  = PITR

Ожидания от базPostgreSQL из коробки

- **High Availability & Fault Tolerance**
  Native: Streaming Replication

- **Durability**Native: Backup + WAL-archive
  = PITR

- **Scalability**Native: вертикальное (с оговорками)

Ожидания от базPostgreSQL из коробки

- **High Availability & Fault Tolerance**
  Native: Streaming Replication
- **Durability**Native: Backup + WAL-archive
  = PITR
- **Scalability**Native: вертикальное (с оговорками)
- **Observability**Native: pg_stat_*, csvlog/jsonlog

Ожидания от базPostgreSQL со сторонними решениями (3-rd Party)

High Availability & Fault Tolerance
Native: Streaming Replication3-rd Party:
Patroni + DCS

Ожидания от базPostgreSQL со сторонними решениями (3-rd Party)

- **High Availability & Fault Tolerance**
  Native: Streaming Replication3-rd Party:
  Patroni + DCS

- **Durability**Native: Backups + WAL-archive = PITR3-rd Party: wal-g

Ожидания от базPostgreSQL со сторонними решениями (3-rd Party)

High Availability & Fault Tolerance
Native: Streaming Replication3-rd Party: Patroni + DCS

DurabilityNative: Backups + WAL-archive = PITR3-rd Party: wal-g

ScalabilityNative: вертикальное (с оговорками)3-rd Party: spqr (с оговорками)/app-side

Ожидания от базPostgreSQL со сторонними решениями (3-rd Party)

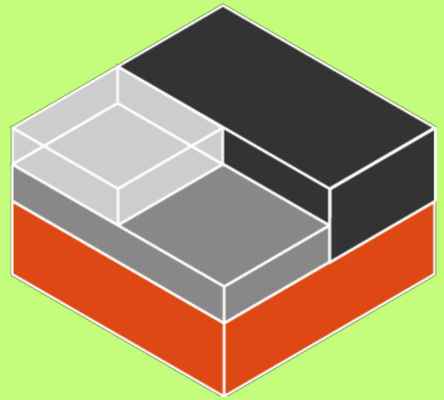
High Availability & Fault Tolerance
Native: Streaming Replication3-rd Party: Patroni + DCS

DurabilityNative: Backups + WAL-archive = PITR3-rd Party: wal-g

ScalabilityNative: вертикальное (с оговорками)3-rd Party: spqr (с оговорками)/app-side

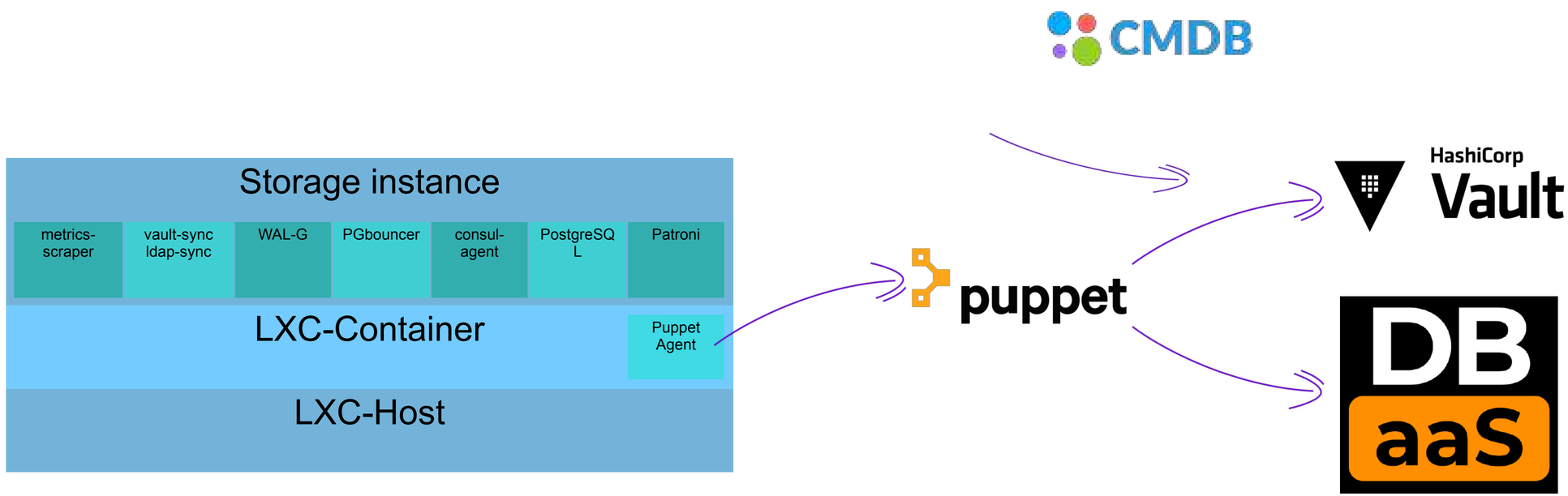
ObservabilityNative: pg_stat_*, csvlog/jsonlog3-rd Party: metrics-scraper

PostgreSQL на LXC
Отправная точка

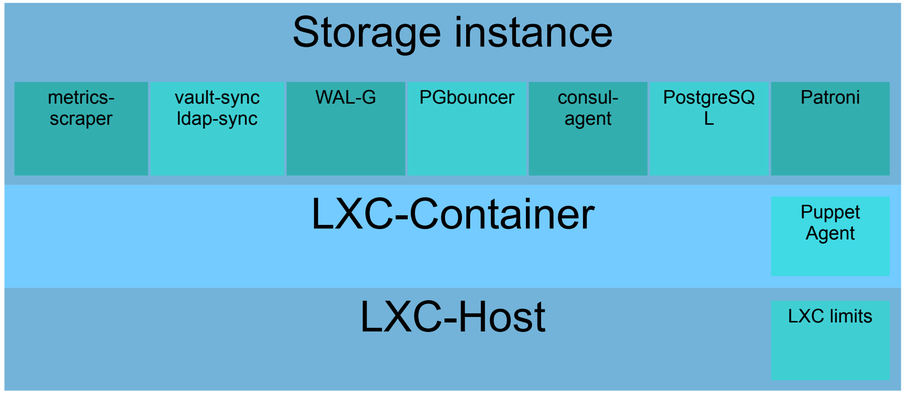


1 2 3 4

PostgreSQL на LXCIaC: Puppet



PostgreSQL на LXCЧто не так?



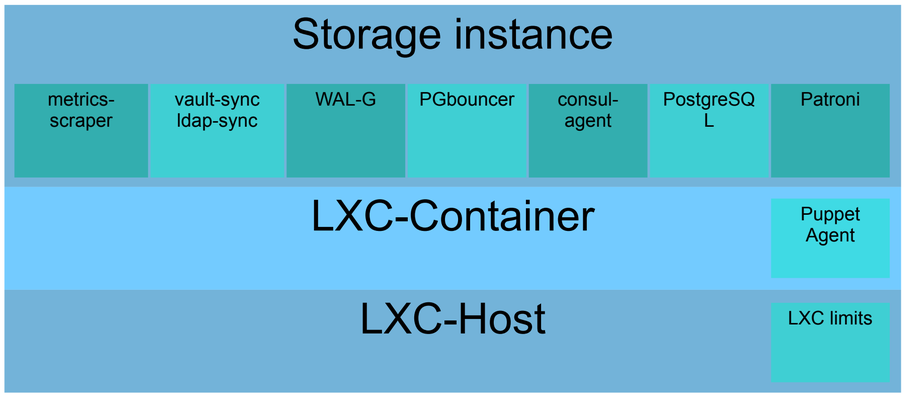
Проблема

Ресурсы планируются вручную

Решение

Пилим свой шедулер для LXC

PostgreSQL на LXCЧто не так?



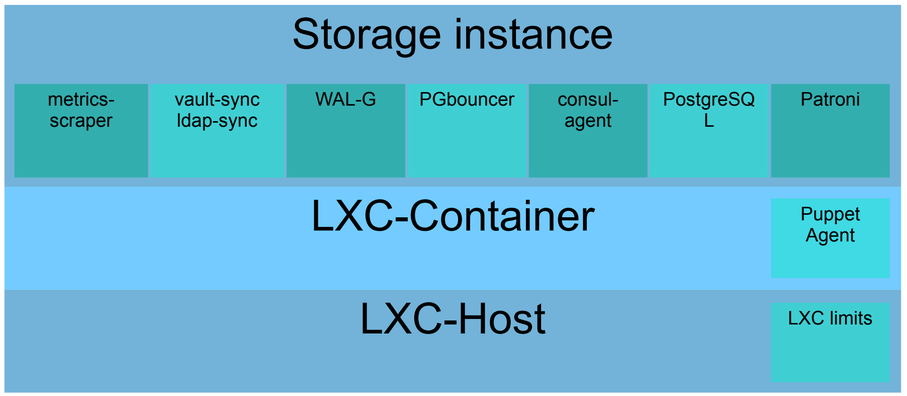
Проблема

- Ресурсы планируются вручную
- Базы создаются вручнуюСрок — 3-7 дней

Решение

- Пилим свой шедулер для LXC
- Пилим свою автоматику над LXC

PostgreSQL на LXCЧто не так?



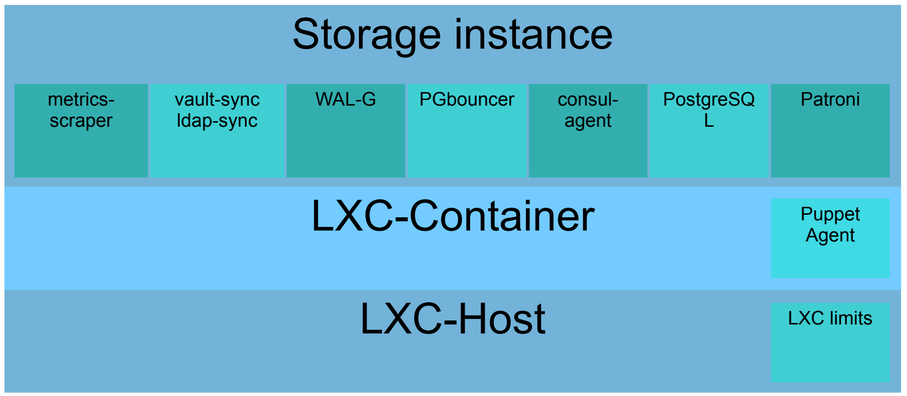
Проблема

- Ресурсы планируются вручную
- Базы создаются вручнуюСрок — 3-7 дней
- Puppet: SPOF/bottlneck

Решение

- Пилим свой шедулер для LXC
- Пилим свою автоматику над LXC
- Immutable Infrastructure

PostgreSQL на LXCЧто не так?



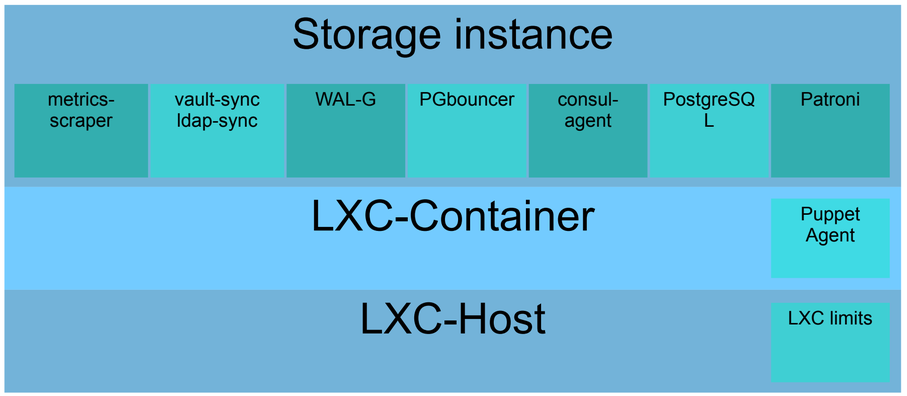
Проблема

- Ресурсы планируются вручную
- Базы создаются вручнуюСрок — 3-7 дней
- Puppet: SPOF/bottlneck
- Consul as DCS: SPOF

Решение

- Пилим свой шедулер для LXC
- Пилим свою автоматику над LXC
- Immutable Infrastructure
- Пилим per-storage DCS

PostgreSQL на LXCЧто не так?



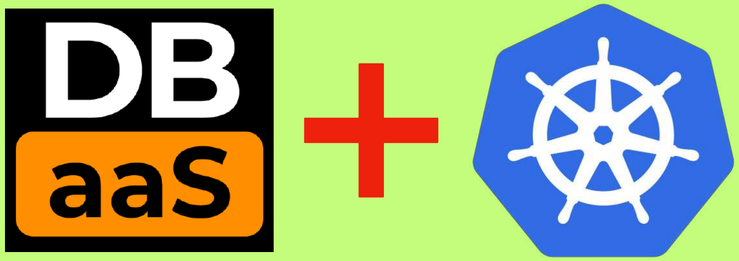
Проблема

- Ресурсы планируются вручную
- Базы создаются вручнуюСрок — 3-7 дней
- Puppet: SPOF/bottleneck
- Consul as DCS: SPOF

Решение

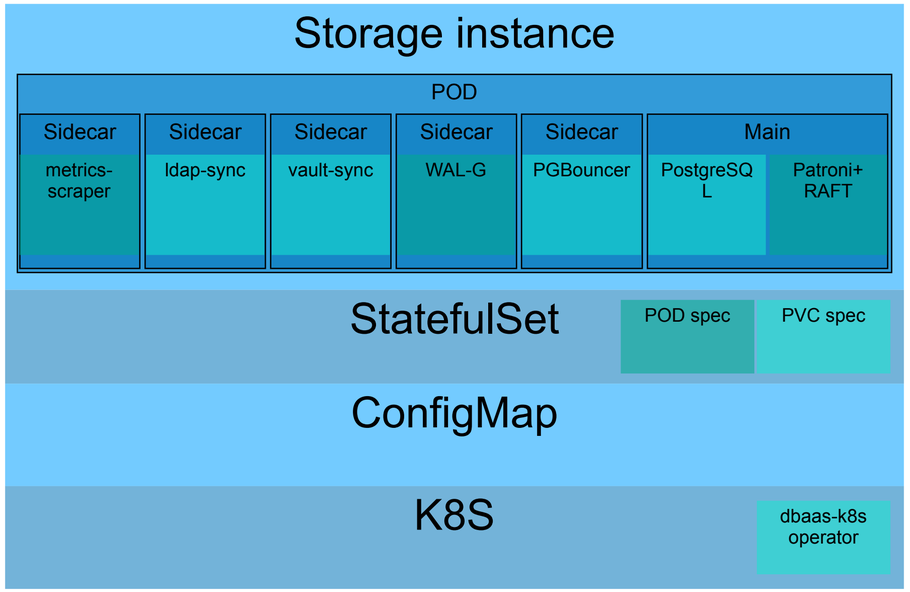
- Пилим свой шедулер для LXC
- Пилим свою автоматику над LXC
- Immutable Infrastructure
- Пилим per-storage DCS

PostgreSQL в DBaaS/k8s



1 2 3 4

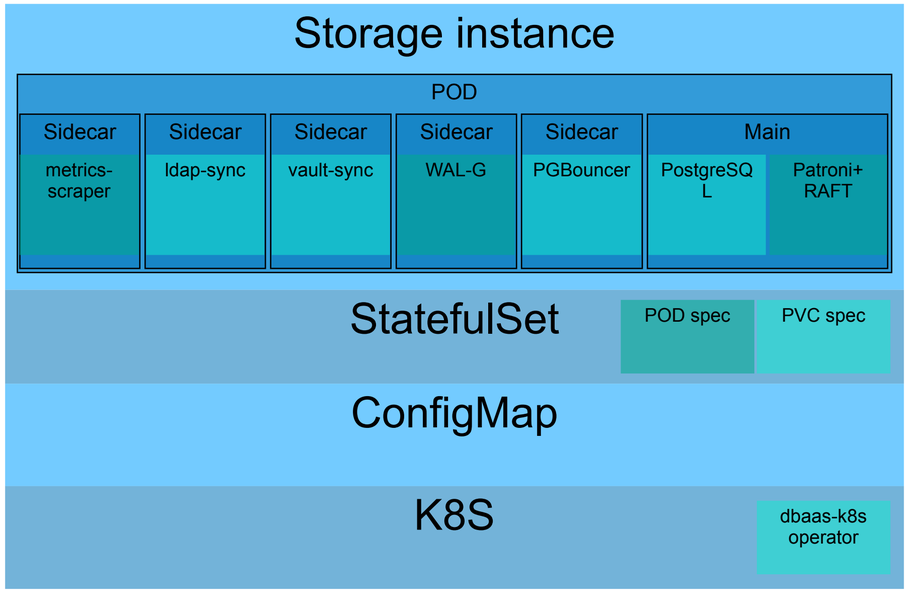
PostgreSQL в DBaaS/k8sНаивный подход



Отличия от LXC

- Перешли на Patroni RAFT.
- Все компоненты убрали в сайдкары.
- Получили scheduling «из коробки».
- Получили механику RollingUpdate от DBaaS.

PostgreSQL в DBaaS/k8sНаивный подход. Что не так?



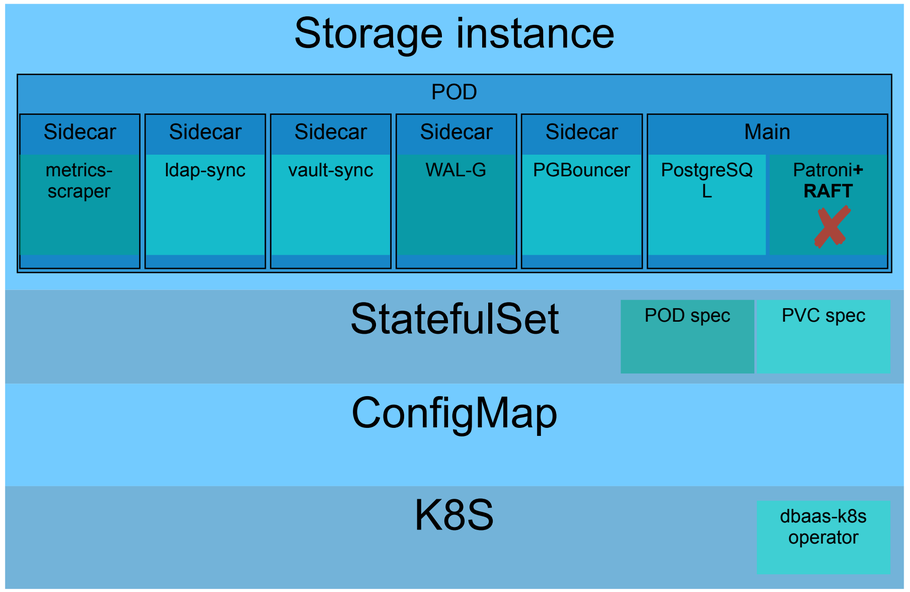
Проблема

Изменения конфиговне доезжают — некому сигналить процессам

Решение

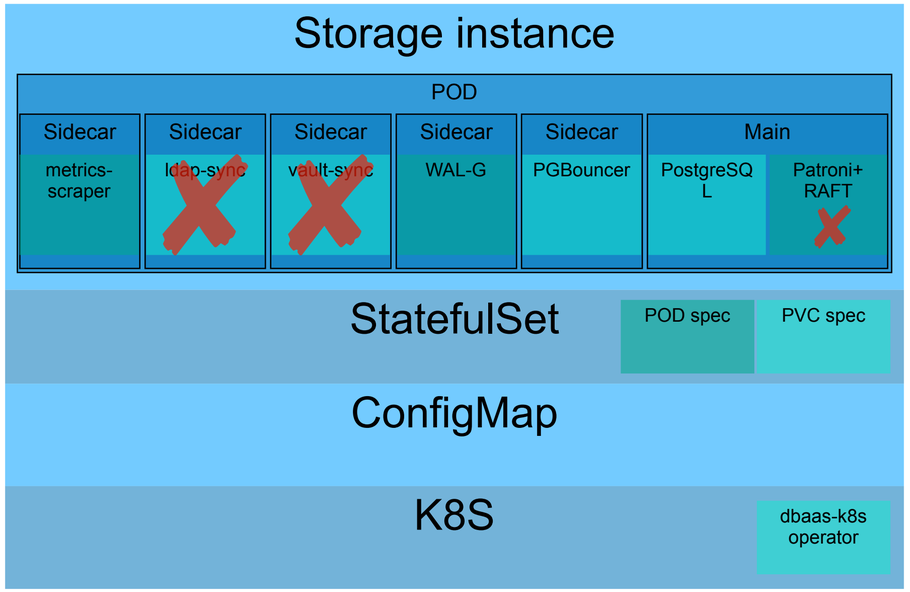
PostgreSQL Agent

PostgreSQL в DBaaS/k8sНаивный подход. Что не так?



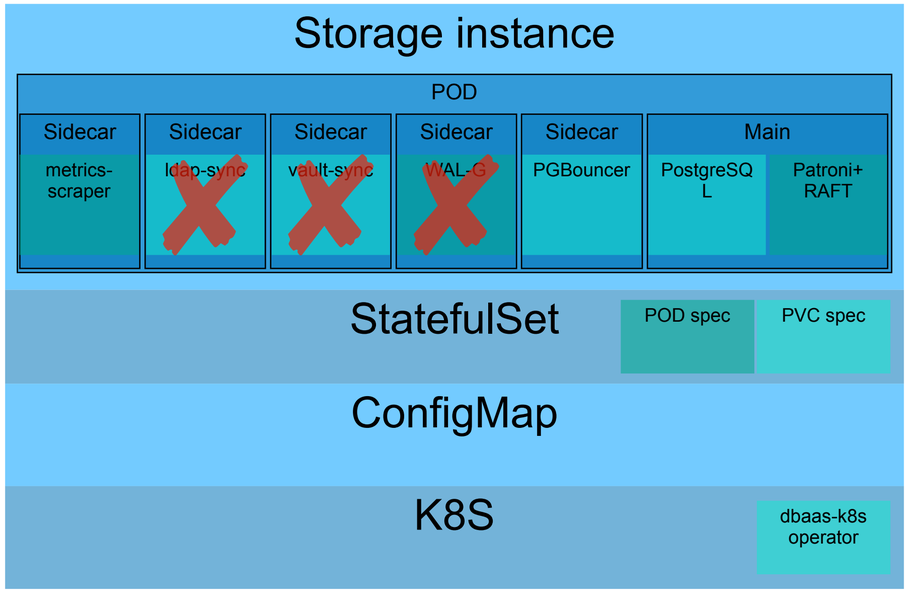
Проблема	Решение
Изменения конфиговне доезжают — некому сигналить процессам	PostgreSQL Agent
Patroni-raft оказался сыроват, а чуть позжеи вовсе deprecated	Встраиваем ETCDв PostgreSQL Agent

PostgreSQL в DBaaS/k8sНаивный подход. Что не так?



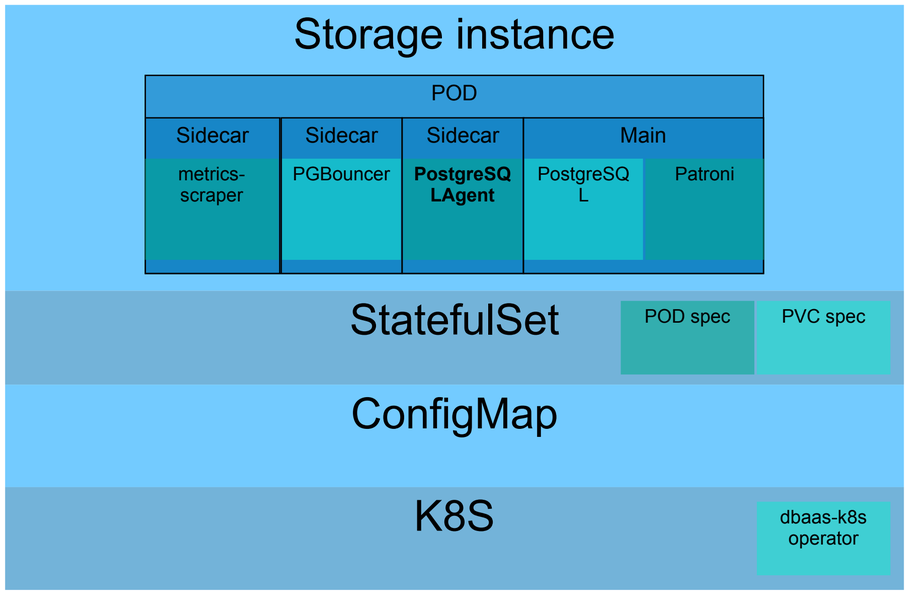
Проблема	Решение
Изменения конфиговне доезжают — некому сигналить процессам	PostgreSQL Agent
Patroni-raft оказался сыроват, а чуть позжеи вовсе deprecated	Встраиваем ETCDв PostgreSQL Agent
Много компонентов	Тащим в PostgreSQL Agent всё, что можем

PostgreSQL в DBaaS/k8sНаивный подход. Что не так?



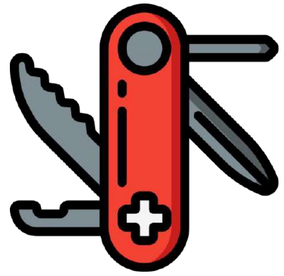
Проблема	Решение
Изменения конфиговне доезжают — некому сигналить процессам	PostgreSQL Agent
Patroni-raft оказался сыроват, а чуть позжеи вовсе deprecated	Встраиваем ETCDв PostgreSQL Agent
Много компонентов	Тащим в PostgreSQL Agent всё, что можем
Нет бэкапов, т.к. в LXCмы полагались на общийconsul-lock	Тащим в PostgreSQL Agent Backup API

PostgreSQL в DBaaS/k8sPostgreSQL Agent: Swiss Army Knife

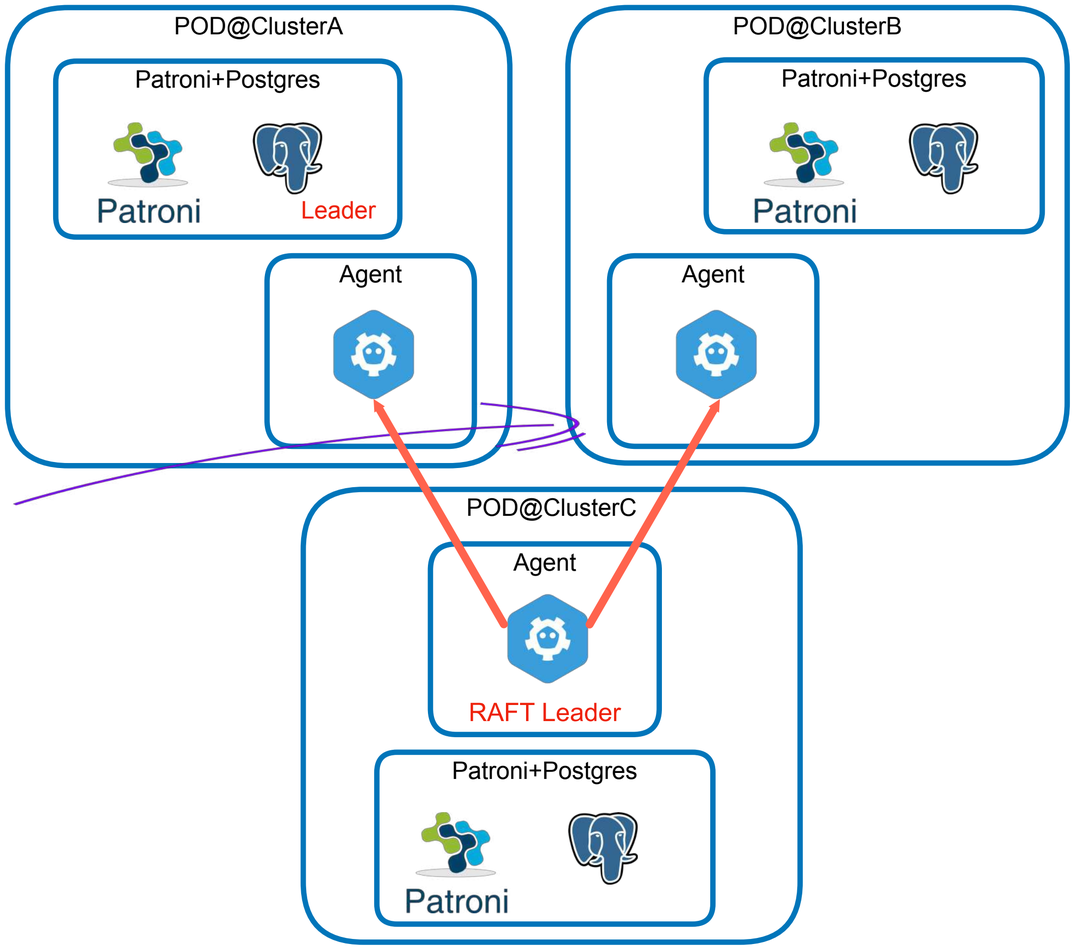


Решаемые задачи

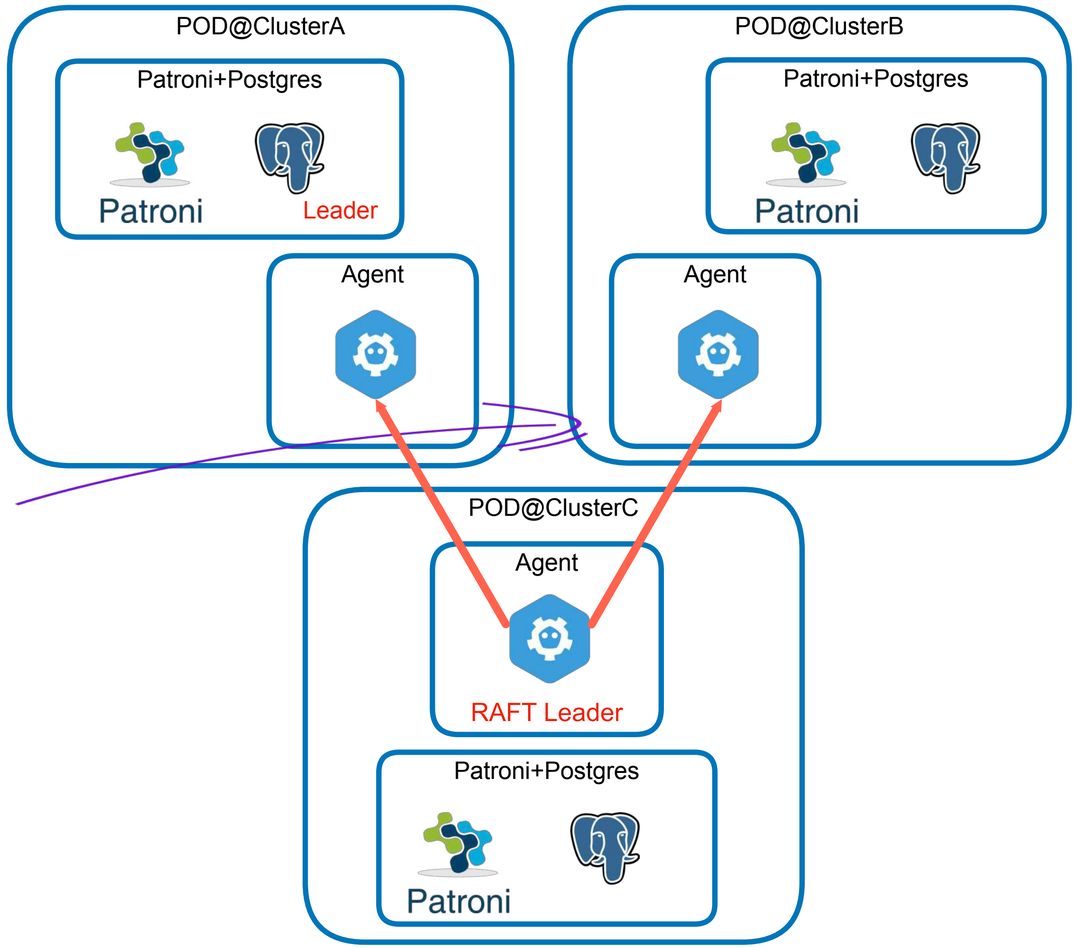
- Синхронизация
- Health-endpoints для k8s и контроллеров
- ETCD для Patroni
- Backup API



PostgreSQL в DBaaS/k8sPostgreSQL Agent: Embedded ETCD



PostgreSQL в DBaaS/k8sPostgreSQL Agent: Embedded ETCD



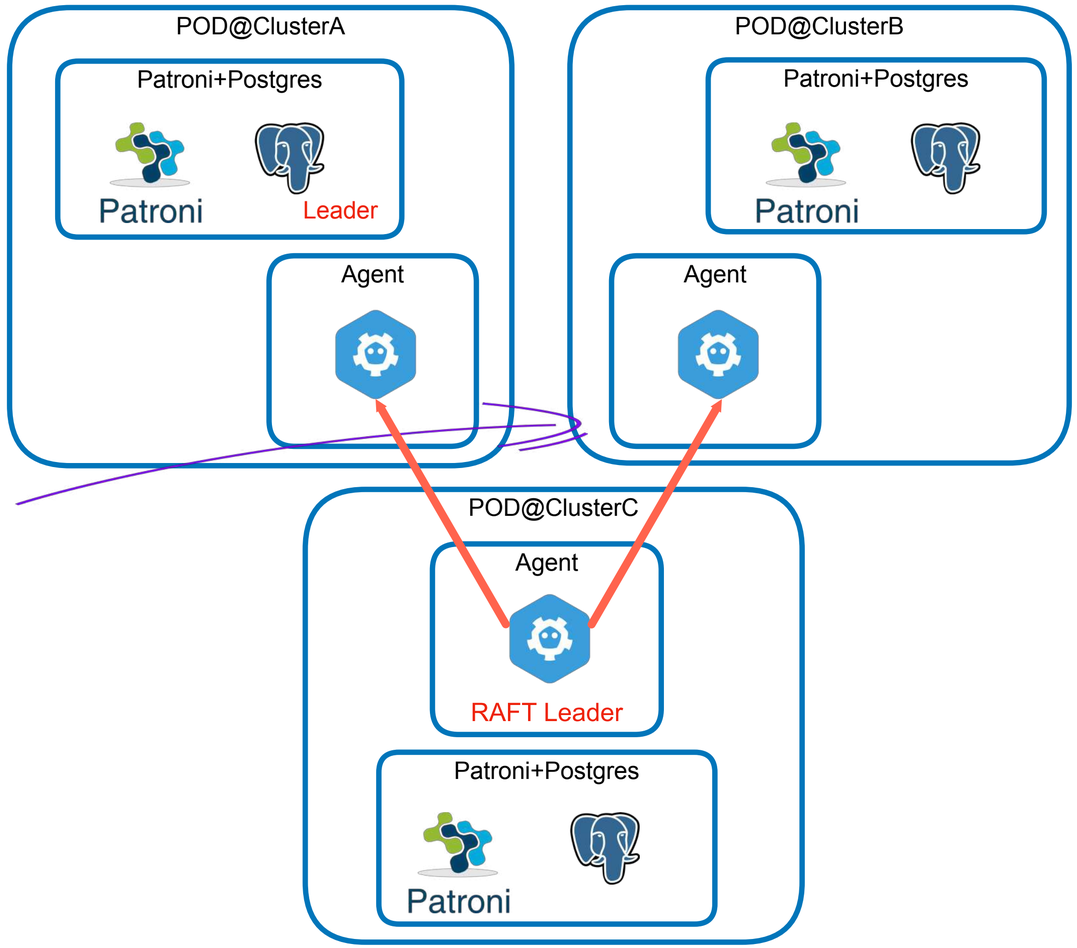
Проблема

ETCD Bootstrap

Решение

Реализуем на основеPostgreSQL Agent API

PostgreSQL в DBaaS/k8sPostgreSQL Agent: Embedded ETCD



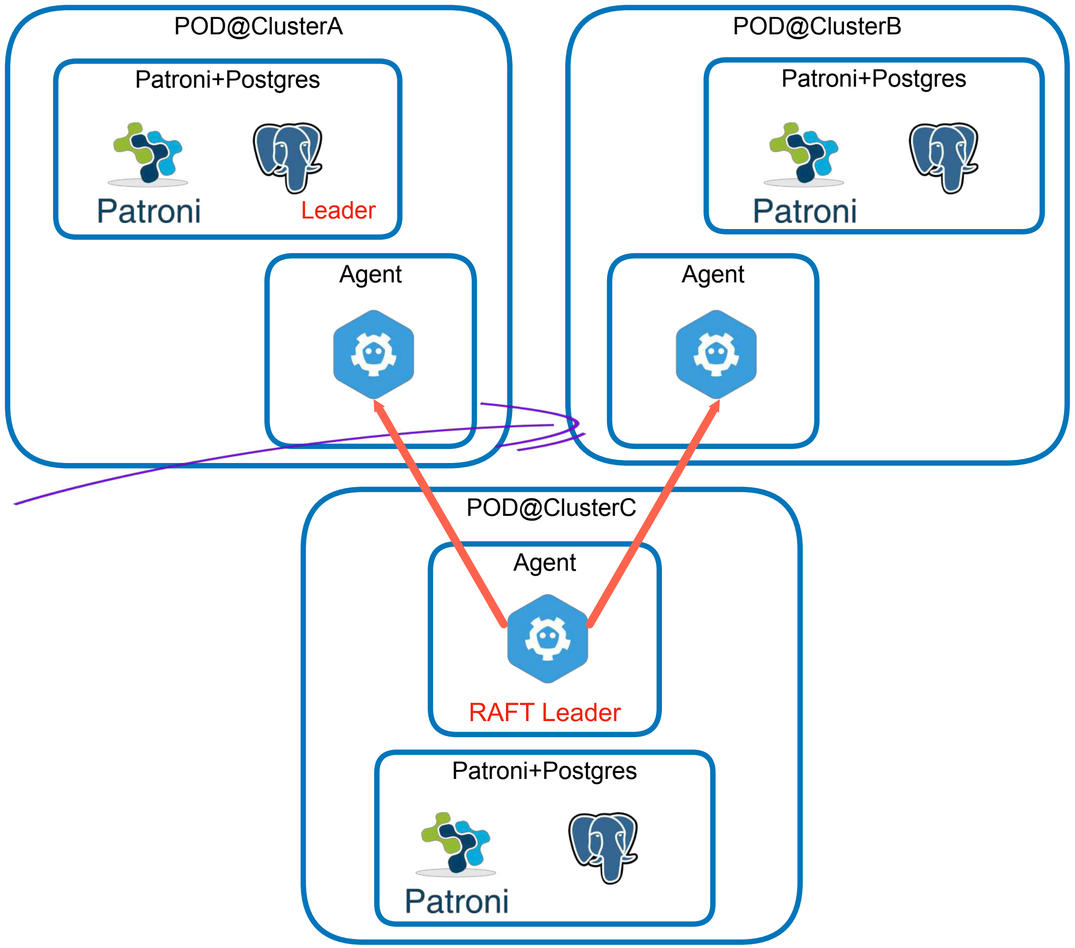
Проблема

- ETCD Bootstrap
- Не все Readiness Probes полезны

Решение

- Реализуем на основеPostgreSQL Agent API
- Оставляем только дляPostgreSQL Agent,остальным — /bin/true

PostgreSQL в DBaaS/k8sPostgreSQL Agent: Embedded ETCD



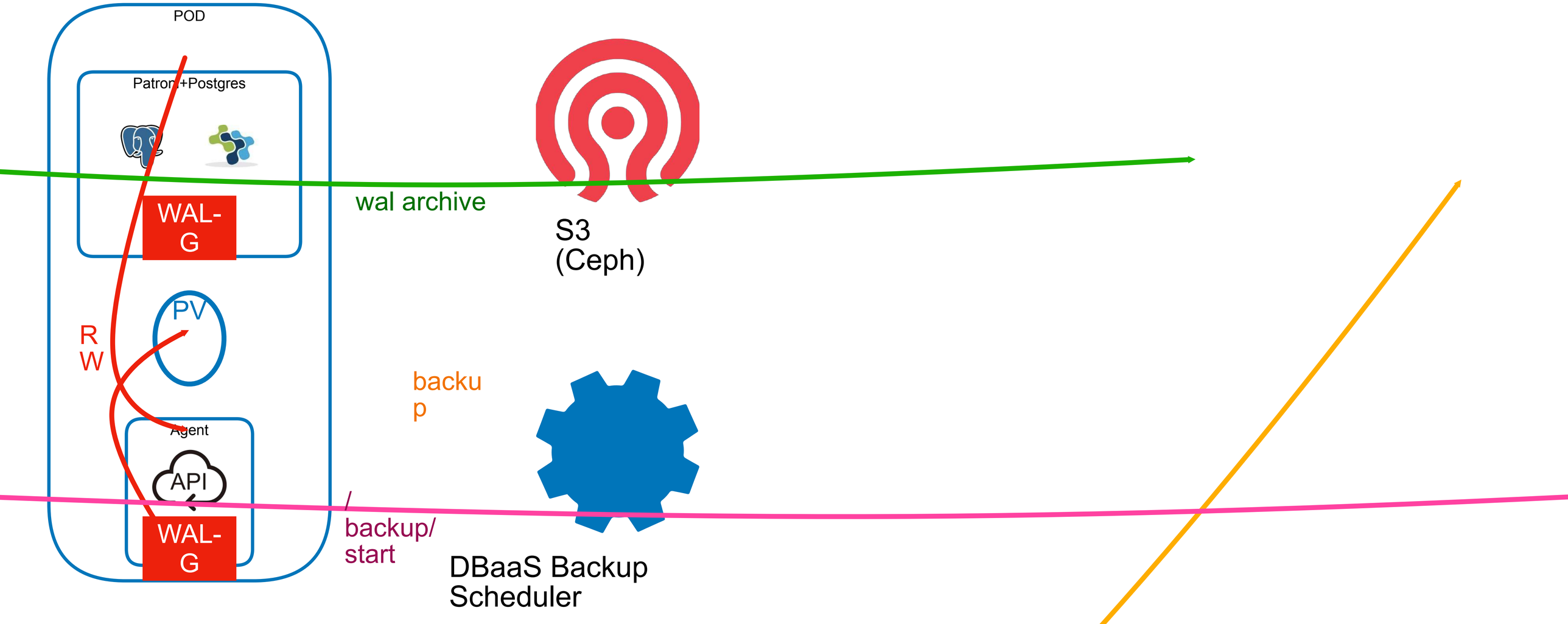
Проблема

- ETCD Bootstrap
- Не все Readiness Probes полезны
- Сложная конструкция

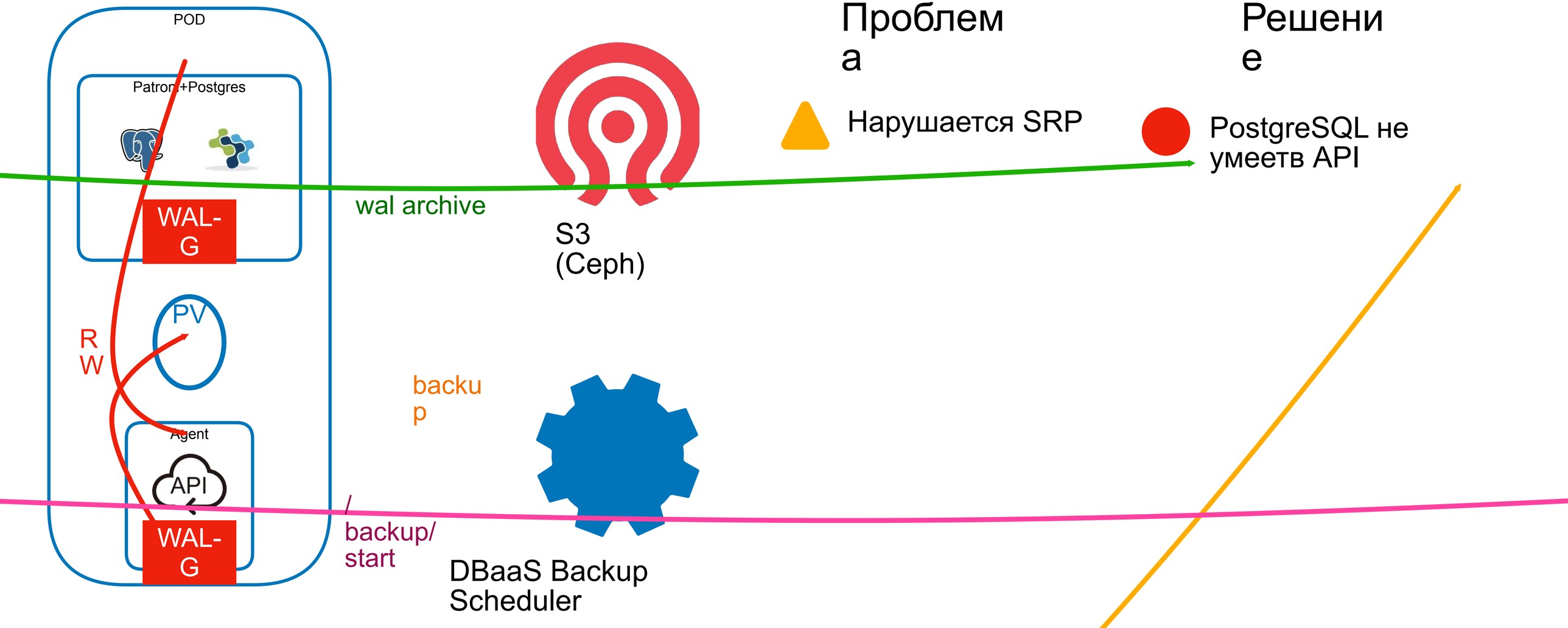
Решение

- Реализуем на основеPostgreSQL Agent API
- Оставляем только дляPostgreSQL Agent,остальным — /bin/true
- Ничего прощене придумали

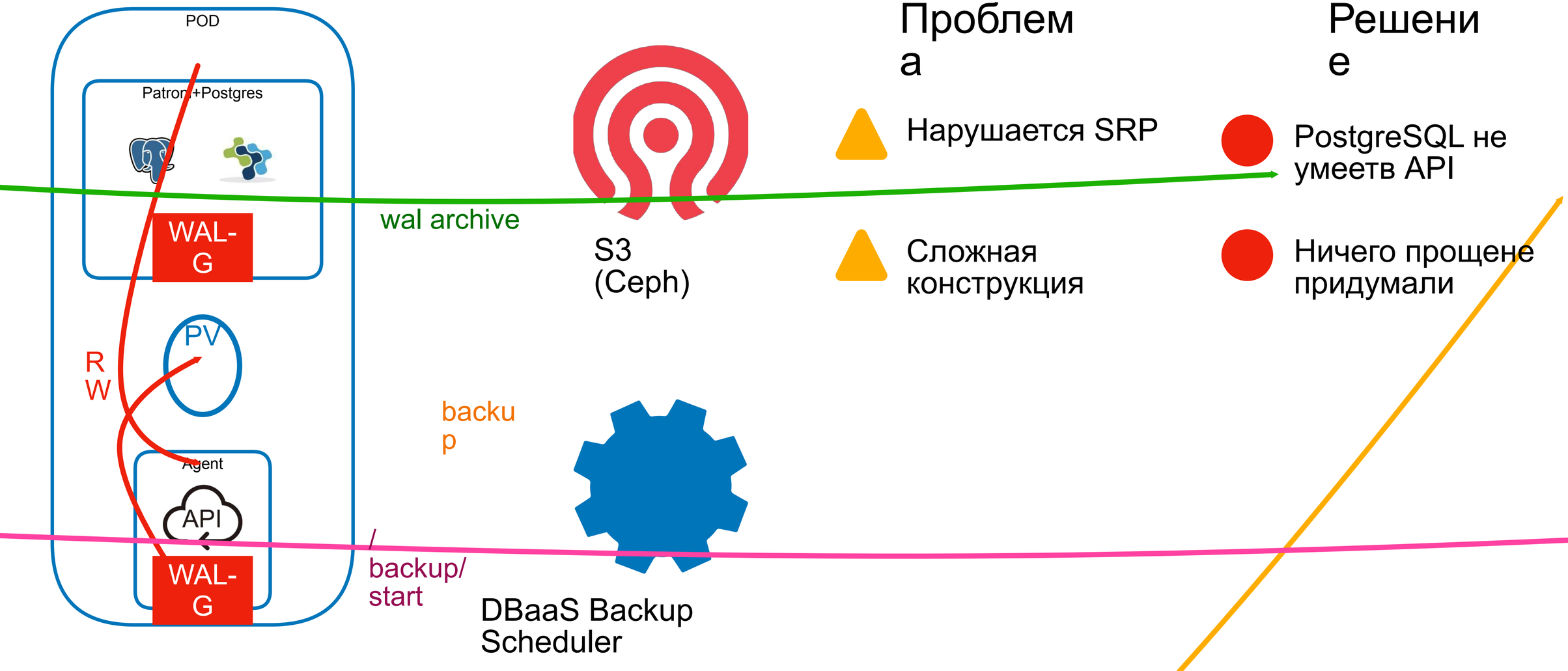
PostgreSQL в DBaaS/k8sPostgreSQL Agent: Backup API



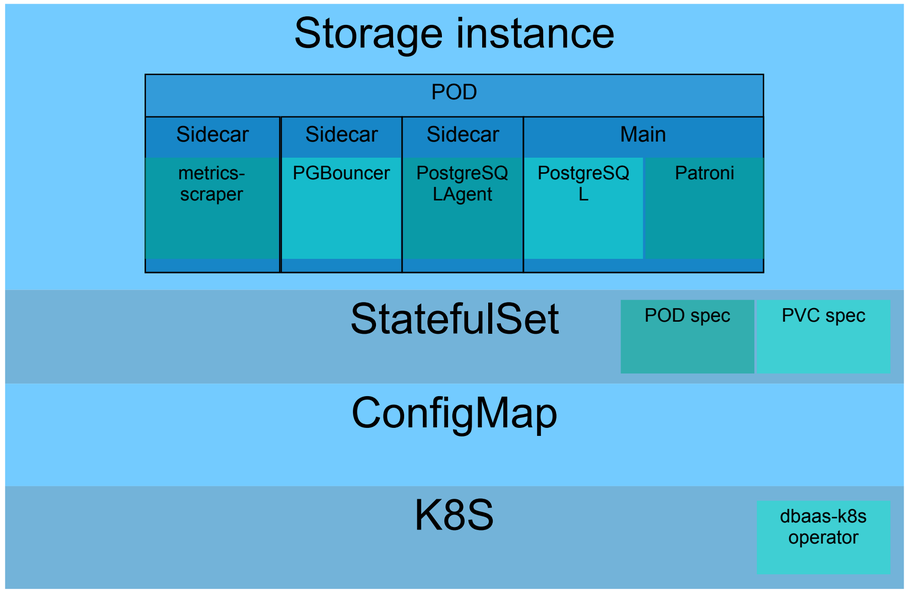
PostgreSQL в DBaaS/k8sPostgreSQL Agent: Backup API



PostgreSQL в DBaaS/k8sPostgreSQL Agent: Backup API



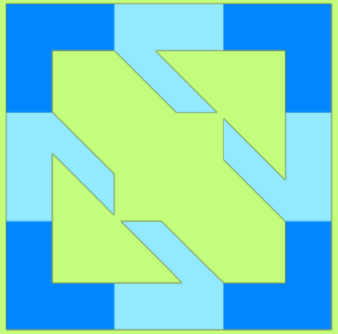
PostgreSQL в DBaaS/k8sЧто в итоге?



PostgreSQL на платформе DBaaS/k8s

- Автоматизированный жизненный цикл
- Время создания базы — 2-4 минуты
- Много разработки, но меньше операционки
- Достаточно сложная реализация
- PostgreSQL всё ещё плохо масштабируется

Что там с Cloud Native?



1 2 3 4

Что там с Cloud Native?CockroachDB «из коробки»

- **High Availability & Fault Tolerance**
  Native: RAFT, Replication
- **Durability**Native: Backup to S3
- **Scalability**Native: вертикальное (с оговорками)Native: Горизонтальное
- **Observability**Native: Prometheus-exporter, Json-лог

Что там у Cloud Native?Адаптация CockroachDB на платформе DBaaS/k8s

avito.tech Москва | 2024

Не все базы одинаковок8s-friendly.

Андрей Лярский

Инженер в SQL/DBA

@oxumorron