ozon{tech

Что под капотом у облачного PostgreSQL в Ozon

Дмитрий Васильев

Эксперт разработки информационных систем группы PostgreSQL DBA, Ozon

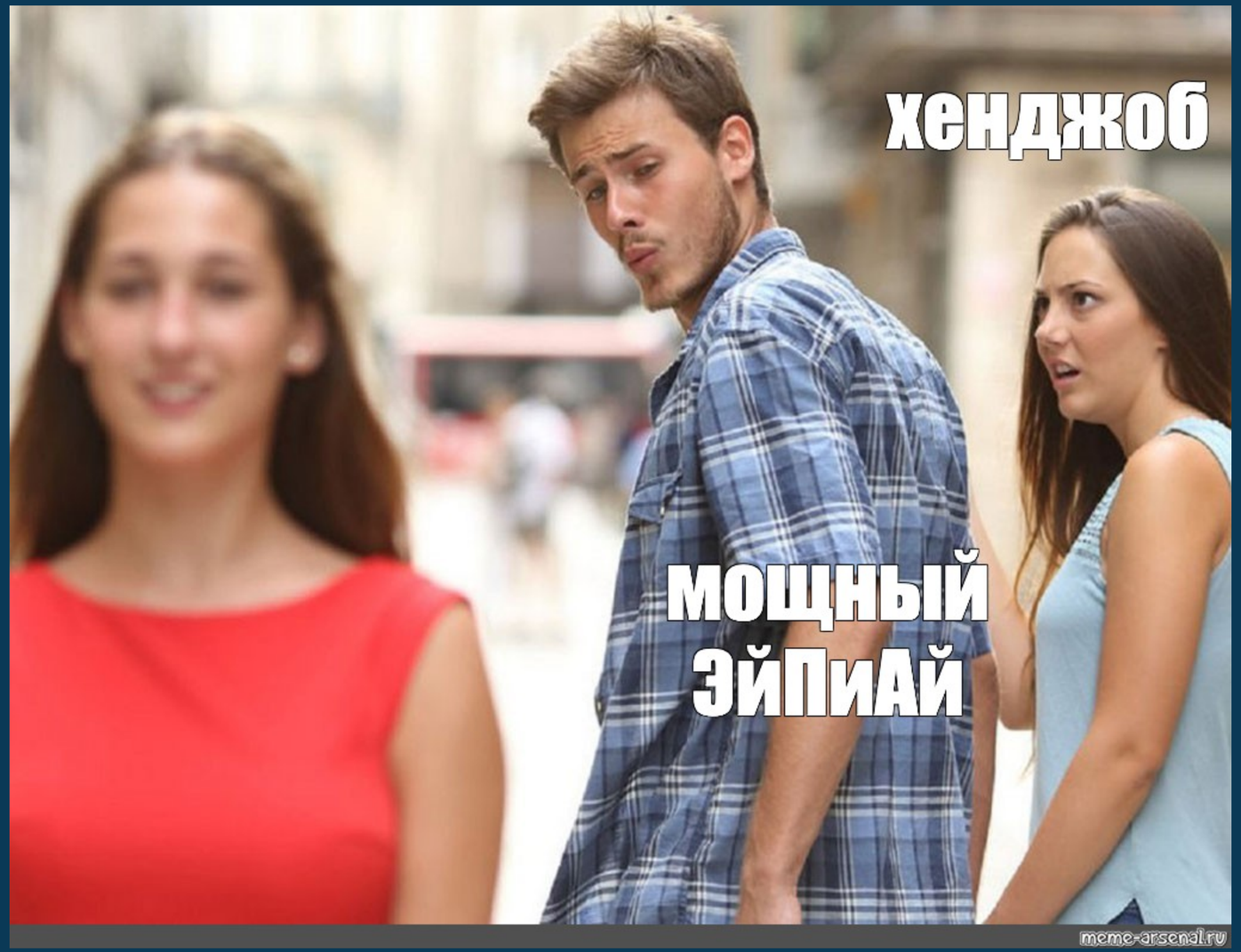
Ozon Tech 2022

PostgreSQL в OZON

ozon{tech

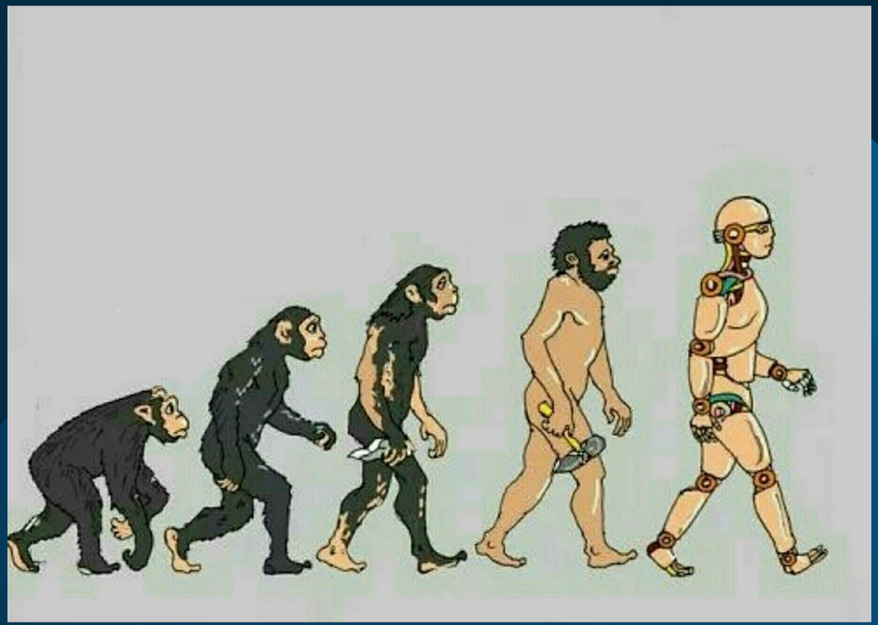
- Почти каждый новый сервис имеет базу PostgreSQL а иногда и не одну (шардирование).
- Каждый месяц в OZON появляется несколько сотен PostgreSQL кластеров.
- Производить инсталляцию вручную:
  - Медленно - мешает развитию
  - Человеческий фактор
  - Рутина

ozon tech



ozon{tech

Эволюция разворачивания PostgreSQL кластера

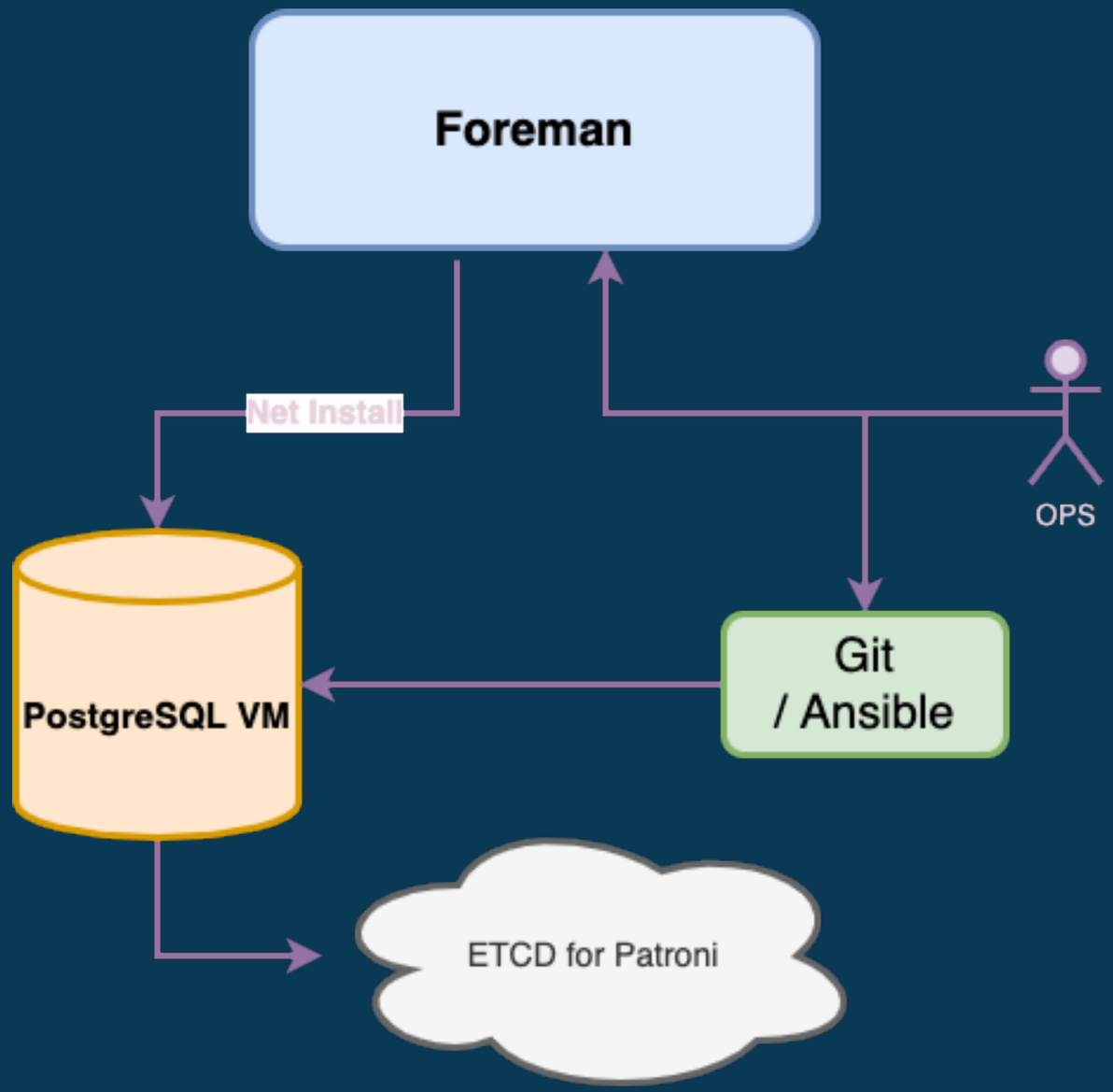


Инсталляция PostgreSQL кластера

ozon{tech

Как всё начиналось

- Первоначальная наливка VM при помощи Foreman
- GroupVars для кластера PostgreSQL через Ansible

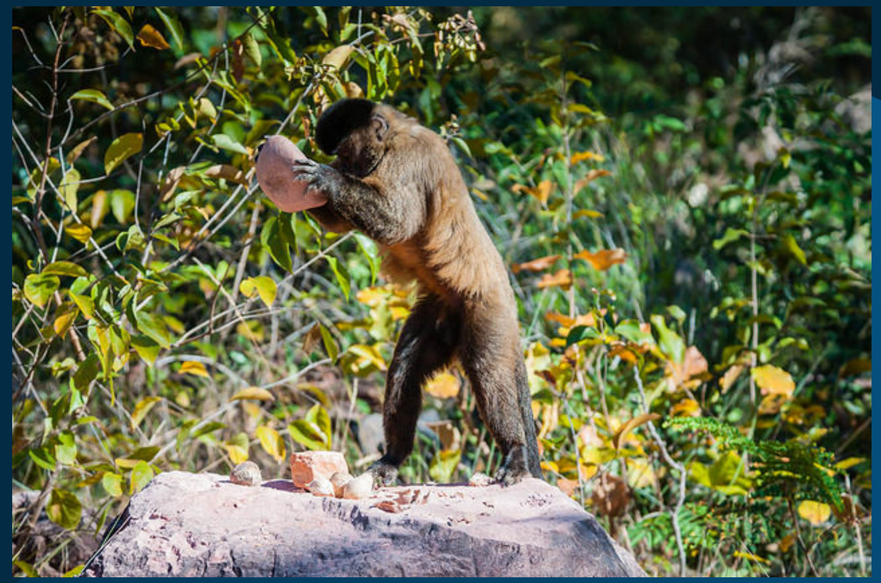


https://www.theforeman.org/

ozon{tech

Тупо автоматизируем ручную работу.

Шаг номер 1: Resource Manager.

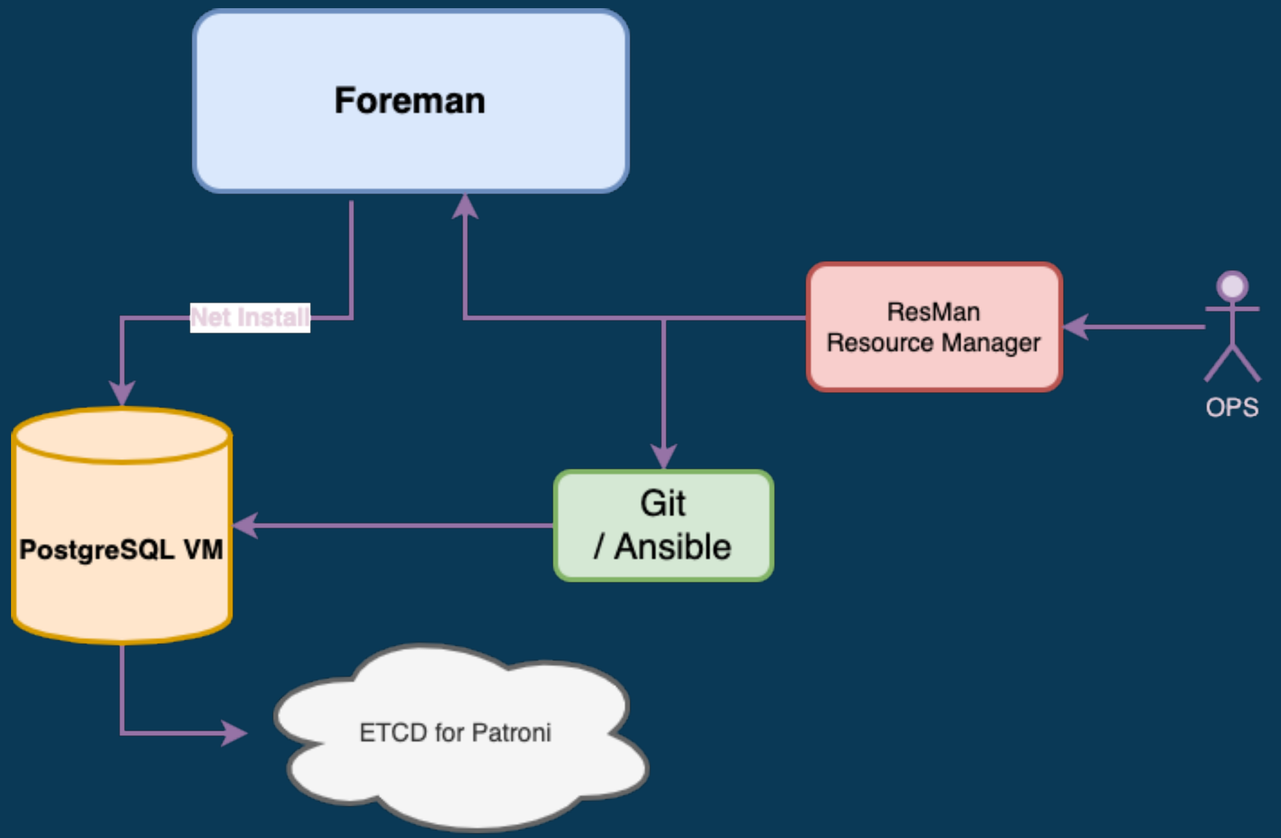


Инсталляция PostgreSQL кластера

ozon tech

Первые шаги к автоматизации

- Добавлен Resource Manager
  - контролирует правильное размещение виртуалок с учетом стойки/ДЦ/выделеных ресурсов на гипервизоре
  - коммитит в Git необходимые GroupVars



ozon{tech

Появилось несколько дата центров.

Разнородное железо под гипервизоры.

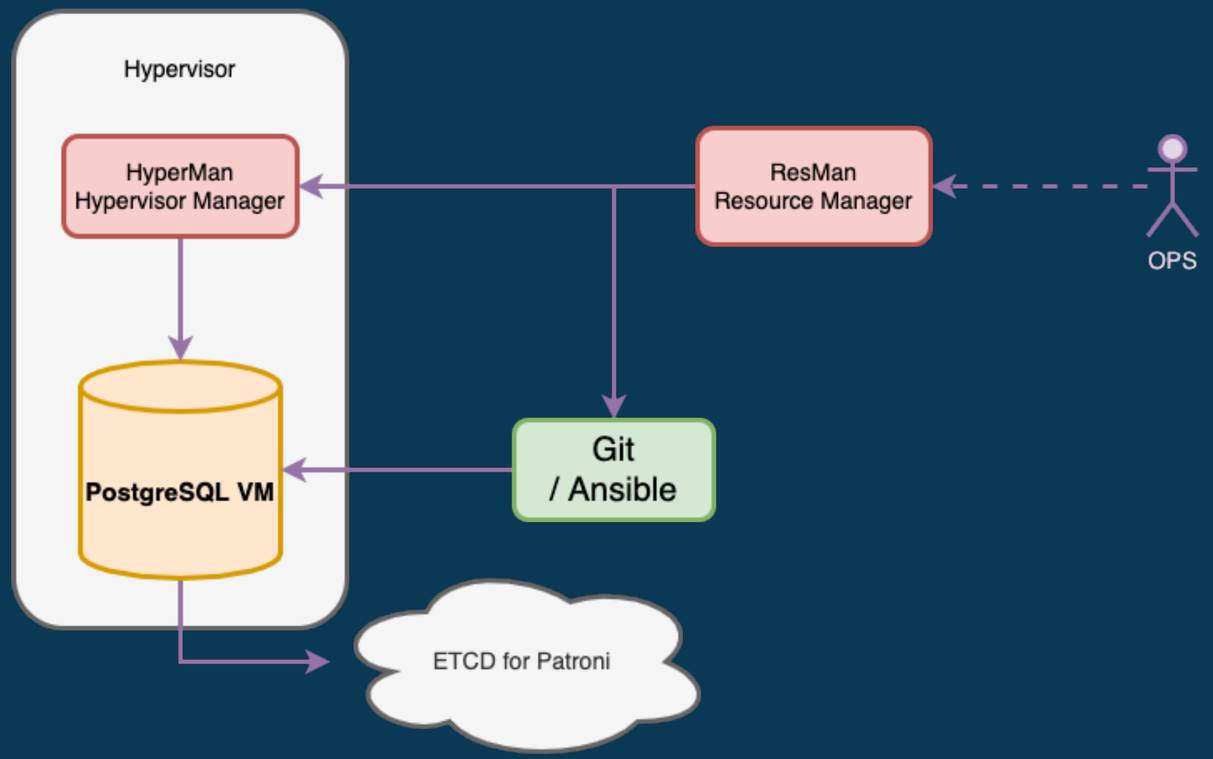
Шаг номер 2: Hyper-Man.

Инсталляция PostgreSQL кластера

ozon{tech

Избавляемся от Foreman

- Добавлен децентрализованный Hypervisor Manager
  - настройка над Libvirt
  - использование “Gold Image”
  - онлайн ресайзы виртуальных машин через CLI
  - Интеграция с qemu-agent



ozon tech

Merge approve стал бутылочным горлышком.

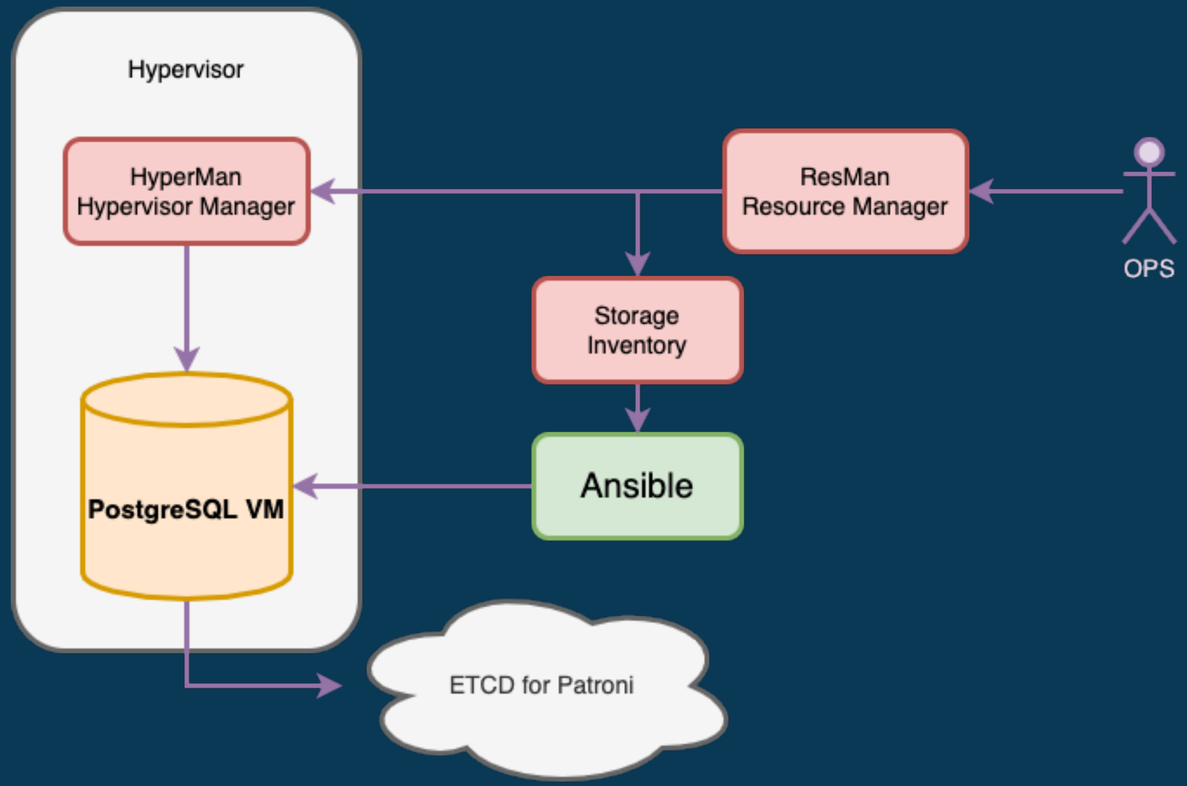
Шаг номер 3: Storage-Inventory.

Инсталляция PostgreSQL кластера

ozon{tech

Избавляемся от коммитов в Git

- Добавлен Storage Inventory
  - API для хранения GroupVars
  - Аудит/История/Откаты



Ansible

ozon{tech

Боль и унижение

1. Работа по сети — очень медленно. Полный деплой всех кластеров - 1 день.
2. Ускоряемся, когда запускаем локально (ssh <hostname> “ansible-playbook”). Полный деплой - 2 часа.
3. Запуск очень дорогой по ресурсам (repo cache → disk, python inventory → CPU).
4. Не удовлетворяет банальным требованиям:
    1. Поставить пакет: лишний раз нужно дергать repo cache.
    2. Потерянные notify в случае падения (падение при недоступности vault, например).

ozon{tech

Puppet, Ansible ... Chiit!

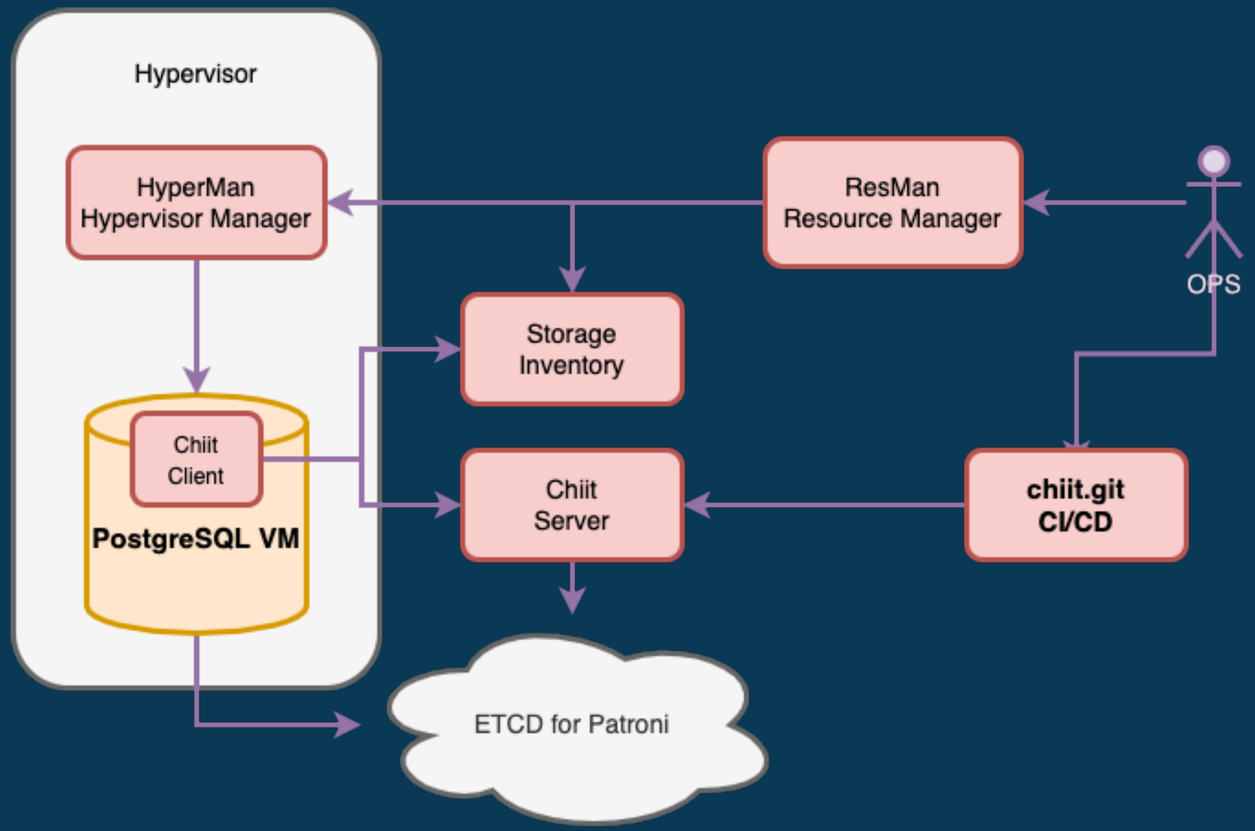
Шаг номер 4: Пишем свой SCM

Инсталляция PostgreSQL кластера

ozon{tech

Chiit: деплой тысяч машин за 5 секунд

- Добавлен новый SCM — Chiit (Change It)
  - Клиент-серверная архитектура
  - Хранение секретов (кэш перед vault)
  - Быстрота работы
  - Клиент сохраняет отчёт о статусе и измененных ресурсах
  - Интеграция с CI/CD:
    - собираем бинарь
    - тестируем через Inspec
    - загружаем бинарь в несколько Ceph
    - делаем запись в Server о релизной версии



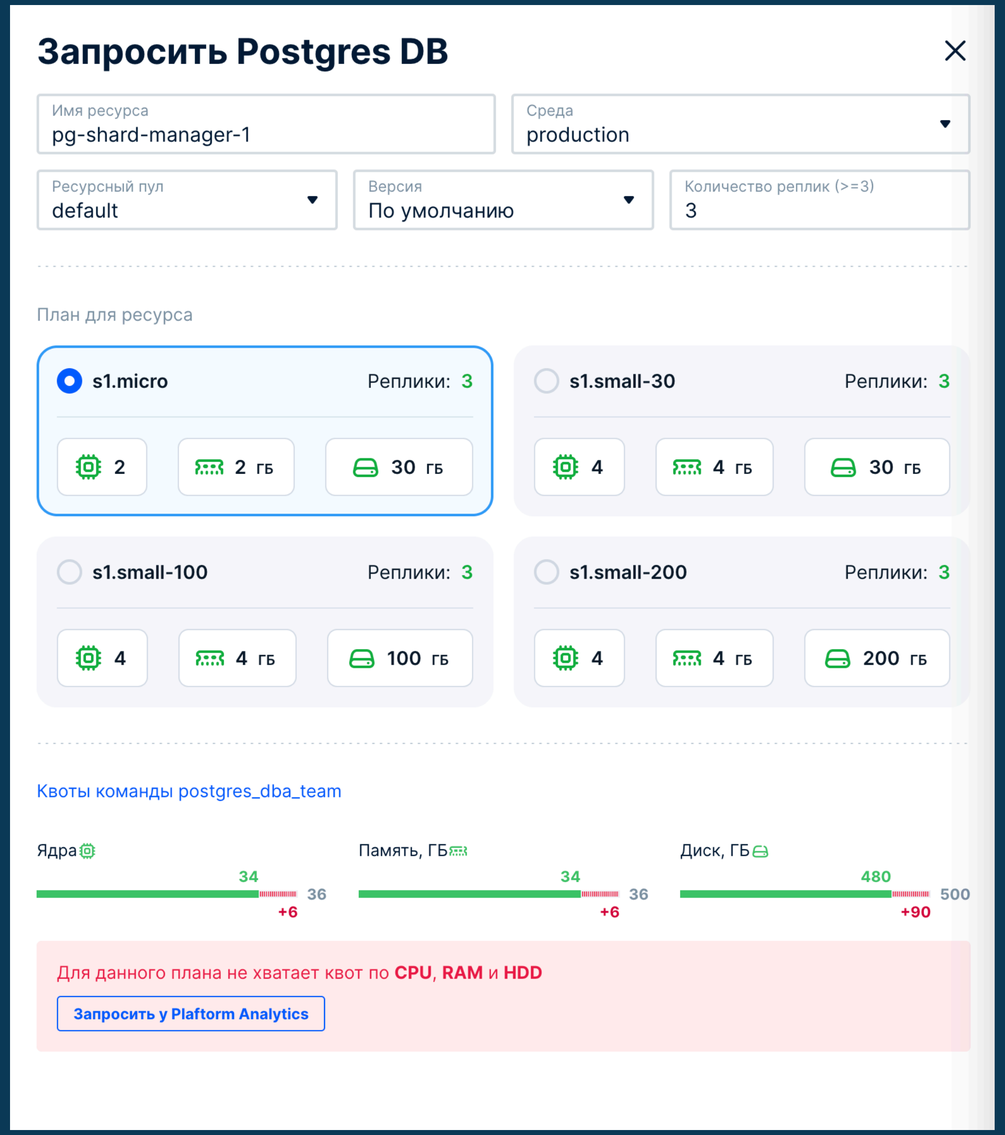
Chiit example: stop using YAML

ozon{tech

```
systemd.NewService(ctx, &systemd.ServiceConfig{
    Name:          "service-name",
    After:         "after",
    ExecStart:     "package-bin",
    User:          "user",
    SyslogIdentifier: "service",
    MaxMemoryMb:   1024,
    CPUQuotaPercent: 50,
    Restart:       systemd.ServiceRestartAlways,
    CPUAffinity:   1,
    MemoryAccounting: true,
    CPUAccounting:    true,
})
apt.Install(ctx, &packages.Package{Name: "package-name",
    Version: inventory.Get( key: "package.version").String()}).Notify(
    func() {
        defers.Runner.AddRestartSystemd(ctx, name: "service-name")
    })
```

Инсталляция PostgreSQL кластера

ozon tech

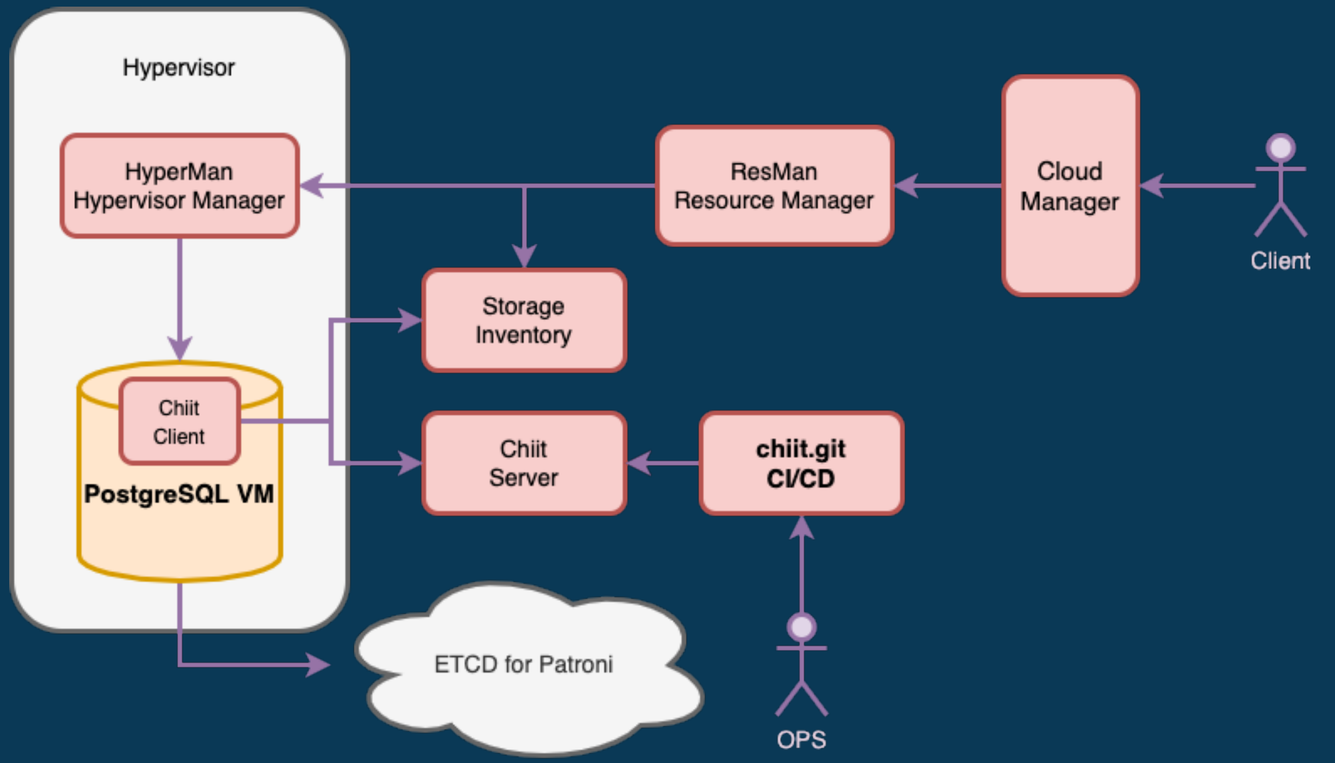


Инсталляция PostgreSQL кластера

ozon{tech

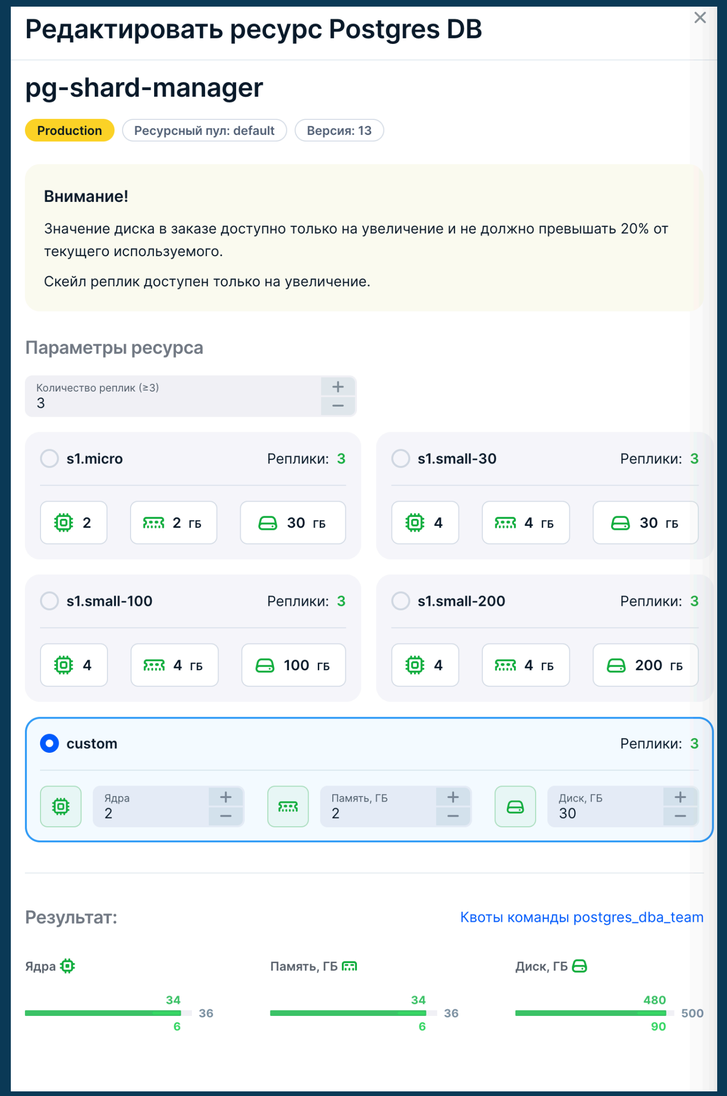
Поворачиваемся лицом к пользователю

- Добавлена интеграция с Cloud Manager Console
  - избавили от необходимости OPS проводить операции даже через cli
  - добавили понятие размера ресурсного пула для команды



Инсталляция PostgreSQL кластера

ozon{tech

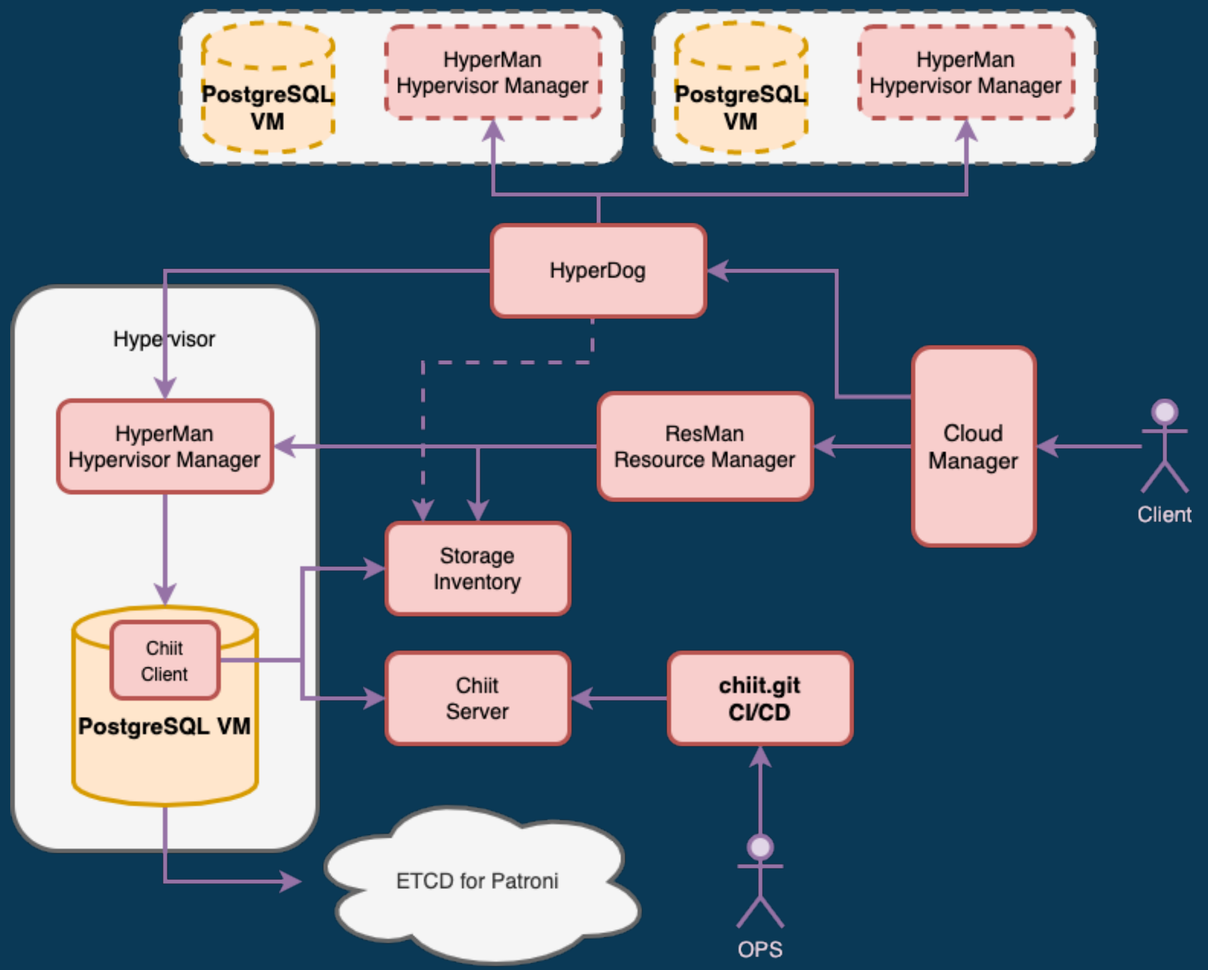


Инсталляция PostgreSQL кластера

ozon{tech

Проблема роста размера кластера

- Добавлен Hyper-Dog (управляет Hyper-Man как кластером PostgreSQL).
  - следит за дисковым авто-ресайзом
  - производит ресайзы (CPU/Memory) кластеров



ozon tech

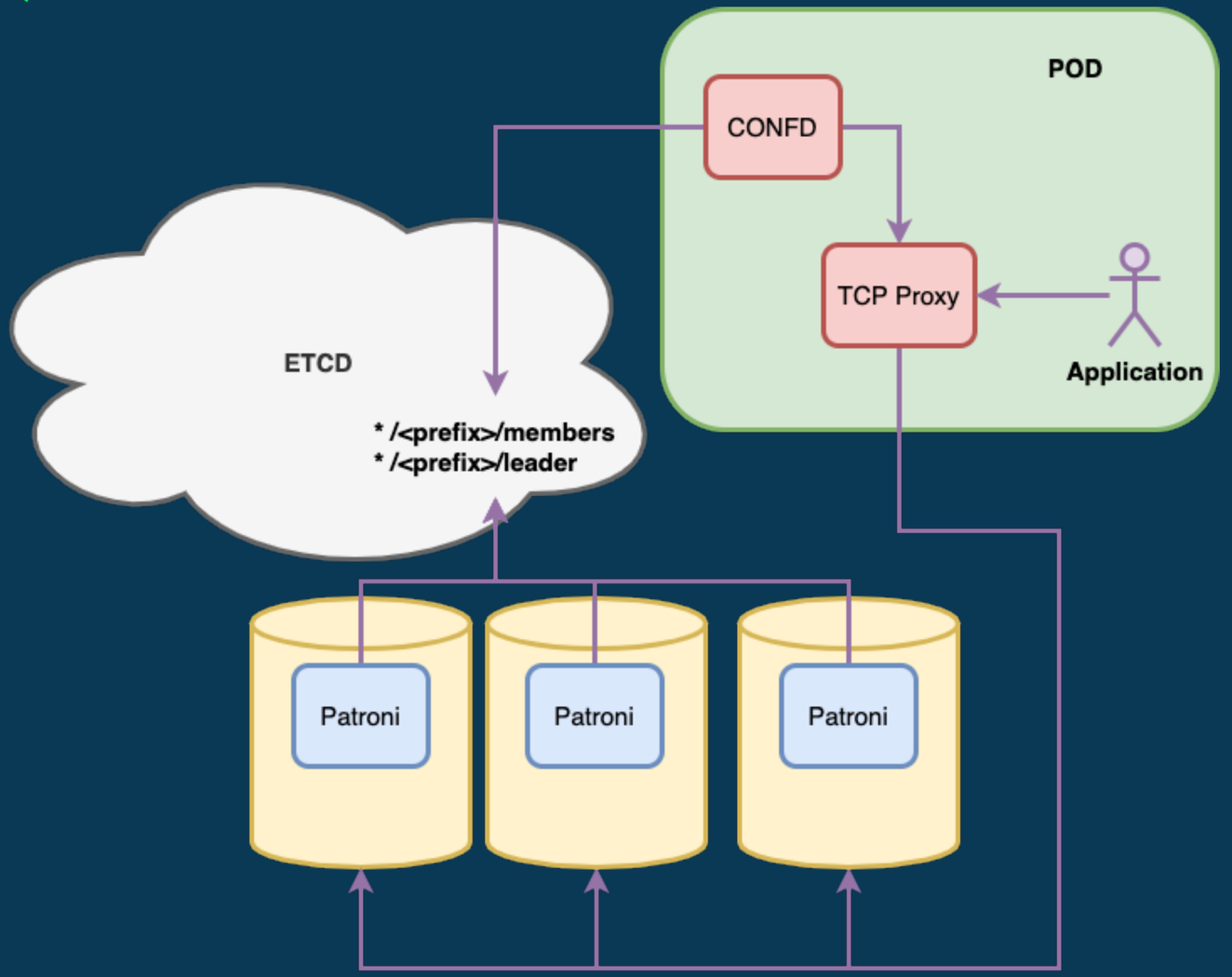
Отказоустойчивость и драйвера

Отказоустойчивость и драйвера

ozon{tech

Подсаживаем в pod sidecar с tcp-proxy (haproxy/nginx)

- ConfD следит за состоянием кластера в ETCD
- ConfD генерирует конфиг proxy с конечными endpoints до PostgreSQL



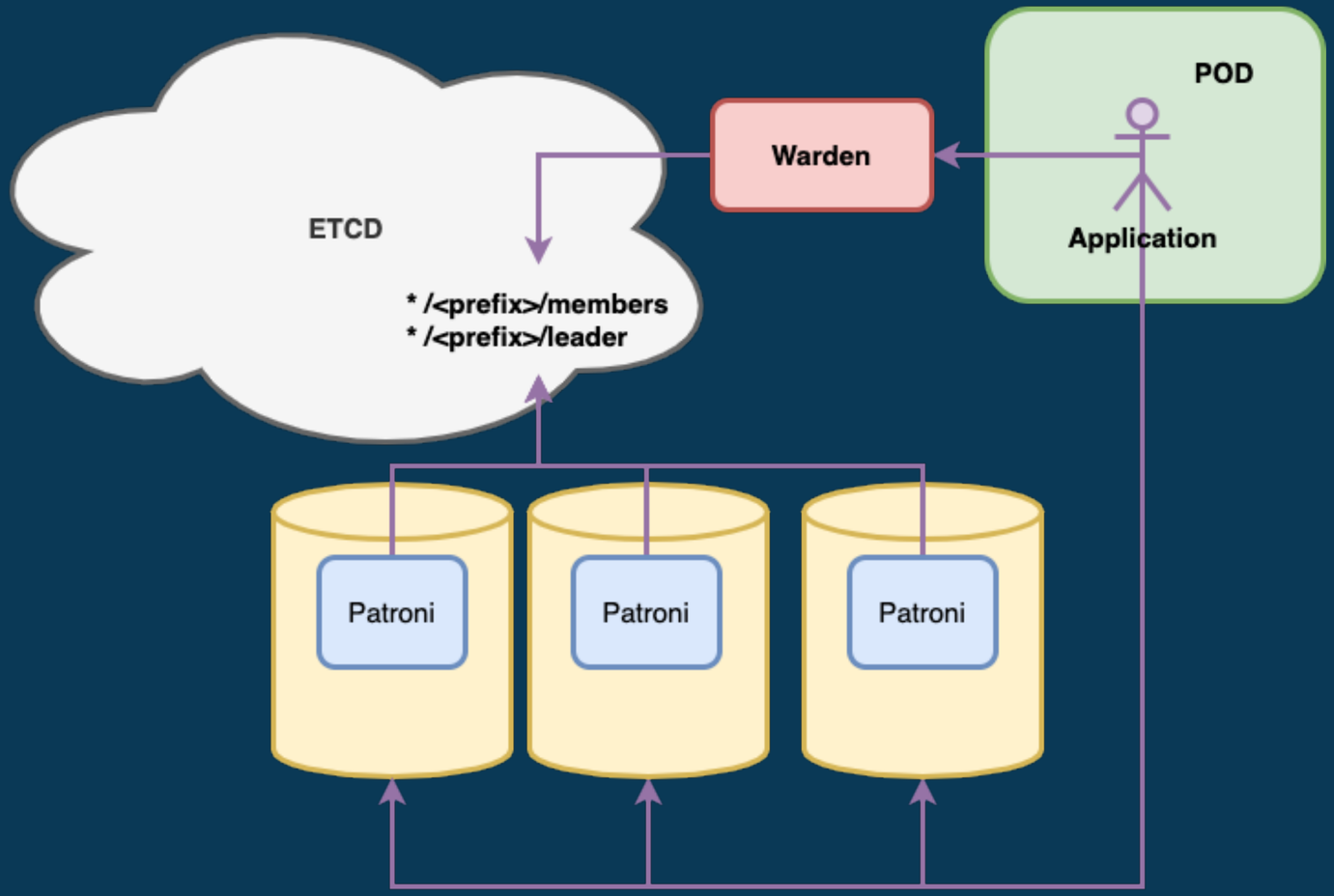
https://github.com/kelseyhightower/confd

Отказоустойчивость и драйвера

ozon{tech

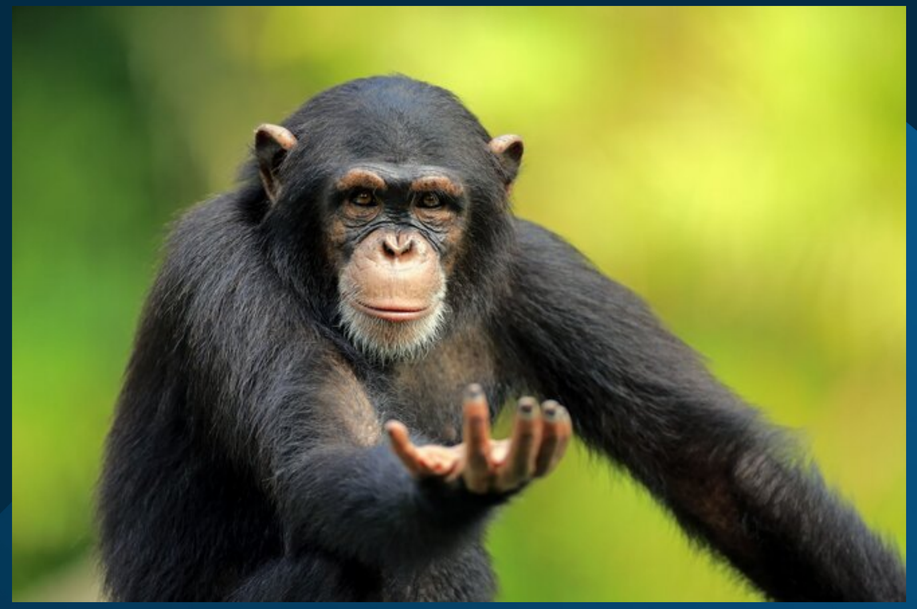
Большое количество коннектов к ETCD, «зависание» некоторых POD'ов

- Добавили Service Discovery для PostgreSQL кластеров
  - У каждого из POD'ов единая картина мира
  - Простота разработки:
    - Cluster.New("cluster-name").GetRole("master").Query()
    - Cluster.New("cluster-name").GetRole("read").Query()
    - Cluster.New("cluster-name").GetRole("sync").Query()



ozon{tech

Изменение размера кластера “на лету”

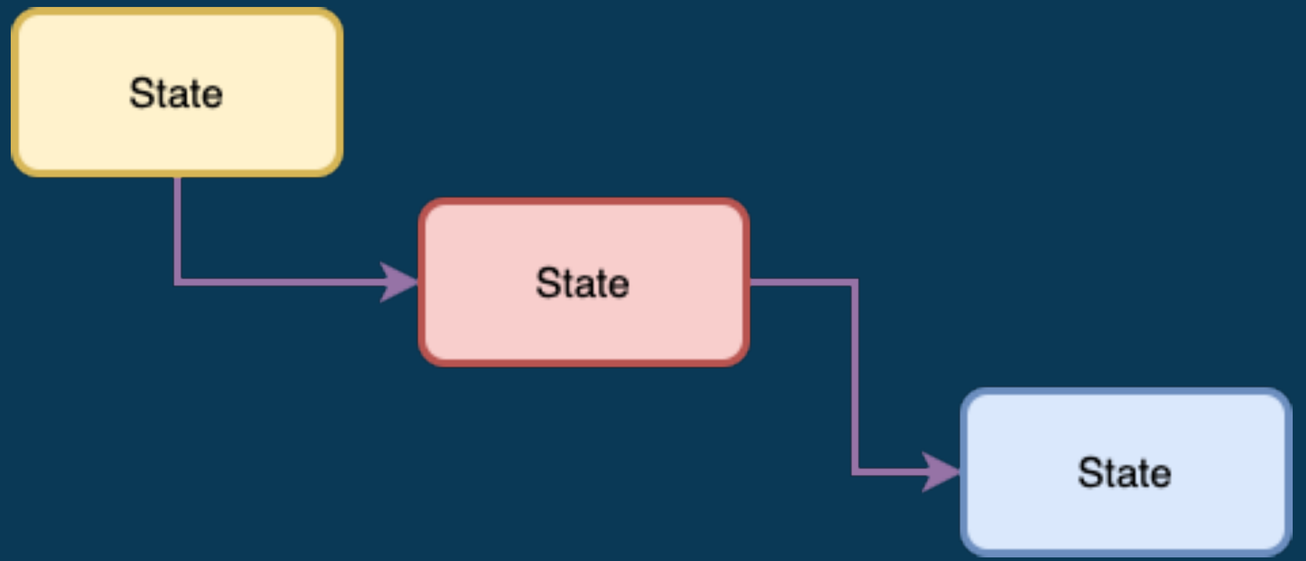


Изменение кластера «на лету»

ozon tech

"Safe scripts"

- Проект, который безопасно позволяет перезагрузить машину из кластера: master/sync/async
- Проверяет на целостность кластер (3 ноды)
- Делает вывод из кластера при помощи tag NoLoadBalance через Service Discovery
- Остановку PostgreSQL учитывая логическую репликацию

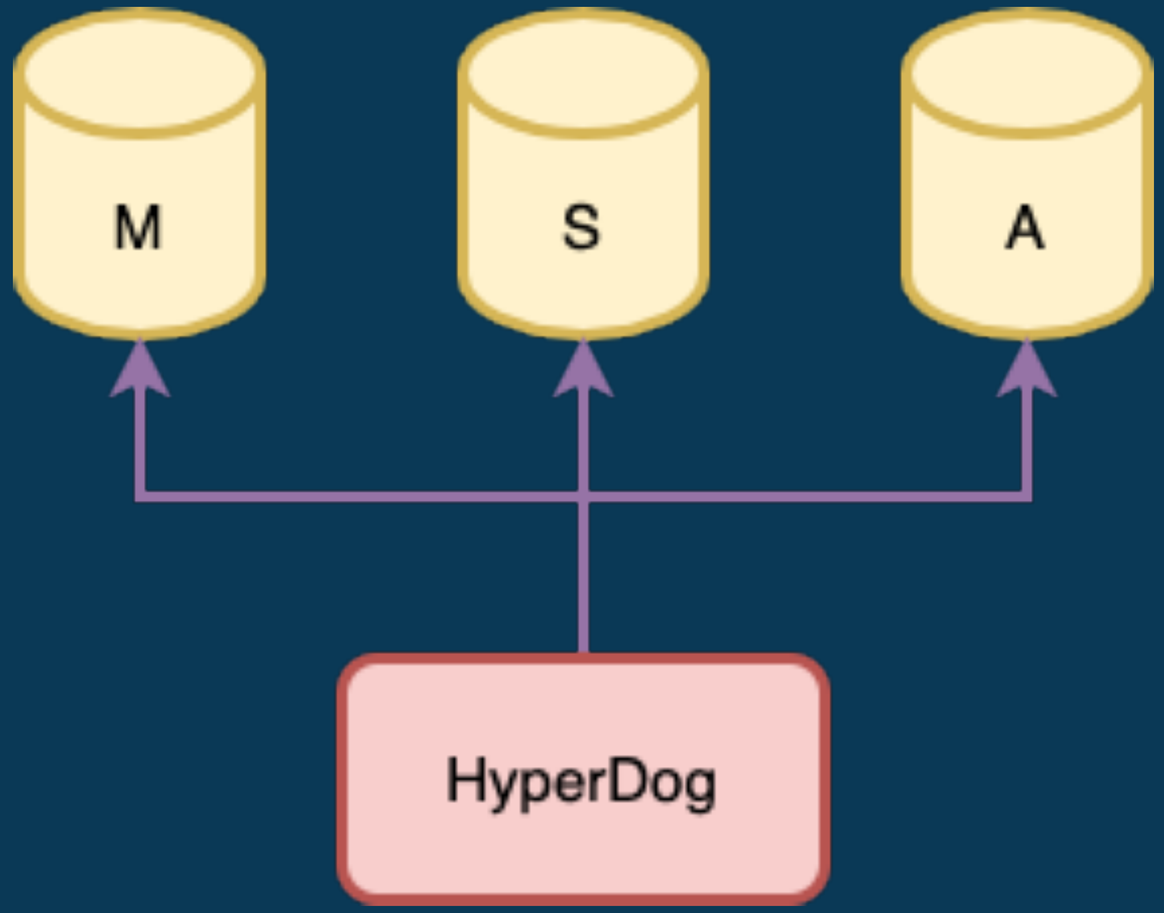


Изменение кластера «на лету»

ozon{tech

Меняем ram/количество CPU

- Делаем изменения последовательно:
  - рестарт Async-реплик с новыми размерами
  - делаем switchover sync-реплики: дожидаемся появления sync-реплики, рестарт с новыми размерами
  - делаем switchover master, рестарт с новыми параметрами

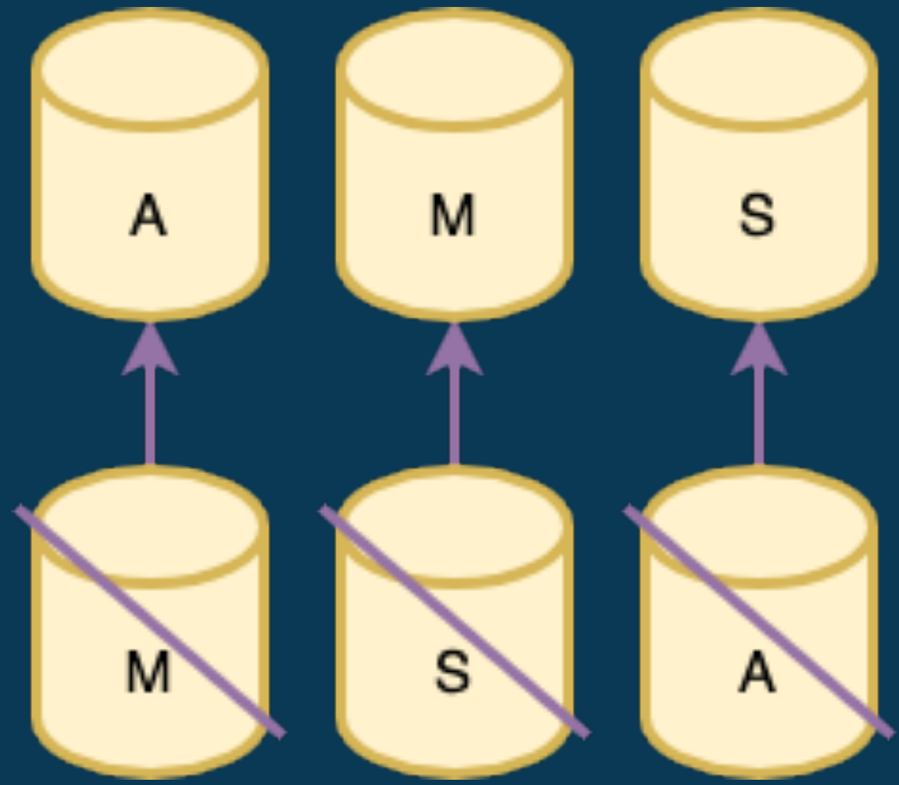


Изменение кластера «на лету»

ozon{tech

Сжатие дисков, shrink.

- Определяем фактический размер базы
- Пересоздаем со switchover async, sync, master



Функциональность облака

ozon{tech

Что в итоге мы умеем?

- CPU, RAM x2 без switchover.
- CPU, RAM - без аффекта для приложения.
- Disk вверх и вниз.

ozon{tech

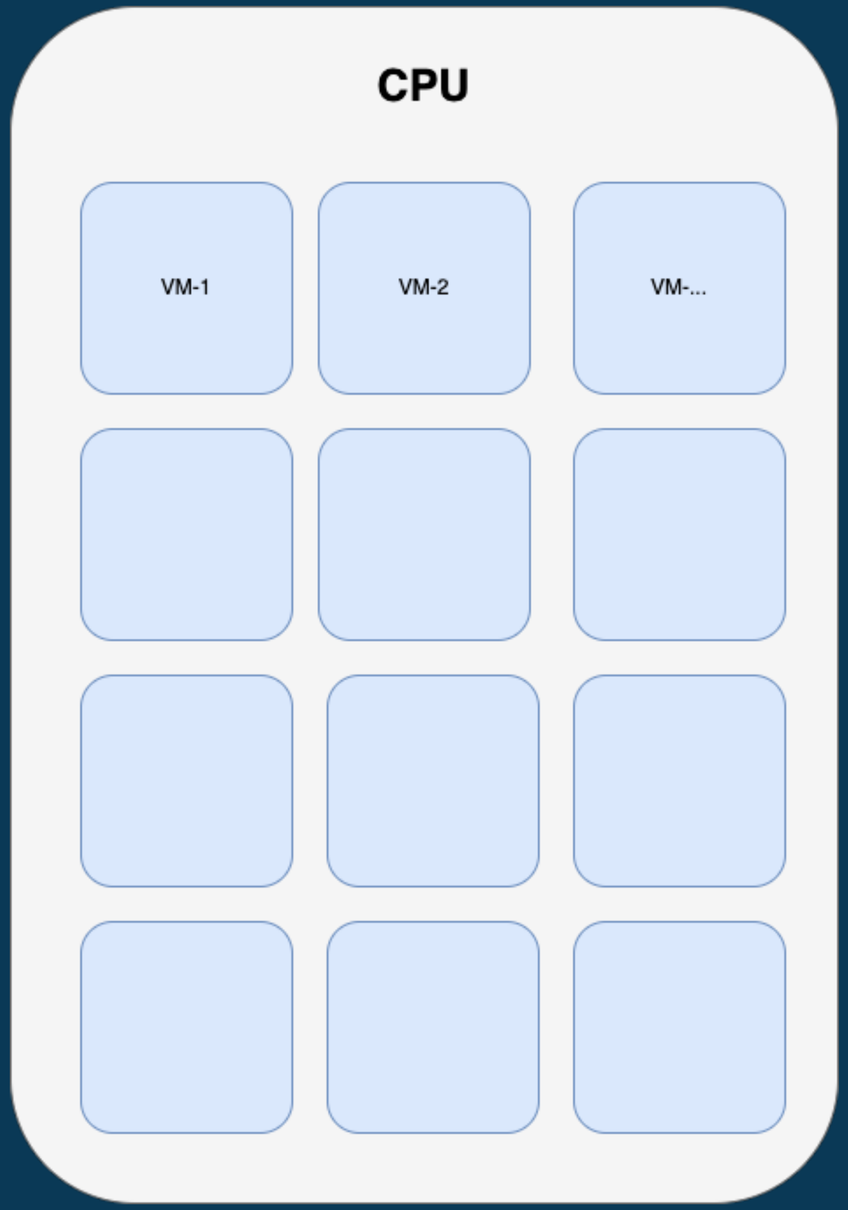
Тюнинг виртуальных машин

Тюнинг CPU

ozon{ech

Первоначально общий пул CPU

- Все системные процессы и виртуальные машины делятся всеми CPU
- Есть оверселлинг, виртуальные машины мешают друг-другу

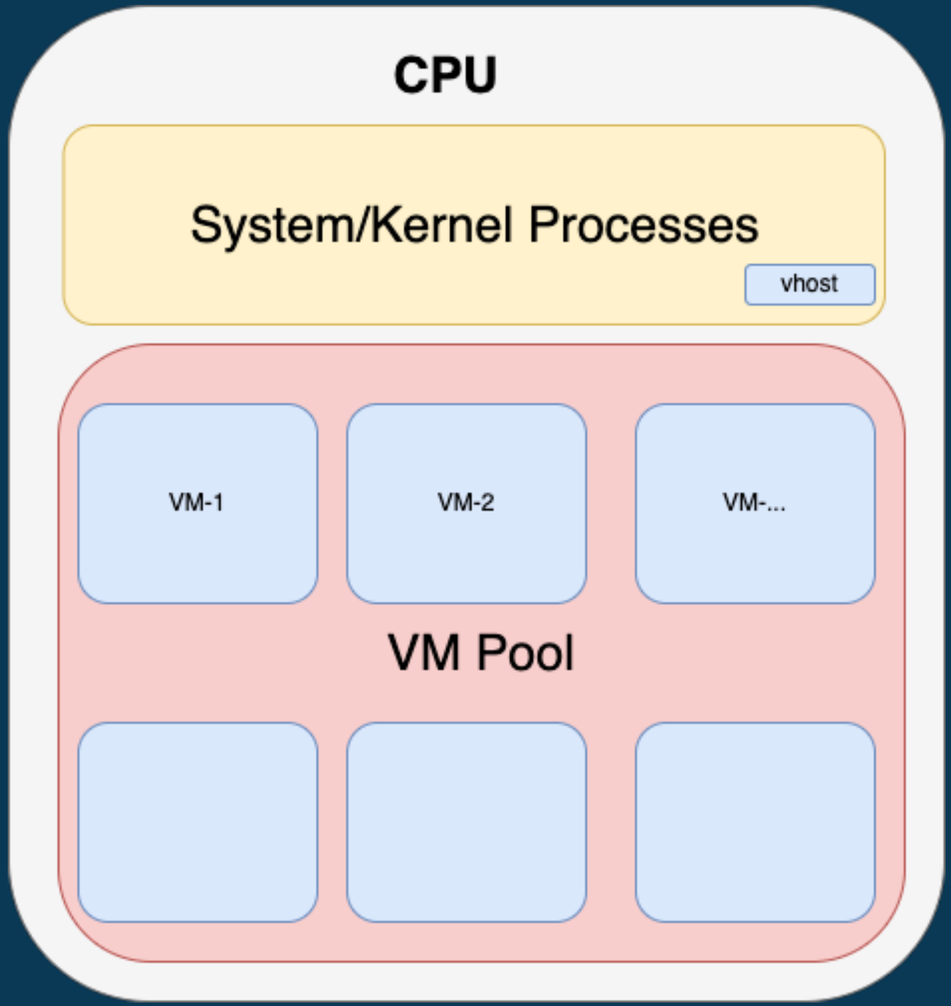


Тюнинг CPU

ozon{tech

Системный пул и пул для виртуальных машин: лечим сеть

- Отдельно есть пул процессоров, которые обслуживают виртуальные машины.
- Отдельно есть пул, в котором находятся системные процессы и всякие kworker, которые обслуживают сеть/диски в виртуальных машинах.

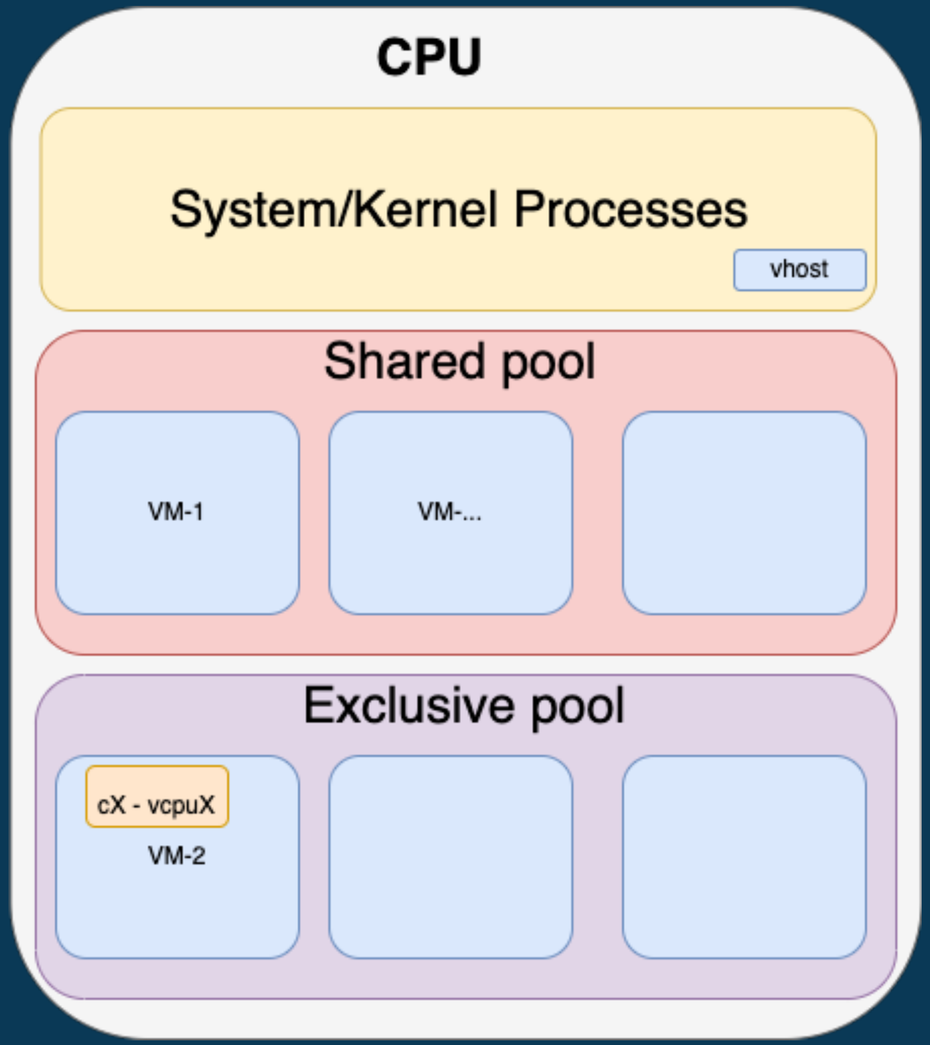


Тюнинг CPU

ozon{tech

Изоляция: пул для виртуальных машин поделен на два

- Эксклюзивный
  - в нем происходит прямое, эксклюзивное выделение каждому vcpu виртуальной машины к реальному CPU
  - имеет эластичный размер, если нет ВМ с эксклюзивными CPU, то размер пула 0
- Шареный
  - все остальные ВМ варятся в общем котле

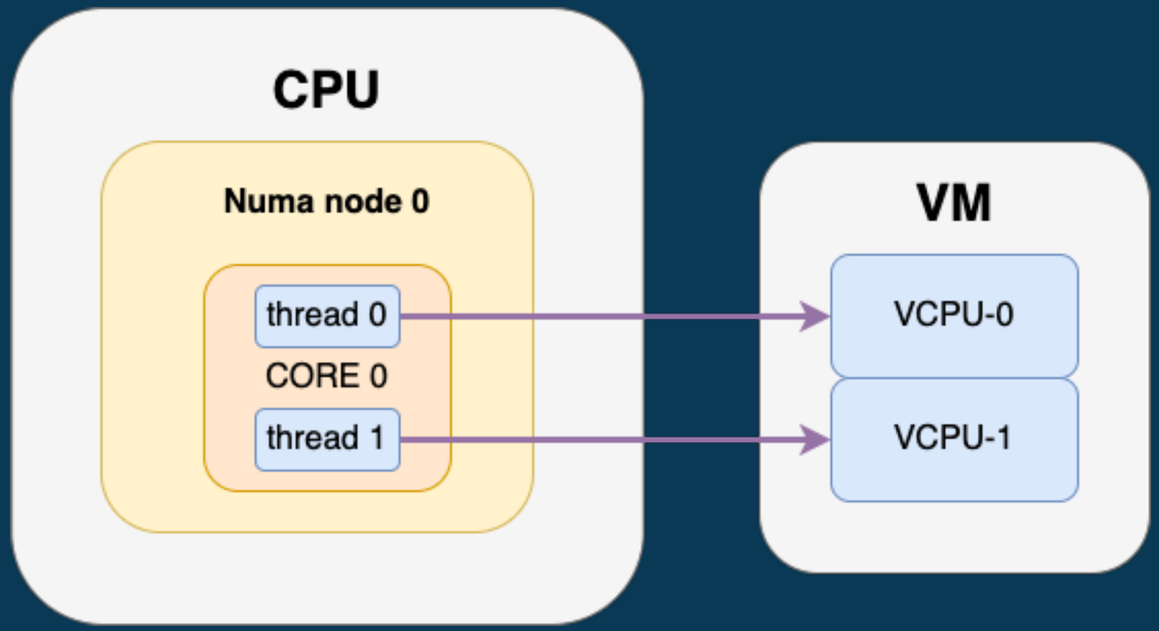


Тюнинг CPU

ozon{tech

Изоляция: учитываем топологию CPU

- Для системного пула
- Для эксклюзивного пула



Тюнинг CPU

ozon{tech

Отключение SMT (Hyper-Threading): 30% по stddev pg_stat_statements

- Для shared pool — деградация производительности.
- Для exclusive pool — увеличение производительности.

```
echo off > /sys/devices/system/cpu/smt/control
```

Тюнинг CPU

ozon{tech

Эластичный System Pool

- В системном пуле кроме vhost-net находятся emulator pin (до +20% stddev pgss для машин в шареном пуле)
- Если system + user cpu time превышает отметку 75%, то накидываем в системный пул дополнительные ядра

Тюнинг сети

ozon{tech

Проблема ещё не решена

- Bridge public mode
  - рандомно фиксировали ЧУЖОЙ трафик в своей ВМ от соседней ВМ, всплески systime
- Macvtap private mode
  - не видим соседей, проблема с шардированием
- SR-IOV
  - неоднородное железо
  - иногда течет libvirt
  - проблема с некоторыми Intel-карточками

https://access.redhat.com/solutions/2822941
https://access.redhat.com/solutions/4847951

ozon{ech

Спасибо за внимание

Васильев Дмитрий, DBA

dmitrivasilyev@ozon.ru