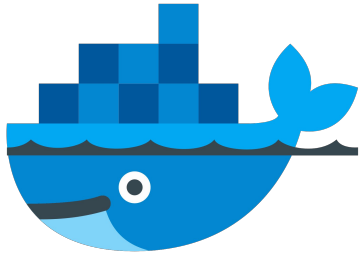
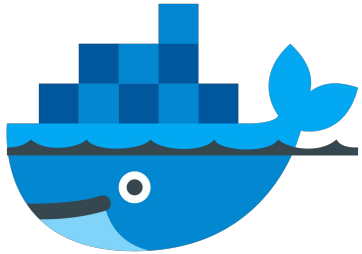
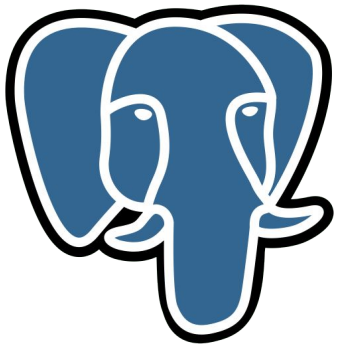
Есть ли дружба между железом и облаком?

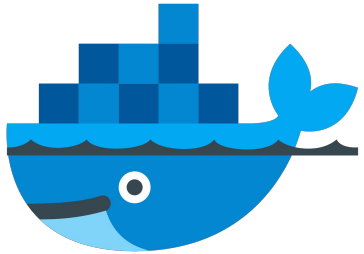
Дехтярёв Евгений

2GIS

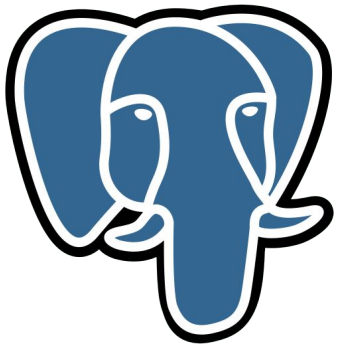


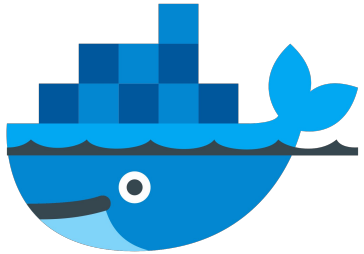




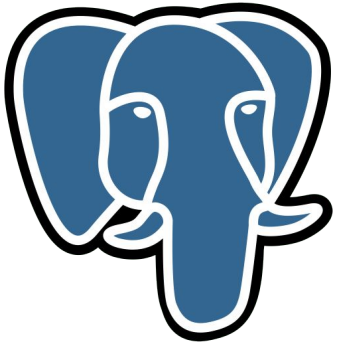


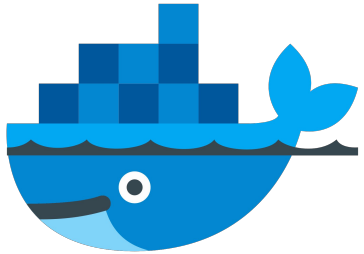

Stack



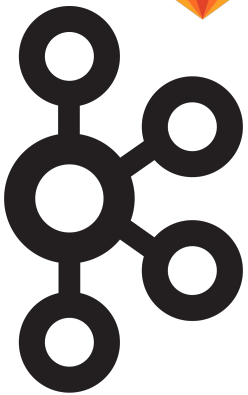


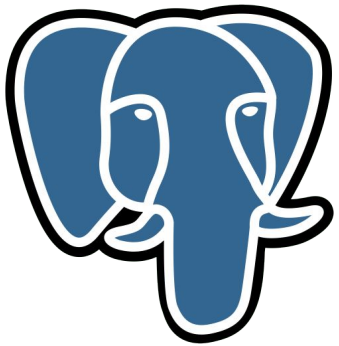

Stack

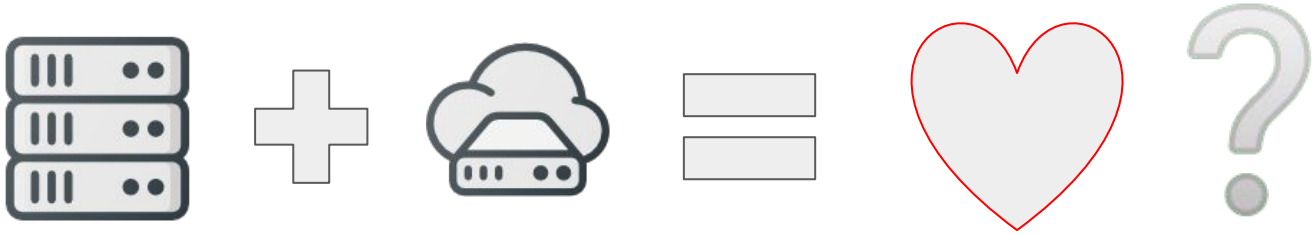





Stack







Поговорим

Задача

Вводные

Решения задачи

Унификация HW и VM

Какие-то выводы

Задача

Задача: SLA 99.9

- Большой заказчик 

Задача: SLA 99.9

- Большой заказчик 
- Трафик x3-5
  - Карта: 10 -> 50 krps
  - Справочник: 3 -> 10 krps
  - Навигация: 2 -> 10 krps

Задача: SLA 99.9

- Большой заказчик
- Трафик x3-5
  - Карта: 10 -> 50 krps
  - Справочник: 3 -> 10 krps
  - Навигация: 2 -> 10 krps
- Требования к
  - надежности
  - скорости ответа

Задача: SLA 99.9

- Большой заказчик
- Трафик x3-5
  - Карта: 10 -> 50 krps
  - Справочник: 3 -> 10 krps
  - Навигация: 2 -> 10 krps
- Требования к
  - надежности
  - скорости ответа
- Ситуационная нагрузка

Задача★: SLA 99.9 и дешево!

- Нет безлимитного бабла
  - Нельзя заранее купить всё

Задача★: SLA 99.9 и дешево!

- Нет безлимитного бабла
  - Нельзя заранее купить всё
- Надо отказоустойчиво
  - Минимум 2 ДЦ
  - Переживаем выход 1 ДЦ

Задача★: SLA 99.9 и дешево!

- Нет безлимитного бабла
  - Нельзя заранее купить всё
- Надо отказоустойчиво
  - Минимум 2 ДЦ
  - Переживаем выход 1 ДЦ
- Эффективно для клиентов
  - Latency поменьше

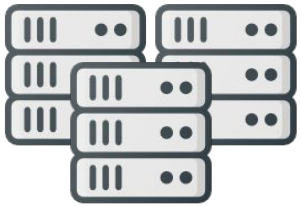
Задача★: SLA 99.9 и дешево!

- Нет безлимитного бабла
  - Нельзя заранее купить всё
- Надо отказоустойчиво
  - Минимум 2 ДЦ
  - Переживаем выход 1 ДЦ
- Эффективно для клиентов
  - Latency поменьше
- Эффективно для эксплуатации
  - Не растить операционку

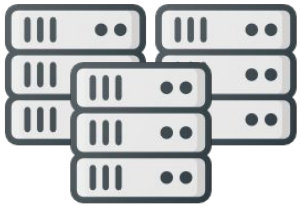
Вводные

Ограничения. **У нас есть**

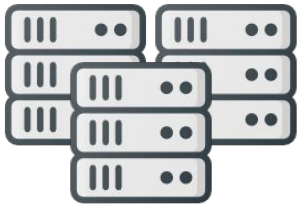
DC1



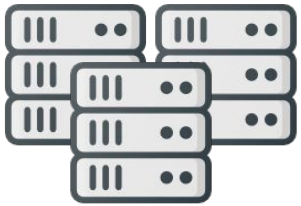
DC2



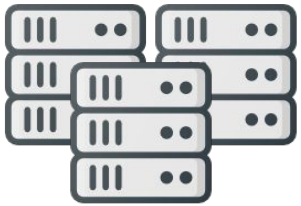
DC4



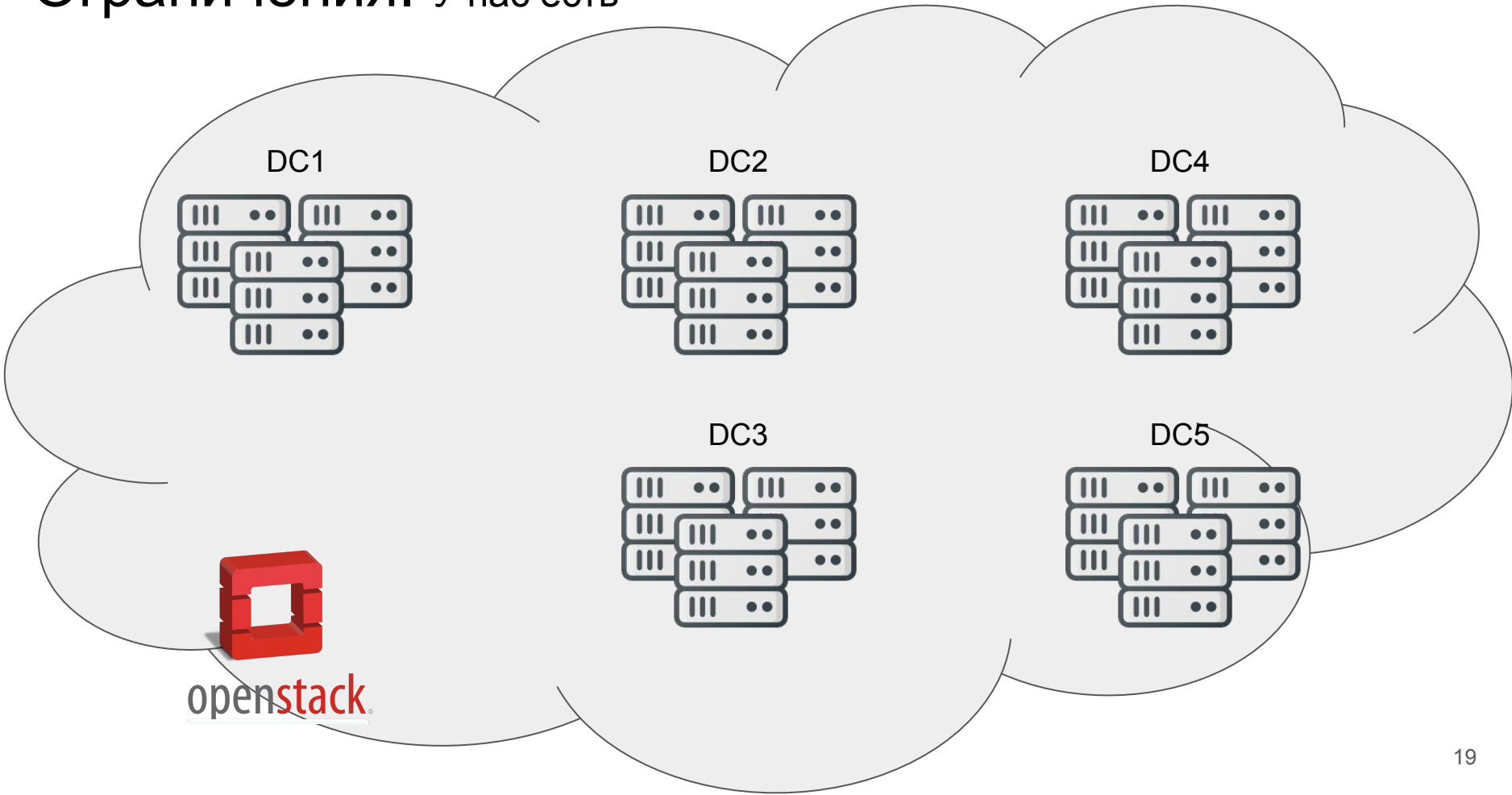
DC3



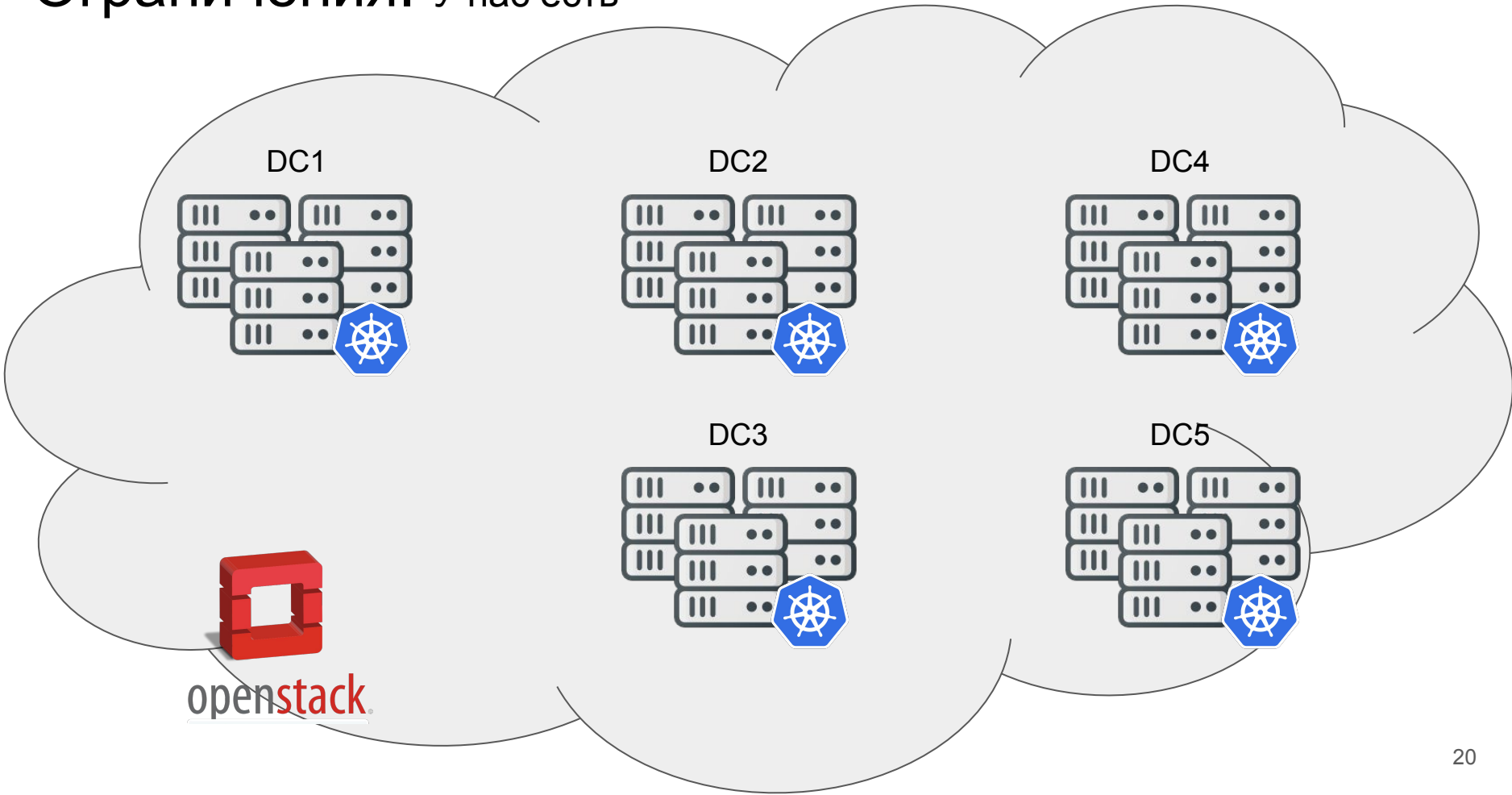
DC5



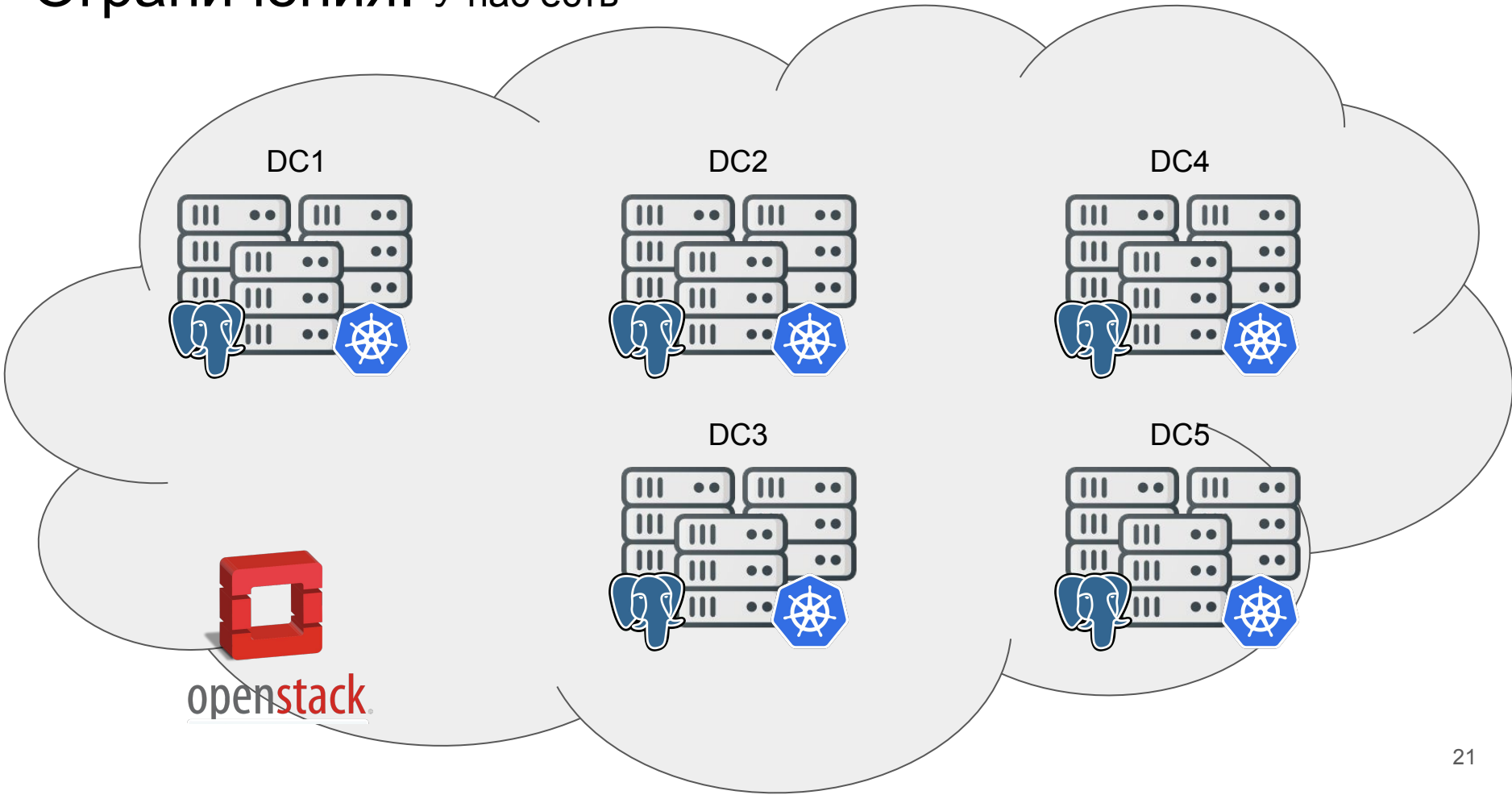
Ограничения. У нас есть



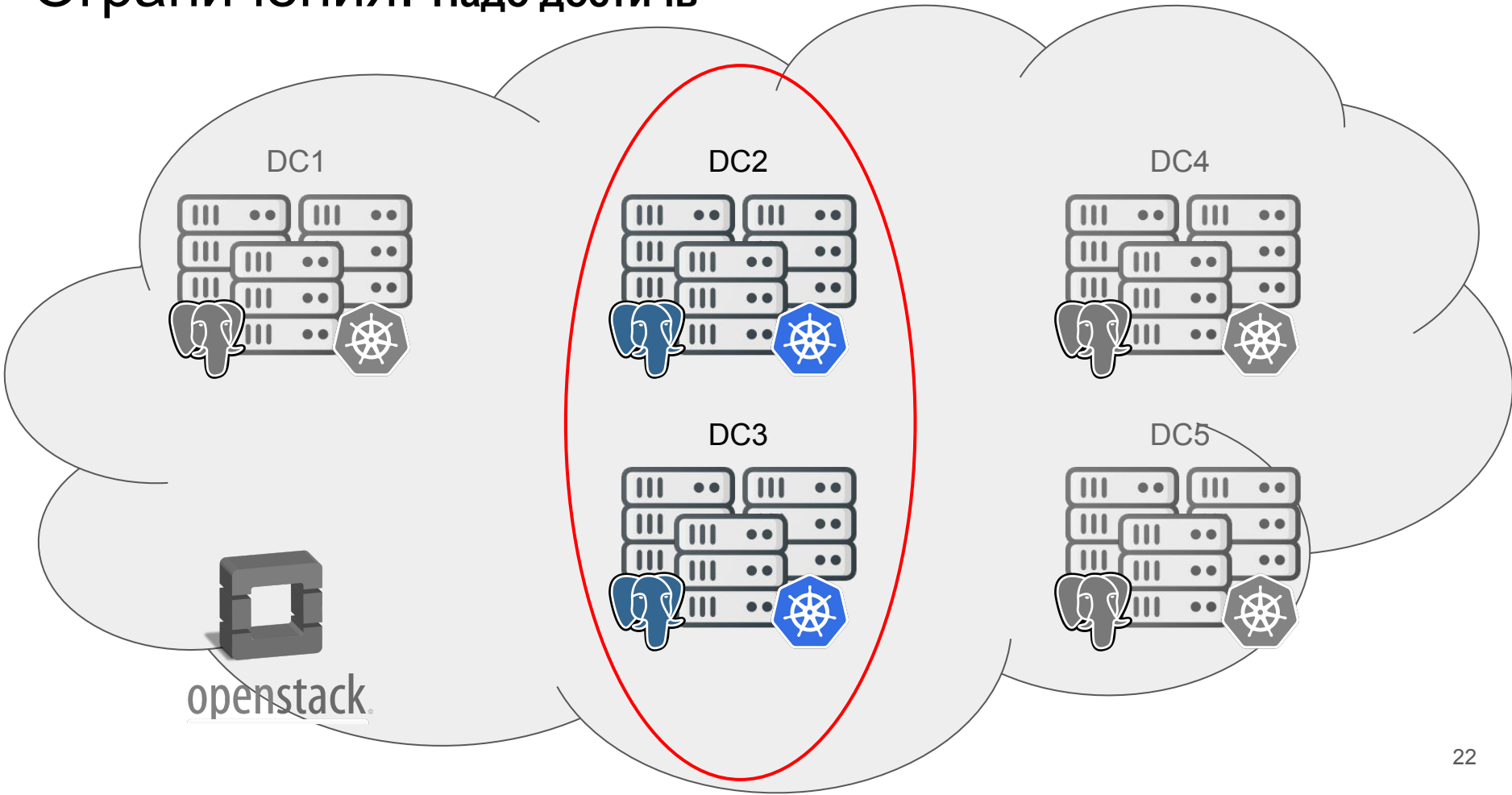
Ограничения. У нас есть



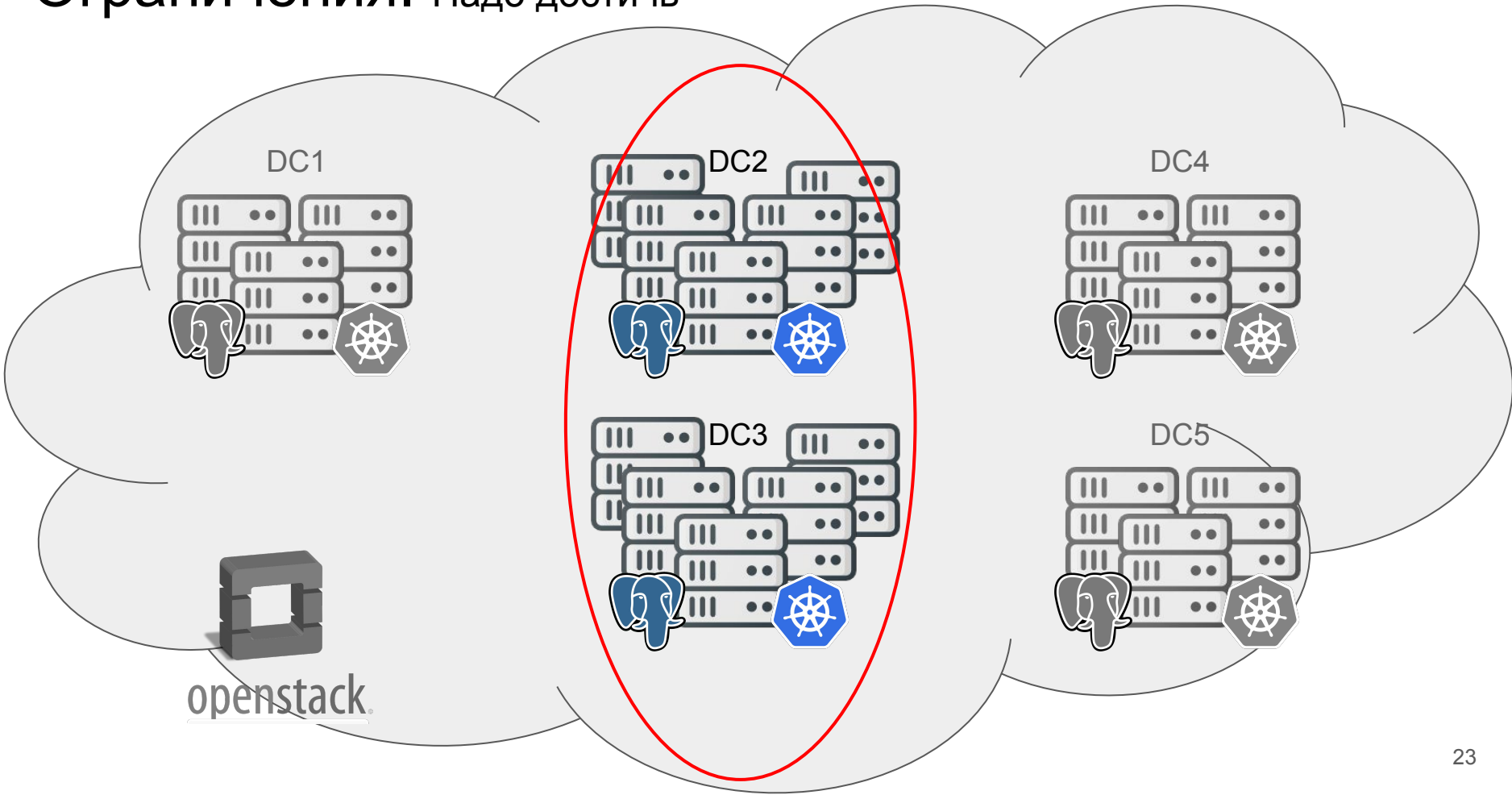
Ограничения. У нас есть



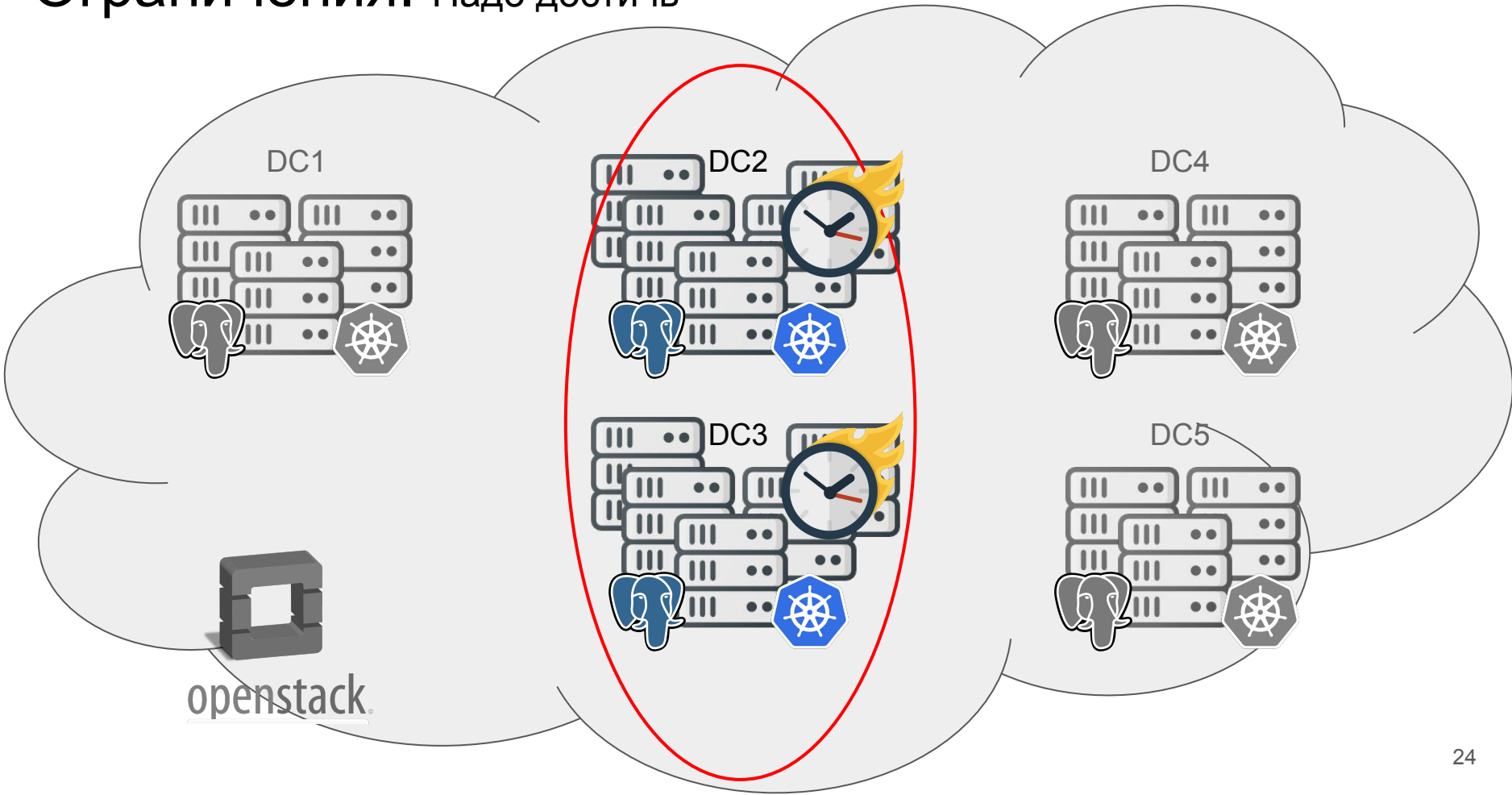
Ограничения. **Надо достичь**



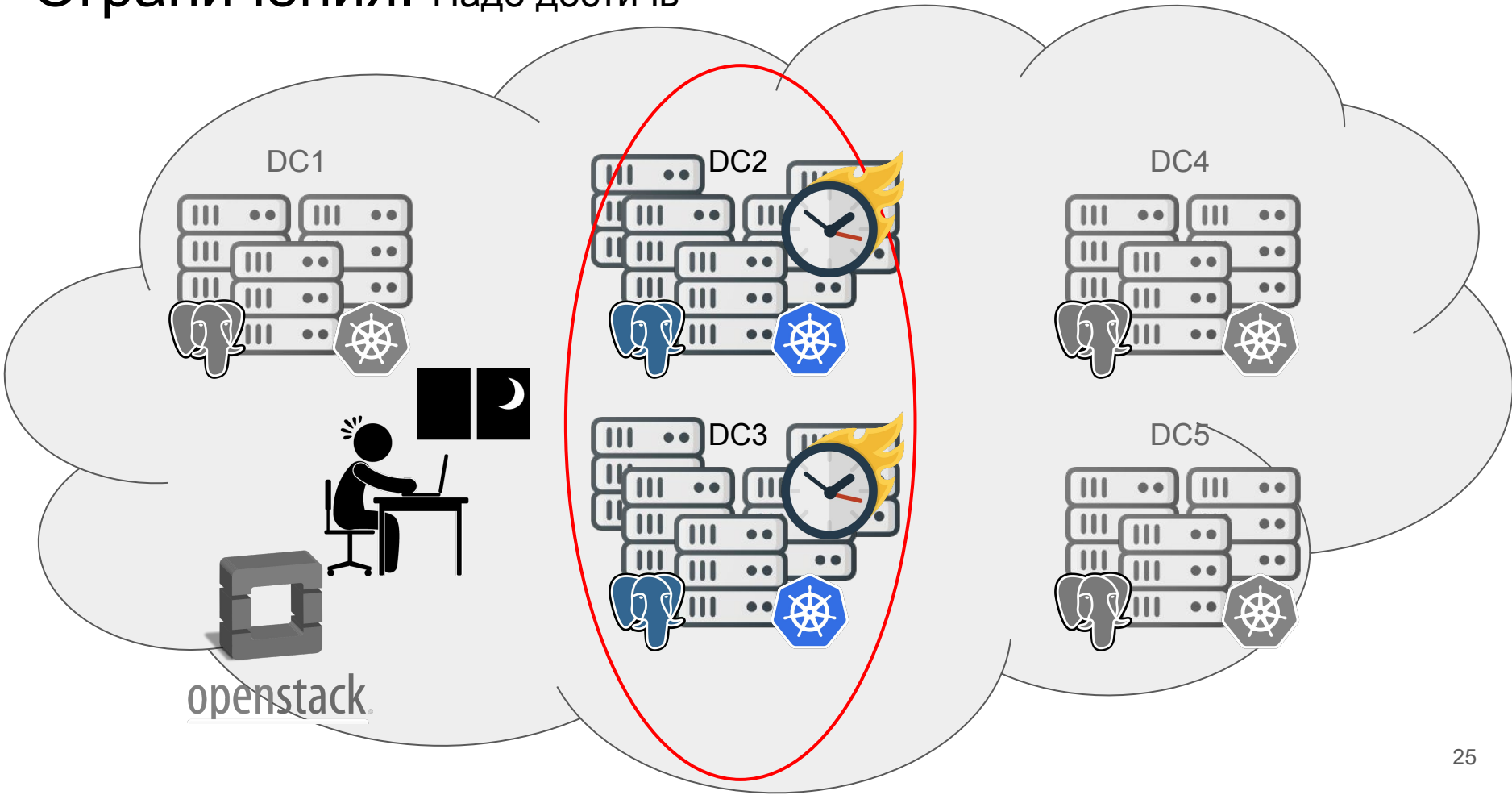
Ограничения. Надо достичь



Ограничения. Надо достичь



Ограничения. Надо достичь



Прикинем возможности

- Железо быстро == Аренда железа в ДЦ
  - Голое железо - дешево
  - Коммутация в том же ДЦ - удобно

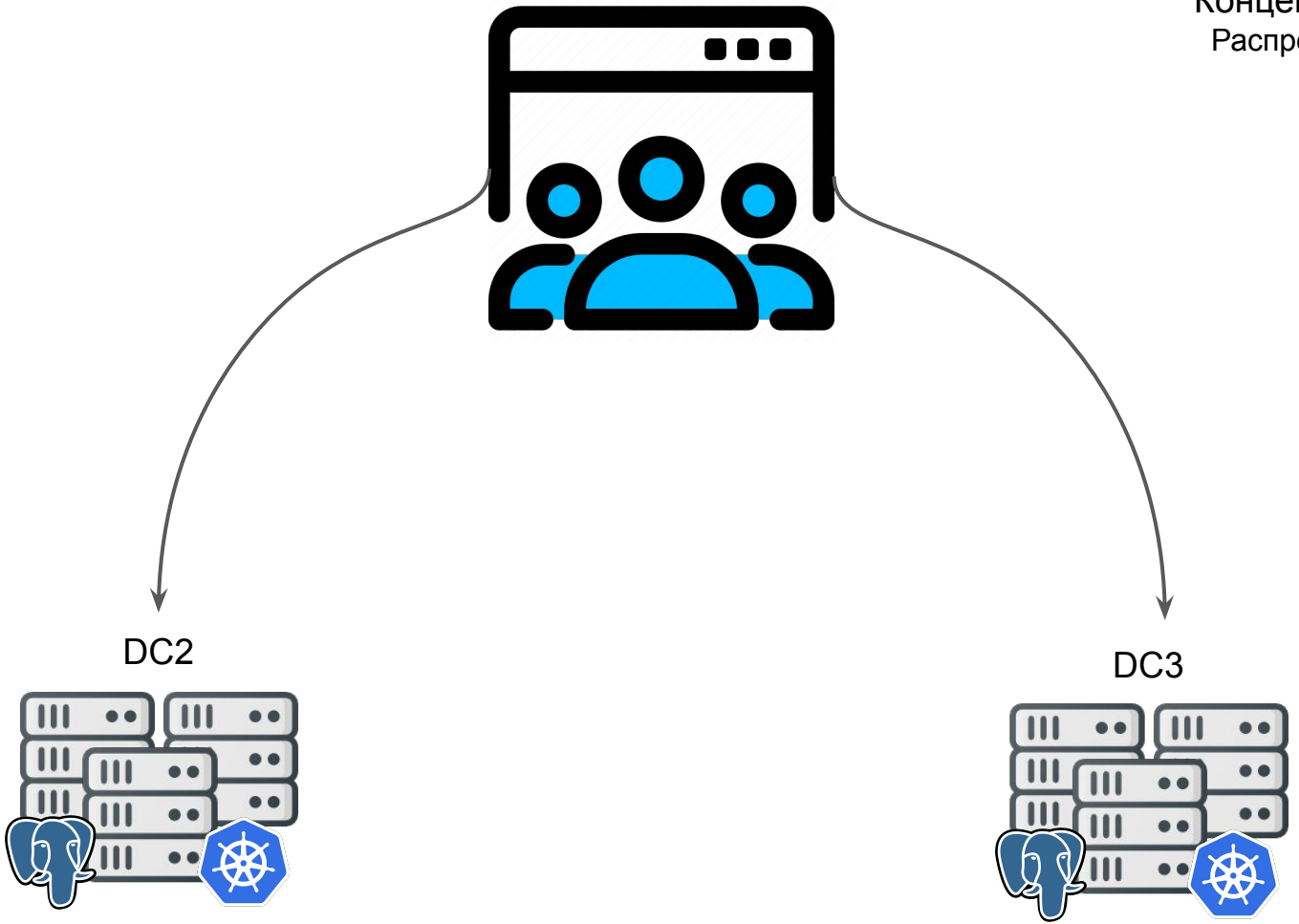
Прикинем возможности

- Железо быстро == Аренда железа в ДЦ
  - Голое железо - дешево
  - Коммутация в том же ДЦ - удобно
- Ресурсы быстро == Аренда облака (SberCloud)
  - Pay-as-you-go - используем когда нужно
  - РФ, прямая коммутация

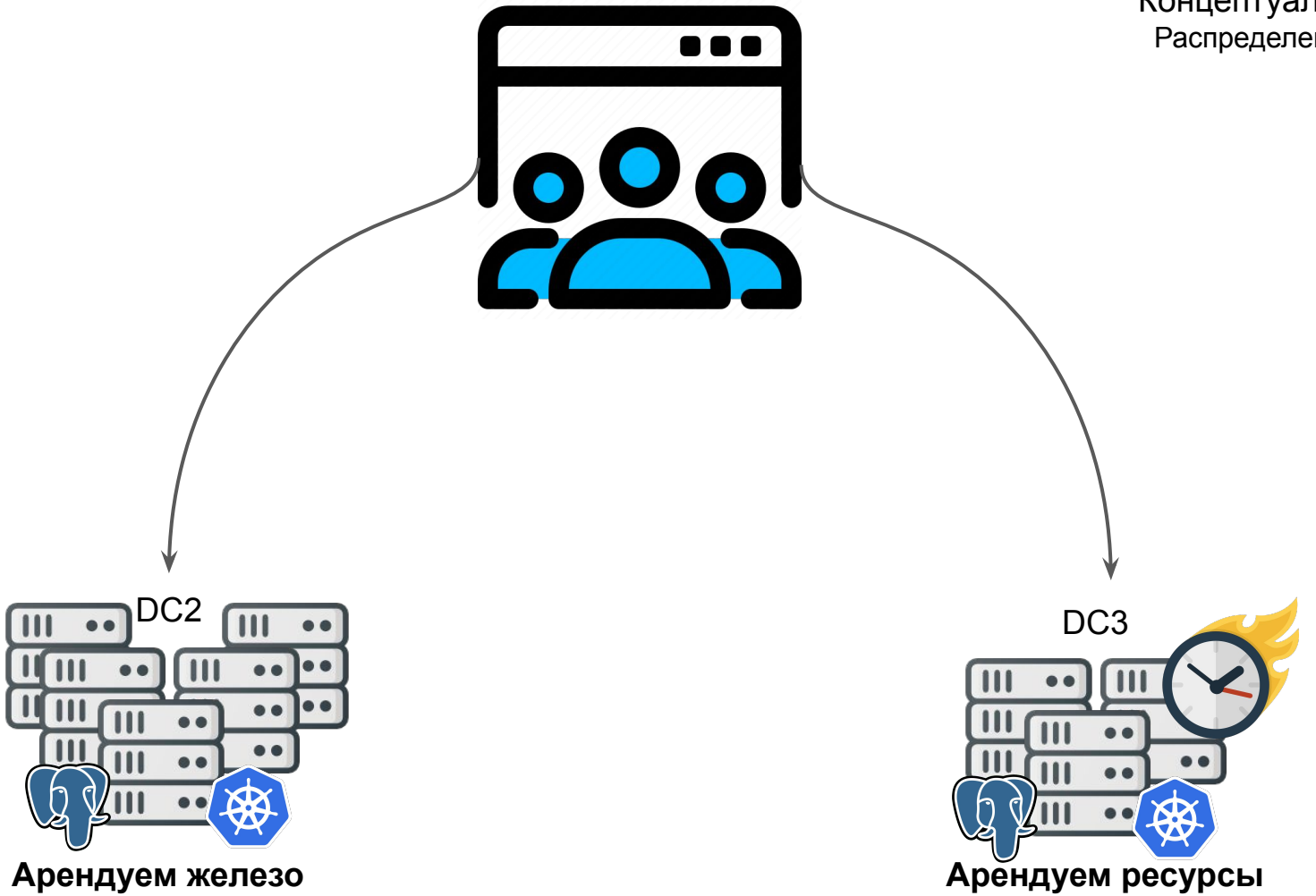
Прикинем возможности

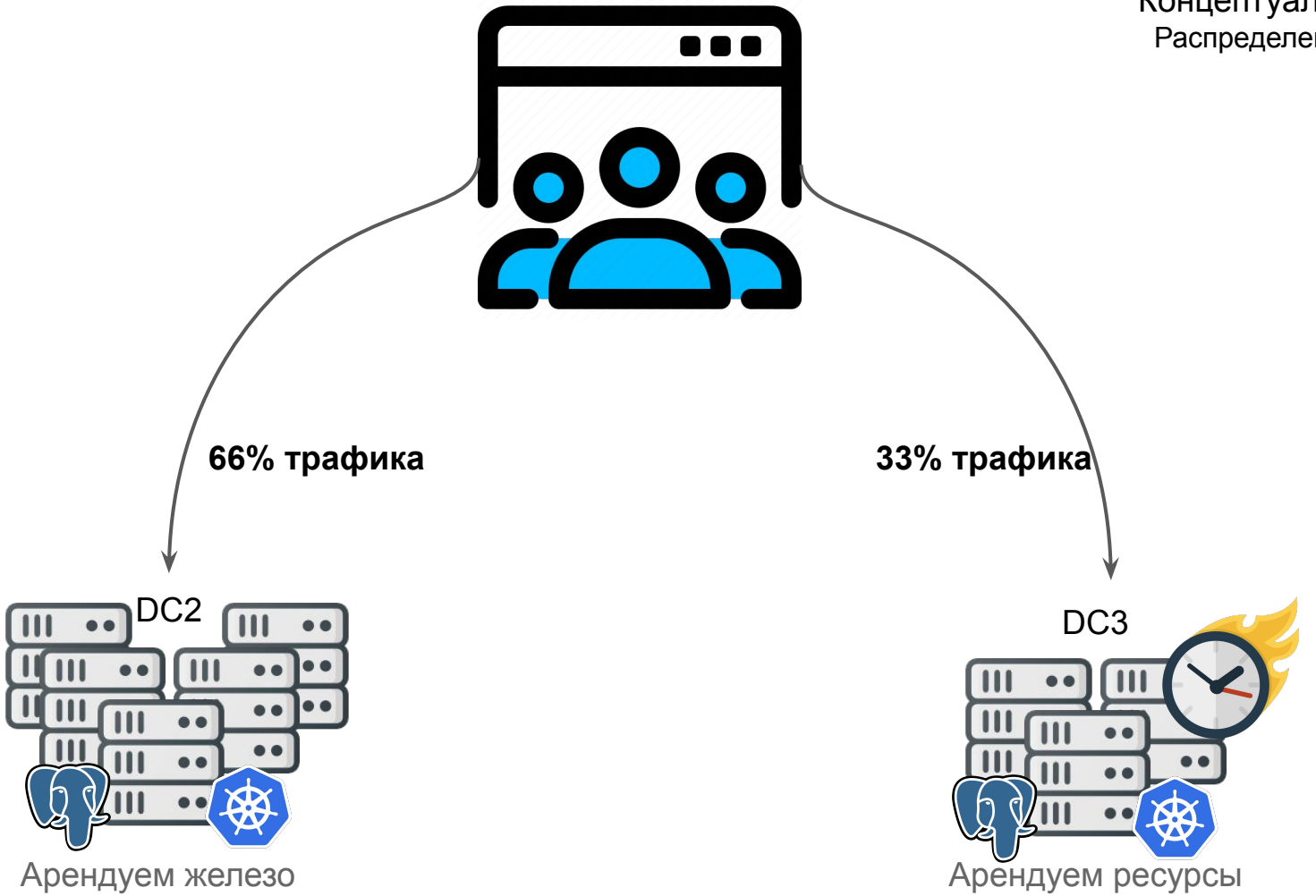
- Железо быстро == Аренда железа в ДЦ
  - Голое железо - дешево
  - Коммутация в том же ДЦ - удобно
- Ресурсы быстро == Аренда облака (Sbercloud)
  - Pay-as-you-go - используем когда нужно
  - РФ, прямая коммутация
- Как выбрать ДЦ?
  - Как распределять трафик?

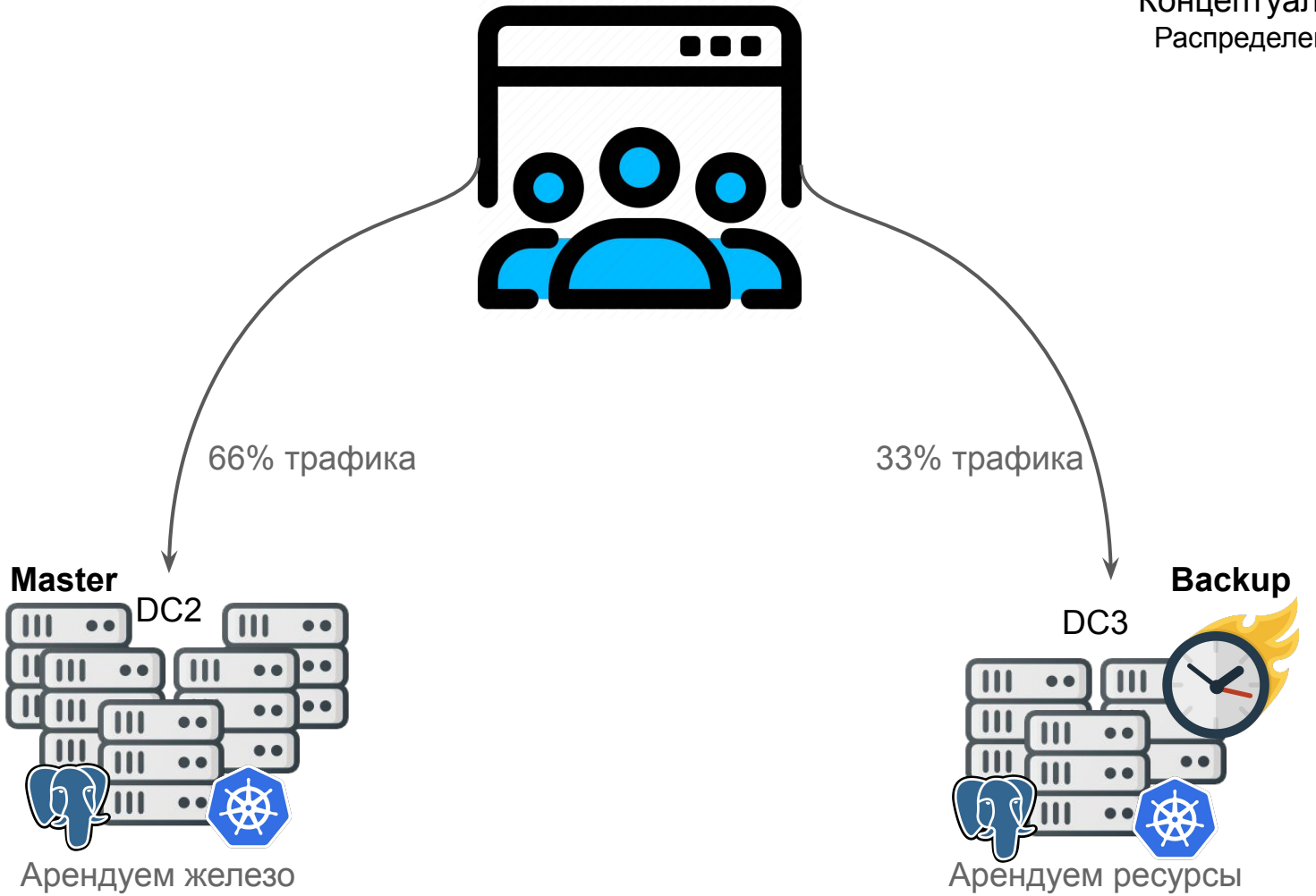
Концептуальный план
Распределения трафика



Концептуальный план
Распределения трафика




66% трафика
33% трафика
DC2
DC3
Арендуем железо
Арендуем ресурсы


66% трафика
33% трафика
Master
DC2
Backup
DC3
Арендуем железо
Арендуем ресурсы

Задача сложная - давай упрощать!

Что конкретно должно работать с SLA 99.9 ?

- 5-6 сервисов
- Работают в k8s

Задача сложная - давай упрощать!

Что конкретно должно работать с SLA 99.9 ?

- 5-6 сервисов
- Работают в k8s
- Часть Stateless
- Часть с СУБД PostgreSQL/Cassandra

Задача сложная - давай конкретику!

Для достижения SLA и ёмкости:

Надо разово увеличить (железо)

- K8s workers & routers
- Конкретные СУБД PostgreSQL
  - Cassandra разово, дальше достаточно только кэша
- Инфраструктурную обвязку
  - Логирование
  - Мониторинг

Задача сложная - давай конкретику!

Для достижения SLA и ёмкости:

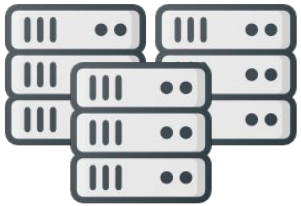
Уметь экономить на постоянной ёмкости (ресурсы)

- K8s workers & routers
- Конкретные СУБД PGSQL
  - ~~Cassandra разово, дальше достаточно только кэша~~
- ~~Инфраструктурную обвязку~~
  - ~~Логирование~~
  - ~~Мониторинг~~

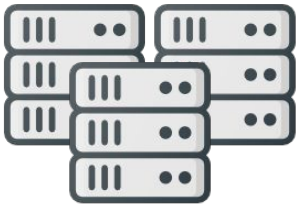
Давайте начинать!

С чего начинали

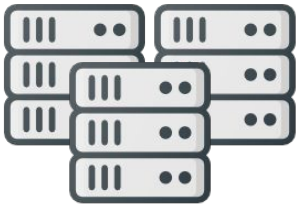
DC1



DC2

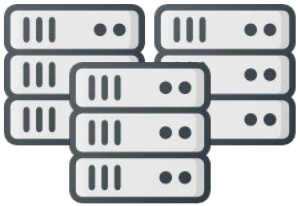


DC4

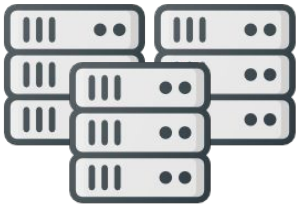


300+ железок

DC3

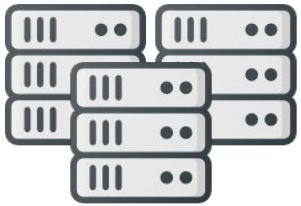


DC5

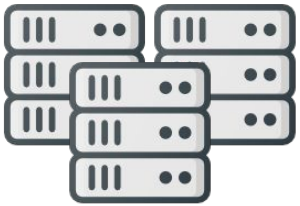


С чего начинали

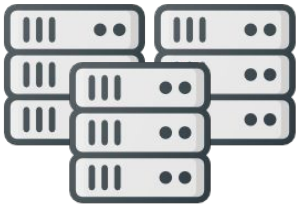
DC1



DC2



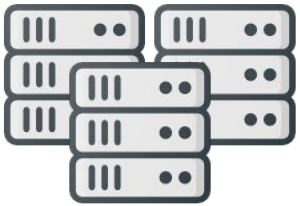
DC4



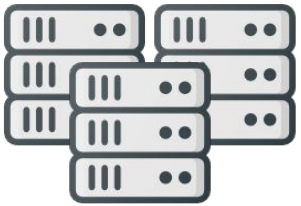
300+ железок

Загрузка через Preseed

DC3

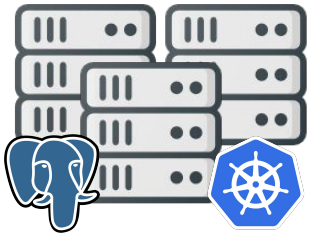


DC5

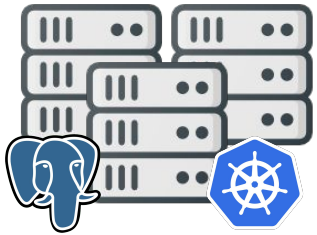


С чего начинали

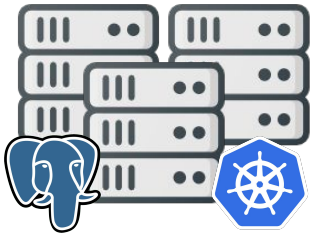
DC1



DC2



DC4



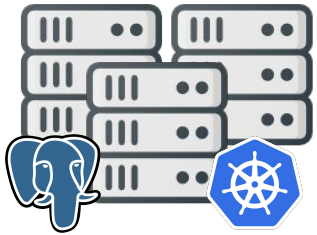
300+ железок

Загрузка через Preseed

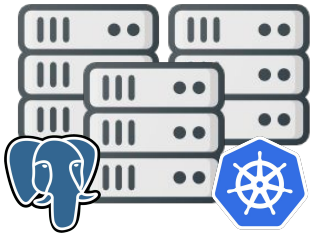
PostgreSQL ♥ Ansible

K8s ♥ Ansible

DC3

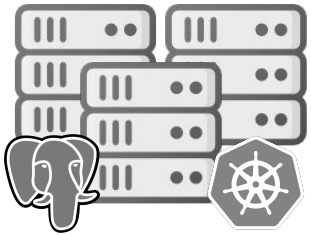


DC5

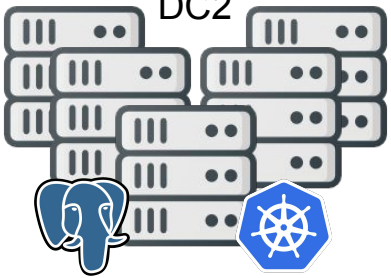


Надо добавить

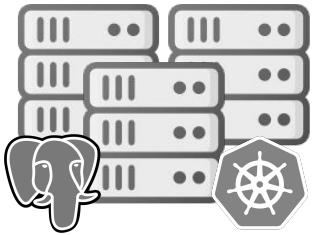
DC1



DC2

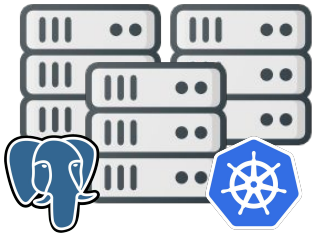


DC4

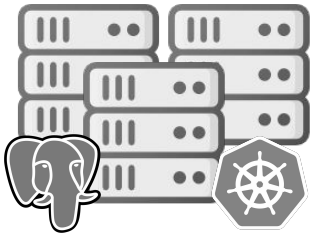


100 железок

DC3

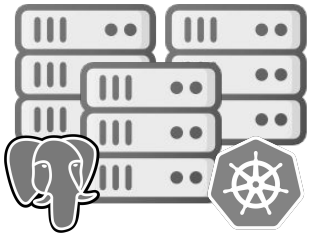


DC5

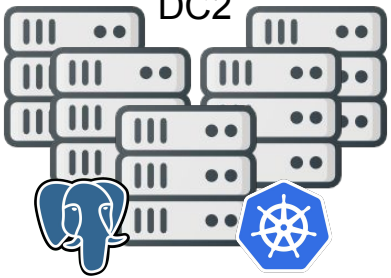


Надо добавить

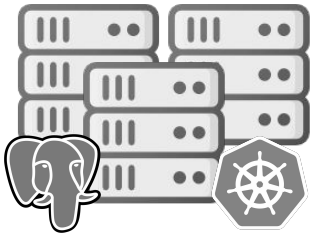
DC1



DC2



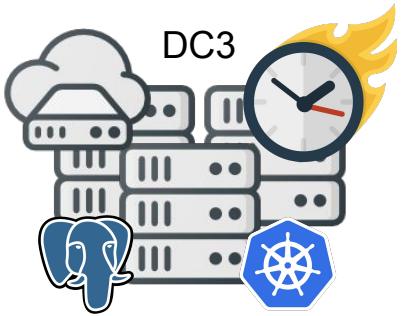
DC4



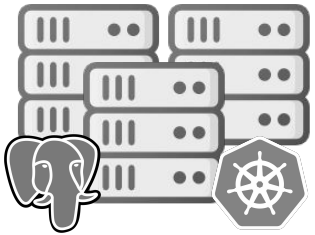
100 железок

Автозапуск ресурсов

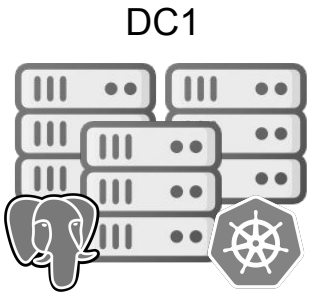
DC3



DC5



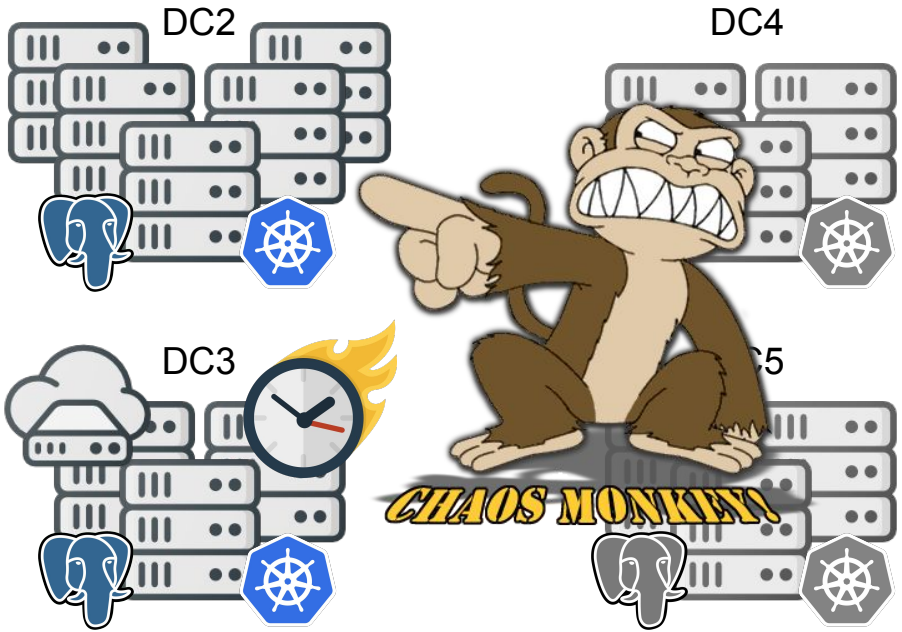
Надо добавить



100 железок

Автозапуск ресурсов

Обновление всех нод



Унифицировать загрузку

Железо

Openstack

SberCloud

Варианты для SLA

Оставить всё как есть

Деплой на Ansible плохо масштабируется

Алиса:

--- А как это будет работать при масштабировании в 10 раз?

Боб:

--- Плохо.

Боб явно делает что-то неправильно.
Не будь как Боб.

Оставить всё как есть

Деплой на Ansible плохо масштабируется

Процесс обновления кластеров сложный и долгий

согласования, надо прожимать кнопки, ждать

Алиса:

--- А как это будет работать при масштабировании в 10 раз?

Боб:

--- Плохо.

Боб явно делает что-то неправильно.
Не будь как Боб.

Оставить всё как есть

Деплой на Ansible плохо масштабируется

Процесс обновления кластеров сложный и долгий

согласования, надо прожимать кнопки, ждать

Дрифт конфигурации нод

Алиса:

--- А как это будет работать при масштабировании в 10 раз?

Боб:

--- Плохо.

Боб явно делает что-то неправильно.
Не будь как Боб.

Оставить всё как есть

Деплой на Ansible плохо масштабируется

Процесс обновления кластеров сложный и долгий

согласования, надо прожимать кнопки, ждать

Дрифт конфигурации нод

Новые ноды добавляются в кластер руками

долго и утомительно при больших объёмах

Алиса:

--- А как это будет работать при масштабировании в 10 раз?

Боб:

--- Плохо.

Боб явно делает что-то неправильно.
Не будь как Боб.

Всё как есть + managed k8s

➕ Оставляем наш деплой k8s в текущем виде

Всё как есть + managed k8s

➕ Оставляем наш деплой k8s в текущем виде

➕ Получаем автоскейлинг достаточно дёшево

есть "из коробки", нужно лишь разобраться в отличиях от апстрима

Всё как есть + managed k8s

➕ Оставляем наш деплой k8s в текущем виде

➕ Получаем автоскейлинг достаточно дёшево

есть "из коробки", нужно лишь разобраться в отличиях от апстрима

➖ +1 отдельный кластер, в который придётся настраивать деплои всем

Всё как есть + managed k8s

➕ Оставляем наш деплой k8s в текущем виде

➕ Получаем автоскейлинг достаточно дёшево

есть "из коробки", нужно лишь разобраться в отличиях от апстрима

➖ +1 отдельный кластер, в который придётся настраивать деплои всем

➖ Непонятки с аутентификацией в кластере

текущая схема требует контроля над apiserver'ом, которого в managed k8s наверняка нет

Всё как есть + managed k8s

➕ Оставляем наш деплой k8s в текущем виде

➕ Получаем автоскейлинг достаточно дёшево

есть "из коробки", нужно лишь разобраться в отличиях от апстрима

➖ +1 отдельный кластер, в который придётся настраивать деплои всем

➖ Непонятки с аутентификацией в кластере

текущая схема требует контроля над apiserver'ом, которого в managed k8s наверняка нет

➖ Необходимость синхронизации конфигурации наших кластеров с предлагаемыми в SberCloud

как самого k8s (в т.ч. версии), так и набора аддонов и их настроек

пользователи не должны знать/учитывать отличия между кластерами веб-платформы

Маленькая революция

Загрузка по сети

Маленькая революция

Загрузка по сети

Сборка своего .iso для быстрой загрузки

Маленькая революция

Загрузка по сети

Сборка своего .iso для быстрой загрузки

Добавляем оператор-обновлятор

Маленькая революция

Загрузка по сети

Сборка своего .iso для быстрой загрузки

Добавляем оператор-обновлятор

Добавляем автоскейлер

Низкоуровневые сложности

Образ и сеть

Надо разобраться

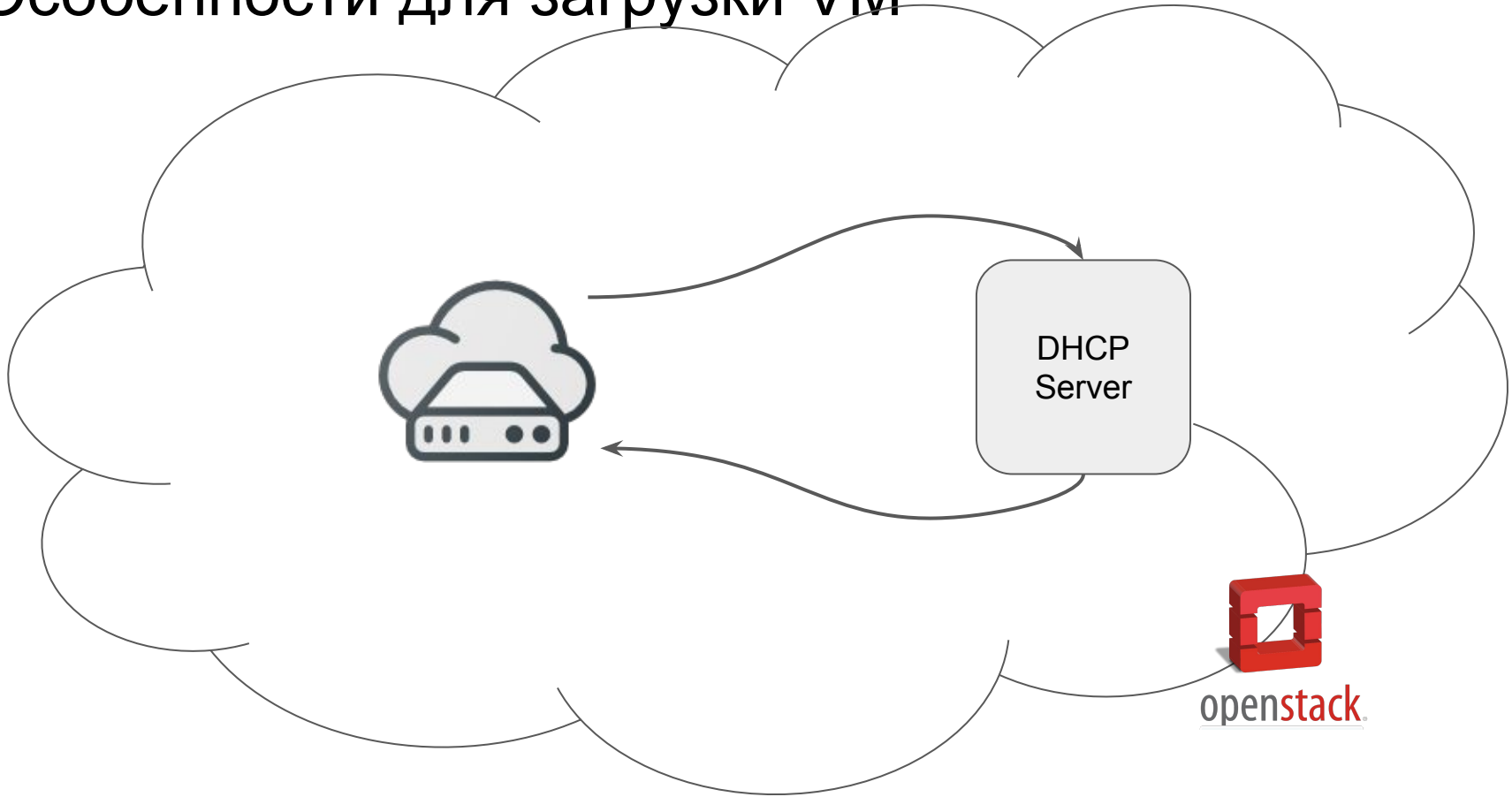
Загрузка PXE + DHCP + TFTP

Сборка iso-образов

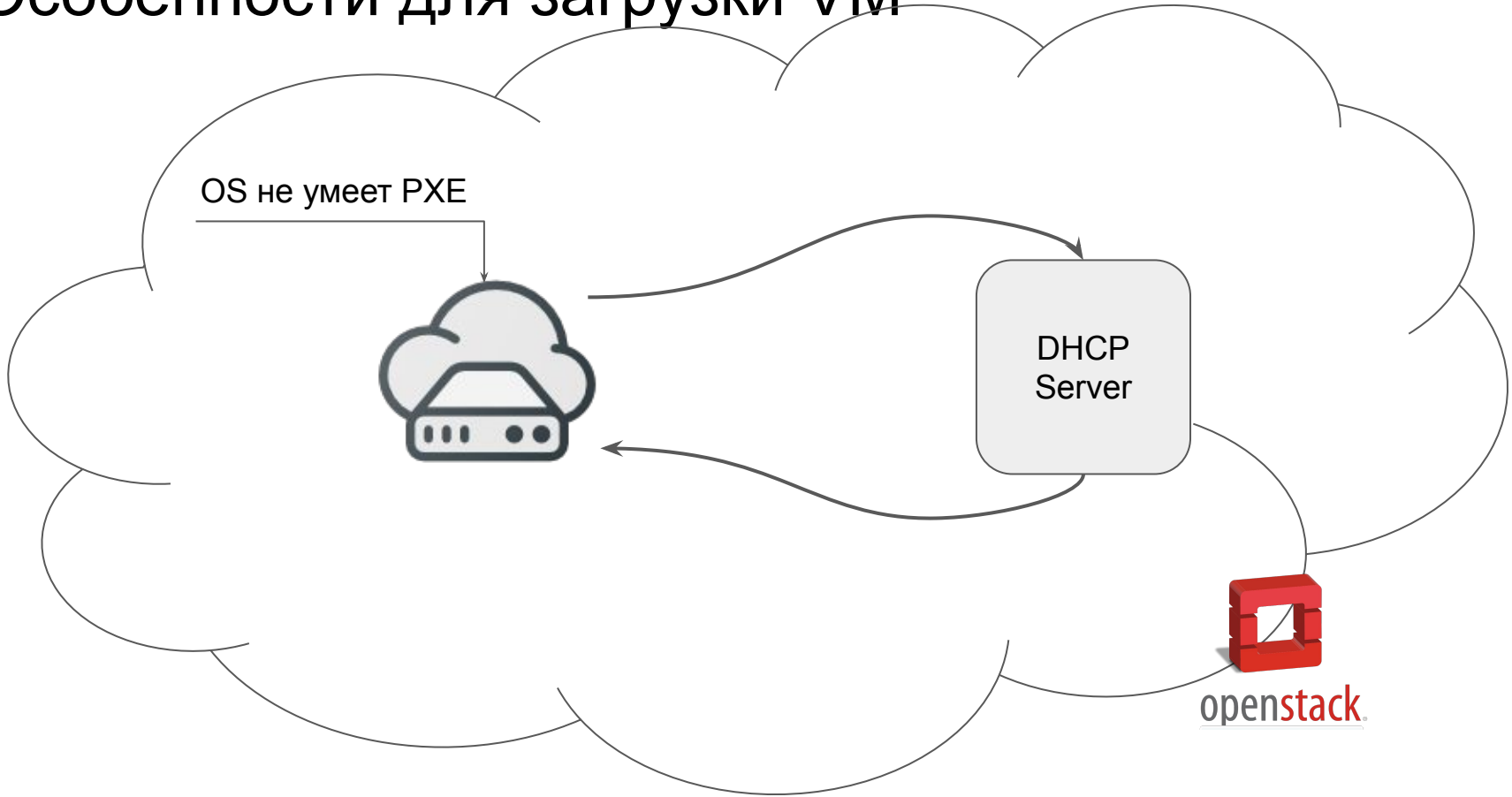
Особенности работы на разных коммутаторах

Алгоритм загрузки PXE

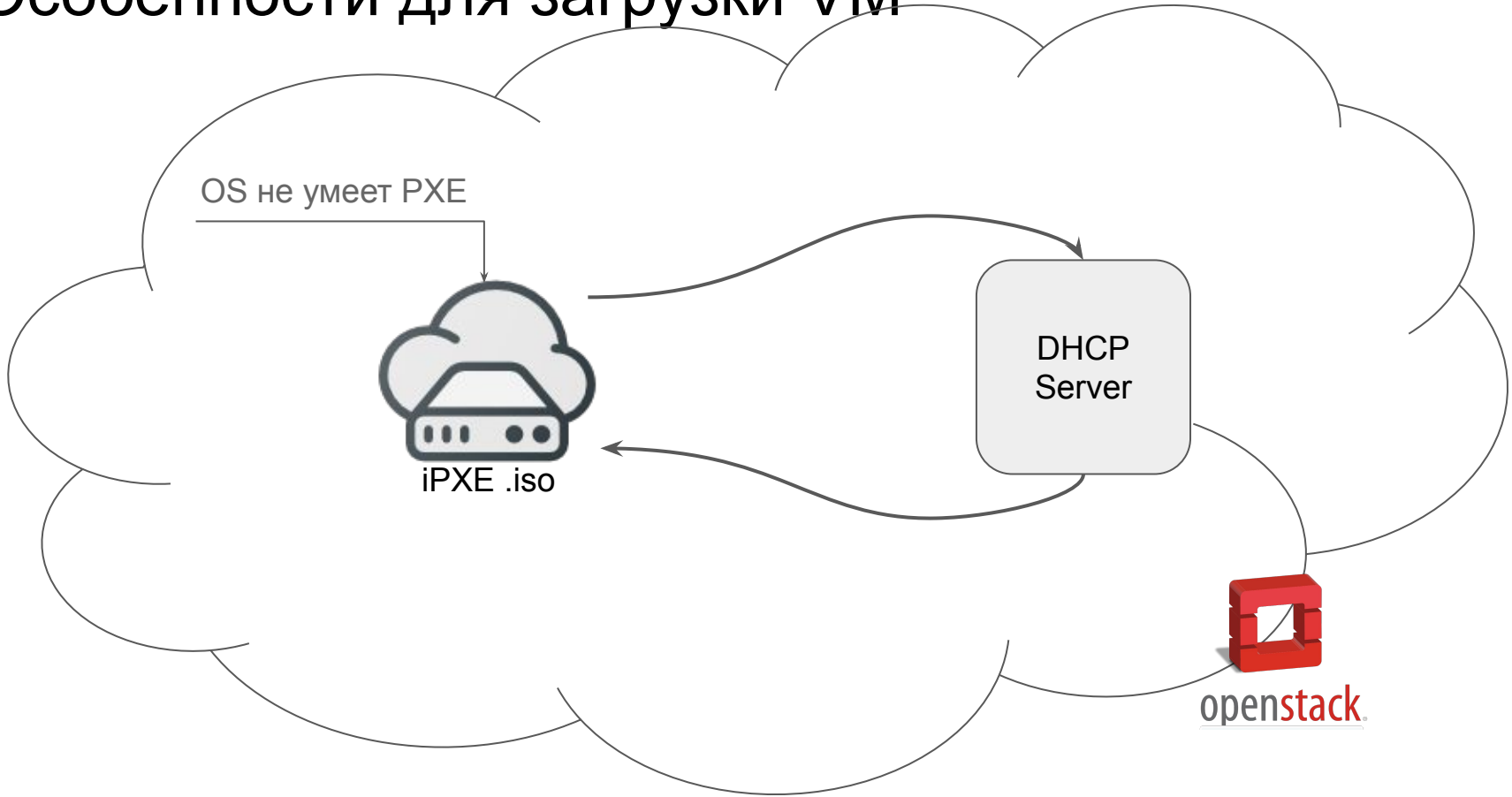
Особенности для загрузки VM



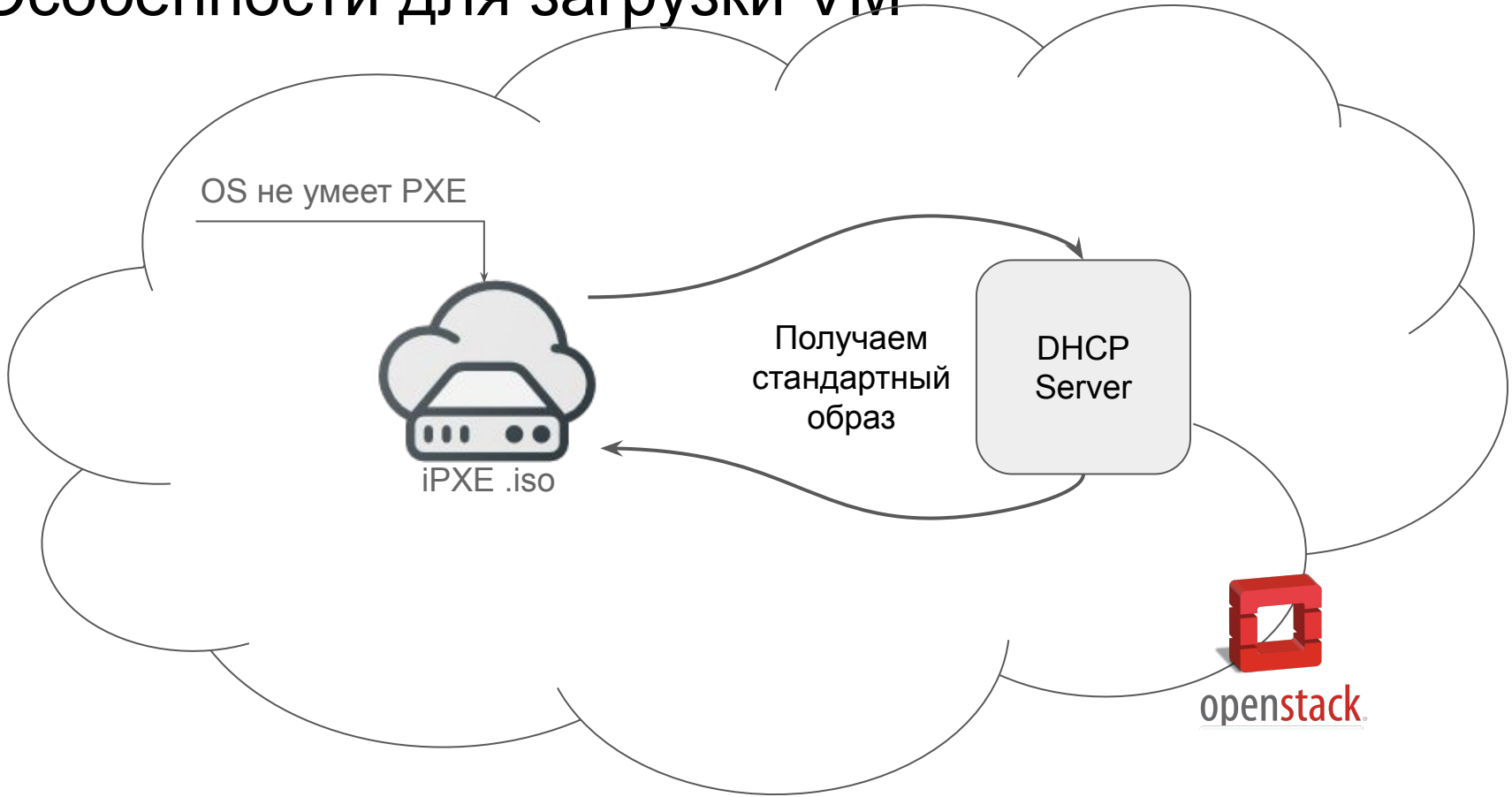
Особенности для загрузки VM



Особенности для загрузки VM



Особенности для загрузки VM



Управление зоопарком

загружать это здорово, теперь попробуй обнови

Как обновлять работающие ноды

Сервис обязательно должен уметь:

- Определять требующие обновления ноды.
- Перезагружать ноды, которые загружаются по PXE.
- Пересоздавать виртуальные машины.

Как обновлять работающие ноды

Сервис обязательно должен уметь:

- Определять требующие обновления ноды.
- Перезагружать ноды, которые загружаются по PXE.
- Пересоздавать виртуальные машины.

Дополнительно было бы здорово видеть:

- Выбор допустимого времени для обновления
  - Gitlab'a лучше обновлять рано утром
- Гибкие настройки интервалов, таймаутов, реакции на ошибки
  - Drain не смог быть выполнен за N минут

Как обновлять работающие ноды

Сервис обязательно должен уметь:

- Определять требующие обновления ноды.
- Перезагружать ноды, которые загружаются по PXE.
- Пересоздавать виртуальные машины.

Дополнительно было бы здорово видеть:

- Выбор допустимого времени для обновления
  - Gitlab'a лучше обновлять рано утром
- Гибкие настройки интервалов, таймаутов, реакции на ошибки
  - Drain не смог быть выполнен за N минут

Остановились на weaveworks/kured

- *Но решение от Flant тоже неплохое!*

Остановились на weaveworks/kured

Умеет:

- drain'ить ноды (в т.ч. с таймаутом);
- ребутать только одну за раз (блокировка посредством аннотации для daemonset'а);
- ребутать только в заданное время (дни недели + время суток);
- смотреть на алерты в Prometheus (блокирует процесс, если что-то пошло не так);
- не ребутать, если на ноде есть pod с заданным лейблом (если есть pod'ы, которые очень дорого выгонять (например, может быть полезно для CronJob'ов));
- отправлять уведомления (в т.ч. в Slack);
- метрики для Prometheus;

Остановились на weaveworks/kured

Умеет:

- drain'ить ноды (в т.ч. с таймаутом);
- ребутать только одну за раз (блокировка посредством аннотации для daemonset'а);
- ребутать только в заданное время (дни недели + время суток);
- смотреть на алерты в Prometheus (блокирует процесс, если что-то пошло не так);
- не ребутать, если на ноде есть pod с заданным лейблом (если есть pod'ы, которые очень дорого выгонять (например, может быть полезно для CronJob'ов));
- отправлять уведомления (в т.ч. в Slack);
- метрики для Prometheus;

Осталось только взять и сделать! :)

Автоскейлинг

Ок, а как скейлиться то?

- Явно - в облаке
- Группа автоскейлинг в SberCloud
- Механизм тот же, что для железа
- Инструмент автоскейлинга (cluster-autoscaler)
  - Знает состояние кластера и переключает цифру в группе автоскейлинга;
  - У него есть модуль для работы с облаком SberCloud. *Надеемся, что он заработает.*

Какие результаты скейлинга?

Какие результаты скейлинга?

Мы не знаем :) Он ещё не готов.

Облако молодое, мы первые с такими запросами

Какие результаты скейлинга?

Мы не знаем :) Он ещё не готов.

Облако молодое, мы первые с такими запросами

Надо патчить облако:

- Часть патчей уже накатили
  - Вопросы с кастомными DHCP опциями
  - Небольшие патчи Terraform провайдера
- Последний должен доехать 11.11.2021

Какие результаты скейлинга?

Мы не знаем :) Он ещё не готов.

Облако молодое, мы первые с такими запросами

Надо патчить облако:

- Часть патчей уже накатили
  - Вопросы с кастомными DHCP опциями
  - Небольшие патчи Terraform провайдера
- Последний должен доехать 11.11.2021

Прямой контакт с проджектами и техническими специалистами Sbercloud есть - **это явный плюс!**

Небольшие итоги

- У нас есть обновлятор нод кластера (*пока нет*)
  - Готовы брать, осталось найти время

Небольшие итоги

- У нас есть обновлятор нод кластера (*пока нет*)
  - Готовы брать, осталось найти время
- У нас есть самонадувающийся кластер k8s (*пока нет*)
  - Ждём патчей Sbercloud

Небольшие итоги

- У нас есть обновлятор нод кластера (*пока нет*)
  - Готовы брать, осталось найти время
- У нас есть самонадувающийся кластер k8s (*пока нет*)
  - Ждём патчей Sbercloud
- При выходе из строя одного ДЦ - SLA 99.9 выполнится!
  - SLA выполняется - пока мы просто немного надули второй ДЦ и раз в месяц чекаем его статус

⅔ Задачи сделано

А как же СУБД ?

RDS и не паримся?

RDS и не паримся?

Это не унифицировано → дорого в поддержке

RDS и не паримся?

Это не унифицировано → дорого в поддержке

Очень дорого в стоимости

78.14 ₽/hour вместо **23.15** ₽/hour

RDS и не паримся?

Это не унифицировано → дорого в поддержке

Очень дорого в стоимости

> 78.14 ₽/hour вместо 23.15 ₽/hour

В целом - всё должно получиться

НО, нам всё равно поддерживать имеющиеся кластера

200Гб PostgreSQL за 2 минуты

- Это возможно?
  - Как вы перельёте данные так быстро?
- Это работает?
  - Старт PostgreSQL + подключение к Master БД

200Гб PostgreSQL за 2 минуты

Как мы это делаем

- Автоскейлер SberCloud стартует VM с нашего .iso
  - 90-120 секунд необходимо на старт

200Гб PostgreSQL за 2 минуты

Как мы это делаем

- Автоскейлер SberCloud стартует VM с нашего .iso
  - 90-120 секунд необходимо на старт
- Snapshot’ы данных при старте VM
  - 100 строк Python

200Гб PostgreSQL за 2 минуты

Как мы это делаем

- Автоскейлер SberCloud стартует VM с нашего .iso
  - 90-120 секунд необходимо на старт
- Snapshot’ы данных при старте VM
  - 100 строк Python
- После старта сразу заводим трафик и СУБД работает

200Гб PostgreSQL за 2 минуты

Как мы это делаем

- Автоскейлер SberCloud стартует VM с нашего .iso
  - 90-120 секунд необходимо на старт
- Snapshot’ы данных при старте VM
  - 100 строк Python
- После старта сразу заводим трафик и СУБД работает
- WAL’ы догоняются в течении 2-3 минут

200Гб PostgreSQL за 2 минуты

Подводные камни

- Для подключения ко всем СУБД есть только один IP
  - Завели его в HAProxy, но это ненормальное распределение трафика
  - Надо сделать автодискавери

200Гб PostgreSQL за 2 минуты

Подводные камни

- Для подключения ко всем СУБД есть только один IP
  - Завели его в HAProxy, но это ненормальное распределение трафика
  - Надо сделать автодискавери
- Данные для snapshot, берутся от конкретной VM
  - сломалась машина, не можем скейлиться

200Гб PostgreSQL за 2 минуты

Подводные камни

- Для подключения ко всем СУБД есть только один IP
  - Завели его в HAProxy, но это ненормальное распределение трафика
  - Надо сделать автодискавери
- Данные для snapshot, берутся от конкретной VM
  - сломалась машина, не можем скейлиться
- База “Прохладная”

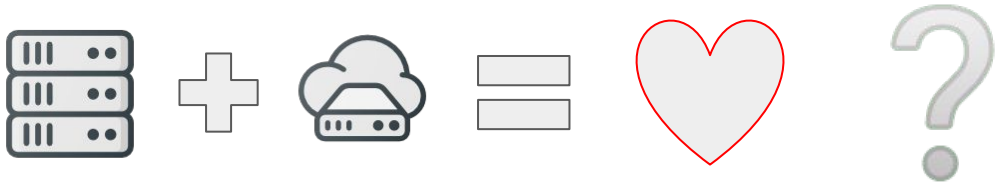
200Гб PostgreSQL за 2 минуты

Подводные камни

- Для подключения ко всем СУБД есть только один IP
  - Завели его в HAProxy, но это ненормальное распределение трафика
  - Надо сделать автодискавери
- Данные для snapshot, берутся от конкретной VM
  - сломалась машина, не можем скейлиться
- База “Прохладная”
- Нужно следить, чтобы не кончился канал на Master СУБД

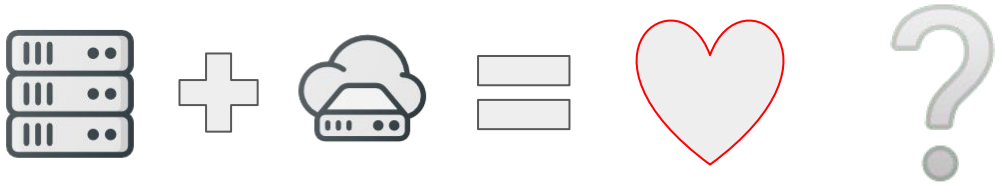
Подводим итоги

Гетерогенные инфраструктуры позволяют быть очень гибкими



Гетерогенные инфраструктуры позволяют быть очень гибкими

Есть решения, которые можно применять для виртуальных и физических ресурсов



Гетерогенные инфраструктуры позволяют быть очень гибкими

Есть решения, которые можно применять для виртуальных и физических ресурсов

Все решения open source

Результаты

- SLA 99.9 - **ОК**
- SLA 99.9★ - пока **не ОК**

Результаты

- SLA 99.9 - **ОК**
- SLA 99.9★ - пока **не ОК**
- Скейл Worker, router - пока **не ОК**
- Скейл СУБД - **ОК**

Результаты

- SLA 99.9 - **ОК**
- SLA 99.9★ - пока **не ОК**
- Скейл Worker, router - пока **не ОК**
- Скейл СУБД - **ОК**
- Унификация работы с VM/HW - **ОК**
- Знания кишок - **бесценны**

Что дальше

- Ждём патча от SberCloud
  - Скейлим worker и router

Что дальше

- Ждём патча от SberCloud
  - Скейлим worker и router
- Внедряем обновления нод с рестартами

Что дальше

- Ждём патча от SberCloud
  - Скейлим worker и router
- Внедряем обновления нод с рестартами
- Просто коммитим IaC, а роботы пусть сами разбираются

Спасибо!

Вопросы?

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

2GIS