

# Создание гибридного облака в ритейле

Презентация для CNews конференции  
«Облачные технологии 2021»





# Структура

1

**СОСТОЯНИЕ  
НА НАЧАЛО  
2019 ГОДА**

2

**ПУБЛИЧНОЕ  
ОБЛАКО**

3

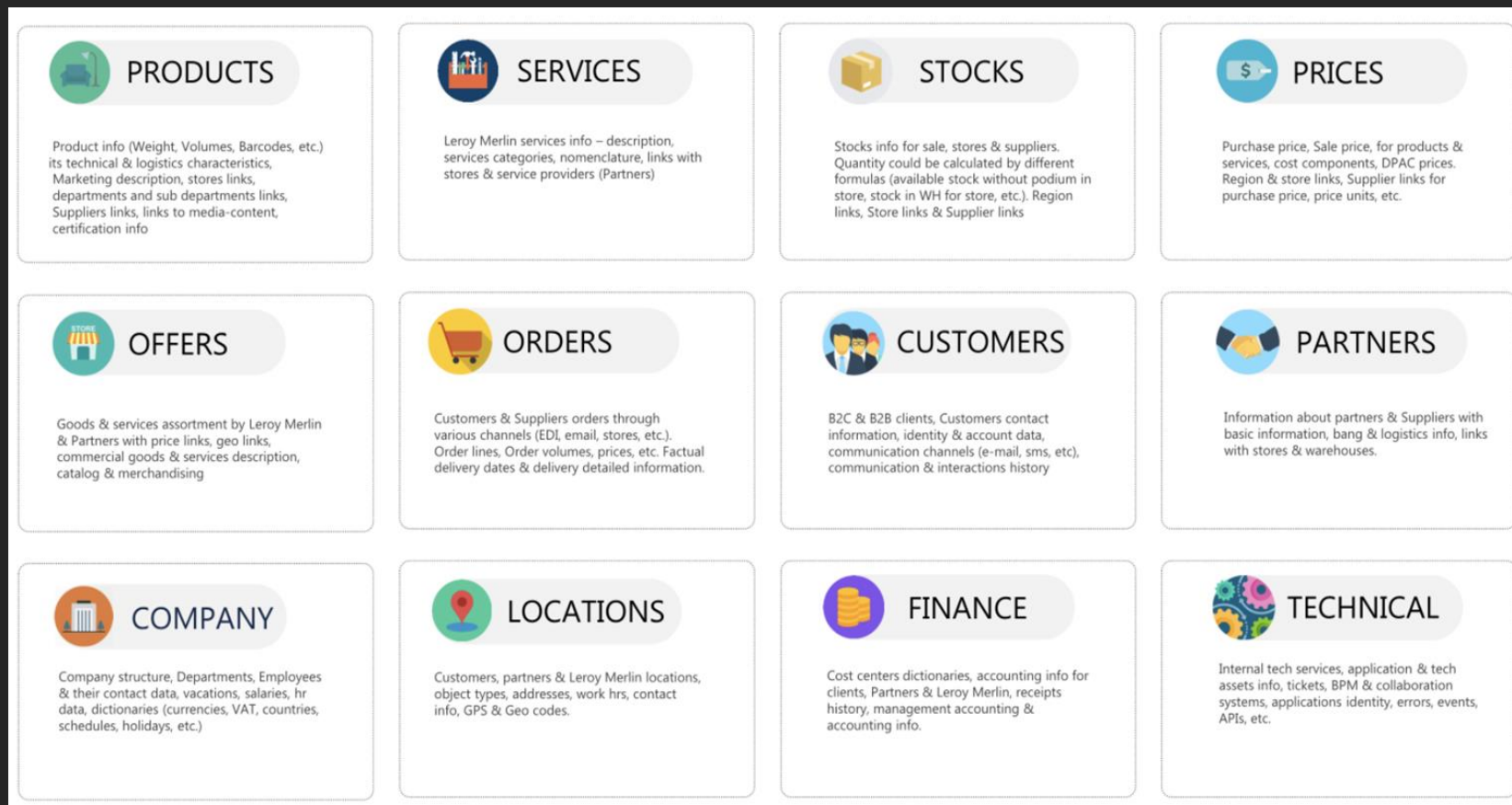
**ПРИВАТНОЕ  
ОБЛАКО**

4

**ГИБРИДНОЕ  
ОБЛАКО**

# О КОМПАНИИ

«Леруа Мерлен» — международная компания - ритейлер, специализирующаяся на продаже товаров для строительства, ремонта и обустройства дома, дачи и сада.



IT: 500+ человек, 19 доменов, 100 продуктовых команд

# Начало

РЕТРОСПЕКТИВА  
СТРАТЕГИИ 2019

## На начало 2019 года

- **Digital transformation** полным ходом
- Замена подрядчиков **in-house** разработкой
- Переход от экспертов к **data driven** принятию решений (команда Data Platform)
- Продуктовые команды (плюс API-зация всех приложений)
- Команда развития DevOps практик
- On-premise VMware (виртуалка через «ServiceDesk API», SLA на тикеты 1 неделя)
- Публичное Яндекс Облако – штучные эксперименты



# Подходы

## Legacy

Эффективная работа в облаке невозможна.

Не может масштабироваться.

Переписывать систему некому.

## Cloud Ready

Проектировалось не под облако, но может работать в облаке.

Есть CI/CD пайплайны и автотесты.

Может масштабироваться.

API подход.

Может стать микросервисом.

## Cloud Native

Приложение изначально проектировалось для облака.

Использует все преимущества облачной архитектуры.

Не содержит облачных антипаттернов

## Data Platform

Приложения для хранения и обработки данных.

Специфичный набор инструментов.



# Миссия команды гибридного облака

РЕТРОСПЕКТИВА  
СТРАТЕГИИ 2019

## Mission

Предоставить полностью функциональную облачную платформу, которая будет иметь **эффективное ценообразование** и будет предоставлять необходимые для бизнеса **гарантированные параметры сервисов**.

## Vision

- IaaS + PaaS
- Cloud agnostic (продуктовые команды не должны ощущать разницу между публичным и приватным облаком)
- Оптимизация затрат
- Преимущества от managed services в публичных облаках
- Гиперскейл в публичных облаках (платим только за то что работает)

# IaaS Strategy

РЕТРОСПЕКТИВА  
СТРАТЕГИИ 2019

## Состояние на Q2 2019

- Все процессы люди делают вручную с минимальной автоматизацией

## Мы поставили себе цель на год (Q3 2019 – Q3 2020)

- **VMware** – существующие среды, legacy и базы данных
- **Yandex Cloud + OpenStack** – cloud native приложения (при желании)
- Команды используют **IaaS в публичном и приватном облаке** без подключения к процессу управления ресурсами инфраструктурных или SD инженеров
- **Web UI консоль** для «песочниц» и как дашборд для менеджеров
- Для dev/prod – **Infrastructure as Code** с применением API/CLI/Terraform
- Командам выставляются ежемесячные счета за ресурсы в облаке

# IaaS/PaaS Strategy

РЕТРОСПЕКТИВА  
СТРАТЕГИИ 2019

## Цели на 2020-2023

- **VMware** только для Legacy
- Базы данных в **VMware, публичных и частных облаках**
- Базы данных используют **multi-datacenter архитектуру** и выдерживают выключение дата центра
- Приложения используют **multi-datacenter** и **active-active (master-master)** архитектуру
- **IaaS Marketplace** для виртуальных машин / дисков / сетей
- **PaaS Marketplace** для БД / Kafka / etc
- **Мониторинг ресурсов и сбор метрик** делается автоматически
- **Ручной труд** инженеров применяется только **для incident management**





# 2 ПУБЛИЧНОЕ ОБЛАКО

# Публичное Облако – первые шаги

## Яндекс облако на момент Q1 2019 г

- **Три ЦОД** в РФ для балансирования и DR (Москва, Владимир, Рязань)
- Полностью **функционирующий IaaS**
- Несколько **DBaaS**
- **Авторизация** через Яндекс.Паспорт
- **Сеть через VPN** в виртуальную машину
- **Убедительный Roadmap** на 2019 – 2020 г



# Публичное Облако глазами разработчиков

## Битва подходов

### Legacy подход

Нужна виртуальная машина

Написали тикет в SD

SLA 1 неделя

Подождали 3 дня

Инженер задал уточняющий вопрос

Подождали еще 3 дня

Инженер сделал, но не совсем то, потому что мы неправильно объяснили, а он неправильно понял

**Получили виртуальную машину через 1 неделю**

### Облачный подход

Нужна виртуальная машина

Выбрали инструмент по вкусу (Web UI, консольный клиент, Terraform, сами подёргали за API)

**Получили виртуальную машину через 3 минуты**



# Публичное Облако – через полгода

## Яндекс облако на момент Q3 2019 г

- Уже **больше DBaaS**
- Появились PaaS
- Авторизация через Яндекс.Паспорт + **сделана Федерация через ADFS**
- **Надежная сеть через два оптических линка 10G + 10G**
- **Interconnect** для обеспечения сетевой связанности Leroy Merlin и сетей в Облаке
- Мы видим как **выполняется Roadmap** на 2019 – 2020
- Появились **первые серьезные проекты**



# Яндекс Облако сейчас

На стороне Яндекс

**30+** реализованных Feature Requests

На стороне Leroy Merlin

**FinOps** инструменты

**Метрики**

**Интеграция DNS**

**Мониторинг + Алертинг**

Скидка



# Публичное Облако: развитие

## Альтернатива:

- Провели **исследования рынка**
- Сравнили цены и функциональные возможности

## Готовы использовать в 2021:

- Сделали пилотный проект
- Подготовили и проверили федеративную авторизацию через ADFS
- Драфт договора с **прогрессивной скидкой**



# 3 ПРИВАТНОЕ ОБЛАКО



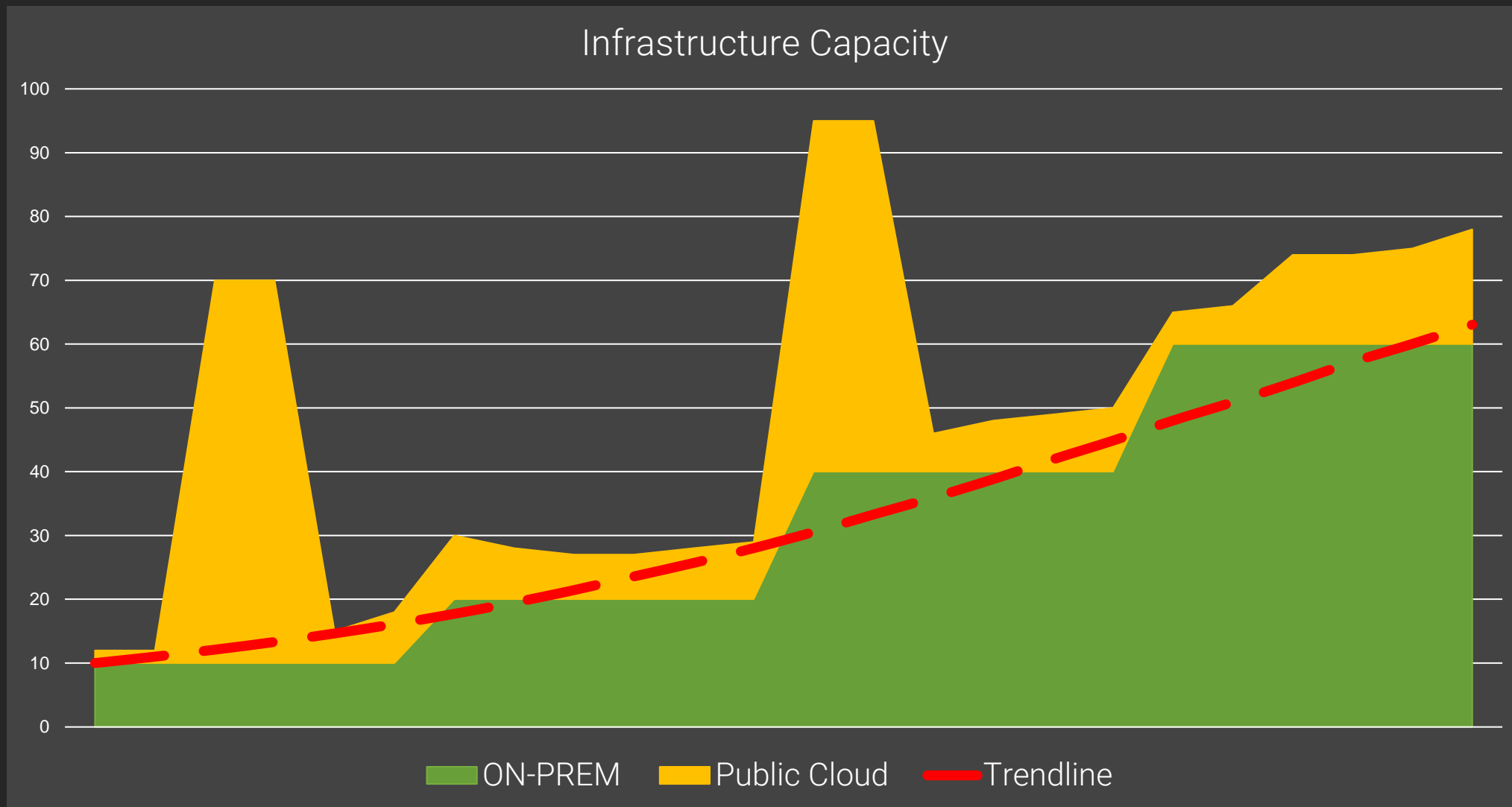
# Зачем нужен гибридный подход

## 1. Cost effectiveness





# Зачем нужен гибридный подход





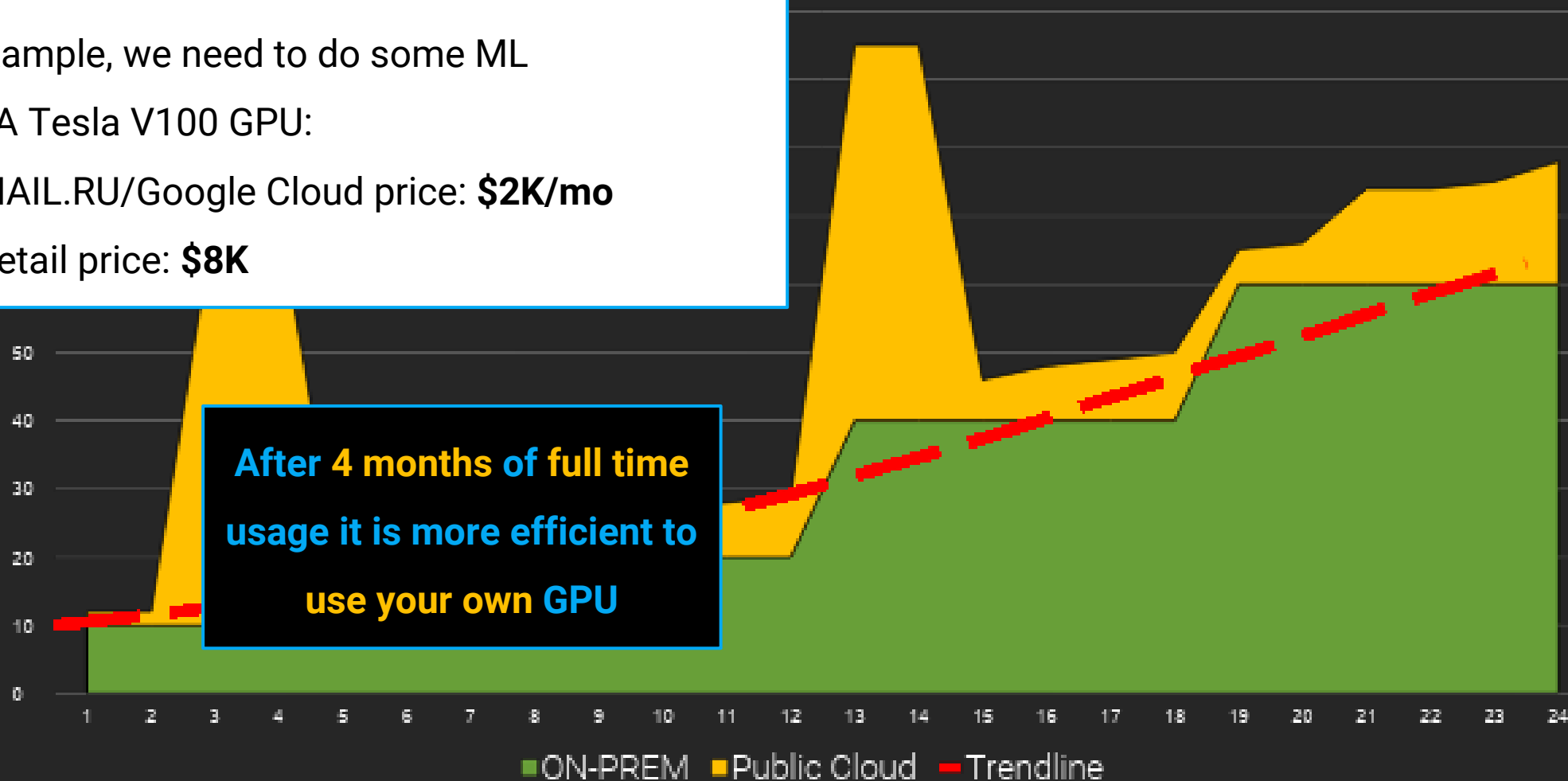
# Экономия на частных ресурсах

Infrastructure Capacity

For example, we need to do some ML

NVIDIA Tesla V100 GPU:

- MAIL.RU/Google Cloud price: \$2K/mo
- Retail price: \$8K



After 4 months of full time usage it is more efficient to use your own GPU



# Зачем нужен гибридный подход

**2. Вы не управляете тем,  
чем не владеете**



# IaaS Requirements

**РЕТРОСПЕКТИВА:  
Приватное облако**

## Infrastructure team requirements

- Industry standard
- High availability setup with fault tolerance
- Horizontal growth with no architecture change
- Vendor support
- Effective hardware utilization
- Guaranteed quality of service (SLA/SLO)
- Billing engine

## Product teams requirements

- Self service web portal for managing resources manually
- API for CI/CD tools integration
- Role-Based Access Control (RBAC)
- Ability to develop third party tools

# OpenStack

- **Web management** portal
- **API/CLI/Terraform** support
- **Scaling** (vertical and horizontal) based on business requirements
- **Multi-tenancy**
- **Resources limit** (quota) management for a product/project
- We can manage SLA/SLO, noisy neighbors and custom quality of service for environments
- Consolidated and fully integrated hardware resources with **balanced workloads** for compute, network, and storage
- **Elimination of single points of failure** in every layer by delivering continuous access to virtual machines
- Collect measurements of used resources to allow **billing**
- **Open source** license
- **Flexibility** – hardware and software components can be modified or customized to meet various unique customer requirements



# Как мы делали OpenStack

## Как создавалась архитектура

- На Серв наложено вето
- Все должно быть **на нашем железе**
- Все должно быть **HA** и **self healing**
- Отказ от гиперконвергентности / разделение control plane и data plane
- **Деплой обновлений – это рутина**, а не катастрофа

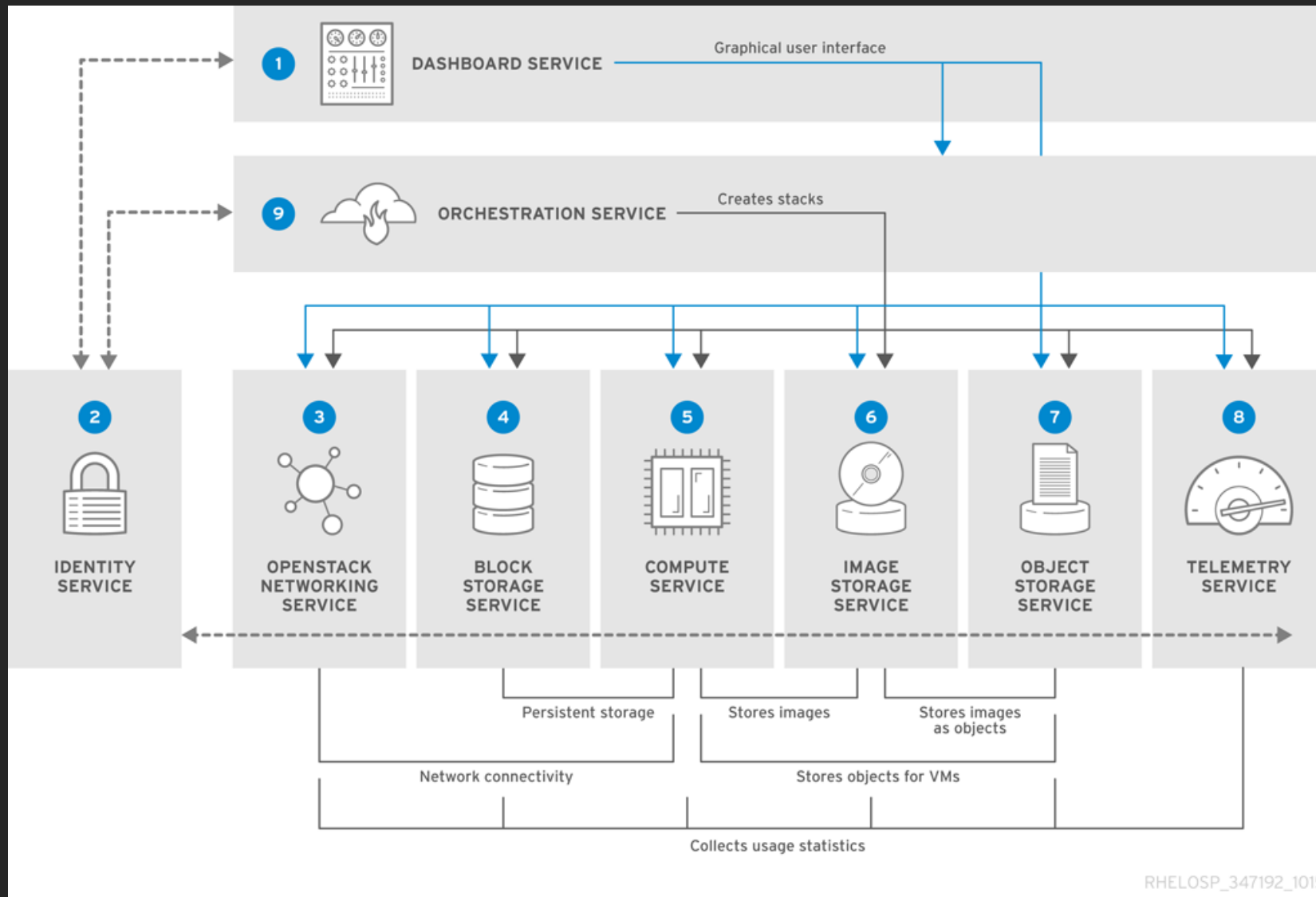
## Технические требования

- Почти все пользователи живут в VLAN (простые сети)
- Kubernetes живет в VLAN+VxLAN
- Гарантированные CPU/IOPS/throughput
- Низкое latency на дисках
- Тестируем и измеряем все сервисы

## Подходы

- Продуктовый подход к платформе частного облака
- Берем всё что можно в нашу зону ответственности (управляем всем железом и TOR коммутаторами)
- Наше взаимодействие с сетевиками происходит на ядре, где они конфигурируют VLAN и L3

# Архитектура OpenStack





# Continuous testing

| Tests duration<br>sec   | success                             | skipped                             | expected<br>failures                | unexpected<br>success               | failures                            |
|-------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2959.315                | 1210                                | 559                                 | 0                                   | 0                                   | 0                                   |
| Filter tests by status: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Near 1 hour





# Как мы делали OpenStack

## DEV кластер v1

- Пока готовятся спецификации и идет закупка, мы занимаемся разработкой
- У нас есть куча списанного магазинного серверного железа + полка Huawei Dorado 6000v3 + 1 сервер Huawei
- Проектирование архитектуры
- Linux ставим руками, остальное через pipeline
- Мы запускаем краш тесты всех видов, результат: проверенная боевыми краш тестами архитектура
- Изначально всё живет в Git (у нас в компании innersource подход)
- Изначально подход infrastructure as a code



# Как мы делали OpenStack

## Кластер – стадия MVP

- Мы сделали краш тесты всех видов, результат: проверенная боевыми тестами архитектура
- Изначально всё живет в Git (у нас в компании innersource подход)
- Изначально подход infrastructure as a code, даже подготовка железа происходит по PXE через Ironic
- Новое железо проходит нагрузочные тесты для выявления производственного брака
- DEV стенд сделан из такого же железа
- К нам пришла команда DevOps с кластерами Kubernetes и мы прошли их тесты
- Мы случайно убили ноду Control Plane. Совсем. Кластер продолжает работать. API отвечает.
- Починили, буднично прогнав по ней пайплайн
- Мы решили обновить Kubernetes. Поминорно было долго. Снесли Kubernetes и поставили его заново. API downtime был меньше часа. Виртуальные машины в облаке в это время работали.



# Как мы делали OpenStack

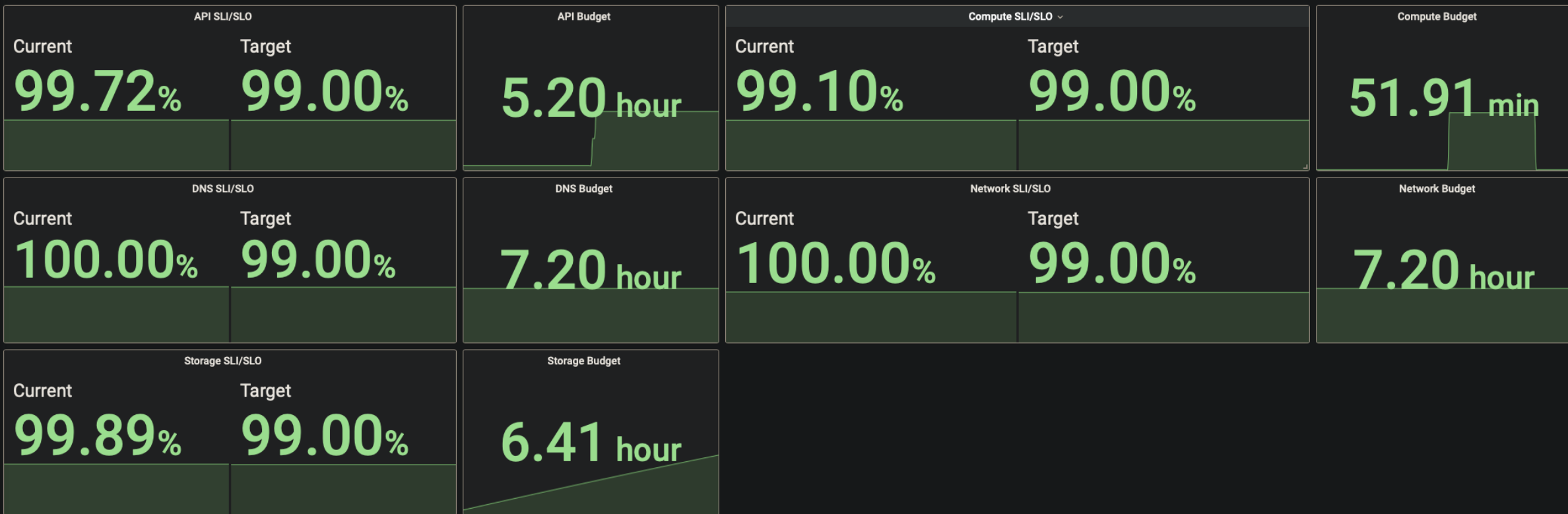
## PROD кластер

- Мониторинг (раз всё в Kubernetes, то будет Prometheus + Prometheus operator)
- Железо мониторим node\_exporter + E-service
- На 1М метриках **перешли на VictoriaMetrics**
- Убрали ложные алерты
- Журналирование (для PROD – эластики в Яндекс Облаке, для DEV – эластики в PROD OpenStack)
- Пришлось дописать вывод логов для ряда компонентов
- Стало сильно легче дебажить сложные истории
- DNSaaS – сделали, работает, команды используют



# SLI/SLO

General SLI

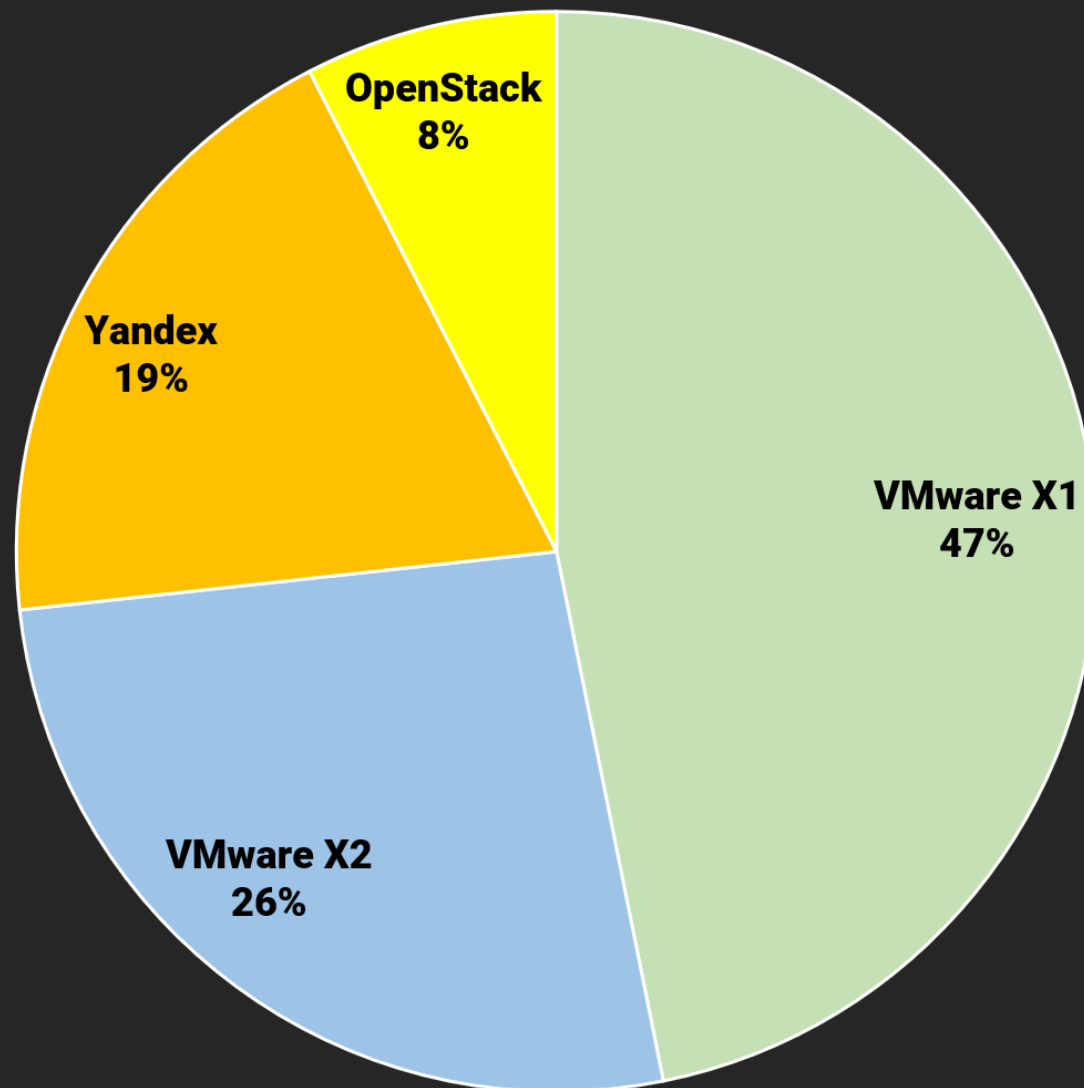


ПРОДУКТОВАЯ ИСТОРИЯ

# 4 ГИБРИДНОЕ ОБЛАКО И ЧТО ДЕЛАТЬ ДАЛЬШЕ



# Распределение мощностей – Q4 2020





# Roadmap 2021

## Приватное облако

- Кластер OpenStack в дата центре X1
- DBaaS совместно с командой DBA

## Публичное облако

- Выровнять функциональные возможности второго облака (Interconnect)
- Управление качеством сервисов
- Сбор фидбэка и реализация потребностей от команд
- Работа с поставщиками облачных услуг

## Для всех облаков

- FinOps методология + FinOps портал



# Ключевые факторы успеха

- **Доверие** со стороны СТО
- **Зрелость команд** в компании (и прежде всего DevOps/Kubernetes)
- Мы не пытались сделать VMware (это невозможно)
- Kubernetes одинаково работает в мультиклауде (VMware, Яндекс Облаке и OpenStack)
- Мы подтвердили расчетами **экономическую эффективность** приватного облака
- **В облаке – облачные подходы** к доступности и к HA/DR (никакого lift and shift)
- Мы **поддерживаем** команды в моменты проблем (спецназ экспертов)
- Мы ни с кем не делим зону ответственности за приватное облако





# Вопросы



[tech.leroymerlin.ru](https://tech.leroymerlin.ru)