

Отказоустойчивость: Отказоустойчивость в облаке



Александр
Зубарев



Александр Зубарев

Председатель цикловой комиссии “Информационной
безопасности инфокоммуникационных систем”

АКТ (ф) СПбГУТ

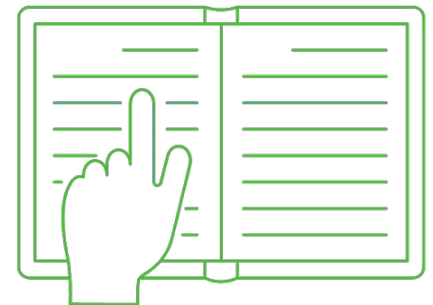
Обо мне

- Эксперт в инфраструктуре, сетях и контейнерах СКА, CCNP Enterprise, GCP Architect, RHCE;
- Построил облако в Казахстане (Транстелеком) и развиваю облачную платформу в Yandex.Cloud;
- Участвовал в проектировании и внедрении 100+ проектов для Enterprise и Service Provider, живущих до сих пор в продакшене.

Предисловие

На этом занятии мы:


- обсудим, зачем строить отказоустойчивые системы;
- рассмотрим основные сценарии отказа инфраструктуры;
- обсудим, какие механизмы для повышения доступности приложения доступны в Yandex.Cloud.





План занятия

1. [Что такое отказоустойчивость и зачем она нужна?](#)
2. [Из чего складывается отказоустойчивость?](#)
3. [Почему сервис может быть недоступен?](#)
4. [Как снизить риски сбоя сервиса?](#)
5. [Как сделать отказоустойчивый сервис в Яндекс.Облаке?](#)
6. [Дополнительные материалы](#)
7. [Итоги](#)
8. [Домашнее задание](#)



**Что такое
отказоустойчивость и зачем
она нужна?**



Немного терминологии

У любого сервиса есть SLA.

SLA — это набор метрик и их допустимых значений между пользователем сервиса и провайдером сервиса.

Одной из метрик* SLA для любого сервиса является его доступность.

Отказоустойчивость – способ увеличения доступности сервиса.

*Помимо доступности, в SLA сервиса могут быть и другие метрики

Когда нужна отказоустойчивость?

Когда недоступность сервиса ведет к финансовым потерям в связи с:

- упущенной выручкой и прибылью;
- потерей пользователей/клиентов;
- негативной репутацией (пользователи ждут 100% Uptime);
- нарушением требований регуляторов;
- нарушением критических бизнес-процессов компании.


Но отказоустойчивость — это дополнительные затраты на сервис, поэтому очень важно применять ее тогда, когда это целесообразно.

Когда не обязательна отказоустойчивость?

Бывают ситуации , где отказоустойчивость может быть избыточной:

- среды разработки и тестирования;
- задачи у которых нет SLA по доступности (например, батч задачи).

Но важно во всех сервисах, где доступность приложения является частью SLA, договориться о метриках этой доступности, даже если от него не требуется высокая доступность.



**Из чего складывается
отказоустойчивость на
примерах?**

Из чего состоит отказоустойчивость?

- избыточность (redundancy);
- мониторинг узлов;
- реакция на сбой (failover);
- возвращение узла в кластер (failback).

Для того чтобы сделать сервис отказоустойчивым, необходимо понимать, из каких компонентов он состоит, чтобы сделать эти компоненты отказоустойчивыми.

Примеры плохой архитектуры

- Всё приложение крутится на одной VM.
- Данные реплицированы, но есть точка отказа.

LAMP Stack



Application
servers

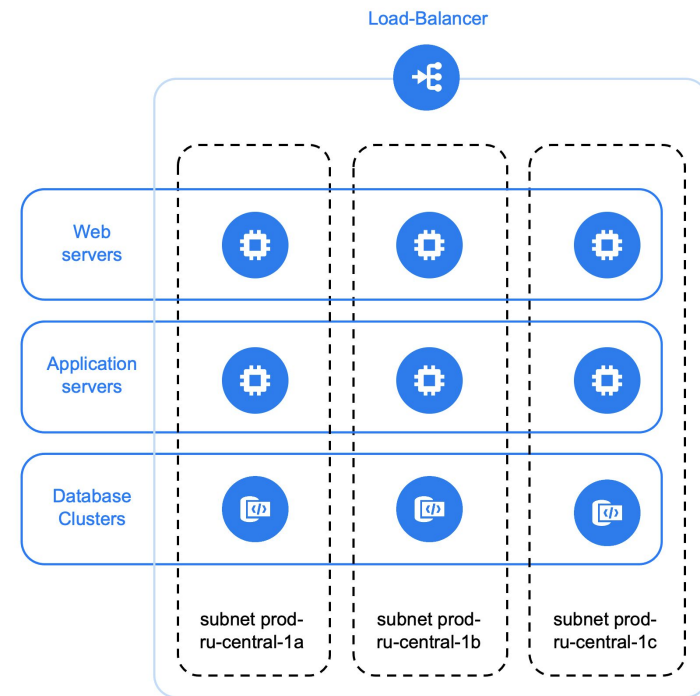


Database
Clusters



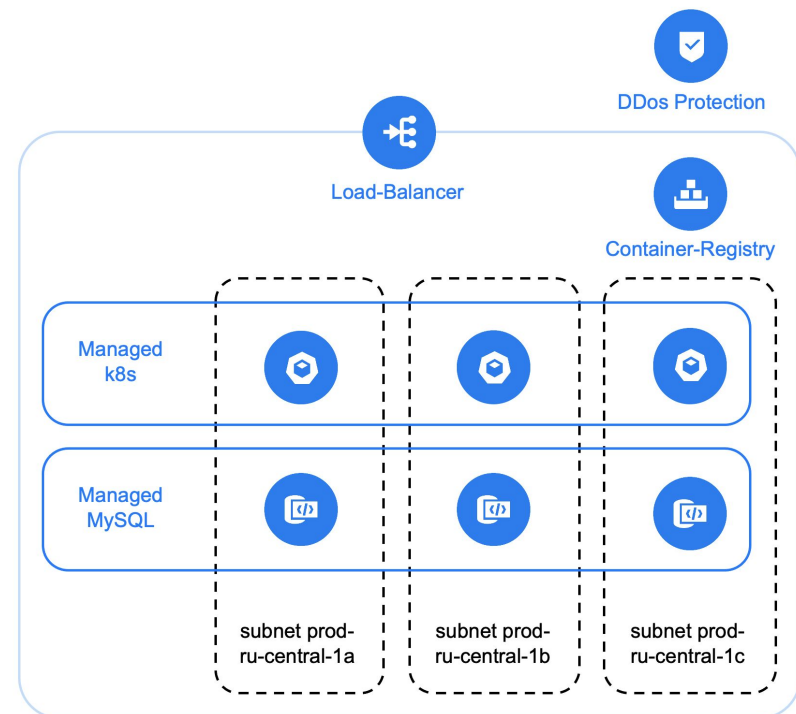
Пример хорошей архитектуры

- Веб-серверы находятся за внешним балансировщиком нагрузки;
- Веб-серверы балансируют трафик на сервера приложений;
- Сервера приложений ходят в мастера и реплики БД.



Пример архитектуры на базе k8s

- Балансировщик защищен услугой DDoS Protection;
- Ноды кластера находятся за балансировщиком нагрузки;
- Ingress Controller принимает входящий трафик от балансировщика и направляет на сервисы;
- Сервисы ходят в мастера и реплики БД.

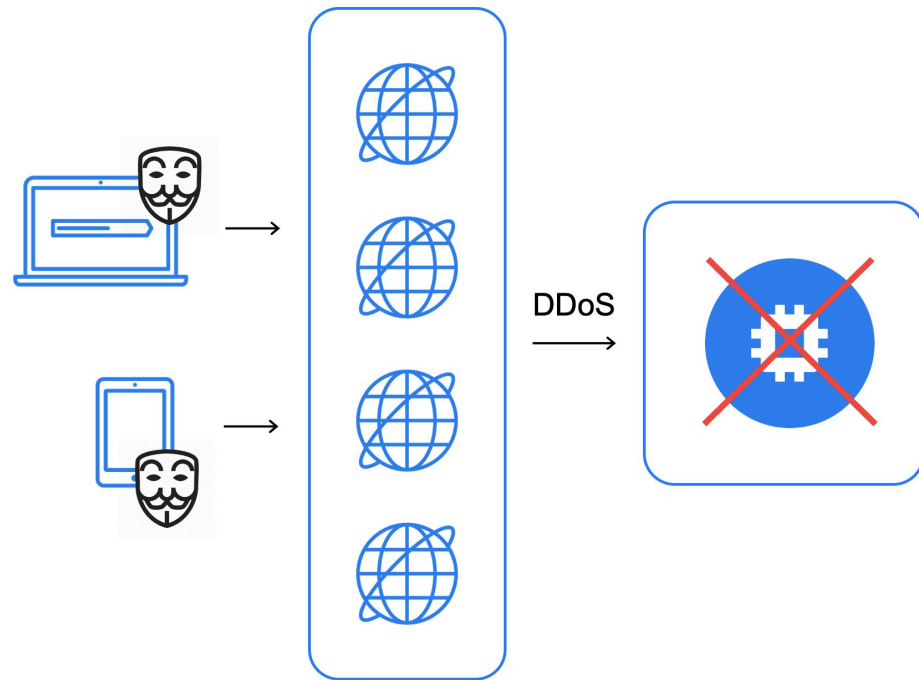




**Почему сервис может быть
недоступен?**

Атака

- Dos
- DDOS



Проблемы из-за инфраструктуры

Сбой на стороне инфраструктуры:

- Отказ физического сервера / стойки;
- Отказ зоны доступности / ДЦ;
- Сетевые проблемы;
- Проблемы с дисковой подсистемой.

Превышение квот и лимитов работы инфраструктуры:

- Понимание разницы между квотами и лимитами;
- Примеры лимитов в УС:
 - Сеть (лимит по flow);
 - Диск (лимит на производительность).

Проблемы из-за настроек сервиса

Сбой на стороне приложения:

- Утечки памяти, утечки на ядре ОС;
- Конец свободного места в файловой системе;
- Баг в новом релизе софта;
- Баг в сторонней библиотеке.

Перегрузка:

- Резкий всплеск активности — Хабразэффект, Черная пятница;
- Постоянный рост нагрузки;
- Следствие сбоя на стороне инфраструктуры.

Как снизить риски сбоя сервиса?

Атака

Dos:

- Анализируйте приложение на уязвимости. Примеры сканеров: Burp Suite, acunetix, nessus;
- Можно заказать pen test от Лаборатории Касперского, GroupIB, BiZone;
- Web application Firewall: Imperva, F5, Nginx plus, Wallarm, Cloudflare.

DDos:

- Яндекс.Облако, Qrator, Cloudflare, Akamai;
- Автомасштабирование: Instance Groups, Managed k8s.

Сбой на стороне инфраструктуры

Сбой сервера:

- Балансировка нагрузки с использованием healthchecks;
- Anti-affinity правила — гарантия того, что копии сервиса запускаются.

Сбой дата центра:

- Балансировка нагрузки на несколько дата-центров;
- Disaster recovery.

Сбой из-за лимитов

Сеть:

- Читайте [документацию](#);
- Используйте средства для уменьшения паразитной нагрузки;
- Горизонтально масштабируйте нагрузку.

Диски:

- Читайте [документацию](#);
- Увеличивайте размер диска и число дисков;
- Горизонтально масштабируйте нагрузку.

Сбой на стороне приложения

ОС:

- Мониторинг ОС (потребление RAM, CPU, свободного места);
- Обновление ОС и ядра;
- Масштабирование места на диске.

Баги:

- Dev/stage среды;
- Юнит тесты, интеграционные тесты;
- Возможность сделать rollback;
- Современные методики деплоя;
- Учения;
- Feature-флаги.

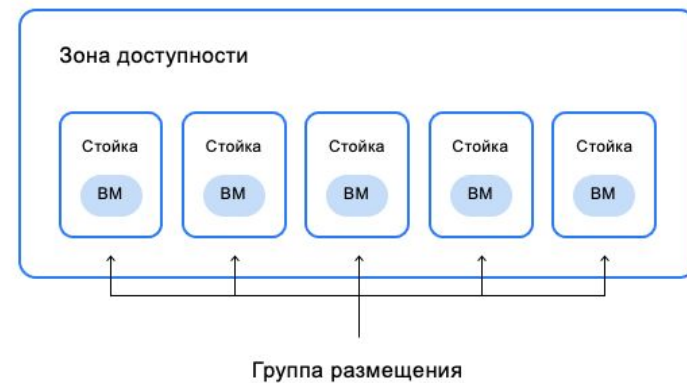
Перегрузка

- **Сайзинг** — приложение должно полноценно уметь обрабатывать нагрузку:
 - При падении нескольких узлов;
 - При падении дата-центра;
- Готовьтесь к возможной неравномерной балансировке;
- Делайте мониторинг нагрузки;
- Делайте нагрузочное тестирование перед вводом в production;
- Приложение должно уметь горизонтально масштабировать входящую нагрузку: автоматически или вручную;
- Аккуратно комбинируйте резервирование и автомасштабирование — алгоритмы автоскейлинга могут не успеть смасштабировать нагрузку при высоких всплесках нагрузки.

Как сделать отказоустойчивый сервис в Яндекс.Облаке?

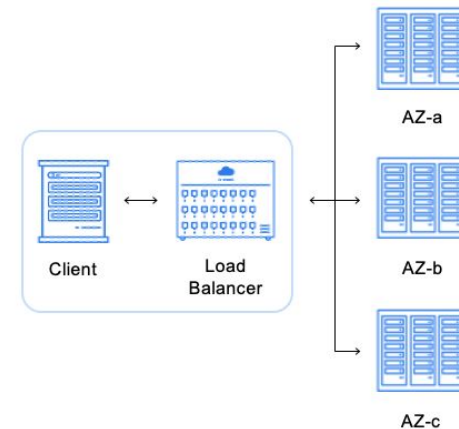
Yandex Compute Cloud

- ВМ и диск — сущность зоны доступности;
- Группа размещения (placement groups) позволяет гарантировать, что ВМ будут находиться в разных стойках.



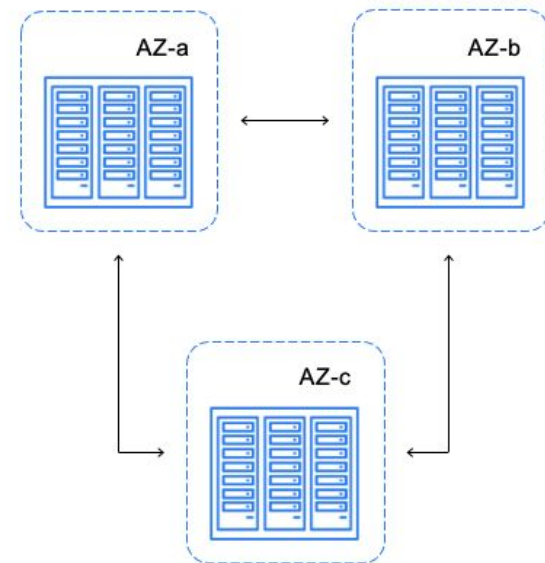
Yandex Load Balancer

- Стабильный статический IP адрес;
- Можно подключить Anti-DDoS;
- Cross AZ балансировка нагрузки;
- Трафик на зоны доступности приходит с помощью ECMP;
- Трафик внутри зоны доступности использует consistent hashing.



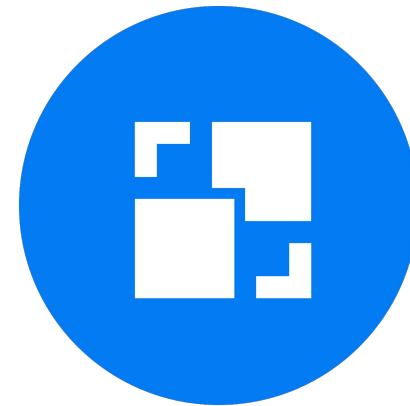
Virtual Private Cloud

- Зона доступности — независимый дата-центр;
- VPC обеспечивает полную IP связность между зонами доступности;
- Latency между зонами;
- Сервис позволяет защитить виртуальные машины с помощью Anti-DDoS.



Instance Groups

- Управляемый сервис для работы с группой виртуальных машин;
- Горизонтальное масштабирование на несколько AZ;
- Автоматическое масштабирование;
- Rolling Update;
- Интеграция с Load Balancer.



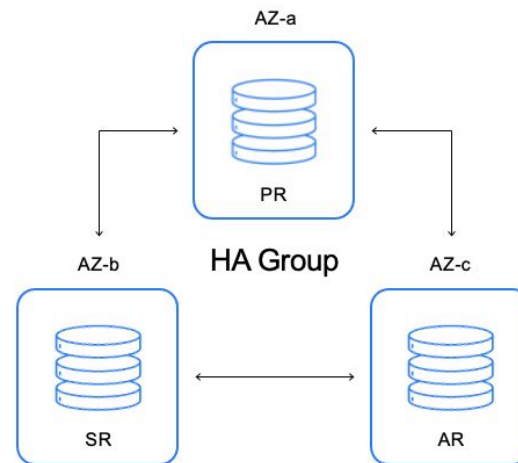
Yandex Managed Kubernetes

- Managed Kubernetes:
 - Отказоустойчивые мастера;
 - Много нативной функциональности для доступности и масштабирования контейнеров;
 - Интеграция с балансировщиком нагрузки;
- Авто масштабирование узлов;
- Интеграция с Container Registry, Load Balancer.

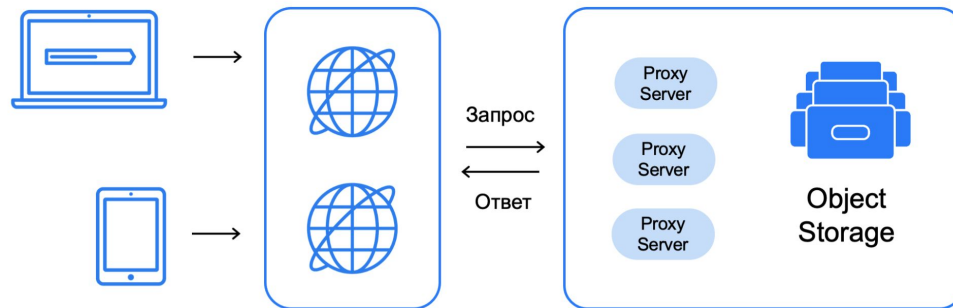


Managed Databases


- Виды конфигураций:
 - Кластер (минимум 2 узла) — разные AZ;
 - Возможность масштабирования вверх.



Yandex Object Storage



- Бесконечно масштабируемый по нагрузке Object Storage;
- Данные реплицированы на 3 ЦОД;
- Есть поддержка SSL и кастомного домена;
- Есть интеграция с CDN.



Дополнительные материалы

Посмотреть

Public Cloud — Гайд по масштабированию.

<https://youtu.be/1fmFjOj4H-4>



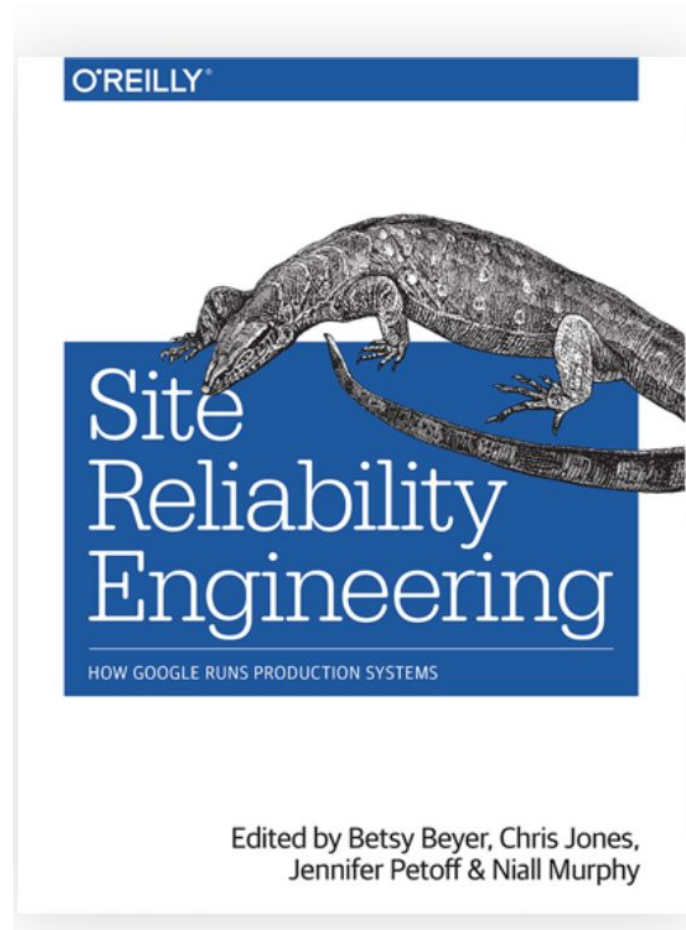
Настройка отказоустойчивой архитектуры в Яндекс.Облаке - Глеб Мищенко.

<https://www.youtube.com/watch?v=40FZ27fUaKo>

Почитать

Google SRE Books

<https://sre.google/books/>





Итоги

Итоги

Сегодня мы:

- поняли, что такое отказоустойчивость и зачем она нужна;
- обсудили сценарии, от которых надо защищать приложение;
- прошли по базовым сервисам Yandex.Cloud, которые позволяют увеличить доступность вашего сервиса





Домашнее задание

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера .
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Александр Зубарев

