

```
class Server::Cloud < Server::Hardware
```

Управляй серверами как объектами со свойствами и методами.

Хештег в твиттере:  
#scalaxy

Нат Гаджибалаев  
— ruby-разработчик  
— евангелист компании Оверсан

nat@oversun.ru  
@xnutsive

# Что такое «облачный хостинг»?

По определению nist.gov, ключевые характеристики:

1. On-demand self service (хотите получить больше — получаете больше)
2. Ubiquitous network access (доступно всегда и отовсюду)
3. Metered use (оплата «по факту» с коротким промежутком тарификации)
4. Elasticity (гибкие модели масштабирования и аренды)
5. Resource pooling (устойчивость к аппаратным сбоям)

## Примеры

?

IaaS

Amazon Web Services

Rackspace Cloud

Terremark

PaaS

Google app engine

Heroku



Code  
Fest

# Что такое IaaS, PaaS и SaaS

- Пользователь арендует конкретное приложение
- У пользователя нет доступа к платформе и ОС
- Администратор полностью контролирует железо, операционную систему, платформу и приложение.

SaaS

- Пользователь арендует платформу, заточенную под конкретную технологию
- Пользователь не контролирует ОС или платформу, а только устанавливает свое приложение в нее

PaaS

- Пользователь арендует инфраструктуру виртуальных серверов
- Пользователь имеет доступ к ОС
- Пользователь занимается управлением инфраструктурой и ОС самостоятельно

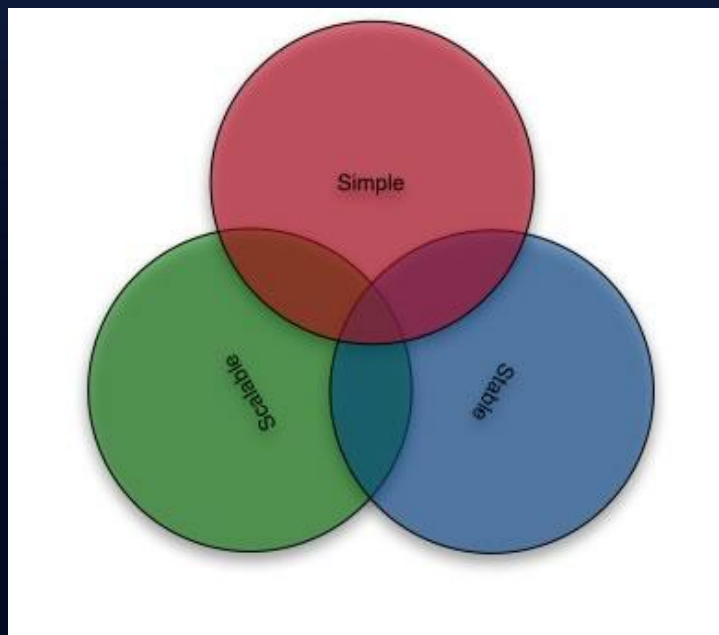
IaaS



Code  
Fest

# Что такое

**масштабирование?**  
Веб-приложение масштабируется под нагрузкой, либо для выполнения прикладных задач. Для некоторых видов масштабирования достаточно просто использовать подходящую облачную платформу. Для других — одного облака будет не достаточно и придется переписать часть приложения.



# Вертикальное

Веб-сервер  
1Гб ram

# масштабирование



Веб-сервер  
8Гб ram

При увеличении нагрузки:

- Кол-во ресурсов одного сервера увеличивается
- Приложения подхватывают и используют новые ресурсы

Плюсы:

- Минимум доработок в существующем коде

Минусы:

- Не все приложения корректно работают при увеличении объема памяти :(
- Скорее всего, будет существовать потолок масштабирования



Code  
Fest

# Горизонтальное

При увеличении нагрузки:  
масштабирование  
— запускает еще несколько виртуальных машин,  
нагрузка распределяется между ними

Плюсы:

- Не плюс, но характеристика: большой шаг масштабирования. Не N ресурсов, а целый сервер
- Нет проблем несовместимости ПО с этой моделью масштабирования

Минусы:

- Требуется использование балансировщика нагрузки
- Слишком большой для рынка РФ шаг масштабирования, негибко

App-сервер  
1Гб ram

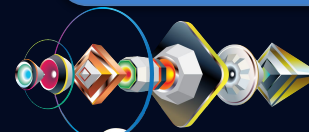


Веб-сервер  
1Гб ram

Веб-сервер  
1Гб ram

Веб-сервер  
1Гб ram

Веб-сервер  
1Гб ram

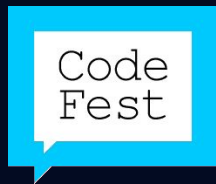


Оверсан

Code  
Fest

# Cloud API

- Для высоконагруженных (очень) проектов, стандартных средств автомасштабирования недостаточно
- Если нужно реагировать на нагрузки очень быстро — нужно делать полный мониторинг самостоятельно
- Если облако не предоставляет средств автомасштабирования, масштабировать инфраструктуру нужно самостоятельно
- Как правило, облачные провайдеры предоставляют программный интерфейс (API) для всех этих операций



# Cloud API:

## примеры

- Amazon web services
- Rackspace cloud
- ...





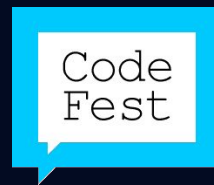
# Оверсан —

# Скапакси

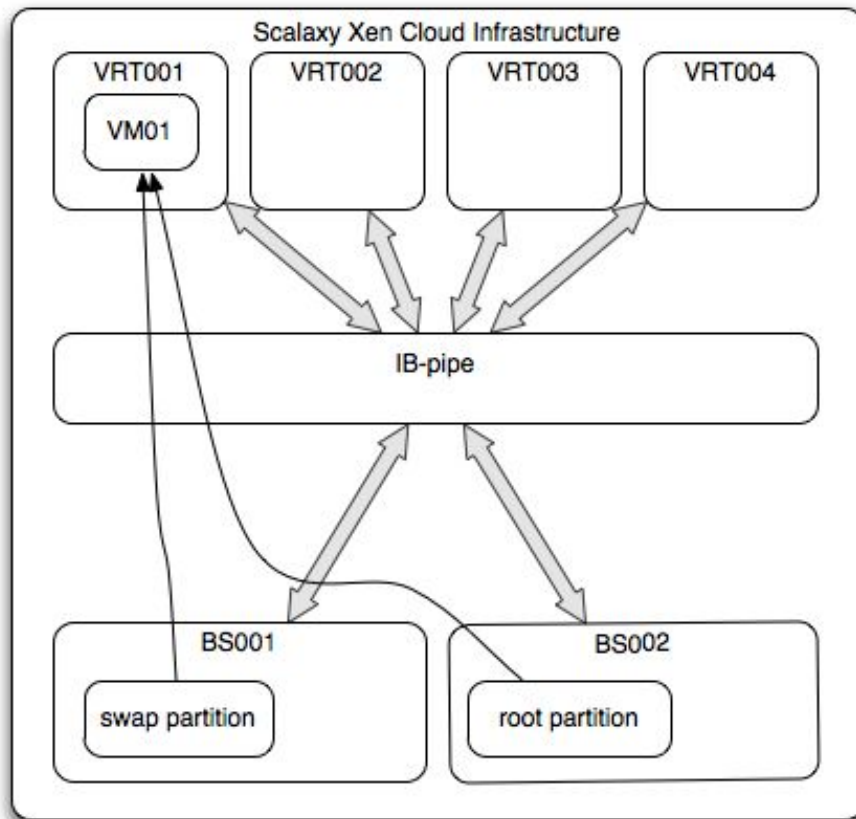
Облачный хостинг

Инфраструктура

- Платформа & сообщество



# Оверсан — Скалакси: инфраструктура



- VRT — физические машины, хосты виртуализации
- BS — физические машины, массивы дисков
- IB-pipe — Infiniband шина внутренней сети на 40 Гбит/с
- VM — пользовательская виртуальная машина

# Виртуальные

## машины

- Единица измерения ресурсов — слот
  - 512 Мб оперативной памяти
  - 5 Мбит/с полоса в Интернет
  - Доступны все 8 ядер процессора
- Размер машины — от 1 до 64 слотов, то есть от 512 Мб до 32 Гб оперативной памяти
- Вертикальное масштабирование без перебоев в работе
- Создание и старт машины — менее чем за 60 секунд
- Изменение размера виртуальной машины — практически мгновенно
- Почасовая оплата выделенных машине ресурсов
- Два вида слотов: development vs production: 0.65 vs 2.0 рублей в час



Code  
Fest

# Виртуальные машины: масштабирован

ие

Изменить мощность виртуальной машины можно:

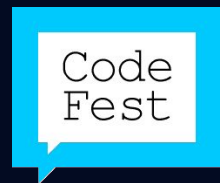
- Вручную на любое количество слотов
  - По расписанию
  - Автоматически (по нагрузке)
- Управляемый минимальный и максимальный размер виртуальной машины



# Виртуальные машины: автомасштабирован

**И** Автоматическое управление количеством выделенных ресурсов, основываясь на текущей загрузке в реальном времени

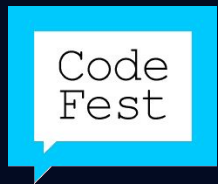
- Постоянный мониторинг параметров виртуальных машин
- Триггер по объему занятой оперативной памяти, загруженности сети



# Блочное

## хранилище

- Неограниченное число блочных устройств
- Объем диска от 1 Гб до 1 Тб
- Гарантированная скорость — 300 iops для рандомной записи
- 8 рублей в месяц за гигабайт, тарификация ежечасно (около 11 копеек)

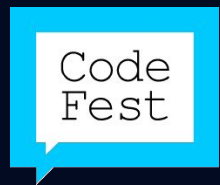


# API

## Скаляр

Что можно сделать с его помощью?

- Как оно реализовано?
- Зачем его использовать?

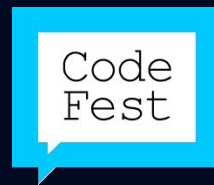


# API

## Скаляр

Что можно сделать с его помощью?

- Просмотреть список машин и их состояние
  - Создать виртуальную машину
  - Запустить или остановить машину
  - Масштабировать машину
  - Создать или удалить блочные устройства
  - Создать, удалить или изменить правила масштабирования
  - Все действия, доступные панели управления
  - **Панель = клиент к API**
- 
- Как оно реализовано?
  - Зачем его использовать?



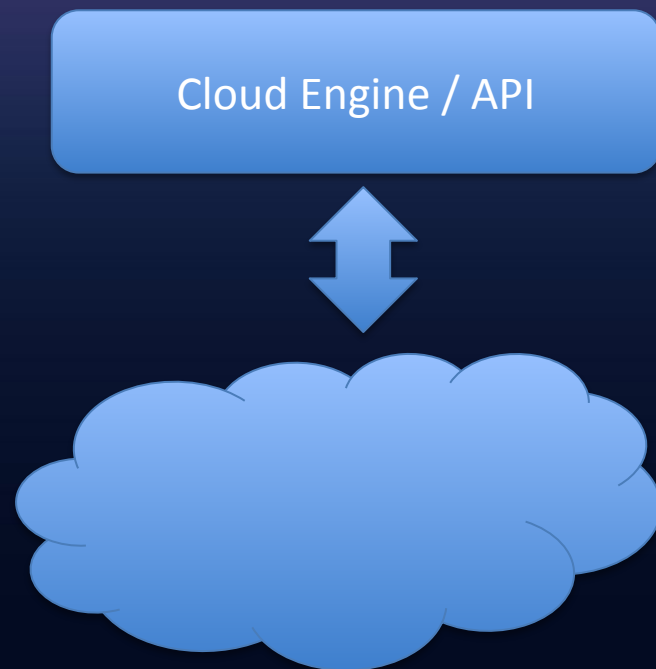


# API Скалакси:

Что можно сделать с его помощью?  
реализация  
• Как оно реализовано?

- HTTPS
- RESTful Web Service
- HTTP-авторизация
- JSON

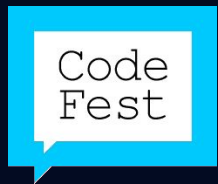
- Зачем его использовать?



# API

## Скалпакси

- Что можно сделать с его помощью?
- Как оно реализовано?
- Зачем его использовать?
  - Высоконагруженные приложения
  - Гибкая балансировка нагрузки
  - Гибридные архитектуры с горизонтальным и вертикальным масштабированием
  - Максимально гибкая экономия ресурсов и денег :)



# API

## Высоконагруженные приложения: Скапакси

- Масштабирование с использованием API и собственной системой мониторинга позволяет сократить время реакции на возрастание нагрузки до нескольких секунд
- Можно включать виртуальные машины, выполняющие отдельные задачи по расписанию, только тогда, когда они фактически нужны
- Программный балансировщик нагрузки может анализировать объем входящего трафика веб-приложения и подключать дополнительные application-серверы по мере необходимости (гибридные архитектуры). Так можно достигнуть максимальной гибкости и экономии



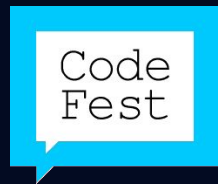
Code  
Fest

# API Скалакси:

## сущности

- Проект (ограниченное использование)
- Инстанс (виртуальная машина)
- Группа инстансов
- Блочное устройство
- Правило масштабирования по расписанию
- Правило автомасштабирования
- Домен
- Запись домена
- Пользователь
- Образы операционных систем

Подробные материалы — на сайте сообщества  
[wiki.scalaxy.ru](http://wiki.scalaxy.ru)



# API Скалакси: работа с REST- интерфейсом

POST-запросы: создание  
объектов

```
curl -u username:password-X POST -H "Content-Type: application/json"  
-d"{name: 'Project name'}" https://www.scalaxy.ru/api/projects.json  
  
{"name":"Project name","id":152}
```

GET-запросы: чтение  
объектов

```
curl -u username:password https://www.scalaxy.ru/api/projects.json  
  
[{"name":"Project name","id":152}]
```



Code  
Fest

# API Скалакси:

## горизонтальный

## скейлинг

Старт  
машин

```
curl -u username:password-X PUT -H "Content-Type: application/json"  
-d"{name: 'Project name'}"  
https://www.scalaxy.ru/api/projects/1/instances/2/start.json
```

Отключение  
машин

```
curl -u username:password-X PUT -H "Content-Type: application/json"  
-d"{name: 'Project name'}"  
https://www.scalaxy.ru/api/projects/1/instances/2/stop.json
```



Code  
Fest

# API Скалакси:

## создание

## инстанса

Создание

сервера:

```
{  
  name: 'name',  
  slots: 4,  
  os_image_id: 10,  
  root_size: 30000000000,  
  password: 'password'  
}
```

```
curl -u username:password-X POST -H "Content-Type:  
application/json" -d"{ name: 'Test VM', slots: 4,  
os_image_id: 10, root_size: 30000000000, password:  
'password' }"  
https://www.scalaxy.ru/api/projects/1/instances.json
```



Code  
Fest

# Roadmap

- Инфраструктура:
  - Скапакси
    - Мультимедийное файловое хранилище и CDN
    - Репликация блочных хранилищ
- Платформа
  - Wiki
  - API integration





# Спасиб

о! <http://scalaxy.ru> — Начать тестирование можно сегодня

Follow us on twitter: @scalaxy, @xnutsive

Весь ваш фидбэк пишите в твиттер по тегу #scalaxy или на почту [nat@scalaxy.ru](mailto:nat@scalaxy.ru) или [experts@scalaxy.ru](mailto:experts@scalaxy.ru)

