

# Автоматизация тестовой инфраструктуры



Антон Галицын, [2GIS](#)

# О себе

- Разработчик в отделе Infrastructure & Operations
- Работал в команде OpenStack
- Занимаюсь автоматизацией деплоя проектов 2ГИСа

У нас есть приложение!

# Приложение

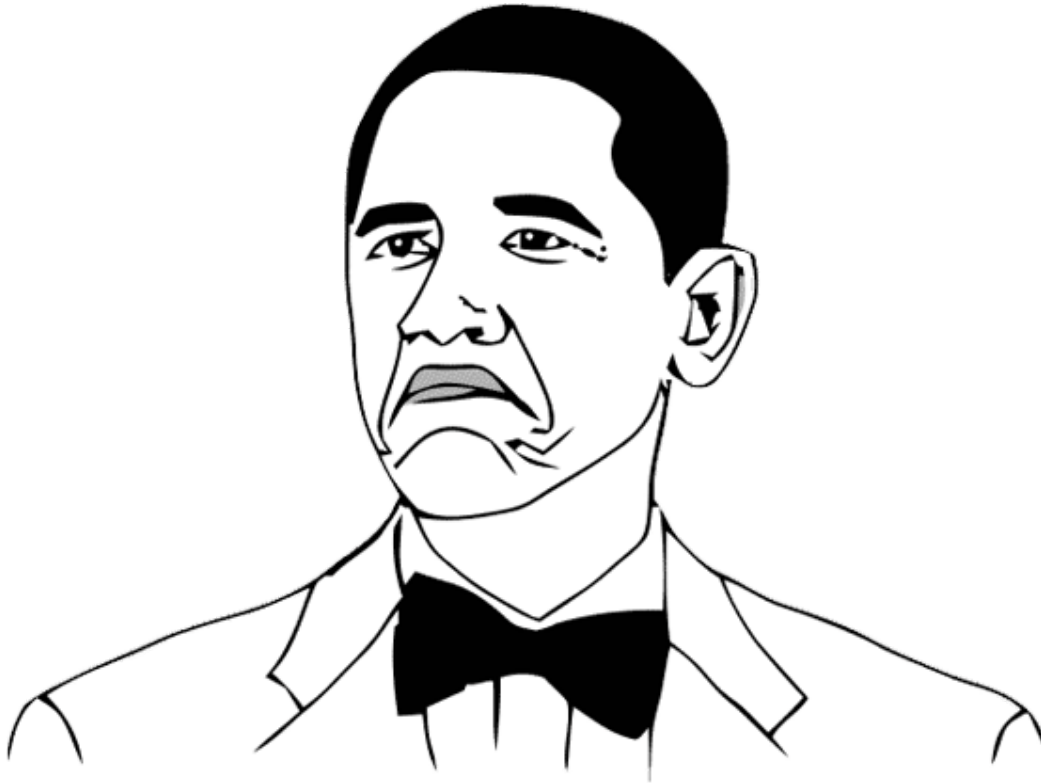
C, gcc 4.9.3, Linux 4.2

```
$ ./codefest
```

```
Codefest is IT conference!
```

Где будем тестировать?

Локально



**NOT BAD**

# Локально

+

- Просто
- А зачем что-то еще?

А что с Windows и OS X?



# Локально

—

- Разные операционные системы

OK, Linux!

**CHALLENGE ACCEPTED**



# УПС...

Some packages could not be installed.

codefest: Depends: libstdc++6 (>= 5.2)  
but 4.9.2-10 is to be installed

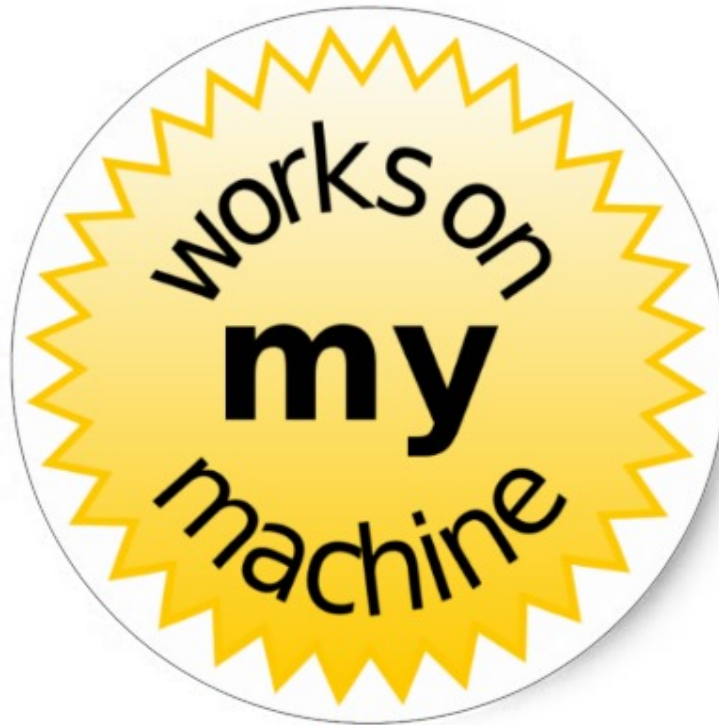
E: Unable to correct problems.

# Локально

—

- Разные операционные системы
- Проблемы с зависимостями

Локально



Локально, с изоляцией



# Локально, с изоляцией

- VirtualBox
- KVM
- LXC
- Docker

# Где будем тестировать?

- Шаг 1 - создать окружение



Любое окружение надо  
конфигурировать

# Конфигурация

- Documentation-driven
- Scripts (bash, python, perl)
- Configuration management (chef, ansible, puppet)

# Где будем тестировать?

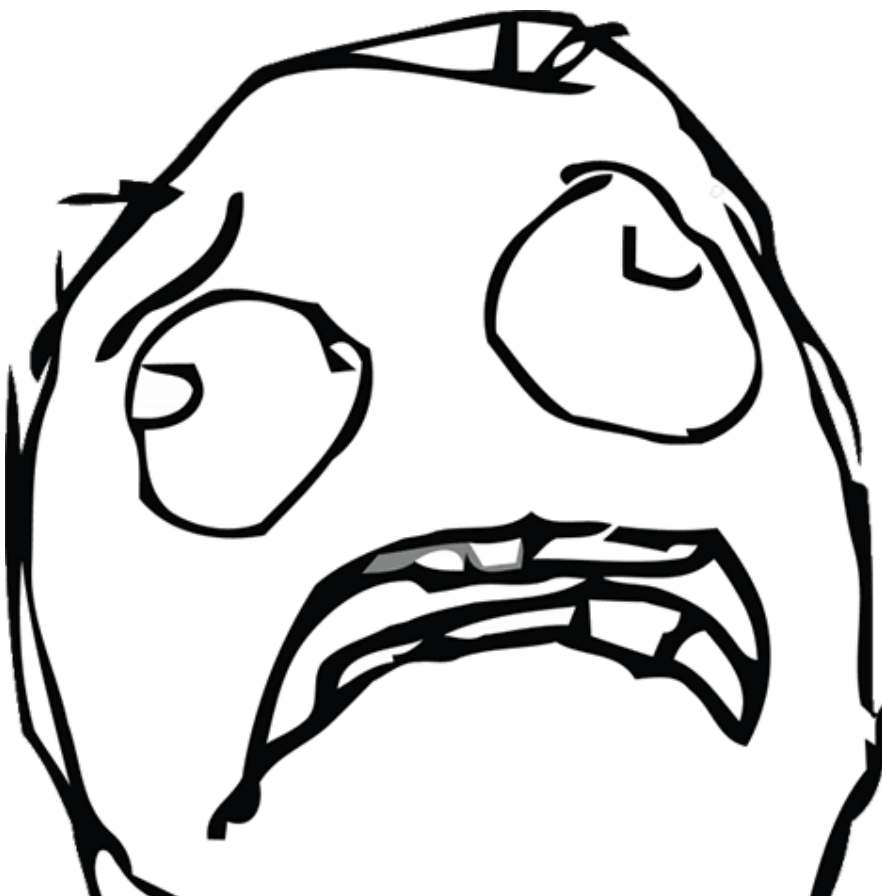
- Шаг 1 - создать окружение
- Шаг 2 - настроить окружение

# Тестируем!

- ✓ Шаг 1 - создать окружение
- ✓ Шаг 2 - настроить окружение

Прогоняем тесты, смотрим результаты

Автотесты показывают  
разные результаты!



# Достоверные результаты автотестов

# Достоверные результаты автотестов

- Результат автотестов повторяем

# Достоверные результаты автотестов

- Результат автотестов повторяем
- Автотесты находят баги



# Проблемы

# Проблемы

- Окружений  $> 1$

# Проблемы

- Окружений  $> 1$
- Неоднородная инфраструктура

# Проблемы

- Окружений > 1
- Неоднородная инфраструктура
- Автотесты оставляют артефакты (tmp файлы, cache, неконсистентная база)

Решение

# Решение

- Список окружений

# Решение

- Список окружений
- Конфигурация в коде

# Решение

- Список окружений
- Конфигурация в коде
- Инфраструктура по требованию



# Решение

- Список окружений
- Конфигурация в коде
- Инфраструктура по требованию
- "Чистые" автотесты

# Решение

- Список окружений
- Конфигурация в коде
- Инфраструктура по требованию
- ~~"Чистые" автотесты~~

# Универсальный инструмент

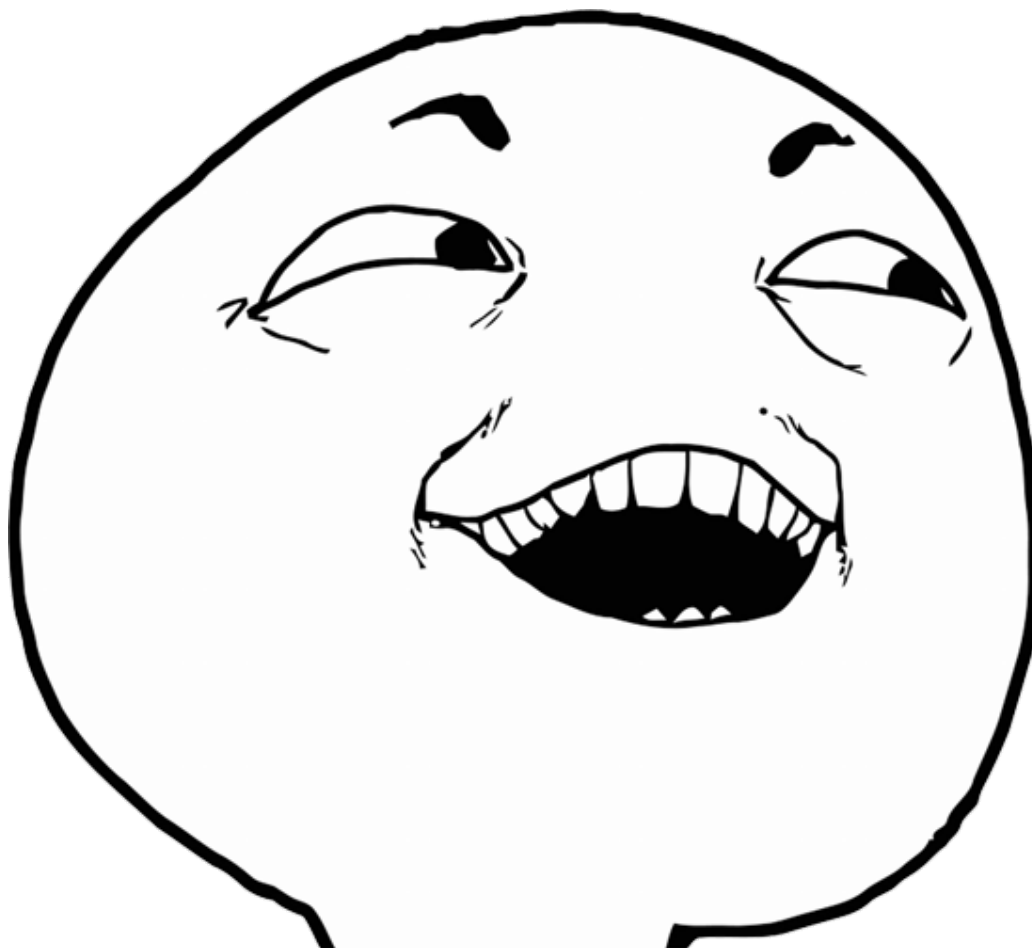
- ✓ Шаг 1 - vagrant provider (Vbox, LXC, Docker, Digital Ocean)
- ✓ Шаг 2 - vagrant provisioner (shell, chef, ansible, puppet)

```
$ vagrant up
```

# Что у нас есть?

- ✓ Список окружений
- ✓ Конфигурация в коде
- ✓ Инфраструктура по требованию

Переписываем приложение!



Web

# Better Web

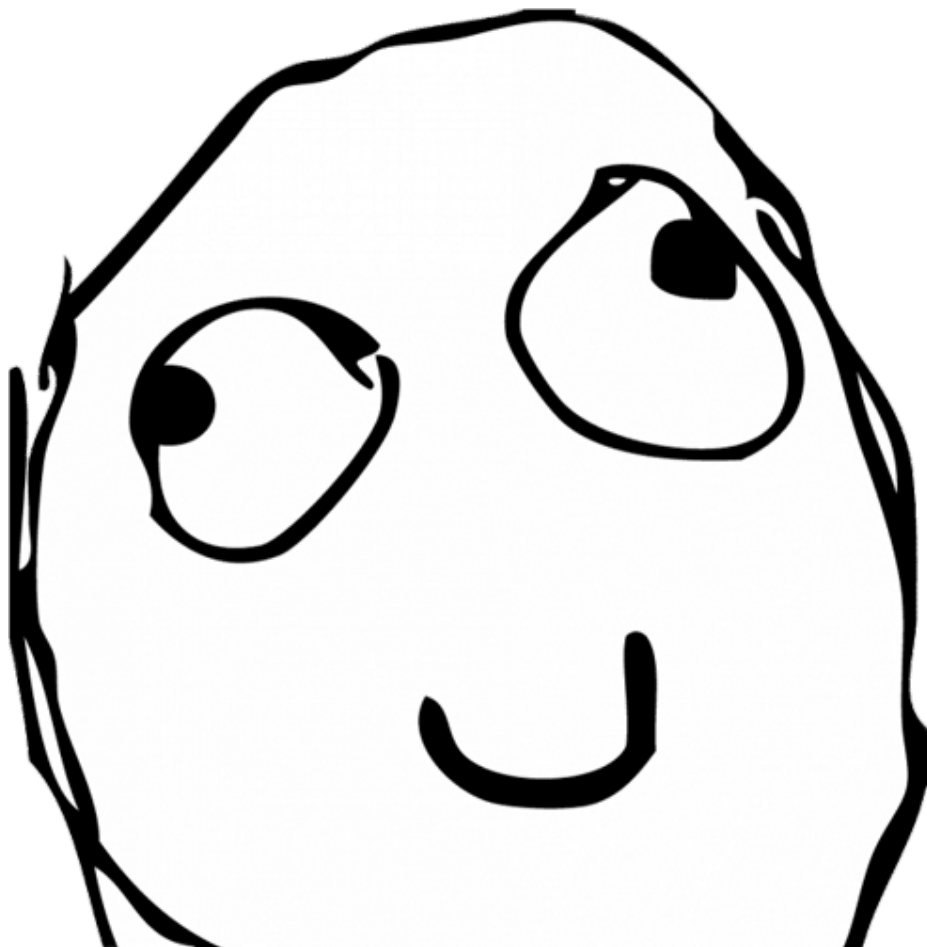
Service-oriented



# Microservices

Где будем тестировать?

Удаленно



# Варианты

# Варианты

- Публичное облако (AWS, Digital Ocean, Rackspace)

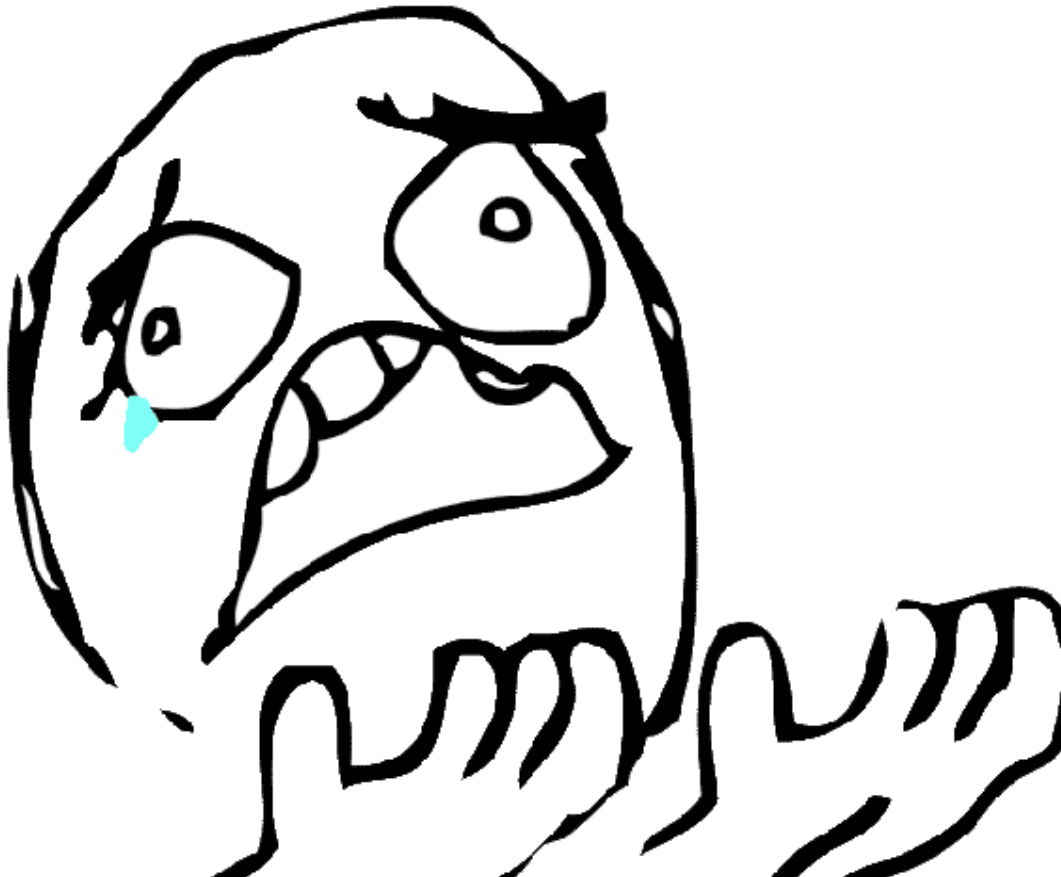
# Варианты

- Публичное облако (AWS, Digital Ocean, Rackspace)
- Приватное облако (VMware, HP)

# Варианты

- Публичное облако (AWS, Digital Ocean, Rackspace)
- Приватное облако (VMware, HP)
- Приватное облако open-source (Apache Cloudstack, OpenStack)

Я просто хотел запустить  
тесты!





Маршруты

Поиск

Проезд

☀️ -1°  
Вечером -7°  
Пора подумать о резине

🔥 Сауны  
🏠 Новостройки  
📄 Все рубрики

WP Store App World



# 2ГИС

- Справочник и карты
- Web, Mobile, Desktop
- 35 команд



# Команды

- Разные языки
- Интеграции



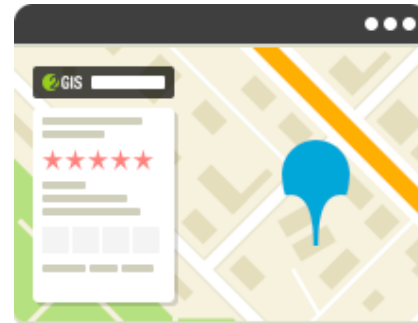
# Потребности команд

- Автоматизированное тестирование
- Достоверные результаты



# Итог

- 35 команд
- Интеграции
- Автоматизированное тестирование



Где будем тестировать?

# Что у нас есть?

- ✓ Свое железо

# Что у нас есть?

- ✓ Свое железо
- ✓ Системные администраторы



# Что у нас есть?

- ✓ Свое железо
- ✓ Системные администраторы
- ✓ Опыт в виртуализации

Инфраструктура компании *не*  
*готова*

# Проблемы инфраструктуры

Конец 2013 года - Proxmox Virtual  
Environment

# Проблемы инфраструктуры

Конец 2013 года - Proxmox Virtual Environment

- Создание виртуалок вручную по тикету

# Проблемы инфраструктуры

Конец 2013 года - Proxmox Virtual Environment

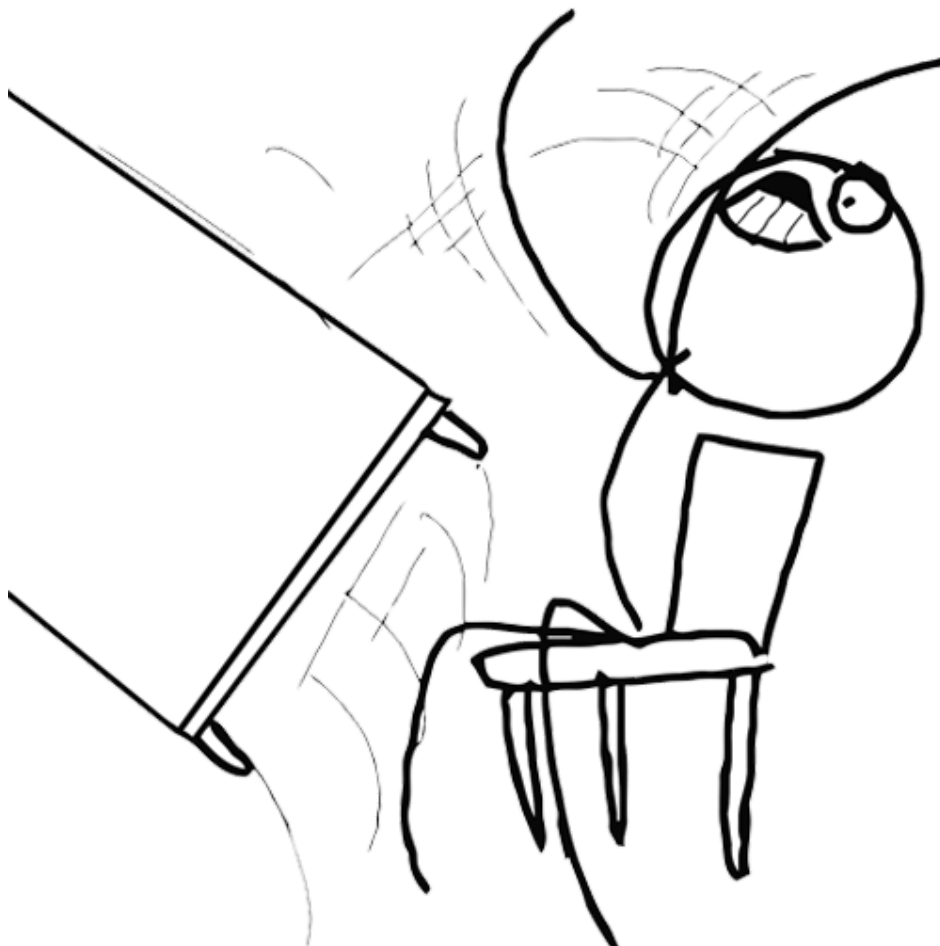
- Создание виртуалок вручную по тикету
- Нет разделения по проектам

# Проблемы инфраструктуры

Конец 2013 года - Proxmox Virtual Environment

- Создание виртуалок вручную по тикету
- Нет разделения по проектам
- Слабый API, платные плагины

Пора что-то менять!



# Требования к решению



# Требования к решению

- Эффективная утилизация железа

# Требования к решению

- Эффективная утилизация железа
- Командные ресурсы

# Требования к решению

- Эффективная утилизация железа
- Командные ресурсы
- Модульность

# Требования к решению

- Эффективная утилизация железа
- Командные ресурсы
- Модульность
- Легко дорабатывать

# Требования к решению

- Эффективная утилизация железа
- Командные ресурсы
- Модульность
- Легко дорабатывать
- API

# Требования к решению

- Эффективная утилизация железа
- Командные ресурсы
- Модульность
- Легко дорабатывать
- API
- Изоляция

# Варианты

- ~~Публичное облако (AWS, Digital Ocean, Rackspace)~~
- ~~Приватное облако (VMware, HP)~~
- ✓ Приватное облако open-source (Apache Cloudstack, OpenStack)





# Что такое OpenStack?

OpenStack - набор сервисов для построения публичного или частного облака.

# OpenStack - open software

- Релиз раз в полгода  
[status.openstack.org](https://status.openstack.org)
- Базовые компоненты  
[github.com/openstack](https://github.com/openstack)
- Инкубатор [github.com/stackforge](https://github.com/stackforge)
- Конференции, встречи, Q&A, mail-lists  
[openstack.org/community](https://openstack.org/community)

# Контрибьюторы OpenStack

[openstack.org/foundation/companies](https://openstack.org/foundation/companies)

- Red Hat
- Rackspace
- IBM
- Intel
- Cisco

# Решение проблем инфраструктуры

# Решение проблем инфраструктуры

- ✓ Эффективная утилизация железа

# Решение проблем инфраструктуры

- ✓ Эффективная утилизация железа
- ✓ Командные ресурсы

# Решение проблем инфраструктуры

- ✓ Эффективная утилизация железа
- ✓ Командные ресурсы
- ✓ Модульность

# Решение проблем инфраструктуры

- ✓ Эффективная утилизация железа
- ✓ Командные ресурсы
- ✓ Модульность
- ✓ Легко дорабатывать



# Решение проблем инфраструктуры

- ✓ Эффективная утилизация железа
- ✓ Командные ресурсы
- ✓ Модульность
- ✓ Легко дорабатывать
- ✓ API

# Решение проблем инфраструктуры

- ✓ Эффективная утилизация железа
- ✓ Командные ресурсы
- ✓ Модульность
- ✓ Легко дорабатывать
- ✓ API
- ✓ Изоляция

# Инфраструктура по требованию

# Инфраструктура по требованию

## Создание стеков из шаблонов

```
server:
  depends_on: [ ip_adress, dns_record ]
  type: OS::Nova::Server
  properties:
    name: { get_param: name }
    flavor: { get_param: flavor }
    image: { get_param: image }
    key_name: { get_param: key_name }
```

# Инфраструктура по требованию

## Создание стеков из шаблонов

```
$ heat stack-create API -f api.yml \  
  -P key_name="id_rsa" \  
  -P name=api-all-in-one \  
  -P flavor=medium \  
  -P image="ubuntu-14.04-x64"
```

Это было начало

# Развитие автоматизации

# Continuous integration

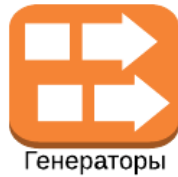




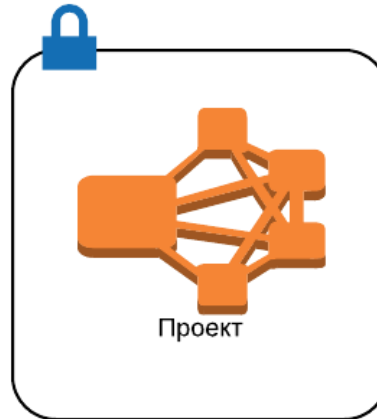
[github.com/2gis/vmmaster](https://github.com/2gis/vmmaster)



# Нагрузочное тестирование



Генераторы



Проект



Отчеты

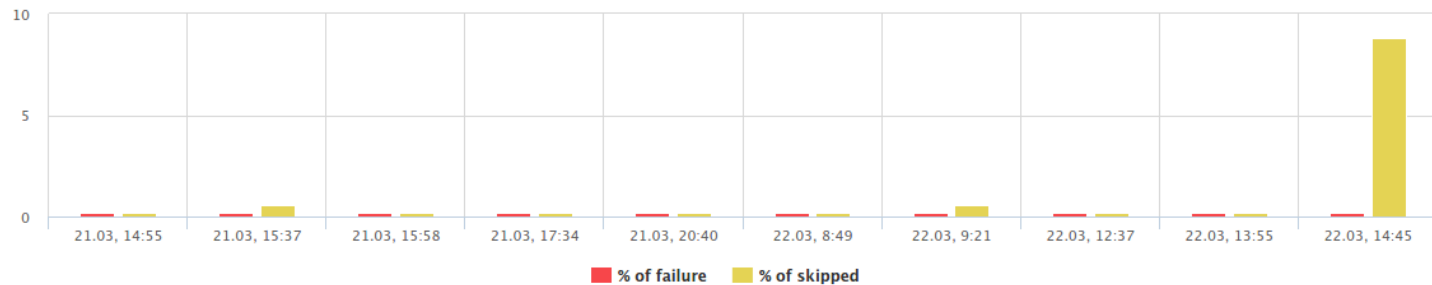


Графики

# github.com/2gis/badger

[WebAPI](#)[Testplans ▾](#)[Short statistics](#)[Metric statistics](#)[Most failed tests](#)[🏠](#) [Login](#)

## "Full regression" statistics



By	Branch	Date	Counts				State	% fail	C
Jenkins	<a href="#">i</a>	22.03 14:45	1486	144	0	0	STARTED	0	
Dima	<a href="#">i</a>	22.03 13:55	40059	0	3	0	FINISHED	0.007	

[github.com/2gis/badger-api](https://github.com/2gis/badger-api)



Load  
Balancer



Badger



UI



API



Queues



Workers



DBs



И мы будем писать еще!

# Выводы

# Выводы

- Используйте однородные окружения

# Выводы

- Используйте однородные окружения
- Храните конфигурации в коде



# Выводы

- Используйте однородные окружения
- Храните конфигурации в коде
- Используйте "чистую" инфраструктуру по требованию

Собирайте достоверные  
результаты!

Спасибо!

Антон Галицын

[a.galitsyn@2gis.ru](mailto:a.galitsyn@2gis.ru)

[github.com/agalitsyn](https://github.com/agalitsyn)

# Нераскрытые темы про OpenStack

- Эксплуатация
- Деплой
- Апгрейд
- Тестирование
- Разработка плагинов
- Расследование инцидентов

