

Шашлыки — это хорошо. А что насчёт виртуализации?

Есть банальная виртуализация — запуск нескольких виртуальных машин на одной физической. Это хорошо. Но мало. Хочется полной абстракции вычислительных ресурсов.

Как это? Очень просто. У вас в руках ноутбук, планшет или телефон. Все ваши задачи работают в большом ЦОДе. Когда им нужно много ресурсов - они их автоматически получают. Когда мало - ресурсы используются под что-то другое.

Вы не задумываетесь, что и как пришлось перенастроить, чтобы выдать вам 4 процессора и 20Гб RAM на два часа. Вы не волнуетесь, не сломалось ли от этого что-нибудь или не пострадал ли кто-то другой. Об этом система думает сама.

Но чтобы так получилось систему нужно строить очень продуманно. В этом челендж. Эта задача еще не решена.

Подробности на слайде «Что значит задача не решена?»

Что значит задача не решена?

Ведь есть же вендора систем виртуализации?

- Есть. Не верьте их маркетингу!
- Динамическое управление в полном объеме официально рекомендуется отключать в промышленных средах.
- *Смотрите примеры их кода на слайдах «Современные системы управления облаком». :-)*

Ведь работают же облака Amazon, RackSpace, Селектел?

- Да, работают. Они объективно лидеры и заслуживают уважения.
- Но есть нюансы:
 - Стоимость облаков сегодня часто оказывается неадекватно высокой по сравнению с традиционными решениями. Мы сами сейчас мигрируем из облака Amazon в собственный клауд. Использовать Amazon оказалось дорого, несмотря на эффект масштаба.
 - Когда люди хотят построить свое облако — сложность задачи пугает даже ведущих интеграторов.
 - *Смотрите примеры того как устроены нынешние облака на слайдах «Анатомия облака».*

Нормальных облаков еще нет. Должны появиться новые решения.

Современные системы управления «облаком» (1/2)

Пример исходного кода одной промышленной системы виртуализации. Управление ресурсами. Периодический мониторинг состояния памяти для управления KSM:

```
=====
```

Try:

```
    memTotal = utils.readTotalFreeCachedMem()[0]
    self._sleep = str(config.getint('ksm', 'ksm_sleep') * 16 * 2**20 \
                                   / memTotal)
```

Except:

```
    self._sleep = '10000'
```

```
=====
```

Магические числа... Что такое 16 и 2**20? Откуда 10000? А ведь этот продукт стоит сотни долларов на процессор... Причем успешно продается.

Вам интересно, откуда тут могут быть exception? Смотрите следующий слайд.

Современные системы управления «облаком» (2/2)

Так откуда берется Exception? Как может упасть сбор данных о памяти?

=====

```
def readTotalFreeCachedMem(readSwap=False):
```

```
    """
```

```
    Parse ``/proc/meminfo`` and return some of it.
```

```
    For a reason unknown to me, ``/proc/meminfo`` is is sometime  
    empty when opened. If that happens, the function retries to open it  
    3 times.
```

```
<comments snipped>
```

```
    """
```

```
    # FIXME the root cause for these retries should be found and fixed
```

```
    tries = 3
```

```
    while True:
```

```
        Tries -= 1
```

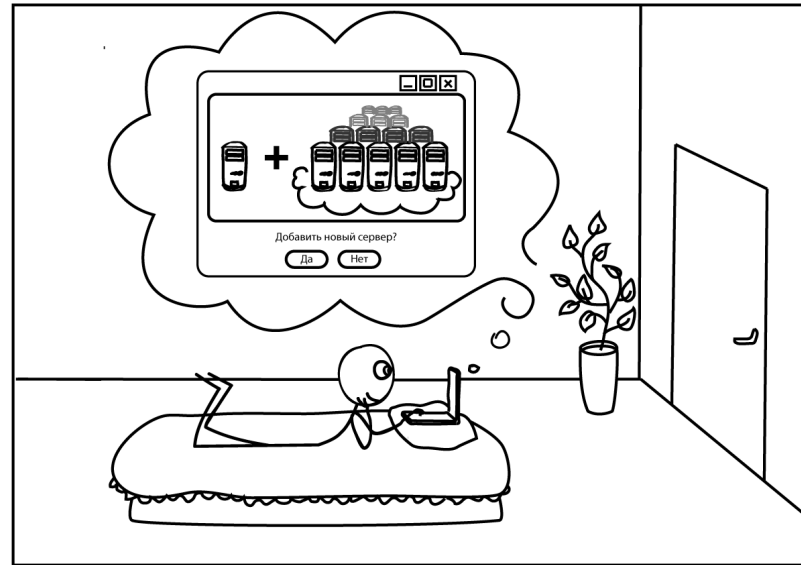
```
<и так далее...>
```

```
=====
```

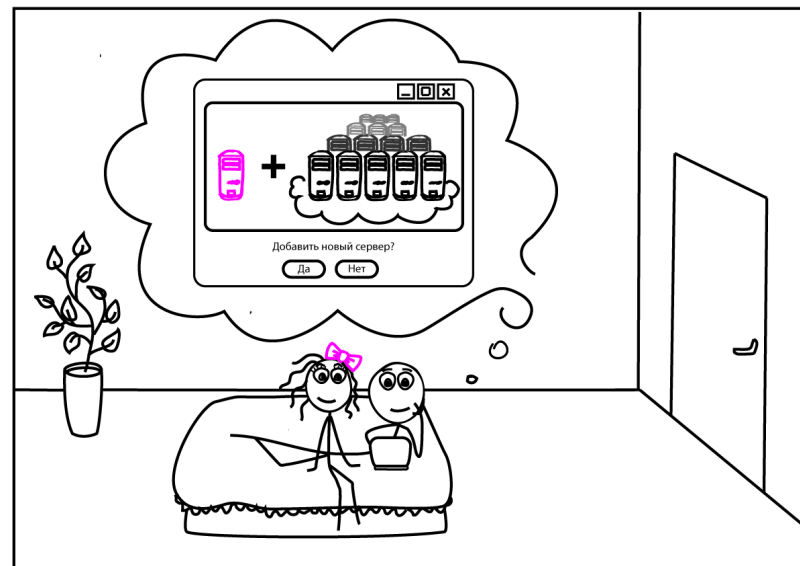
Очень мило. «Unknown reason» и «FIXME». И это продакшн-код.

Анатомия современного «облака» (1/3)

**Сначала облако
преподносят так:**

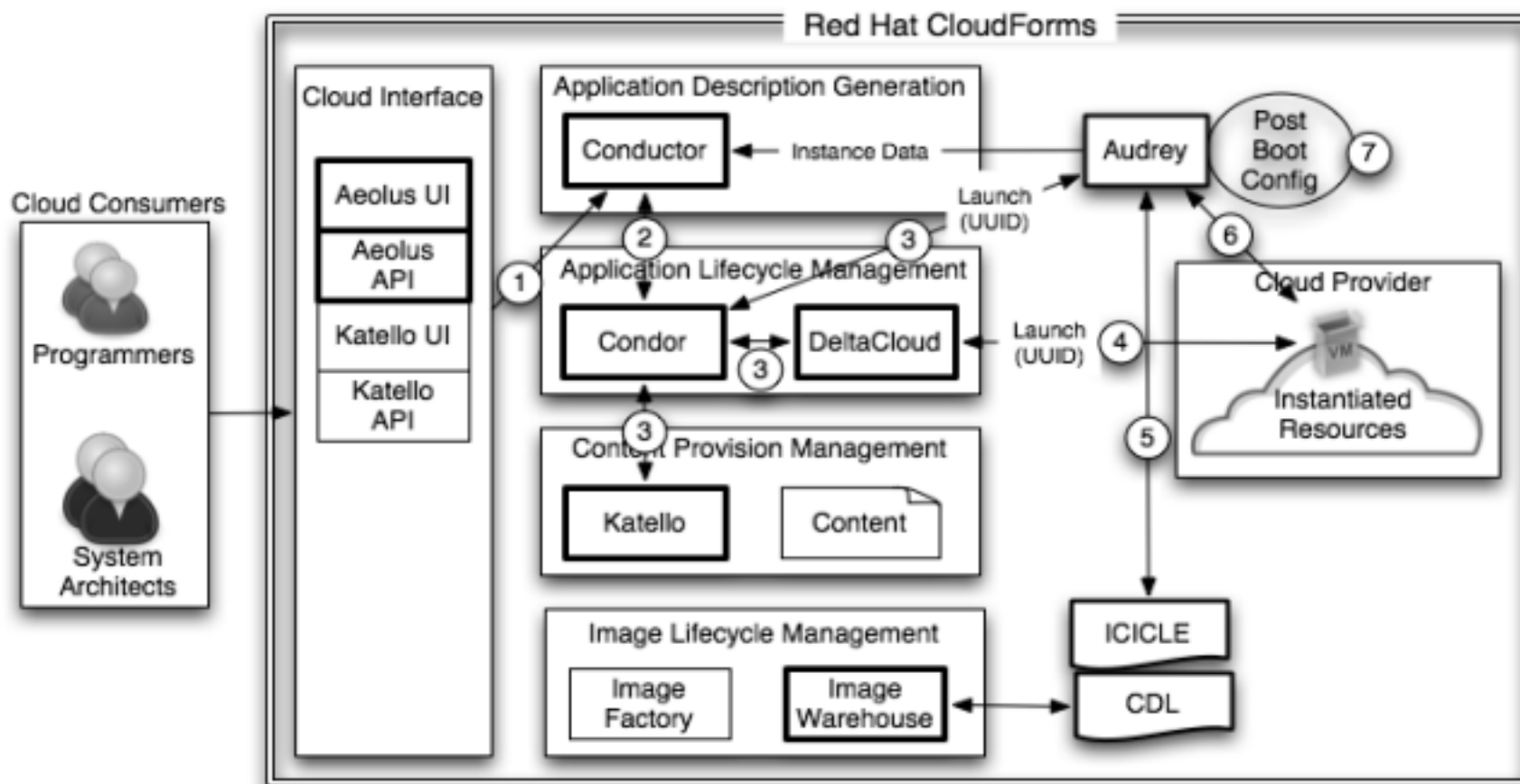


Или даже так:



Анатомия современного «облака» (2/3)

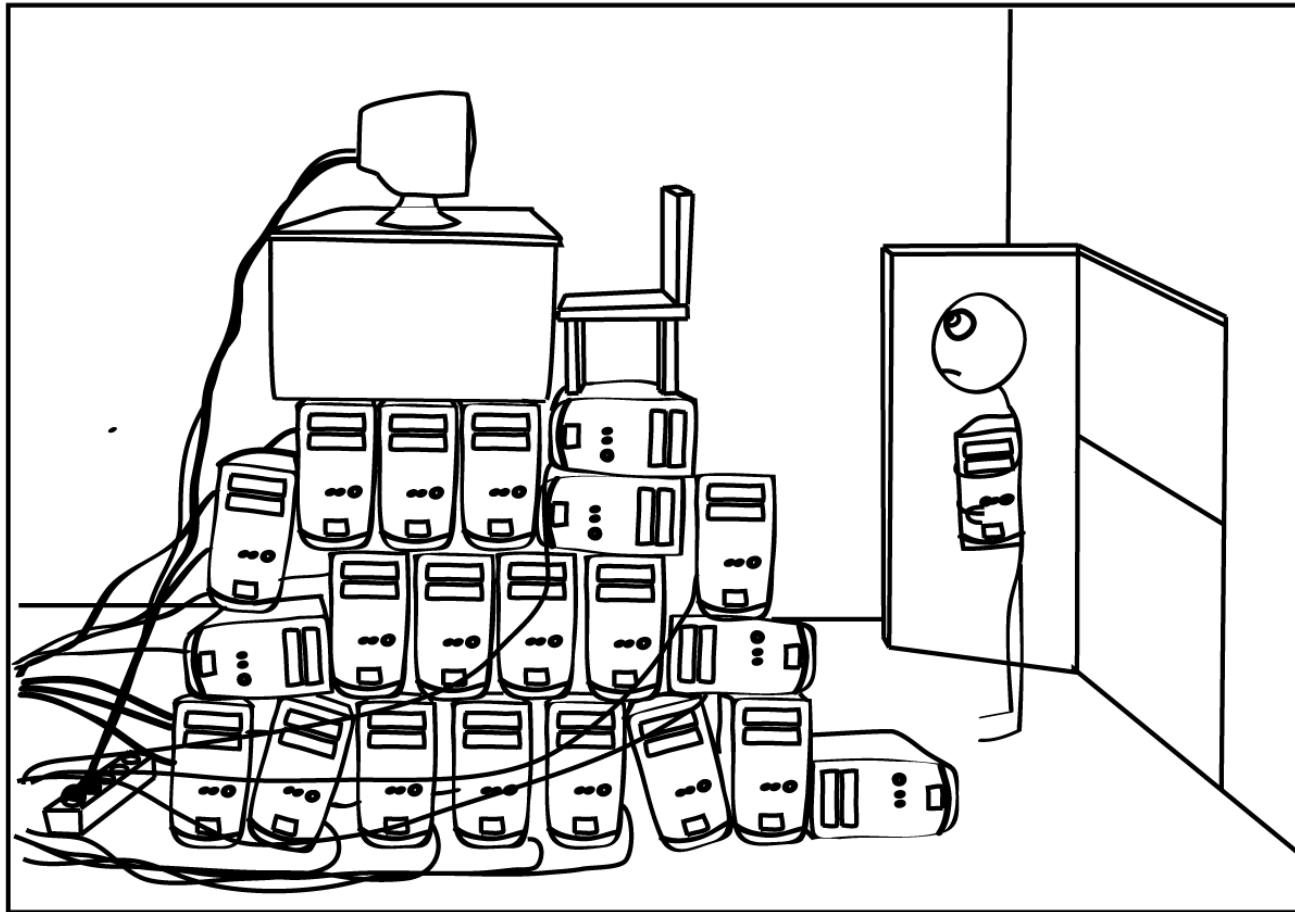
Но потом вы заглядываете внутрь, а там...



Вы думаете это облако? Да нет. Это только шлюз к нему. И по каждому квадратику мануал на 250 страниц с указанием, чего делать нельзя. В общем, лютый, бешеный ...

Анатомия современного «облака» (3/3)

В общем, пока получаем облако на китайцах.



Вы в роли китайца.

Что реально имеем на сегодняшний день

Все не так уж плохо. Просто надо сразу признать, что серьезные облака — пока что тема исследовательская.

- Есть статьи исследователей IBM, Red Hat, VMware, Parallels, прочих представителей индустрии. *Спросите окружающих, распечатки где-то ходят по рукам.*
- Есть исследовательские проекты в Беркли, Стэнфорде, MIT. *Статьи и ссылки тоже найдутся где-то вокруг.*
- Есть результаты полученные уже физтехами. *Графики и комментарии к ним опять же найдутся у окружающих.*

Где здесь области для творчества? Как минимум:

- Безопасность в облаках. Здесь нет еще даже методологии.
- Динамическое управление ресурсами в облаках. Здесь есть основы, но нужны работающие технологии, в которые всерьез поверят.
- Управление и мониторинг в облаке. Здесь традиционные подходы сейчас начинают сильно ломаться и перерождаться.

**Это горячая тема. В ней есть куда приложить все то, чему вас учили эти годы — от матана до кодинга.
*Есть еще вопросы? Задайте их окружающим.***