# Life in the Cloud

## Историите на системният администратор

Владимир Витков

2014.11.15 / BlagoevgradConf



Cloud услугите навлизат все повече и повече и всички твърдят че ще ви помогнат да си вършите по-бързо и по-ефективно работата. Че ще намалят разходите ви и ще увеличат безгранично капацитета.

Дали е така или не ще определим накрая. А между временно ще покажем плюсове, минуси, грешки и голяма част от работата която Системният администратор ще не ще трябва да свърши за да се случат облачните неща. Ако имате чадър ... забравете го. В облака не вали, там е буря.

# Кой е пред вас

- Владимир
- Системен Администратор
- Привърженик на FOSS но не и зеалот
- Експериментатор
- Относително приятен човек за разговор*





# Основи

# Що е това Cloud

- Хостинг - изберете си доставчик
- Инфраструктура под наем (IaaS) - Amazon / Rackspace
- Платформа под наем (PaaS) - Engine Yard / Heroku
- Софтуер под наем (SaaS) - Office365 / Pingdom



Хостинг - класически познат като уеб хостинг. Ресурси за отдалечено ползване.

IaaS - виртуални машини с които може да правите каквото искате. Доста работа, държат се като обикновенни сървъри.

PaaS - платформа, която може да прави неща. Качвате си кода (с инструкции) и той работи.

SaaS - Приложения под наем.

# Къде и как се използва

- Обработка на данни
- Допълнителен капацитет
- Намаляне на капиталовите разходи



Някои примери за употреба и стратегии за ефективно използване. Сезонни натоварвания са особенно добър пример.

# Предимства и недостатъци

- Там е и е винаги наличен
- Лесно се експериментира
- Няма (голяма) нужда от Системен администратор
- Като супермаркет е
- Забравят се неща
- Свързаността спира
- Дори и в облака се случва да спре тока



Налично място за експерименти, винаги има място

Но се плаща и обикновенно няма големи гаранции за наличност (SLA)

Зависите от всички доставчици по трасето а и не е нечувано да стане беля

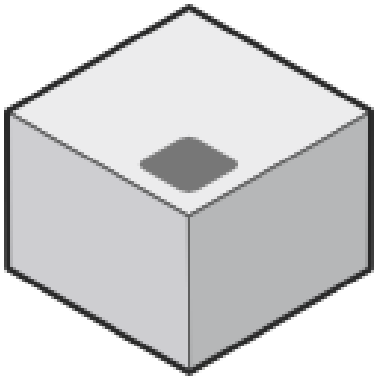
# The APP





## Имаме идея

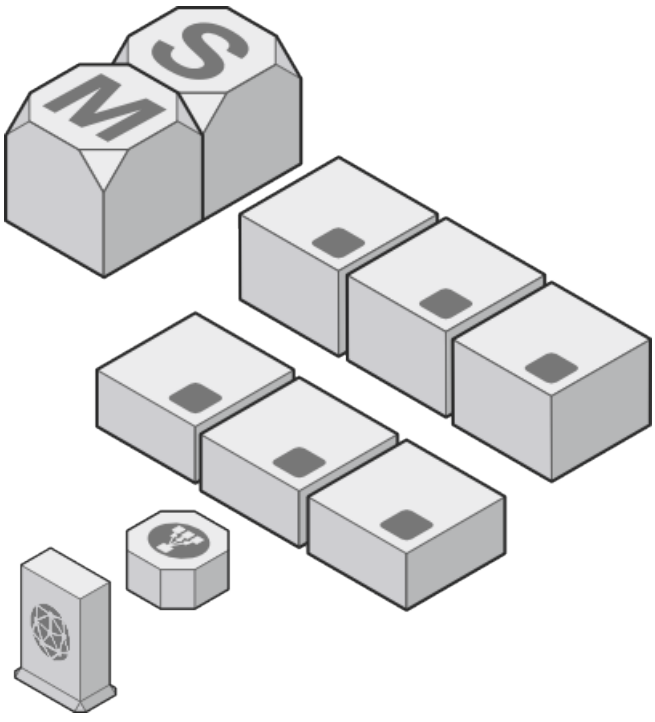
- Единична машина
- Всичко заедно



Блестяща идея, нахвърляне на гол скелет, разработване на тестово приложение. Всичко това се случва обикновенно на една машина (често на разработчика). В тази фаза обикновенно няма големи идеи и предвиждане на трафик и разпределена архитектура



# Production Ready



# Production Ready

- Разделяне на компоненти
- Web/App/DB

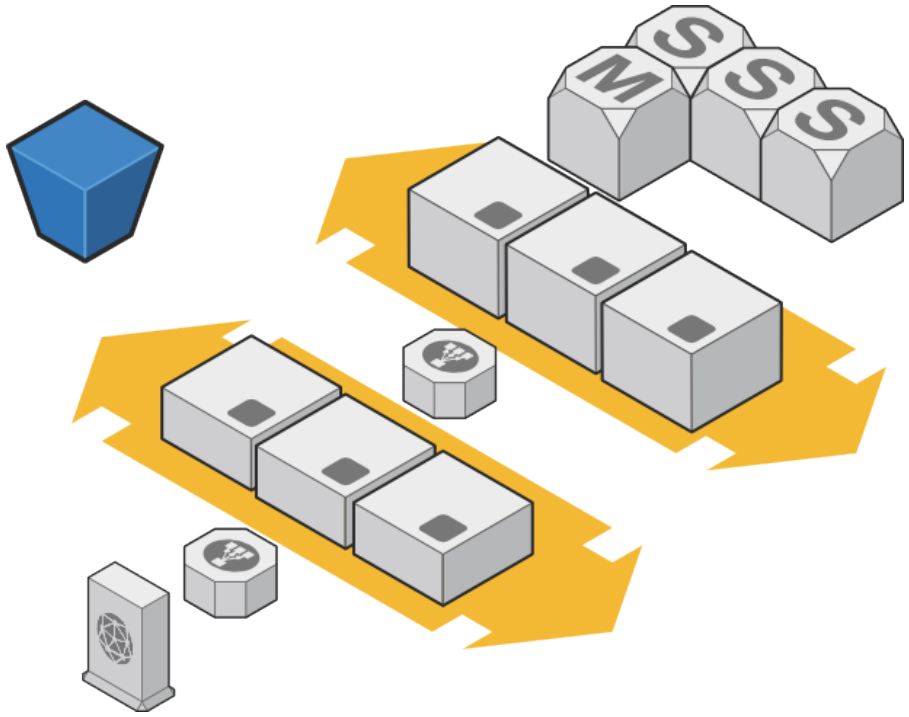


Приложението вече е голямо, изтествано и следва подготовка за нормална употреба. Разделят се отделните компоненти, избират се размери на машини, преценява се трафика и други.

Ползи от Cloud - лесна подмяна на размера, лесна промяна на броя машини

Обикновенно се ползва 3 tier архитектура с балансъри между нивата.

# Now the traffic comes






Основи The APP Проблеми и решения Живот в облака Някои истории Ресурси Q/A

# Now the traffic comes

- Много Frontend-и
- Много Application сървъри
- Master/Slave(s) за базата
- Oh *BEEEP* the traffic keeps comming

С нарастване на популярността се увеличава броят на машини, автоматизират се някои задачи. Идва момент в който трафика е неудържим и администраторите с рудиментарна автоматизация започват да изнемогват.

Започват да се мислят решения за автоматично скалиране на всяко ниво от архитектурата



# Проблеми и решения

# Прости компоненти

- Тръгват по-бързо
- Искат по-малко ресурси
- По-Лесни са за управление
- По-Лесно се търсят грешки в тях



Малките компоненти са по-пъргави, и по лесни за управление. Компонент за сваляне на файлове по адрес, компонент за местене по правила. Това че компонента е малък не значи че трябва да изпълнява само една дейност. Напълно нормално е да изпълнява няколко отделни дейности стига те да са логично организирани.

# Прости компоненти (2)

- Зависят един от друг
- Повече неща за управление
- Повече усилие за съвместимост
- Системата става трудна за управление



Но за сметка на това са по-голям брой и управлението им като цяло довежда до нови проблеми. Enter DevOps.

# Решения

- Опашки
- Твърди API-та
- Developers, Developers, Developers
- Кеширане



Рано или късно се стига до няколко стандартни решения.

Използване на малки компоненти, всеки компонент има твърдо дефинирано (и стабилно) API, използване на опашки за разделяне на компонентите и свободен растеж, кеширане, централизирано съхранение на параметри/конфигурации на системата и други.

Основни водещи трябва да са разработчиците, но това не винаги е така. Често се налага администратори с опит да насочват процеса.

# Живот в облака





## DO's

- Никога не вярвайте на Sales/Marketing
- Бекъпи
- Стратегия за възстановяване (DR)
- Не се заключвайте
- Сървърът не е незаменим

Облакът е динамично животно без край. Маркетинга винаги ще ви залива с информация която е не напълно вярна. Никога ама никога никога не пропускайте архивните копия. Измислете си и следвайте стратегия за бизнес стабилност и възстановяване (DR/BC). Сървърите са евтини и ефимерни, не ги мислете прекалено.



# DO's (2)

- Дискът може да изчезне*
- Имате си място за игра
- SWAP considered harmfull*
- Test, Test, Test
- Backup, Backup, Backup

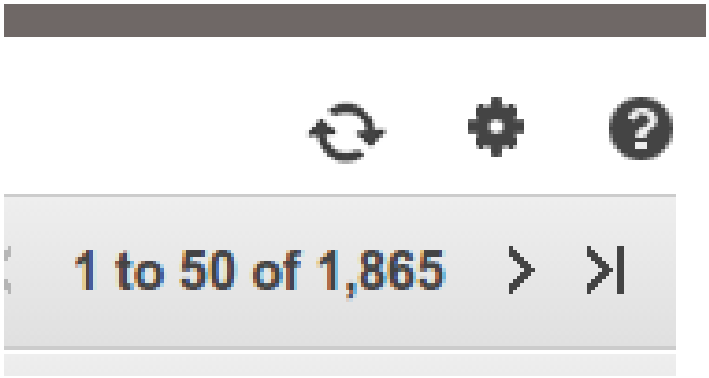
Както и в реалните машини така и във виртуалните могат да имат проблеми, дисковете се чупят, и не са перманентни (освен ако не си ги направите такива).

Всеки един доставчик на клауд услуги има безплатни опции. Да ограничени са но ги има. Тествайте на тях. Ако имате локални дискове, ползвайте ги разумно.



# DONT's

- Не разчитайте до безкрайност
- Сървърите са бетон
- Мрежата е супер ЯКА
- Те си имат няколко захранвания
- Не прекалявайте с дизайна



Сървърите са ефимерни, и безкрайно ненадеждни. Във всяка една система има проблеми. Много и различни проблеми. Ако прекалявате с ранния дизайн ще имате проблеми. Процеса е винаги итеративен и няма смисъл да се втурвате от самото начало.



# DONT's (2)

- Не забравяйте неизползвани ресурси
- Липсващ мониторинг
- Шумен мониторинг
- Не следете всичко лично
- Не игнорирайте възможностите които имате

Облака е готин, облака ни позволява да правим какво ли не, но не ни плаща сметките. По-точно прави ги големи ако сме невнимателни.

Мониторинг - без него не може, с него не става. Конфигурирайте го правилно за да си нямате проблеми.

Възползвайте се от всичко което ви дава облака, но не зависете от него. Измислете си процедури и начини да реализирате всичко и сами.



# Някои истории

# Аз само рестартирах машината

- Без процедура
- Без инструменти
- На ръка
- Без архиви
- Проста билд машина (лесно)





# Трябва ми бекъпа на dev12

- А правил ли си
- А разчиствал ли си стария
- А казал ли си
- Ама там е всичко
- Админа е малко параноик

# AWS Total Zone failure

- Once upon a time there was AWS eu-west-1a
- It is no more
- Sleep . . . for someone else
- Let the pain flow trough me





# AWS Power failure

- Инцидент
- 600 VM Dead
- Mass watchdog failure and disconnect
- Аварийно възстановяване на капацитет
- Последващи проблеми с разчистването
- Oh they are phoenixes (with no mind)

# AWS Network Failure

- Хмм нещо прецъка
- 200+ машини изчезнаха
- Този път бяхме подготвени
- 45 мин полуавтоматично възстановяване
- Инфраструкрурни промени





# Blizzard Planning failure

- 1 Mil players in 1 yer
- Nope - 3 months
- Reactive panic mode
- Large scale production refactoring

# Parse 12 mil CV

- Комерсиален Продукт
- Perl/XML/C
- Heavy as shit
- Пусни повече ... Затлачване
- Автоматизирано скалиране
- Profit





# Кракване на пароли

- g1/i1/hi1
- Клъстер
- Почти без пари
- Elcomsoft
- WPA Cracking

# Правене на пакети

- As easy as a pie*
- Процедури
- Автоматизация
- wanna-build/buildd
- Clean Room
- Култура





# Тестове/Компилиране

- Cloud is nice
- 10 машини ще направят тестовете по-бързо
- Jenkins/Hudson master + on demand slaves
- Dev is happy
- Accounting is happy
- There are no builders (Yeah price is too high)

# Ресурси





## Благодарности

- AWS
- Nik
- Линукс За Българи - http://www.linux-bg.org
- It-Tour - http://it-tour.bg/
- Някой който със сигурност съм забравил

# Връзки

- https://github.com/zeridon/presentations/life-in-the-cloud
- http://www.linux-bg.org
- http://goo.gl/A6m8OC
- https://aws.amazon.com/message/2329B7/
- http://youtu.be/xyPzTywUBsQ





# Q/A

# Контакти


- Владимир Витков
- vvitkov@linux-bg.org
- http://www.getoto.net/me/