

# Приключения DevOps инженера в мире BSOD

Alex Chistyakov, ex-Principal Eng. @ Git in Sky

# Что такое DevOps?



# Что такое DevOps?

- Человек?

# Что такое DevOps?

- Человек?
- Практика?

# Что такое DevOps?

- Человек?
- Практика?
- Методология?

# Что такое DevOps?

- Человек?
- Практика?
- Методология?
- Идеология?

# Что такое DevOps?

- CAMS

# Что такое DevOps?

- Culture
- Automation
- Measurement
- Sharing



# О чем будет этот доклад?

- Culture
- **Automation**
- Measurement
- Sharing

# О чем будет этот доклад?

- Culture
- **Automation**
- Measurement
- Sharing
- **И немного личных переживаний**

# О чем будет этот доклад?



# Выводы

- В Windows есть свой собственный configuration management фреймворк
- Он работает на удивление хорошо
- Это **ПОЧТИ** не больно

# Прекрасная снежинка прекрасна



Или нет



# Повторимость



# Список ингредиентов

- Vagrant (<http://vagrantup.com>)



# Vagrant

- Написан на Ruby командой Митчелла Хашимото
- Представляет собой eDSL для организации связи между провайдером виртуализации и системой провижнинга (той самой CM system)
- Неплохо подходит для создания переносных повторяемых лабораторий

# Список ингредиентов

- Vagrant (<https://vagrantup.com>)
- Packer (<https://www.packer.io>)

# Packer

- Написан на Go командой Митчелла Хашимото
- Представляет собой утилиту создания базовых образов для различных провайдеров виртуализации по их условно-декларативному описанию

# Список ингредиентов

- Vagrant (<https://vagrantup.com>)
- Packer (<https://www.packer.io>)
- CM system (система управления конфигурацией)

# CM system

- Написана без участия Митчелла Хашимото
- Чтобы было не скучно, их написано много\* разных

\* несколько

# Список ингредиентов

- Vagrant (<https://vagrantup.com>)
- Packer (<https://www.packer.io>)
- CM system (система управления конфигурацией)
- Магия!



# Магия!

- Магия вызывает удивление (и 5 стадий принятия)
- Ненавижу магию
- Удивлялся я в этом проекте постоянно

# CM systems: 4 основные

- Puppet (Ruby)
- Chef (Ruby)
- SaltStack (Python)
- Ansible (Python)



# CM systems: 4 основные

- Puppet (Ruby)
- Chef (Ruby)
- SaltStack (Python)
- Ansible (Python)

Agent-based vs. agentless

# Выбор: SaltStack

- Выбирал не я, а заказчик
- Выбор неожиданно хорош, SaltStack точно использовался коллегами для управления Windows-инфраструктурой
- (Все 4 системы могут быть использованы)

# Краткий курс SaltStack

- modules: кусочки кода на Python для исполнения команд вручную
- state modules: идемпотентные `{{modules}}` не вручную
- states: списки того, что нужно применить
- pillars: структуры данных для states

алгоритмы + структуры данных

# Краткий курс SaltStack

- Сервер на управляющей ноде
- Клиент на управляемых нодах
- Транспортom является ZeroMQ
- Это большой плюс\* - файлы точно доедут

\* если инфраструктура не очень велика

# SaltStack и Windows

- Есть набор state и non-state модулей для Windows
- Я использовал модули:
- `dsc.set_lcm_config` (non-state)
- `dsc.run_config` (non-state)
- Что именно они делают, будет ясно позже
- Любой non-state модуль дергается из `state module.run`

# Как собрать базовый образ

- Инсталляционный CD
- Boxcutter (<https://github.com/boxcutter/windows>)
- Магия!
- Примерно 40 минут на попытку
- Примерно 2 раза из 3 VirtualBox падал из-за драйвера видеокарты

# Как собрать базовый образ

- Мы делаем лабораторию с AD
- Одна машина - domain controller, вторую вводим в домен
- Проблема duplicate SID
- **Решение: при первом запуске с базового образа менять SID**

# Как собрать базовый образ

- Я модифицировал конфиги boxcutter под наши нужды
- Конфиги boxcutter:
- По сути, императивная свалка cmd и PowerShell-скриптов, монтируемая через виртуальный floppy drive и запускаемая в нужном порядке



# Воркфлоу

- `vagrant up`
- Машина с CentOS 7 и salt server на ней
- Две машины с Windows на которых провижнером типа shell установлен salt client, NuGet и DSC ресурсы из PowerShell Gallery
- `salt '*' state.highstate` на сервере

# Магия! PowerShell DSC

- PowerShell DSC - декларативное описание состояния
- По сути - встроенная CM system для Windows
- DSC Resource Kit

(<https://github.com/PowerShell/DscResources>)

# Конфигурация в стиле DSC

- Файл с описанием узлов и описанием необходимых DSC resources с установлением связей между списком ресурсов и узлами
- В одном таком файле можно описать конфигурацию всего домена (я этим не пользовался, SaltStack такое не поддерживал, но я заставил)

# Анатомия DSC ресурса

- Обычный PS скрипт, к которому приписана метайнформация
- В скрипте по соглашению функции
- **Get-TargetResource**, Set-TargetResource, Test-TargetResource
- “Если не test, то set”

# Особенности error handling в PS

- Terminating и non-terminating errors
  - **Нужно все non-terminating сделать terminating**
  - В начале скрипта:
  - `$ErrorActionPreference = "Stop"`
  - Иначе non-terminating error терминирует DSC скрипт
- O\_O

# SaltStack и Windows еще раз

- `dsc.set_lcm_config` - конфигурирует LCM
- `dsc.run_config` - компилирует PS DSC скрипт в набор MOF-файлов и передает их в LCM
- LCM - Local Configuration Manager

# Как сконфигурировать LSM

- Разрешить **автоматический рестарт**
- Разрешить продолжать конфигурирование системы после автоматического рестарта

# Что именно сделал я

- Поставил AD services
- Создал новый домен
- Поставил DNS и DHCP services
- Ввел клиента в домен (**выключить DHCP в Vagrant**)
- Поставил Exchange
- Завел distribution groups



# Что именно сделал я

- Запатчил `dsc.run_config` (файл `win_dsc.py`)
- Запатчил DSC модуль `MSFT_xExchInstall`
- Написал свой DSC ресурс `xDistributionGroup`
- С боями прорывался к своим

# Воркфлоу еще раз

- vagrant раскидывает файлы
- SaltStack раскидывает файлы
- Всю настоящую работу делает PS DSC LCM
- К привычному нормальному человеку миру Unix это не имеет никакого отношения

Повторимость: ну и как оно?

# Повторимость: ну и как оно?

- ЭТО НЕ ДОЛЖНО ПОВТОРИТЬСЯ
- НЕ ДЕЛАЙТЕ ЭТО СО МНОЙ
- NEVER AGAIN!

# Повторимость: ну и как оно?

- Объективно, несмотря на некоторые неочевидные для нормального человека проблемы, добиться результата

**МОЖНО**

# Ваши вопросы?

- [alexclear@gmail.com](mailto:alexclear@gmail.com)
- Skype, Telegram: demeliorator
- У меня есть канал в Telegram:  
<http://t.me/lhommequipleure>

# Спасибо, что были со мной!

- Есть чувства, которые лучше переживать не в одиночку