

Содержание

Предисловие	16
Предисловие ко второму изданию	18
Предисловие к первому изданию	20
Глава 1. Введение.....	23
Примечание о версиях	24
Ansible: область применения.....	24
Как работает Ansible	25
Какие преимущества дает Ansible?	26
Простота синтаксиса	27
Отсутствие необходимости установки на удаленных хостах.....	27
Основан на технологии принудительной настройки	27
Управление небольшим числом серверов	28
Встроенные модули.....	28
Очень тонкий слой абстракции	29
Не слишком ли проста система Ansible?	30
Что я должен знать?.....	31
О чем не рассказывается в этой книге	31
Установка Ansible.....	32
Подготовка сервера для экспериментов.....	33
Использование Vagrant для подготовки сервера.....	33
Передача информации о сервере в Ansible	36
Упрощение задачи с помощью файла ansible.cfg.....	37
Что дальше	40
Глава 2. Сценарии: начало	41
Подготовка	41
Очень простой сценарий.....	42
Файл конфигурации Nginx.....	44
Создание начальной страницы	44
Создание группы веб-серверов	45
Запуск сценария.....	45
Сценарии пишутся на YAML	47
Начало файла	47
Комментарии	47

Строки	47
Булевы выражения	47
Списки	48
Словари	48
Объединение строк.....	49
Структура сценария.....	49
Операции	50
Задачи.....	52
Модули.....	53
Резюме.....	54
Есть изменения? Отслеживание состояния хоста	54
Становимся знатоками: поддержка TLS	55
Создание сертификата TLS	56
Переменные	56
Создание шаблона с конфигурацией Nginx.....	58
Обработчики	59
Запуск сценария	60
Глава 3. Реестр: описание серверов	63
Файл реестра	63
Вводная часть: несколько машин Vagrant	64
Поведенческие параметры хостов в реестре.....	67
ansible_connection.....	67
ansible_shell_type.....	67
ansible_python_interpreter	68
ansible_*_interpreter	68
Переопределение поведенческих параметров по умолчанию	68
Группы, группы и еще раз группы.....	68
Пример: развертывание приложения Django.....	70
Псевдонимы и порты	72
Группировка групп	72
Имена хостов с номерами (домашние питомцы и стадо)	72
Переменные хостов и групп: внутренняя сторона реестра.....	73
Переменные хостов и групп: создание собственных файлов	75
Динамический реестр	76
Интерфейс сценария динамического реестра.....	77
Написание сценария динамического реестра.....	78
Предопределенные сценарии реестра	81
Деление реестра на несколько файлов.....	82
Добавление элементов во время выполнения с помощью add_host	
и group_by	82
add_host	82
group_by	83

Глава 4. Переменные и факты	85
Определение переменных в сценариях	85
Вывод значений переменных	86
Регистрация переменных	86
Факты	89
Просмотр всех фактов, доступных для сервера	90
Вывод подмножества фактов	91
Любой модуль может возвращать факты	92
Локальные факты	93
Использование модуля <code>set_fact</code> для задания новой переменной	94
Встроенные переменные	94
<code>hostvars</code>	95
<code>inventory_hostname</code>	95
<code>groups</code>	96
Установка переменных из командной строки	96
Приоритет	97
Глава 5. Введение в Mezzanine: тестовое приложение	99
Почему сложно разворачивать приложения в промышленном окружении	99
База данных PostgreSQL	103
Сервер приложений Gunicorn	103
Веб-сервер Nginx	104
Диспетчер процессов Supervisor	105
Глава 6. Развертывание Mezzanine с помощью Ansible	106
Вывод списка задач в сценарии	106
Организация устанавливаемых файлов	107
Переменные и скрытые переменные	108
Использование цикла (<code>with_items</code>) для установки большого количества пакетов	109
Добавление выражения <code>become</code> в задачу	111
Обновление кэша диспетчера пакетов <code>apt</code>	111
Извлечение проекта из репозитория Git	113
Установка Mezzanine и других пакетов в <code>virtualenv</code>	115
Короткое отступление: составные аргументы задач	117
Настройка базы данных	120
Создание файла <code>local_settings.py</code> из шаблона	121
Выполнение команд <code>django-manage</code>	124
Запуск своих сценариев на Python в контексте приложения	125
Настройка конфигурационных файлов служб	127
Активация конфигурации Nginx	130
Установка сертификатов TLS	130
Установка задания <code>cron</code> для Twitter	131

Сценарий целиком	132
Запуск сценария на машине Vagrant	136
Устранение проблем.....	136
Не получается извлечь файлы из репозитория Git	136
Недоступен хост с адресом 192.168.33.10.xip.io	137
Bad Request (400).....	137
Установка Mezzanine на нескольких машинах	137
Глава 7. Роли: масштабирование сценария	138
Базовая структура роли	138
Примеры ролей: database и mezzanine.....	139
Использование ролей в сценариях.....	139
Предварительные и заключительные задачи.....	140
Роль database для развертывания базы данных	141
Роль mezzanine для развертывания Mezzanine	143
Создание файлов и директорий ролей с помощью ansible-galaxy.....	148
Зависимые роли.....	148
Ansible Galaxy.....	149
Веб-интерфейс.....	149
Интерфейс командной строки.....	150
Добавление собственной роли.....	151
Глава 8. Сложные сценарии	152
Команды changed_when и failed_when.....	152
Фильтры	155
Фильтр default.....	156
Фильтры для зарегистрированных переменных	156
Фильтры для путей к файлам	156
Создание собственного фильтра	157
Подстановки.....	158
file	159
pipe.....	160
env.....	160
password.....	160
template	161
csvfile	161
dnstxt	162
redis_kv.....	163
etcd.....	164
Написание собственного плагина	164
Сложные циклы	164
with_lines	165
with_fileglob.....	165

with_dict	166
Циклические конструкции как плагины подстановок	167
Управление циклами	167
Выбор имени переменной цикла	167
Управление выводом	168
Подключение	169
Динамическое подключение	170
Подключение ролей	171
Блоки	172
Обработка ошибок с помощью блоков	172
Шифрование конфиденциальных данных при помощи Vault	175
Глава 9. Управление хостами, задачами и обработчиками	178
Шаблоны для выбора хостов	178
Ограничение обслуживаемых хостов	179
Запуск задачи на управляющей машине	179
Запуск задачи на сторонней машине	180
Последовательное выполнение задачи на хостах по одному	180
Пакетная обработка хостов	182
Однократный запуск	183
Стратегии выполнения	183
linear	184
free	185
Улучшенные обработчики	186
Обработчики в pre_tasks и post_tasks	186
Принудительный запуск обработчиков	187
Выполнение обработчиков по событиям	189
Сбор фактов вручную	195
Получение IP-адреса хоста	195
Глава 10. Плагины обратного вызова	197
Плагины стандартного вывода	197
actionable	198
debug	198
dense	199
json	199
minimal	200
online	200
selective	200
skippy	200
Другие плагины	201
foreman	201
hipchat	202

jabber.....	202
junit.....	202
log_plays.....	203
logentries	203
logstash	203
mail	204
osx_say	204
profile_tasks	204
slack.....	205
timer	205
Глава 11. Ускорение работы Ansible	206
Мультиплексирование SSH и ControlPersist	206
Включение мультиплексирования SSH вручную	207
Параметры мультиплексирования SSH в Ansible	208
Конвейерный режим	209
Включение конвейерного режима	210
Настройка хостов для поддержки конвейерного режима	210
Кэширование фактов	211
Кэширование фактов в файлах JSON	213
Кэширование фактов в Redis	213
Кэширование фактов в Memcached.....	214
Параллелизм	214
Асинхронное выполнение задач с помощью Async	215
Глава 12. Собственные модули.....	217
Пример: проверка доступности удаленного сервера.....	217
Использование модуля script вместо написания своего модуля.....	217
Где хранить свои модули.....	218
Как Ansible вызывает модули	218
Генерация автономного сценария на Python с аргументами (только модули на Python)	219
Копирование модуля на хост	219
Создание файла с аргументами на хосте (для модулей не на языке Python)	219
Вызов модуля	219
Ожидаемый вывод.....	220
Ожидаемые выходные переменные	220
Реализация модулей на Python	221
Анализ аргументов.....	222
Доступ к параметрам	223
Импортирование вспомогательного класса AnsibleModule	223
Свойства аргументов.....	224

AnsibleModule: параметры метода инициализатора	226
Возврат признака успешного завершения или неудачи	229
Вызов внешних команд.....	229
Режим проверки (пробный прогон)	230
Документирование модуля.....	231
Отладка модуля.....	233
Создание модуля на Bash	234
Альтернативное местоположение интерпретатора Bash	235
Примеры модулей	236
Глава 13. Vagrant	237
Полезные параметры настройки Vagrant	237
Перенаправление портов и приватные IP-адреса	237
Перенаправление агента.....	239
Сценарий наполнения Ansible	239
Когда выполняется сценарий наполнения	239
Реестр, генерируемый системой Vagrant	240
Наполнение нескольких машин одновременно.....	241
Определение групп.....	242
Локальные сценарии наполнения	243
Глава 14. Amazon EC2	244
Терминология	246
Экземпляр	246
Образ машины Amazon.....	246
Теги.....	246
Учетные данные пользователя	247
Переменные окружения.....	247
Файлы конфигурации.....	248
Необходимое условие: библиотека Python Boto.....	248
Динамическая инвентаризация	249
Кэширование реестра.....	251
Другие параметры настройки	251
Автоматические группы	251
Определение динамических групп с помощью тегов.....	252
Присваивание тегов имеющимся ресурсам	252
Создание более точных названий групп.....	253
EC2 Virtual Private Cloud (VPC) и EC2 Classic	254
Конфигурирование ansible.cfg для использования с ec2	255
Запуск новых экземпляров	255
Пары ключей EC2.....	257
Создание нового ключа.....	257
Выгрузка существующего ключа	258

Группы безопасности	258
Разрешенные IP-адреса	260
Порты групп безопасности	260
Получение новейшего AMI	261
Добавление нового экземпляра в группу	262
Ожидание запуска сервера.....	264
Создание экземпляров идемпотентным способом	265
Подведение итогов	265
Создание виртуального приватного облака	267
Динамическая инвентаризация и VPC	272
Создание AMI	272
Использование модуля ec2_ami	272
Использование Packer	273
Другие модули	277
Глава 15. Docker	278
Объединение Docker и Ansible	279
Жизненный цикл приложения Docker	280
Пример применения: Ghost.....	281
Подключение к демону Docker	281
Запуск контейнера на локальной машине.....	281
Создание образа из Dockerfile.....	282
Управление несколькими контейнерами на локальной машине	284
Отправка образа в реестр Docker.....	285
Запрос информации о локальном образе	287
Развертывание приложения в контейнере Docker.....	288
Postgres	288
Веб-сервер	289
Веб-сервер: Ghost	290
Веб-сервер: Nginx	291
Удаление контейнеров	291
Прямое подключение к контейнерам	292
Контейнеры Ansible	293
Контейнер Conductor.....	293
Создание образов Docker	294
Настройка container.yml	295
Запуск на локальной машине	297
Публикация образов в реестрах	298
Развертывание контейнеров в промышленном окружении.....	300
Глава 16. Отладка сценариев Ansible	301
Информативные сообщения об ошибках	301
Отладка ошибок с SSH-подключением	302

Модуль debug	303
Интерактивный отладчик сценариев.....	304
Модуль assert.....	305
Проверка сценария перед запуском.....	307
Проверка синтаксиса.....	307
Список хостов	307
Список задач	308
Проверка режима.....	308
Вывод изменений в файлах	308
Выбор задач для запуска	309
Пошаговое выполнение	309
Выполнение с указанной задачи.....	309
Теги.....	310
Глава 17. Управление хостами Windows.....	311
Подключение к Windows	311
PowerShell.....	312
Модули поддержки Windows.....	314
Наш первый сценарий.....	315
Обновление Windows	316
Добавление локальных пользователей	317
Итоги	320
Глава 18. Ansible для сетевых устройств	321
Статус сетевых модулей	322
Список поддерживаемых производителей сетевого оборудования.....	322
Подготовка сетевого устройства.....	322
Настройка аутентификации через SSH.....	323
Как работают модули.....	325
Наш первый сценарий.....	326
Реестр и переменные для сетевых модулей	327
Локальное подключение	328
Подключение к хосту.....	329
Переменные для аутентификации	329
Сохранение конфигурации	330
Использование конфигураций из файлов	331
Шаблоны, шаблоны, шаблоны	334
Сбор фактов	336
Итоги	338
Глава 19. Ansible Tower: Ansible для предприятий	339
Модели подписки	340
Пробная версия Ansible Tower	340

Какие задачи решает Ansible Tower.....	341
Управление доступом.....	341
Проекты.....	342
Управление инвентаризацией.....	342
Запуск заданий из шаблонов	344
RESTful API.....	346
Интерфейс командной строки Ansible Tower	347
Установка	347
Создание пользователя.....	348
Запуск задания.....	350
Послесловие	351
Приложение А. SSH.....	352
«Родной» SSH	352
SSH-агент.....	352
Запуск ssh-agent.....	353
macOS	353
Linux	354
Agent Forwarding.....	354
Команда sudo и перенаправление агента	356
Ключи хоста	357
Приложение В. Использование ролей IAM для учетных данных EC2.....	361
Консоль управления AWS	361
Командная строка.....	362
Глоссарий	365
Библиография	368
Предметный указатель	369
Об авторах.....	380
Колофон	381

Я буквально проглотил рукопись первого издания «Установка и работа с Ansible» за несколько часов: Лорин прекрасно справился с задачей описания всех аспектов Ansible, и я был рад услышать, что он решил объединиться с Рене для подготовки второго издания. Эти два автора проделали громадную работу, чтобы показать нам, как пользоваться невероятно удобной утилитой, и я не могу вспомнить ни одного момента, которого бы они не охватили в полной мере.

– *Ян-Пит Менс (Jan-Piet Mens), консультант*

Впечатляющая глубина освещения Ansible. Эта книга прекрасно подойдет не только начинающим, но и опытным специалистам, желающим понять все тонкости использования продвинутых возможностей. Фантастический источник информации для стремящихся повысить свой уровень владения Ansible.

– *Мэтт Джейнс (Matt Jaynes),
ведущий инженер, High Velocity Ops*

Самое замечательное в Ansible – возможность начать с простого прототипа и быстро продвигаться к намеченной цели. Однако со временем начинает ощущаться нехватка знаний, которые порой трудно получить.

«Установка и работа с Ansible» – очень ценный источник, восполняющий эту нехватку и разъясняющий особенности Ansible с самых основ до сложностей работы с YAML и Jinja2. А благодаря наличию массы практических примеров она позволяет получить представление, как другие автоматизируют свои окружения.

В течение последних нескольких лет, проводя теоретические и практические занятия, я всегда рекомендовал эту книгу своим коллегам и клиентам.

– *Даг Вьерс (Dag Wieers),
консультант и инженер-системотехник
в области систем на основе Linux,
долгое время участвовавший в разработке Ansible*

Эта книга помогает быстро приступить к использованию системы управления конфигурациями Ansible и описывает ее во всех подробностях. В ней приводится большое количество подсказок и практических советов и охватывается широкий круг вариантов использования, включая AWS, Windows и Docker.

– *Инго Йохим (Ingo Jochim),
руководитель отдела облачных реализаций, itelligence GMS/CIS*

Лорин и Рене проделали большую работу, написав эту книгу. Авторы берут читателя за руку и ведут его через наиболее важные этапы создания и управления проектов Ansible. Эта книга намного больше, чем справочник, – она охватывает ряд важнейших концептуальных тем, отсутствующих в официальной документации. Это превосходный источник знаний для начинающих и практических идей для более опытных пользователей Ansible.

– *Доминик Бартон (Dominique Barton),
инженер DevOps в confirm IT solutions*

Предисловие

Разработка системы Ansible началась в феврале 2012-го с создания простого побочного проекта, и ее стремительное развитие стало приятным сюрпризом. Сейчас над продуктом работает порядка тысячи человек (а идеи принадлежат даже большему числу людей), и он широко используется практически во всех странах мира. И наверняка вам удастся обнаружить, по крайней мере, нескольких человек, использующих его, в сообществе знакомых вам ИТ-специалистов.

Привлекательность Ansible объясняется ее простотой. И правда, Ansible не несет в себе новых, но объединяет все лучшее из уже существующих идей, разработанных другими экспертами, делая их чуть более доступными.

Создавая Ansible, я старался найти для нее место где-то между решениями автоматизации ИТ-задач (естественная реакция на огромные коммерческие пакеты программного обеспечения) и простыми сценариями, минимально необходимыми для выполнения своей работы. Кроме того, мне хотелось заменить систему управления конфигурациями, развертыванием и организацией проектов и нашу библиотеку произвольных, но важных сценариев командной оболочки единой системой. Вот в чем состояла идея.

Могли ли мы убрать важные архитектурные компоненты из стека автоматизации? Устранив демоны управления и переложив работу на OpenSSH, система могла бы начать управление компьютерами незамедлительно без установки агентов на контролируемые машины. Кроме того, система стала бы более надежной и безопасной.

Я заметил, что в предыдущих попытках создания систем автоматизации простые вещи заметно усложнялись, а написание сценариев автоматизации часто и надолго уводило меня в сторону от того, чему бы я хотел посвятить больше времени. В то же время мне не хотелось получить систему, на изучение которой не нужны месяцы.

Честно говоря, мне больше нравится писать программы, чем заниматься управлением системами автоматизации. Мне хотелось бы тратить на автоматизацию как можно меньше времени, чтобы высвободить его на решение более интересных задач. Ansible – это не та система, с которой приходится работать сутки напролет. Используя ее, вы сможете зайти, что-то поправить, выйти и продолжить заниматься своими делами. Я надеюсь, что эта черта Ansible понравится вам.

Хотя я потратил много времени, стараясь сделать документацию для Ansible исчерпывающей, всегда полезно взглянуть на одни и те же вещи под разными углами. Полезно увидеть практическое применение справочной документации. В книге «Установка и работа с Ansible» Лорин представляет Ansible, используя идиоматический подход, в точности как следовало бы изучать эту

систему. Лорин работал с Ansible практически с самого начала, и я очень благодарен ему за его вклад.

Я также безмерно благодарен каждому, кто принимал участие в проекте до настоящего времени, и каждому, кто подключится к нему в будущем. Наслаждайтесь книгой и получайте удовольствие от управления вашим компьютерным флотом! И не забудьте установить cowsay!

– Майкл ДеХаан (*Michael DeHaan*)
Создатель Ansible (программной части),
бывший технический директор компании Ansible, Inc.
апрель 2015

Предисловие ко второму изданию

За время, прошедшее с момента выхода первого издания (еще в 2014 году), в мире Ansible произошли большие изменения. Проект Ansible достиг следующей старшей версии 2.0. Также большие изменения произошли за рамками проекта: Ansible, Inc. – компания, стоящая за проектом Ansible, – была приобретена компанией Red Hat. Это никак не повлияло на разработку проекта Ansible: он так же активно развивается и привлекает новых пользователей.

Мы внесли множество изменений в это издание. Наиболее заметным стало появление пяти новых глав. Теперь книга охватывает плагины обратного вызова, хосты под управлением Windows, сетевое оборудование и Ansible Tower. Мы добавили в главу «Сложные сценарии» так много нового, что пришлось разбить ее на две части и добавить главу «Настройка хостов, запуск и обработчики». Мы также переписали главу «Docker», включив в нее описание новых модулей Docker.

Мы обновили все примеры кода для совместимости с Ansible 2.3. Например, устаревшую инструкцию `sudo` мы повсюду заменили более новой `become`. Мы также удалили ссылки на устаревшие модули, такие как `docker`, `ec2_vpc` и `ec2_api_search`, и заменили их более новыми. Глава «Vagrant» теперь охватывает локальные сценарии вызова Ansible, глава «Amazon EC2» – Packer Ansible, механизм удаленного вызова, глава «Ускорение работы Ansible» – асинхронные задания, а глава «Отладка сценариев Ansible» – новые средства отладки, появившиеся в версии 2.1.

Также было внесено множество мелких изменений. Например, мы отказались от использования контрольных сумм MD5 в OpenSSH и перешли на хэши SHA256, внося соответствующие изменения в примеры. Наконец, мы исправили ошибки, обнаруженные нашими читателями.

ПРИМЕЧАНИЕ К СТИЛЮ ИЗЛОЖЕНИЯ

Первое издание книги было написано одним автором, и в нем часто использовалось местоимение «я» первого лица. Это издание написано уже двумя авторами, поэтому употребление местоимения в первом лице кое-где может показаться странным. Тем не менее мы решили не исправлять его, потому что в большинстве случаев оно используется для выражения мнения одного из авторов.

БЛАГОДАРНОСТИ

От Лорин

Мои благодарности Яну-Пит Менсу (Jan-Piet Mens), Мэтту Джейнсу (Matt Jaynes) и Джону Джарвису (John Jarvis) за отзывы в процессе написания книги. Спасибо Айзаку Салдана (Isaac Saldana) и Майку Ровану (Mike Rowan) из SendGrid за поддержку этого начинания. Благодарю Майкла ДеХаана (Michael DeHaan) за создание Ansible и поддержку сообщества, которое разрослось вокруг продукта, а также за отзыв о книге, включая объяснения, почему в качестве названия было выбрано *Ansible*. Спасибо моему редактору Брайану Андерсону (Brian Anderson) за его безграничное терпение в работе со мной.

Спасибо маме и папе за их неизменную поддержку; моему брату Эрику (Eric), настоящему писателю в нашей семье; двум моим сыновьям Бенджамину (Benjamin) и Джулиану (Julian). И наконец, спасибо моей жене Стейси (Stacy) за все.

От Рене

Спасибо моей семье, моей жене Симоне (Simone) за любовь и поддержку, моим трем деткам, Джил (Gil), Сарине (Sarina) и Лиан (Léanne), за свет и радость, что они привнесли в мою жизнь; спасибо всем, кто внес свой вклад в развитие Ansible, спасибо вам за ваш труд; и особое спасибо Маттиасу Блейзеру (Matthias Blaser), познакомившему меня с Ansible.

Предисловие к первому изданию

ПОЧЕМУ Я НАПИСАЛ ЭТУ КНИГУ

Когда я писал свое первое веб-приложение, используя Django, популярный фреймворк на Python, я запомнил чувство удовлетворения, когда приложение наконец-то заработало на моем компьютере. Я запустил команду `django manage.py runserver`, указал в браузере `http://localhost:8000` и увидел свое веб-приложение во всей его красе.

Потом я подумал про все эти... *моменты*, которые необходимо учесть, чтобы просто запустить это чертово приложение на Linux-сервере. Кроме Django и моего приложения, мне потребовалось установить Apache и модуль `mod_python`, чтобы Apache мог работать с приложениями Django. Затем мне пришлось установить правильные значения в конфигурационном файле Apache, заставлявшие мое приложение работать и правильно обслуживать статичные компоненты.

Это было несложно – немного усилий, и готово. Мне не хотелось завязнуть в работе с файлами конфигурации, я лишь хотел, чтобы мое приложение работало. И оно работало, и все было прекрасно... пока через несколько месяцев мне не понадобилось запустить его снова на другом сервере и проделать всю ту же работу с самого начала.

В конце концов, я осознал, что все, что я делал, я делал неправильно. Правильный способ решать такого рода задачи имеет название, и это название – *управление конфигурациями*. Самое замечательное в управлении конфигурациями – полученные знания всегда сохраняют свою актуальность. Больше нет необходимости рыться в поисках нужной страницы в документации или копаться в старых записях.

Недавно коллега заинтересовался применением Ansible для внедрения нового проекта и спросил, как можно использовать идею Ansible на практике, кроме того что указано в официальной документации. Я не знал, что посоветовать почитать, и решил написать книгу, которая восполнит этот пробел, – и вот вы видите эту книгу перед собой. Увы, для него эта книга вышла слишком поздно, но я надеюсь, она окажется полезной для вас.

КОМУ АДРЕСОВАНА ЭТА КНИГА

Эта книга для всех, кто работает с Linux- или Unix-подобными серверами. Если вы когда-либо использовали термины *системное администрирование*, *раз-*

вертывание, управление конфигурациями или (вздых) *DevOps*, вы обязательно найдете для себя что-то полезное.

Хотя я изучал Linux-серверы, моя квалификация связана с разработкой программного обеспечения. А это значит, что все примеры в книге более тяготеют к внедрению программного обеспечения, хотя мы с Эндрю Клей Шафером (Andrew Clay Shafer, [webops]) пришли к тому, что внедрение и конфигурация не имеют четкой границы.

СТРУКТУРА КНИГИ

Я не большой фанат общепринятых принципов структурирования книг: глава 1 охватывает то-то и то-то, глава 2 охватывает это и то и тому подобное. Я подозреваю, что никто не читает этих строк (я лично – никогда), гораздо проще заглянуть в оглавление.

Книга построена так, что каждая последующая глава опирается на предыдущую. Таким образом, я предполагаю, что вы будете читать книгу от начала и до конца. Книга написана в основном в стиле учебного пособия и дает возможность выполнять примеры на вашем компьютере в процессе чтения. Большинство примеров основано на веб-приложениях.

ОБОЗНАЧЕНИЯ И СОГЛАШЕНИЯ, ПРИНЯТЫЕ В ЭТОЙ КНИГЕ

В книге действуют следующие типографские соглашения:

Курсив

Указывает на новые термины, названия файлов и их расширения.

Моноширинный шрифт

Используется для листингов программ, а также в обычном тексте для обозначения элементов программы, таких как имена переменных или функций, баз данных, типов данных, переменных окружения, инструкций и ключевых слов.

Моноширинный полужирный шрифт

Служит для выделения команд или другого текста, который должен быть набран самим пользователем.

Моноширинный курсив

Указывает на текст, который нужно заменить данными пользователя, или значениями, определяемыми контекстом.



Так обозначаются примечания общего характера.



Так обозначаются советы и рекомендации.



Так обозначаются предупреждения и предостережения.

СКАЧИВАНИЕ ИСХОДНОГО КОДА ПРИМЕРОВ

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф на странице с описанием соответствующей книги.

Мы высоко ценим, хотя и не требуем, ссылки на наши издания. В ссылке обычно указываются имя автора, название книги, издательство и ISBN, например: «*Хохштейн Л., Мозер Р.* Запускаем Ansible. М.: О’Reilly; ДМК Пресс, 2018. Copyright © 2017 O’Reilly Media, Inc., 978-1-491-97980-8 (англ.), 978-5-97060-513-4 (рус.)».

Если вы полагаете, что планируемое использование кода выходит за рамки изложенной выше лицензии, пожалуйста, обратитесь к нам по адресу dmkpress@gmail.com.

ОТЗЫВЫ И ПОЖЕЛАНИЯ

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

СПИСОК ОПЕЧАТОК

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Глава 1

Введение

Сейчас интересное время для работы в ИТ-индустрии. Мы не поставляем нашим клиентам программное обеспечение, установив его на одну-единственную машину и совершая дежурные звонки раз в день¹. Вместо этого мы медленно превращаемся в системных инженеров.

Сейчас мы устанавливаем программные приложения, связывая воедино службы, которые работают в распределенной компьютерной сети и взаимодействуют по разным сетевым протоколам. Типичное приложение может включать веб-серверы, серверы приложений, систему кэширования данных в оперативной памяти, очереди задач, очереди сообщений, базы данных SQL, системы хранения данных, NoSQL-хранилища и балансировщики нагрузки.

Мы также должны убедиться в наличии достаточного количества ресурсов, и в случае падения системы (а она будет падать) мы элегантно выйдем из ситуации. Также имеются второстепенные службы, которые нужно разворачивать и поддерживать, такие как служба журналирования, мониторинга и анализа. Имеются и внешние службы, с которыми нужно устанавливать взаимодействие, например с интерфейсами «инфраструктура как сервис» (Infrastructure-as-a-Service, IaaS) для управления экземплярами виртуальных машин².

Мы можем связать эти службы вручную: «прикрутить» нужные серверы, зайдя на каждый из них, установив пакеты приложений, отредактировав файлы конфигурации, и т. д. Но это серьезный труд. Такой процесс требует много времени, способствует появлению множества ошибок, да и просто утомляет, особенно в третий или четвертый раз. А работа вручную над более сложными задачами, как, например, установка облака OpenStack для вашего приложения, – так и просто сумасшествие. Есть способ лучше.

Если вы читаете эту книгу, значит, уже загорелись идеей управления конфигурациями и теперь рассматриваете Ansible как средство управления. Кем

¹ Да, мы согласны, никто и никогда на самом деле так не поставлял программное обеспечение.

² Рекомендую превосходные книги «The Practice of Cloud System Administration» и «Designing Data-Intensive Applications» по созданию и поддержке этих типов распределенных систем.

бы вы ни были, разработчиком или системным администратором, ищущим лучшего средства автоматизации, я думаю, вы найдете в лице Ansible превосходное решение ваших проблем.

ПРИМЕЧАНИЕ О ВЕРСИЯХ

Все примеры кода в этой книге были протестированы в версии Ansible 2.3.0.0, которая на момент написания книги являлась самой свежей. Поскольку поддержка предыдущих версий является важной целью проекта Ansible, эти примеры должны поддерживаться и последующими версиями в неизменном виде.

Откуда взялось название «Ansible»?

Название заимствовано из области научной фантастики. Ansible – это устройство связи, способное передавать информацию быстрее скорости света. Писатель Урсула Ле Гуин впервые представила эту идею в своем романе «Планета Роканнона», а остальные писатели-фантасты подхватили ее.

Если быть более точным, Майкл ДеХаан позаимствовал название Ansible из книги Орсона Скотта Карда «Игра Эндера». В этой книге Ansible использовался для одновременного контроля большого числа кораблей, удаленных на огромные расстояния. Подумайте об этом как о метафоре контроля удаленных серверов.

ANSIBLE: ОБЛАСТЬ ПРИМЕНЕНИЯ

Систему Ansible часто описывают как средство управления конфигурациями, и обычно она упоминается в том же контексте, что и *Chef*, *Puppet* и *Salt*. Когда мы говорим об управлении конфигурациями, то часто подразумеваем некое описательное состояние серверов, а затем фиксацию их реального состояния с использованием специальных средств: необходимые пакеты приложений установлены, файлы конфигурации содержат ожидаемые значения и имеют требуемые разрешения в файловой системе, необходимые службы работают и т. д. Подобно другим средствам управления, Ansible предоставляет предметно-ориентированный язык (Domain Specific Language, DSL), который используется для описания состояний серверов.

Эти инструменты также можно использовать для *развертывания* программного обеспечения. Под развертыванием мы часто подразумеваем процесс получения двоичного кода из исходного (если необходимо), копирования необходимых файлов на сервер(ы) и запуск служб. *Capistrano* и *Fabric* – два примера инструментов с открытым кодом для развертывания приложений. Ansible тоже является превосходным инструментом как для развертывания, так и для управления конфигурациями программного обеспечения. Использование единой системы управления конфигурациями и развертыванием значительно упрощает жизнь системным администраторам.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)