

# Оглавление

<b>Предисловие .....</b>	<b>18</b>
П.1. Обзор методологии непрерывного развертывания и книги.....	18
П.2. Зачем нужно непрерывное развертывание? .....	19
П.2.1. Короткая история .....	19
П.2.2. Непрерывное развертывание способно помочь в таких ситуациях.....	21
П.3. Кому адресована эта книга? .....	23
П.4. Краткое содержание глав.....	23
П.5. Как читать эту книгу .....	25
<b>Благодарности .....</b>	<b>28</b>
<b>Об авторе.....</b>	<b>29</b>
<b>От издательства .....</b>	<b>30</b>
<b>Часть I. Основы.....</b>	<b>31</b>
<b>Глава 1. Непрерывное развертывание: что и как?.....</b>	<b>32</b>
1.1. Введение: что такое непрерывное развертывание? .....	32
1.2. Почему процесс выпуска ПО настолько сложен .....	32
1.2.1. Непрерывная интеграция вселяет надежду.....	33
1.2.2. Медленные и опасные процессы .....	33
1.2.3. Есть возможность ускорения.....	33
1.3. Ценность непрерывного развертывания .....	34
1.3.1. Регулярность.....	35
1.3.2. Контролируемость.....	35
1.3.3. Регрессии.....	36

1.4. Преимущества непрерывного развертывания .....	37
1.4.1. Непрерывное развертывание сокращает сроки внедрения .....	37
1.4.2. Один пример .....	37
1.4.3. Реализация услуги и ее внедрение в эксплуатацию.....	38
1.4.4. Следующий шаг .....	38
1.4.5. Непрерывное развертывание дает конкурентные преимущества .....	38
1.4.6. Без непрерывного развертывания .....	38
1.4.7. Непрерывное развертывание и Lean Startup .....	39
1.4.8. Влияние непрерывного развертывания .....	40
1.4.9. Непрерывное развертывание для минимизации рисков.....	41
1.4.10. Быстрая обратная связь и низкие затраты.....	45
1.5. Создание и организация конвейера непрерывного развертывания .....	46
1.5.1. Пример .....	49
1.6. В заключение .....	50
Ссылки .....	50
<b>Глава 2. Подготовка инфраструктуры .....</b>	<b>52</b>
2.1. Введение .....	52
2.1.1. Автоматизация инфраструктуры: пример .....	53
2.2. Сценарии установки .....	54
2.2.1. Проблемы классических сценариев установки.....	54
2.3. Chef .....	57
2.3.1. Chef и Puppet .....	59
2.3.2. Другие альтернативы .....	60
2.3.3. Технические основы.....	61
Основные термины Chef .....	62
Повара, поваренные книги и рецепты.....	63
Роли .....	66
2.3.4. Chef Solo .....	68
2.3.5. Chef Solo: заключение.....	70
2.3.6. Knife и Chef Server .....	70
2.3.7. Chef Server: заключение.....	75

2.4. Vagrant .....	76
2.4.1. Пример с Chef и Vagrant .....	78
2.4.2. Vagrant: заключение .....	80
2.5. Docker .....	80
2.5.1. Решение на основе Docker .....	81
Контейнеры Docker и виртуализация .....	83
Цели Docker .....	83
Взаимодействия между контейнерами Docker .....	83
2.5.2. Создание контейнеров Docker .....	84
Файлы Dockerfile .....	85
Сборка и запуск образов Docker .....	86
2.5.3. Запуск примера приложения с помощью Docker .....	87
Дополнительные команды Docker .....	88
2.5.4. Docker и Vagrant .....	89
Комплектация контейнеров с помощью Vagrant .....	89
2.5.5. Docker Machine .....	92
2.5.6. Сложные конфигурации Docker .....	94
Docker Registry .....	94
Docker в кластере .....	95
2.5.7. Docker Compose .....	96
2.6. Неизменяемый сервер .....	99
2.6.1. Недостатки идемпотентности .....	99
2.6.2. Неизменяемый сервер и Docker .....	100
2.7. Инфраструктура как код .....	101
2.7.1. Тестирование инфраструктуры как кода .....	103
Serverspec .....	103
Test Kitchen .....	103
ChefSpec .....	103
2.8. Платформа как услуга .....	104
2.9. Хранение информации и базы данных .....	106
2.9.1. Обработка схем .....	107
2.9.2. Тестовые и базовые данные .....	108
2.10. В заключение .....	109
Ссылки .....	110

---

<b>Часть II. Конвейер непрерывного развертывания .....</b>	<b>113</b>
<b>Глава 3. Автоматизация сборки и непрерывная интеграция .....</b>	<b>114</b>
3.1. Введение .....	114
3.1.1. Автоматизация сборки: пример .....	115
3.2. Автоматизация сборки и инструменты сборки .....	115
3.2.1. Инструменты сборки в мире Java .....	116
3.2.2. Ant .....	117
3.2.3. Maven .....	118
Версии и мгновенные снимки .....	121
Выпуск версий с помощью Maven .....	122
3.2.4. Gradle .....	123
Gradle Wrapper .....	125
3.2.5. Другие инструменты сборки .....	126
3.2.6. Выбор правильного инструмента .....	127
3.3. Модульные тесты .....	129
3.3.1. Создание хороших модульных тестов .....	130
3.3.2. Разработка через тестирование .....	133
3.3.3. Чистый код и искусство программирования .....	134
3.4. Непрерывная интеграция .....	135
3.4.1. Jenkins .....	136
Расширение с помощью плагинов .....	138
Плагин SCM Sync Configuration .....	138
Плагин Environment Injector .....	138
Плагин Job Configuration History .....	139
Плагин Clone Workspace SCM .....	139
Плагин Build Pipeline .....	139
Плагин Amazon EC2 .....	140
Плагин Job DSL .....	140
Создание собственных плагинов .....	142
3.4.2. Инфраструктура непрерывной интеграции .....	142
3.4.3. В заключение .....	144
3.5. Оценка качества кода .....	147
3.5.1. SonarQube .....	149
Интеграция в конвейер .....	149

3.6. Управление артефактами .....	152
3.6.1. Интеграция в процедуру сборки.....	154
3.6.2. Дополнительные особенности репозитория.....	156
3.7. В заключение.....	157
Ссылки .....	158
<b>Глава 4. Приемочные тесты .....</b>	<b>161</b>
4.1. Введение .....	161
4.1.1. Приемочные испытания: пример .....	161
4.2. Пирамида тестирования.....	162
4.3. Что такое «приемочные тесты»?.....	166
4.3.1. Автоматизированные приемочные испытания .....	166
4.3.2. Больше, чем просто увеличение эффективности .....	167
4.3.3. Тестирование вручную.....	168
4.3.4. Какую роль играет заказчик? .....	168
4.3.5. Приемочные и модульные тесты .....	169
4.3.6. Окружения для тестирования.....	170
4.4. Приемочные тесты через ГИП.....	171
4.4.1. Проблемы тестирования графического интерфейса .....	171
4.4.2. Абстракции против хрупких тестов через графический интерфейс.....	172
4.4.3. Автоматизация с помощью Selenium.....	173
4.4.4. Программный интерфейс веб-драйвера.....	173
4.4.5. Тестирование без веб-браузера: HtmlUnit .....	173
4.4.6. Программный интерфейс веб-драйвера Selenium .....	174
4.4.7. Selenium IDE.....	174
4.4.8. Проблемы автоматизации тестов через графический интерфейс.....	176
4.4.9. Выполнение тестов графического интерфейса .....	176
4.4.10. Преобразование тестов в программный код .....	176
4.4.11. Изменение тестов вручную.....	177
4.4.12. Тестовые данные.....	177
4.4.13. Шаблон Page Object .....	178

---

4.5. Альтернативные инструменты тестирования графического интерфейса .....	179
4.5.1. PhantomJS .....	179
4.5.2. Windmill .....	179
4.6. Текстовые приемочные тесты .....	181
4.6.1. Behavior-Driven Development .....	181
4.6.2. Адаптеры .....	184
4.7. Альтернативные фреймворки .....	186
4.8. Стратегии приемочных испытаний .....	187
4.8.1. Выбор инструмента .....	187
4.8.2. Быстрая обратная связь .....	188
4.8.3. Охват тестами .....	189
4.9. В заключение .....	189
Ссылки .....	191
<b>Глава 5. Тестирование пропускной способности .....</b>	<b>192</b>
5.1. Введение .....	192
5.1.1. Тестирование пропускной способности: пример .....	192
5.2. Тестирование пропускной способности — как? .....	193
5.2.1. Цели тестирования пропускной способности .....	193
5.2.2. Данные и окружения .....	194
5.2.3. Тестирование производительности должно выполняться только в конце реализации? .....	194
5.2.4. Тесты пропускной способности = управление рисками .....	194
5.2.5. Имитация поведения пользователей .....	195
5.2.6. Документирование требований к производительности .....	195
5.2.7. Аппаратное обеспечение для тестирования пропускной способности .....	196
5.2.8. Облачные решения и виртуализация .....	197
5.2.9. Минимизация рисков за счет непрерывного тестирования .....	198
5.2.10. Тестирование пропускной способности — целесообразно ли? .....	198
5.3. Реализация тестов пропускной способности .....	199
5.4. Тестирование пропускной способности с помощью Gatling .....	201
5.4.1. Различия между демонстрацией и практическим применением .....	205

5.5. Альтернативы инструменту Gatling .....	207
5.5.1. Grinder .....	207
5.5.2. Apache JMeter.....	207
5.5.3. Tsung .....	208
5.5.4. Коммерческие решения.....	208
5.6. В заключение .....	209
Ссылки .....	209
<b>Глава 6. Исследовательское тестирование .....</b>	<b>210</b>
6.1. Введение .....	210
6.1.1. Исследовательское тестирование: пример .....	210
6.2. Цель исследовательского тестирования.....	210
6.2.1. Иногда ручное тестирование оказывается предпочтительнее ...	211
6.2.2. Тестирование заказчиком .....	211
6.2.3. Ручное тестирование нефункциональных требований .....	212
6.3. Как это сделать?.....	212
6.3.1. Руководство по проведению тестирования.....	213
6.3.2. Автоматизированное окружение .....	213
6.3.3. Демонстрационные примеры как основа .....	213
6.3.4. Пример: приложение электронной коммерции .....	214
6.3.5. Бета-тестирование .....	214
6.3.6. Сеансовые тесты .....	215
6.4. В заключение .....	218
Ссылки .....	218
<b>Глава 7. Развертывание — ввод в эксплуатацию.....</b>	<b>219</b>
7.1. Введение .....	219
7.1.1. Развертывание: пример.....	220
7.2. Ввод в эксплуатацию и откат .....	220
7.2.1. Преимущества.....	220
7.2.2. Недостатки.....	221
7.3. Развертывание исправлений.....	222
7.3.1. Преимущества.....	222
7.3.2. Недостатки.....	222

7.4. Сине-зеленое развертывание.....	223
7.4.1. Преимущества.....	224
7.4.2. Недостатки.....	224
7.5. Канареечное развертывание.....	225
7.5.1. Преимущества.....	226
7.5.2. Недостатки.....	226
7.6. Автоматическое развертывание.....	227
7.6.1. Преимущества.....	228
7.6.2. Недостатки.....	229
7.7. Виртуализация.....	229
7.7.1. Физические хосты.....	231
7.8. Вне круга веб-приложений.....	231
7.9. В заключение.....	233
Ссылки.....	233
<b>Глава 8. Эксплуатация.....</b>	<b>234</b>
8.1. Введение.....	234
8.1.1. Эксплуатация — пример.....	235
8.2. Проблемы в период эксплуатации.....	235
8.3. Файлы журналов.....	237
8.3.1. Что следует журналировать?.....	238
8.3.2. Инструменты для обработки файлов журналов.....	240
8.3.3. Журналирование в примере приложения.....	241
8.4. Анализ журналов примера приложения.....	242
8.4.1. Анализ с применением Kibana.....	245
8.4.2. ELK — масштабируемость.....	246
8.5. Другие технологии обработки журналов.....	251
8.6. Продвинутое технологии обработки журналов.....	252
8.6.1. Анонимизация.....	253
8.6.2. Производительность.....	253
8.6.3. Время.....	253
8.6.4. Эксплуатационная база данных.....	254
8.7. Мониторинг.....	254



8.8. Отображение числовых характеристик с помощью Graphite.....	255
8.9. Характеристики, измеряемые в примере приложения.....	258
8.9.1. Структура примера.....	258
8.10. Другие решения для мониторинга.....	260
8.11. Прочие проблемы, возникающие во время эксплуатации приложений.....	262
8.11.1. Сценарии.....	262
8.11.2. Приложения в вычислительном центре клиента.....	262
8.12. В заключение.....	263
Ссылки.....	264

### **Часть III. Управление, организация и архитектура решения непрерывного развертывания..... 267**

<b>Глава 9.</b> Внедрение методологии непрерывного развертывания на предприятии.....	268
9.1. Введение.....	268
9.2. Непрерывное развертывание с самого начала.....	268
9.3. Систематизация потока ценностей.....	269
9.3.1. Систематизация потока ценностей описывает последовательность событий.....	270
9.3.2. Оптимизация.....	271
9.4. Дополнительные меры для оптимизации.....	273
9.4.1. Капиталовложения в качество.....	273
9.4.2. Затраты.....	273
9.4.3. Выгоды.....	274
9.4.4. Запрещайте сохранение изменений в случае ошибки сборки! ...	274
9.4.5. Останови конвейер.....	275
9.4.6. Пять почему.....	275
9.4.7. DevOps.....	276
9.5. В заключение.....	278
Ссылки.....	278

### **Глава 10.** Непрерывное развертывание и DevOps..... 279

10.1. Введение.....	279
---------------------	-----

---

10.2. Что такое DevOps? .....	279
10.2.1. Проблемы .....	280
10.2.2. Точка зрения клиента .....	281
10.2.3. Первопроходцы: Amazon .....	281
10.2.4. DevOps .....	282
10.3. Непрерывное развертывание и DevOps .....	283
10.3.1. DevOps: не только непрерывное развертывание .....	284
10.3.2. Индивидуальная ответственность и самоорганизация .....	285
10.3.3. Технологические решения .....	286
10.3.4. Меньше централизованного управления .....	286
10.3.5. Плюрализм технологий .....	287
10.3.6. Обмен специалистами между командами .....	287
10.3.7. Архитектура .....	288
10.4. Непрерывное развертывание без DevOps? .....	289
10.4.1. Завершение конвейера непрерывного развертывания .....	290
10.5. В заключение .....	292
Ссылки .....	292
<b>Глава 11. Непрерывное развертывание, DevOps и архитектура ПО .....</b>	<b>293</b>
11.1. Введение .....	293
11.2. Архитектура программного обеспечения .....	293
11.2.1. Зачем нужна архитектура программного обеспечения? .....	294
11.3. Оптимизация архитектуры для непрерывного развертывания .....	296
11.3.1. Деление на мелкие единицы развертывания .....	297
11.4. Интерфейсы .....	298
11.4.1. Закон Постела, или принцип надежности .....	300
11.4.2. Страховка от сбоев .....	300
11.4.3. Состояние .....	301
11.5. Базы данных .....	302
11.5.1. Поддержание стабильности схемы базы данных .....	303
11.5.2. База данных = компонент .....	304
11.5.3. Представления и хранимые процедуры .....	304
11.5.4. Отдельная база данных для каждого компонента .....	305
11.5.5. Базы данных NoSQL .....	305

11.6. Микрослужбы .....	305
11.6.1. Микрослужбы и непрерывное развертывание.....	306
11.6.2. Внедрение непрерывного развертывания с микрослужбами ....	307
11.6.3. Микрослужбы способствуют внедрению непрерывного развертывания .....	307
11.6.4. Организация .....	308
11.7. Внедрение новых возможностей.....	308
11.7.1. Отдельные ветви для новых возможностей .....	309
11.7.2. Переключение возможностей .....	309
11.7.3. Преимущества .....	310
11.7.4. Примеры использования переключения возможностей.....	311
11.7.5. Недостатки .....	312
11.8. В заключение .....	312
Ссылки .....	313
<b>Глава 12. Заключение: основные преимущества .....</b>	<b>315</b>
Ссылки .....	316