

УДК 004.42
ББК 32.972
Д94

Дэвис К.
Д94 Шаблоны проектирования для облачной среды / пер. с англ. Д. А. Беликова. – М.: ДМК Пресс, 2020. – 388 с.: ил.

ISBN 978-5-97060-807-4

Эта книга посвящена облачным платформам, которые обеспечивают многие преимущества – практически нулевое время простоя, бесконечную масштабируемость, короткие циклы обратной связи, отказоустойчивость и контроль затрат. Применяя конструкции для облачной среды, разработчики могут создавать гибкие, легко адаптируемые, веб-масштабируемые распределенные приложения, которые обрабатывают огромный пользовательский трафик и объем данных.

Автор рассматривает методы и шаблоны, ориентированные на приложения для облачной среды – с учетом их жизненного цикла, управления конфигурацией в масштабах облака, обновления без простоев.

Для работы с книгой читателю необходимы базовые навыки в области проектирования программного обеспечения и умение понимать код, написанный на Java или похожем языке.

Издание будет полезно всем, кого интересует развертывание систем на различных облачных платформах.

УДК 004.42
ББК 32.972

Original English language edition published by Manning Publications USA, USA. Copyright © 2019 by Manning Publications Co. Russian-language edition copyright © 2020 by ДМК Пресс. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-617-29532-4 (англ.)
ISBN 978-5-97060-807-4 (рус.)

Copyright © 2019 by Manning Publications Co
© Оформление, издание, перевод, ДМК Пресс, 2020

Содержание

Предисловие	11
От автора	12
Благодарности	14
Об этой книге	16
Об авторе	20
Об иллюстрации на обложке	21
Часть I. Контекст облачной среды	22
Глава 1. Вы продолжаете использовать это слово: определение понятия «cloud-native»	23
1.1. Современные требования к приложениям	27
1.1.1. Нулевое время простоя.....	27
1.1.2. Сокращенные контуры обратной связи	28
1.1.3. Мобильная и мультидевайсная поддержка.....	28
1.1.4. Устройства, подключенные к сети, также известные как интернет вещей.....	29
1.1.5. Управление с помощью данных.....	29
1.2. Знакомство с программным обеспечением для облачной среды	30
1.2.1. Определение понятия «cloud-native»	31
1.2.2. Ментальная модель программного обеспечения для облачной среды ...	33
1.2.3. Программное обеспечение для облачной среды в действии.....	38
1.3. Cloud-native и мир во всем мире	43
1.3.1. Cloud и cloud-native	43
1.3.2. Что не относится к понятию «cloud-native»?	44
1.3.3. Облачная среда нам подходит.....	45
Резюме.....	48
Глава 2. Запуск облачных приложений в рабочем окружении	49
2.1. Препятствия	50
2.1.1. Снежинки.....	51
2.1.2. Рискованное развертывание	53
2.1.3. Изменение – это исключение.....	57
2.1.4. Нестабильность рабочего окружения	57
2.2. Стимулирующие факторы	58
2.2.1. Непрерывная доставка	59
2.2.2. Повторяемость	63
2.2.3. Безопасное развертывание	68
2.2.4. Изменение – это правило	72
Резюме.....	75

Глава 3. Платформа для облачного ПО	76
3.1. Эволюция облачных платформ	77
3.1.1. Все началось с облака	77
3.1.2. Тональный вызов	79
3.2. Основные принципы платформы для облачной среды	82
3.2.1. Вначале поговорим о контейнерах	82
3.2.2. Поддержка «постоянно меняющихся»	84
3.2.3. Поддержка «сильно распределенных»	87
3.3. Кто что делает?	91
3.4. Дополнительные возможности платформы для облачной среды	94
3.4.1. Платформа поддерживает весь жизненный цикл разработки программного обеспечения	94
3.4.2. Безопасность, контроль над изменениями, соответствие требованиям (функции управления)	97
3.4.3. Контроль за тем, что идет в контейнер	100
3.4.4. Обновление и исправление уязвимостей	102
3.4.5. Контроль над изменениями	104
Резюме	106
Часть II. Шаблоны для облачной среды	107
Глава 4. Событийно-ориентированные микросервисы: не только запрос/ответ	109
4.1. (Обычно) нас учат императивному программированию	110
4.2. Повторное знакомство с событийно-ориентированными вычислениями	112
4.3. Моя глобальная поваренная книга	113
4.3.1. Запрос/ответ	113
4.3.2. Событийно-ориентированный подход	119
4.4. Знакомство с шаблоном Command Query Responsibility Segregation	129
4.5. Разные стили, схожие проблемы	131
Резюме	133
Глава 5. Избыточность приложения: горизонтальное масштабирование и отсутствие фиксации состояния	134
5.1. У приложений для облачной среды есть много развернутых экземпляров	136
5.2. Приложения с фиксацией текущего состояния в облаке	137
5.2.1. Разложение монолита на части и привязка к базе данных	139
5.2.2. Плохая обработка состояния сеанса	142
5.3. HTTP-сессии и «липкие» сессии	155
5.4. Службы с фиксацией текущего состояния и приложения без фиксации состояния	158
5.4.1. Службы с фиксацией состояния – это специальные службы	158
5.4.2. Создание приложений без сохранения состояния	160
Резюме	165

Глава 6. Конфигурация приложения: не только переменные среды.....	166
6.1. Почему мы вообще говорим о конфигурации?	167
6.1.1. Динамическое масштабирование – увеличение и уменьшение количества экземпляров приложения	168
6.1.2. Изменения инфраструктуры, вызывающие изменения в конфигурации	168
6.1.3. Обновление конфигурации приложения с нулевым временем простоя	169
6.2. Уровень конфигурации приложения	171
6.3. Инъекция значений системы/среды	176
6.3.1. Давайте посмотрим, как это работает: использование переменных среды для конфигурации	176
6.4. Внедрение конфигурации приложения	184
6.4.1. Знакомство с сервером конфигурации.....	185
6.4.2. Безопасность добавляет больше требований.....	193
6.4.3. Давайте посмотрим, как это работает: конфигурация приложения с использованием сервера конфигурации.....	193
Резюме	195
Глава 7. Жизненный цикл приложения: учет постоянных изменений	197
7.1. Сочувствие к операциям.....	199
7.2. Жизненный цикл одного приложения и жизненные циклы нескольких приложений	200
7.2.1. Сине-зеленые обновления.....	203
7.2.2. Последовательные обновления	205
7.2.3. Параллельное развертывание	205
7.3. Координация между различными жизненными циклами приложения	209
7.4. Давайте посмотрим, как это работает: периодическая смена реквизитов доступа и жизненный цикл приложения	212
7.5. Работа с эфемерной средой выполнения	221
7.6. Видимость состояния жизненного цикла приложения	223
7.6.1. Давайте посмотрим, как это работает: конечные точки работоспособности и проверки.....	228
7.7. Внесерверная обработка данных.....	231
Резюме	234
Глава 8. Доступ к приложениям: сервисы, маршрутизация и обнаружение сервисов.....	235
8.1. Сервисная абстракция	238
8.1.1. Пример сервиса: поиск в Google	239
8.1.2. Пример сервиса: наш агрегатор блогов.....	240
8.2. Динамическая маршрутизация	242
8.2.1. Балансировка нагрузки на стороне сервера.....	242
8.2.2. Балансировка нагрузки на стороне клиента	243
8.2.3. Свежесть маршрутов	244
8.3. Обнаружение служб	247

8.3.1. Обнаружение служб в сети	250
8.3.2. Обнаружение сервисов с балансировкой нагрузки на стороне клиента	251
8.3.3. Обнаружение сервисов в Kubernetes	253
8.3.4. Давайте посмотрим, как это работает: использование обнаружения сервисов	255
Резюме	258

Глава 9. Избыточность взаимодействия: повторная отправка запроса и другие циклы управления.....

9.1. Повторная отправка запроса.....	261
9.1.1. Основной шаблон.....	261
9.1.2. Давайте посмотрим, как это работает: простая повторная отправка запроса	262
9.1.3. Повторная отправка запроса: что может пойти не так?	266
9.1.4. Создание шквала повторных отправок запроса	267
9.1.5. Давайте посмотрим, как это работает: создание шквала п овторных отправок запроса.....	268
9.1.6. Как избежать шквала повторных отправок запросов: добрые клиенты	278
9.1.7. Давайте посмотрим, как это работает: стать более доброжелательным клиентом	279
9.1.8. Когда не нужно использовать повторную отставку запроса.....	284
9.2. Альтернативная логика	285
9.2.1. Давайте посмотрим, как это работает: реализация альтернативной логики.....	286
9.3. Циклы управления	291
9.3.1. Типы циклов управления	292
9.3.2. Контроль над циклом управления	293
Резюме	295

Глава 10. Лицом к лицу с сервисами: предохранители и API-шлюзы....

10.1. Предохранители	297
10.1.1. Предохранитель для программного обеспечения	298
10.1.2. Реализация предохранителя	299
10.2. API-шлюзы.....	312
10.2.1. API-шлюзы в программном обеспечении для облачной среды	314
10.2.2. Топология шлюза API.....	316
10.3. Сервисная сеть.....	318
10.3.1. Сайдкар	318
10.3.2. Уровень управления	320
Резюме	323

Глава 11. Поиск и устранение неполадок: найти иголку в стоге сена....

11.1. Ведение журналов приложений	325
11.2. Метрики приложений	329

11.2.1. Извлечение метрик	330
11.2.2. Размещение метрик	333
11.3. Распределенная трассировка	336
11.3.1. Вывод трассировщика	339
11.3.2. Компоновка трассировок с помощью Zipkin	342
11.3.3. Детали реализации	346
Резюме	347

Глава 12. Данные в облачной среде: разбиение

монолитных данных	349
12.1. Каждому микросервису нужен кеш	352
12.2. Переход от протокола «запрос/ответ» к событийно-ориентированному подходу	355
12.3. Журнал событий	357
12.3.1. Давайте посмотрим, как это работает: реализация событийно-ориентированных микросервисов	359
12.3.2. Что нового в темах и очередях?	372
12.3.3. Полезные данные события	375
12.3.4. Идемпотентность	377
12.4. Порождение событий	378
12.4.1. Путешествие еще не окончено	378
12.4.2. Источник истины	380
12.4.3. Давайте посмотрим, как это работает: реализация порождения событий	382
12.5. Это лишь поверхностное знакомство	385
Резюме	385
Предметный указатель	387