

*Я посвящаю эту книгу всем женщинам,
которые делают карьеру в естественно-научных,
инженерно-технических и математических дисциплинах.*

Нет ничего невозможного, если упорно трудиться.

Краткое оглавление

1	■ Добро пожаловать в Spring Cloud	27
2	■ Обзор мира микросервисов через призму Spring Cloud ..	63
3	■ Создание микросервисов с использованием Spring Boot ...	91
4	■ Добро пожаловать в Docker.....	129
5	■ Управление конфигурациями с использованием Spring Cloud Configuration Server.....	150
6	■ Обнаружение служб.....	191
7	■ Когда случаются неприятности: шаблоны устойчивости с использованием Spring Cloud и Resilience4j.....	223
8	■ Маршрутизация служб с использованием Spring Cloud Gateway	259
9	■ Безопасность микросервисов	294
10	■ Событийно-ориентированная архитектура и Spring Cloud Stream	332
11	■ Распределенная трассировка с использованием Spring Cloud Sleuth и Zipkin	367
12	■ Развертывание микросервисов	404

Содержание

<i>Предисловие от издательства</i>	16
<i>Предисловие</i>	17
<i>Благодарности</i>	19
<i>Об этой книге</i>	21
1 <i>Добро пожаловать в Spring Cloud</i>	27
1.1. Эволюция архитектуры микросервисов	28
1.1.1. N-уровневая архитектура	28
1.1.2. Что такое монолитная архитектура?.....	29
1.1.3. Что такое микросервис?.....	30
1.1.4. Зачем менять способ создания приложений?.....	32
1.2. Микросервисы со Spring.....	34
1.3. Что мы будем создавать?	36
1.4. О чем эта книга?	37
1.4.1. Что вы узнаете в этой книге	37
1.4.2. Почему эта книга актуальна для вас?	38
1.5. Облачные приложения и приложения на основе микросервисов	39
1.5.1. Создание микросервиса с помощью Spring Boot	39
1.5.2. Что такое облачные вычисления?.....	44
1.5.3. В чем преимущества облачных вычислений и микросервисов?.....	46
1.6. Микросервисы – это больше чем код	49
1.7. Базовый шаблон разработки микросервисов	50
1.8. Шаблоны маршрутизации	52
1.9. Устойчивость клиентов.....	54

1.10. Шаблоны безопасности	55
1.11. Шаблоны журналирования и трассировки	56
1.12. Шаблон сбора метрик приложения.....	58
1.13. Шаблоны сборки/развертывания микросервисов	59
Итоги	61
2 <i>Обзор мира микросервисов через призму Spring Cloud</i>	63
2.1. Что такое Spring Cloud?	64
2.1.1. <i>Spring Cloud Config</i>	65
2.1.2. <i>Spring Cloud Service Discovery</i>	66
2.1.3. <i>Spring Cloud LoadBalancer и Resilience4j</i>	66
2.1.4. <i>Spring Cloud API Gateway</i>	67
2.1.5. <i>Spring Cloud Stream</i>	67
2.1.6. <i>Spring Cloud Sleuth</i>	67
2.1.7. <i>Spring Cloud Security</i>	68
2.2. Пример использования Spring Cloud	68
2.3. Приемы создания облачных микросервисов	71
2.3.1. <i>База кода</i>	74
2.3.2. <i>Зависимости</i>	75
2.3.3. <i>Конфигурация</i>	76
2.3.4. <i>Вспомогательные службы</i>	76
2.3.5. <i>Сборка, выпуск, выполнение</i>	77
2.3.6. <i>Процессы</i>	78
2.3.7. <i>Привязка портов</i>	78
2.3.8. <i>Масштабируемость</i>	79
2.3.9. <i>Одноразовость</i>	79
2.3.10. <i>Сходство окружений разработки/эксплуатации</i>	80
2.3.11. <i>Журналирование</i>	80
2.3.12. <i>Задачи администрирования</i>	81
2.4. Актуальность наших примеров.....	81
2.5. Создание микросервиса с использованием Spring Boot и Java	82
2.5.1. <i>Подготовка окружения</i>	83
2.5.2. <i>Начало создания проекта</i>	83
2.5.3. <i>Запуск приложения Spring Boot: класс инициализации</i>	88
Итоги	90
3 <i>Создание микросервисов с использованием Spring Boot</i>	91
3.1. Точка зрения архитектора: проектирование микросервисной архитектуры.....	92
3.1.1. <i>Декомпозиция бизнес-задачи</i>	92

3.1.2. Детализация служб	95
3.1.3. Определение интерфейсов служб	98
3.2. Когда не следует использовать микросервисы	99
3.2.1. Сложность распределенных систем	99
3.2.2. Беспорядочный рост виртуальных серверов или контейнеров	99
3.2.3. Тип приложения	100
3.2.4. Транзакции и согласованность данных	100
3.3. Точка зрения разработчика: создание микросервиса с использованием Spring Boot и Java	100
3.3.1. Встраивание дверного проема в микросервис: контроллер Spring Boot	101
3.3.2. Добавление интернационализации в службу лицензий	112
3.3.3. Реализация Spring HATEOAS для отображения связанных ссылок	115
3.4. Точка зрения инженера DevOps: сборка выполняемых артефактов	118
3.4.1. Сборка службы: упаковка и развертывание микросервисов	120
3.4.2. Инициализация службы: управление конфигурацией микросервисов	122
3.4.3. Регистрация и обнаружение службы: взаимодействие клиентов с микросервисами	123
3.4.4. Мониторинг состояния микросервиса	124
3.5. Объединение точек зрения	127
Итоги	128
4 Добро пожаловать в Docker	129
4.1. Контейнеры или виртуальные машины?	130
4.2. Что такое Docker?	132
4.3. Файлы Dockerfile	135
4.4. Docker Compose	136
4.5. Интеграция Docker с микросервисами	138
4.5.1. Создание образа Docker	138
4.5.2. Создание образов Docker со Spring Boot	144
4.5.3. Запуск служб с помощью Docker Compose	147
Итоги	148
5 Управление конфигурациями с использованием Spring Cloud Configuration Server	150
5.1. Об управлении конфигурациями (и сложностью)	151
5.1.1. Архитектура управления конфигурацией	152

5.1.2. Варианты реализации.....	154
5.2. Настройка Spring Cloud Configuration Server	156
5.2.1. Настройка класса инициализации <i>Spring Cloud Config</i> ...	161
5.2.2. Использование <i>Spring Cloud Config Server</i> с файловой системой	161
5.2.3. Создание конфигурационных файлов для службы	163
5.3. Интеграция Spring Cloud Config с клиентом Spring Boot.....	168
5.3.1. Настройка зависимостей <i>Spring Cloud Config Service</i> в службе лицензий.....	170
5.3.2. Настройка службы лицензий для взаимодействий с <i>Spring Cloud Config</i>	170
5.3.3. Подключение к источнику данных с использованием <i>Spring Cloud Config Server</i>	175
5.3.4. Чтение настроек с использованием <i>@ConfigurationProperties</i>	179
5.3.5. Обновление настроек с использованием <i>Spring Cloud Config Server</i>	180
5.3.6. Использование <i>Spring Cloud Configuration Server</i> с <i>Git</i>	182
5.3.7. Интеграция <i>Vault</i> со службой <i>Spring Cloud Config</i>	183
5.3.8. Пользовательский интерфейс <i>Vault</i>	184
5.4. Защита конфиденциальных настроек в конфигурации	187
5.4.1. Настройка симметричного шифрования.....	187
5.4.2. Шифрование и дешифрование настроек.....	188
5.5. Заключительные мысли	190
Итоги	190
6 Обнаружение служб	191
6.1. Где моя служба?	193
6.2. Обнаружение служб в облаке	195
6.2.1. Архитектура механизма обнаружения служб.....	196
6.2.2. Обнаружение служб с использованием <i>Spring</i> и <i>Netflix Eureka</i>	200
6.3. Создание службы Spring Eureka.....	202
6.4. Регистрация служб в Spring Eureka	207
6.4.1. REST API <i>Eureka</i>	211
6.4.2. Панель управления <i>Eureka</i>	212
6.5. Использование механизма обнаружения служб	214
6.5.1. Поиск экземпляров служб с <i>Spring Discovery Client</i>	216
6.5.2. Вызов служб с использованием шаблона <i>Spring REST</i> с поддержкой <i>Load Balancer</i>	218
6.5.3. Вызов служб с использованием <i>Netflix Feign</i>	220
Итоги	222

7	<i>Когда случаются неприятности: шаблоны устойчивости с использованием Spring Cloud и Resilience4j</i>	223
7.1.	Шаблоны устойчивости на стороне клиента	225
7.1.1.	<i>Балансировка нагрузки на стороне клиента</i>	226
7.1.2.	<i>Размыкатель цепи</i>	226
7.1.3.	<i>Резервная реализация</i>	227
7.1.4.	<i>Герметичные отсеки</i>	227
7.2.	Почему устойчивость клиента важна	228
7.3.	Реализация с Resilience4j	232
7.4.	Подготовка службы лицензий к использованию Spring Cloud и Resilience4j	233
7.5.	Реализация размыкателя цепи	234
7.5.1.	<i>Добавление размыкателя цепи для обработки вызовов службы организаций</i>	240
7.5.2.	<i>Настройка размыкателя цепи</i>	240
7.6.	Использование резервной реализации	241
7.7.	Реализация шаблона герметичных отсеков	244
7.8.	Реализация шаблона повторных попыток	248
7.9.	Реализация шаблона ограничителя частоты	249
7.10.	ThreadLocal и Resilience4j	252
	Итоги	257
8	<i>Маршрутизация служб с использованием Spring Cloud Gateway</i>	259
8.1.	Что такое сервисный шлюз?	260
8.2.	Введение в Spring Cloud Gateway	263
8.2.1.	<i>Настройка проекта шлюза Spring Boot</i>	264
8.2.2.	<i>Настройка Spring Cloud Gateway для взаимодействий с Eureka</i>	266
8.3.	Настройка маршрутов в Spring Cloud Gateway	268
8.3.1.	<i>Автоматическое отображение маршрутов с помощью механизма обнаружения служб</i>	268
8.3.2.	<i>Отображение маршрутов вручную с помощью механизма обнаружения служб</i>	270
8.3.3.	<i>Динамическая загрузка настроек маршрутизации</i>	273
8.4.	Настоящая мощь Spring Cloud Gateway: фабрики предикатов и фильтров	274
8.4.1.	<i>Встроенные фабрики предикатов</i>	275
8.4.2.	<i>Встроенные фабрики фильтров</i>	276
8.4.3.	<i>Добавление своих фильтров</i>	278

8.5. Создание предварительного фильтра	281
8.6. Использование идентификатора корреляции в службах...	284
8.6.1. <i>UserContextFilter: перехват входящих HTTP-запросов</i>	286
8.6.2. <i>UserContext: обеспечение доступности HTTP-заголовков в службах</i>	287
8.6.3. <i>RestTemplate и UserContextInterceptor: обеспечение передачи идентификатора корреляции нижестоящим службам</i>	289
8.7. Создание заключительного фильтра, добавляющего идентификатор корреляции	290
Итоги	293

9

Безопасность микросервисов	294
9.1. Что такое OAuth2?	295
9.2. Введение в Keycloak	297
9.3. Начнем с малого: использование Spring и Keycloak для защиты единственной конечной точки	299
9.3.1. <i>Добавление Keycloak в Docker</i>	299
9.3.2. <i>Настройка Keycloak</i>	300
9.3.3. <i>Регистрация клиентского приложения</i>	303
9.3.4. <i>Настройка пользователей O-stock</i>	308
9.3.5. <i>Аутентификация пользователей приложения O-stock</i>	310
9.4. Защита службы организаций с использованием Keycloak....	314
9.4.1. <i>Добавление в службы JAR-файлов Spring Security и Keycloak</i>	314
9.4.2. <i>Настройка связи службы с сервером Keycloak</i>	315
9.4.3. <i>Определение пользователей, кому разрешено обращаться к службе</i>	315
9.4.4. <i>Передача токена доступа</i>	320
9.4.5. <i>Анализ нестандартного поля в JWT</i>	326
9.5. Некоторые заключительные рассуждения о безопасности микросервисов	328
9.5.1. <i>Используйте HTTPS/Secure Sockets Layer (SSL) для взаимодействий между службами</i>	329
9.5.2. <i>Используйте шлюз для организации доступа к микросервисам</i>	329
9.5.3. <i>Разделите свои службы на общедоступные и закрытые</i>	330
9.5.4. <i>Ограничьте поверхность атаки на ваши микросервисы, заблокировав ненужные сетевые порты</i>	330
Итоги	331

10	Событийно-ориентированная архитектура и Spring Cloud Stream	332
10.1.	Обмен сообщениями, событийно-ориентированная архитектура и микросервисы	333
10.1.1.	<i>Передача событий об изменении состояния с использованием синхронного подхода запрос/ответ</i>	334
10.1.2.	<i>Передача событий об изменении состояния с использованием сообщений</i>	337
10.1.3.	<i>Недостатки архитектуры на основе сообщений</i>	339
10.2.	Введение в Spring Cloud Stream.....	340
10.3.	Простые издатель и получатель сообщений	342
10.3.1.	<i>Настройка Apache Kafka и Redis в Docker</i>	343
10.3.2.	<i>Публикация сообщений в службе организаций</i>	344
10.3.3.	<i>Получение сообщений в службе лицензий</i>	351
10.3.4.	<i>Тестирование передачи сообщений между службами</i>	355
10.4.	Пример использования Spring Cloud Stream: распределенное кеширование	356
10.4.1.	<i>Использование Redis в роли кеша</i>	357
10.4.2.	<i>Определение собственных каналов</i>	363
	Итоги	366
11	Распределенная трассировка с использованием Spring Cloud Sleuth и Zipkin	367
11.1.	Spring Cloud Sleuth и идентификатор корреляции.....	369
11.1.1.	<i>Подключение Spring Cloud Sleuth к службам лицензий и организаций</i>	370
11.1.2.	<i>Особенности трассировки в Spring Cloud Sleuth</i>	370
11.2.	Агрегирование журналов и Spring Cloud Sleuth	372
11.2.1.	<i>Интеграция Spring Cloud Sleuth и стека ELK</i>	374
11.2.2.	<i>Настройка Logback в службах</i>	376
11.2.3.	<i>Определение и запуск приложений ELK в Docker</i>	380
11.2.4.	<i>Настройка Kibana</i>	383
11.2.5.	<i>Поиск идентификаторов трассировки Spring Cloud Sleuth в Kibana</i>	386
11.2.6.	<i>Добавление идентификатора корреляции в HTTP-ответ с помощью Spring Cloud Gateway</i>	388
11.3.	Распределенная трассировка с использованием Zipkin	390
11.3.1.	<i>Настройка зависимостей Spring Cloud Sleuth и Zipkin</i>	391

11.3.2. Настройка в службах ссылки на сервер Zipkin	391
11.3.3. Настройка сервера Zipkin.....	392
11.3.4. Настройка уровней трассировки.....	393
11.3.5. Использование Zipkin для трассировки транзакций	394
11.3.6. Визуализация более сложных транзакций	397
11.3.7. Трассировка операций обмена сообщениями.....	398
11.3.8. Добавление дополнительных операций.....	400
Итоги	403
12 Развертывание микросервисов	404
12.1. Архитектура конвейера сборки/развертывания	406
12.2. Настройка базовой инфраструктуры для O-stock в облаке	410
12.2.1. Создание базы данных PostgreSQL с использованием Amazon RDS	413
12.2.2. Создание кластера Redis в Amazon	416
12.3. После подготовки инфраструктуры: развертывание O-stock и ELK	418
12.3.1. Создание экземпляра EC2 с помощью ELK	418
12.3.2. Развертывание стека ELK в экземпляре EC2.....	422
12.3.3. Создание кластера EKS.....	423
12.4. Конвейер сборки/развертывания в действии	430
12.5. Создание конвейера сборки/развертывания.....	432
12.5.1. Настройка GitHub.....	433
12.5.2. Сборка наших служб в Jenkins.....	434
12.5.3. Создание сценария конвейера.....	439
12.5.4. Создание сценариев для конвейера развертывания Kubernetes.....	441
12.6. Заключительные рассуждения о конвейере сборки/развертывания	442
Итоги	444
Приложение А.....	445
Модель зрелости Ричардсона	446
Spring NATEOAS.....	448
Внешняя конфигурация	448
Непрерывная интеграция и непрерывная доставка.....	449
Мониторинг	450
Журналирование.....	451
API-шлюзы	451

Приложение В	453
Тип разрешения: пароль	454
Тип разрешения: учетные данные клиента	455
Тип разрешения: код авторизации.....	456
Тип разрешения: неявный.....	459
Как обновляются токены.....	461
Приложение С	463
С.1. Введение в мониторинг с использованием Spring Boot Actuator	464
С.1.1. Добавление зависимостей Spring Boot Actuator.....	464
С.1.2. Включение конечных точек Spring Boot Actuator.....	464
С.2. Настройка Micrometer и Prometheus	465
С.2.1. Введение в Micrometer и Prometheus.....	466
С.2.2. Интеграция с Micrometer и Prometheus	467
С.3. Настройка Grafana	469
С.4. Итоги обсуждения	474
Предметный указатель	475

Предисловие от издательства

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 на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и 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 обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить следующие издания 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.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

Эта книга – часть моей мечты внести свой вклад в развитие той области, которой я больше всего увлекаюсь, – информатики и, в частности, разработки программного обеспечения. Эти дисциплины демонстрируют свою исключительную важность во взаимосвязанном и глобальном настоящем. Мы ежедневно наблюдаем невероятные преобразования, которые они вызывают во всех сферах человеческой деятельности. Но зачем писать об архитектуре микросервисов, если есть много других достойных тем?

Слово «микросервисы» имеет множество толкований. В этой книге я подразумеваю под микросервисами распределенные, слабо связанные программные службы, которые выполняют ограниченное количество четко определенных задач. Микросервисы стали появляться как альтернатива монолитным приложениям, помогающая бороться с традиционными проблемами сложности в большой кодовой базе, разбивая ее на мелкие, четко ограниченные части.

В течение своей 13-летней карьеры я занималась разработкой программного обеспечения на разных языках и в разных программных архитектурах. Архитектуры, с которых я начинала, в настоящее время практически устарели. Современный мир заставляет нас постоянно двигаться вперед, и инновации в области разработки программного обеспечения развиваются ускоренными темпами. По этой причине, находясь в состоянии постоянного поиска новейших знаний и практик, несколько лет тому назад я решила окунуться в мир микросервисов. С тех пор я использовала эту архитектуру чаще других из-за ее преимуществ (таких как масштабируемость, скорость и удобство сопровождения). Успешный опыт работы в области микросервисов побудил меня взять на себя задачу написать эту книгу, чтобы систематизировать свои знания и поделиться ими с вами.

Как разработчик программного обеспечения, я понимаю, насколько важно постоянно приобретать и применять новые знания. Прежде чем взяться за эту книгу, я решила поделиться своими выводами и начала публиковать статьи о микросервисах в блоге коста-риканской компании (на моей родине), занимающейся разработкой программного обеспечения. Когда я писала эти статьи, я поняла, что нашла новую страсть и цель в своей профессиональной карьере. Через несколько месяцев после публикации одной

из моих статей я получила электронное письмо от издательства Manning Publications с предложением написать второе издание этой, которым и делюсь с вами сегодня.

Первое издание этой книги было написано Джоном Карнеллом (John Carnell), непревзойденным профессионалом с многолетним опытом разработки программного обеспечения. Я написала это второе издание на основе той книги, добавив мои собственные видение и понимание. Второе издание «*Микросервисов Spring*» покажет вам, как реализовать разнообразные шаблоны проектирования, чтобы получить успешную архитектуру микросервисов с помощью Spring – фреймворка, предлагающего готовые решения для многих распространенных задач разработки, с которыми вы неизбежно столкнетесь как разработчик микросервисов. А теперь давайте начнем захватывающее путешествие в мир микросервисов со Spring.

Благодарности

Я глубоко благодарна за возможность поработать над этой книгой, которая позволила мне поделиться с вами моими знаниями и одновременно поучиться самой. Я благодарна издательству Manning Publications за доверие и за то, что позволили мне поделиться своим опытом со многими людьми. Особое спасибо Майклу Стивенсу (Michael Stephens) за то, что дал мне эту фантастическую возможность; Джону Карнеллу (John Carnell) за его поддержку, опыт и знания; Роберту Веннеру (Robert Wenner), моему научному редактору, за его ценный вклад; и Лесли Трайтс (Lesley Trites), моему редактору, за то, что она помогала и поддерживала меня на протяжении всего процесса.

Я хочу сказать спасибо Стефану Пирнбауму (Stephan Pirnbaum) и Джону Гатри (John Guthrie), моим техническим рецензентам, которые проверяли мои рукописи и помогли поднять качество этой книги. Спасибо также моему редактору проекта Дейдре Хиам (Deirdre Hiam); литературному редактору Френсису Бурану (Frances Buran); корректору Кэти Теннант (Katie Tennant); рецензирующему редактору Алексу Драгосавлевичу (Aleks Dragosavljevic) и всем-всем рецензентам (это: Адитья Кумар (Aditya Kumar), Аль Пезевски (Al Pezewski), Алекс Лукас (Alex Lucas), Арпит Ханделвал (Arpit Khandelwal), Бонни Малек (Bonnie Malec), Кристофер Карделл (Christopher Kardell), Дэвид Морган (David Morgan), Гилберто Таккари (Gilberto Taccari), Харинат Кунтамуккала (Harinath Kuntamukkala), Айан Кэмпбелл (Iain Campbell), Капил Дев С. (Kapil Dev S), Константин Еремин (Konstantin Eremin), Кшиштоф Камичек (Krzysztof Kamyczek), Марко Умек (Marko Umek), Мэтью Грин (Matthew Greene), Филипп Виалатт (Philippe Vialatte), Пьер-Мишель Ансель (Pierre-Michel Ansel), Рональд Борман (Ronald Borman), Сатей Кумар Саху (Satej Kumar Sahu), Стефан Пирнбаум

(Stephan Pirnbaum), Тан Ви (Tan Wee), Тодд Кук (Todd Cook) и Виктор Дуран (V́ctor Durán)) – ваши предложения помогли улучшить эту книгу.

Хочу также поблагодарить моих маму, папу и всю мою семью, которые поддерживали и вдохновляли меня и которые, показывая пример преданности своей работе, помогли мне стать настоящим профессионалом. Я благодарна Илаю (Eli), который всегда был рядом со мной в те долгие дни работы, и моим друзьям, постоянно поддерживавшим меня на протяжении всего этого процесса.

И последнее, но не в последнюю очередь: я благодарю каждого из вас за приобретение этой книги и за то, что позволили мне поделиться с вами своими знаниями. Надеюсь, вам понравится читать так же, как мне понравилось писать. Я надеюсь, что эта книга станет ценным вкладом в вашу профессиональную карьеру.

Об этой книге

Книга «*Микросервисы Spring*», второе издание написана для разработчиков на Java/Spring, которым нужны практические советы и примеры создания и ввода в эксплуатацию приложений на основе микросервисов. Работая над этой книгой, мы хотели сохранить центральную идею, заложенную в первое издание, – применение основных шаблонов микросервисов, соответствующих новейшим практикам и примерам Spring Boot и Spring Cloud. Практически в каждой главе вы найдете конкретные шаблоны проектирования микросервисов, а также примеры реализации Spring Cloud.

Кому адресована эта книга

- Разработчикам на Java, имеющим некоторый опыт (1–3 года) создания распределенных приложений.
- Имеющим опыт (более 1 года) использования Spring.
- Желаящим научиться создавать приложения на основе микросервисов.
- Интересующимся особенностями использования микросервисов для создания облачных приложений.
- Стремящимся узнать, являются ли Java и Spring подходящими технологиями для создания приложений на основе микросервисов.
- Всем, кому интересно узнать, что необходимо для развертывания приложения на основе микросервисов в облаке.

Структура книги

Эта книга состоит из 12 глав и 3 приложений.

- Глава 1 знакомит с архитектурой микросервисов как важным и актуальным подходом к созданию приложений, особенно облачных.
- Глава 2 рассказывает о технологиях Spring Cloud, которые мы будем использовать, и описывает порядок создания облачных микросервисов с учетом передовых практик известного руководства «Приложение двенадцати факторов». В этой главе также приводится пример создания простого микросервиса на основе REST с помощью Spring Boot.
- Глава 3 показывает микросервисы с точки зрения архитектора, прикладного программиста и инженера DevOps, а также демонстрирует реализацию некоторых передовых методов в нашем первом микросервисе на основе REST.
- Глава 4 знакомит с миром контейнеров, подчеркивая основные различия между контейнерами и виртуальными машинами (VM). Она также показывает, как запускать микросервисы в контейнерах, используя плагины Maven и команды Docker.
- Глава 5 описывает особенности управления конфигурацией микросервисов с помощью Spring Cloud Config. Spring Cloud Config помогает организовать хранение конфигураций микросервисов в едином репозитории с учетом их версий и воспроизводить их во всех экземплярах микросервисов.
- Глава 6 знакомит с шаблоном маршрутизации обнаружения служб. Здесь вы узнаете, как использовать Spring Cloud и службу Netflix Eureka, чтобы обеспечить независимость местоположения ваших служб от клиентов, которые их используют. Вы также узнаете, как реализовать балансировку нагрузки на стороне клиента с помощью Spring Cloud LoadBalancer и клиента Netflix Feign.
- Глава 7 посвящена защите потребителей ваших микросервисов, когда один или несколько экземпляров микросервисов выходят из строя или находятся в нерабочем состоянии. В этой главе показано, как с помощью Spring Cloud и Resilience4j реализовать шаблоны размыкателя цепи (circuit breaker), отката к резервной реализации (fallback) и герметичных отсеков (bulkhead).
- Глава 8 охватывает шаблон шлюза маршрутизации (gateway routing). Здесь, используя Spring Cloud Gateway, мы создадим единую точку входа для всех наших микросервисов и увидим, как использовать фильтры Spring Cloud Gateway для создания политик, которые могут применяться ко всем службам, доступным через шлюз.

- Глава 9 рассказывает, как реализовать аутентификацию и авторизацию с помощью Keycloak. Здесь рассматриваются некоторые основные принципы OAuth2 и способы использования Spring и Keycloak для защиты архитектуры микросервисов.
- Глава 10 рассматривает реализацию асинхронного обмена сообщениями между микросервисами с помощью Spring Cloud Stream и Apache Kafka. Она также показывает, как использовать Redis для кеширования запросов.
- Глава 11 показывает, как реализовать шаблоны журналирования, такие как корреляция журналов (log correlation), агрегирование журналов (log aggregation) и трассировка, с помощью Spring Cloud Sleuth, Zipkin и ELK Stack.
- Глава 12 является краеугольным камнем этой книги. Здесь мы соберем воедино все службы, создаваемые на протяжении всей книги, и развернем их в Amazon Elastic Kubernetes Service (Amazon EKS). Мы также обсудим автоматизацию процессов сборки и развертывания микросервисов с помощью таких инструментов, как Jenkins.
- Приложение А перечисляет дополнительные передовые практики строительства микросервисных архитектур и объясняет «Модель зрелости Ричардсона».
- Приложение В содержит дополнительные материалы по OAuth2. OAuth2 – чрезвычайно гибкая модель аутентификации, и в этом приложении дается краткий обзор различных способов использования OAuth2 для защиты приложений и микросервисов.
- Приложение С рассказывает, как контролировать микросервисы Spring Boot с помощью таких технологий, как Spring Boot Actuator, Micrometer, Prometheus и Grafana.

В целом мы советуем всем разработчикам прочитать главы 1, 2 и 3, где приводится важная информация о передовых методах реализации микросервисов с использованием Spring Boot и Java 11. Если вы плохо знакомы с Docker, то мы настоятельно рекомендуем внимательно прочитать главу 4, которая кратко представляет все ключевые концепции Docker, используемые в книге.

В остальной части книги обсуждаются некоторые шаблоны проектирования микросервисов, такие как обнаружение служб, распределенная трассировка, API-шлюз и др. Было бы желательно читать главы по порядку и опробовать примеры кода у себя. Если вам жалко тратить время на ввод кода примеров вручную, то можете загрузить его из репозитория GitHub по адресу <https://github.com/ihuaylupo/manning-smia>.

О примерах кода

Эта книга содержит много примеров исходного кода в виде листингов и фрагментов в обычном тексте. В обоих случаях исходный код оформляется моноширинным шрифтом, чтобы его можно было отличить от обычного текста. Код всех примеров доступен в репозитории GitHub <https://github.com/ihuaylupo/manning-smia>.

Примеры хранятся в отдельных папках, по одной для каждой главы. Также обратите внимание, что весь код в этой книге создан для Java 11 и предполагает использование Maven в качестве основного инструмента сборки и Docker в качестве программной платформы контейнеризации. В каждой папке для каждой главы имеется файл README.md, где вы найдете следующую информацию:

- краткое введение в главу;
- перечень необходимых инструментов;
- раздел «How to Use», описывающий порядок использования;
- команду для сборки примеров;
- команду для запуска примеров;
- контакты и дополнительную информацию.

Одна из основных концепций, которых мы старались придерживаться на протяжении всей книги, заключается в том, что примеры кода в каждой главе должны работать полностью независимо от других глав. Что это значит? Вы можете, например, взять код из главы 10 и запустить его, минуя примеры из предыдущих глав. Вы увидите, что для каждой службы, созданной в каждой главе, имеется соответствующий образ Docker. В каждой главе мы используем Docker Compose для запуска образов Docker, чтобы гарантировать воспроизводимость среды выполнения.

Во многих случаях оригинальный исходный код был переформатирован; мы добавили переносы строк и изменили ширину отступов, чтобы уместить строки кода по ширине книжной страницы. В редких случаях даже этого оказалось недостаточно, поэтому в листингах вы увидите маркеры продолжения строки (➔). Кроме того, во многих листингах в книге, которые объясняются в тексте, мы убрали комментарии. Также многие листинги сопровождаются дополнительными аннотациями, подчеркивающими наиболее важные идеи.

Живое обсуждение книги

Приобретая книгу «*Микросервисы Spring*», второе издание, вы получаете бесплатный доступ к частному веб-форуму, организованному издательством Manning Publications, где можно оставлять коммен-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru