

Тебе, папа.

Нам очень не хватает твоей мягкости, мудрости и смирения.

*Кроме того, это ты научил меня программировать,
поэтому любые ошибки в этой книге
технически считаются твоими ошибками!*

Отзывы о книге «Облачный Go»

Автор книги проделал большую работу, подробно описав высокоуровневую концепцию «облачных приложений» и приемы ее реализации с использованием современного языка программирования Go. В результате получилась захватывающая и вдохновляющая книга.

– *Ли Атчисон (Lee Atchison)*
Владелец Atchison Technology LLC

Это первая книга, из встречавшихся мне, которая с такой широтой и глубиной освещает современные приемы реализации облачных вычислений. Представленные здесь шаблоны сопровождаются наглядными примерами решения реальных задач, с которыми инженеры сталкиваются ежедневно.

– *Альваро Атьенза (Alvaro Atienza)*
Инженер по надежности, Flatiron Health

На страницах этой книги ясно (и с юмором) отражен богатый опыт Мэтта в искусстве и науке построения надежных систем в принципиально ненадежном мире. Присоединяйтесь к нему, и он познакомит вас с фундаментальными строительными блоками и приемами конструирования систем, позволяющими создавать масштабные и надежные системы из эфемерных и ненадежных компонентов современной облачной инфраструктуры.

– *Дэвид Никпонски (David Nicponski)*
Главный инженер, Robinhood

За последние несколько лет наметились две важные тенденции: язык Go все чаще используется для разработки не только серверных компонентов, но и инфраструктуры; а инфраструктура перемещается в облако. В этой книге кратко описывается современное состояние сочетания этих двух факторов.

– *Натали Пистунович (Natalie Pistunovich)*
Ведущий пропагандист передовых методов разработки, Aerospike

Я начал читать эту книгу, почти ничего не зная о Go, и закончил, чувствуя себя экспертом. Я бы даже сказал, что, просто прочитав эту книгу, я стал намного более опытным инженером.

– *Джеймс Куигли (James Quigley)*
Инженер по надежности систем, Bloomberg

Содержание

От издательства	17
Об авторе	18
Об иллюстрации на обложке	19
Предисловие	20
Часть I. ОБЛАЧНОЕ ОКРУЖЕНИЕ	24
Глава 1. Что такое «облачное» приложение?	25
История развития до настоящего времени.....	26
Что значит быть «облачным»?.....	28
Масштабируемость.....	28
Слабая связанность.....	29
Устойчивость.....	30
Управляемость.....	32
Наблюдаемость.....	33
Что особенного в облачном окружении?.....	34
Итоги.....	35
Глава 2. Почему Go правит облачным миром	36
Как появился Go.....	36
Особенности облачного мира.....	37
Композиция и структурная типизация.....	37
Понятность.....	39
Модель взаимодействия последовательных процессов.....	40
Быстрая сборка.....	41
Стабильность языка.....	42
Безопасность памяти.....	42
Производительность.....	43
Статическая компоновка.....	44
Статическая типизация.....	45
Итоги.....	46
Часть II. ОБЛАЧНЫЕ КОНСТРУКЦИИ В GO	47
Глава 3. Основы языка Go	48
Базовые типы данных.....	48
Логические значения.....	49

Простые числа	49
Комплексные числа	50
Строки	50
Переменные	51
Сокращенная форма объявления переменных	51
Нулевые значения.....	52
Пустой идентификатор	53
Константы	54
Контейнеры: массивы, срезы и ассоциативные массивы.....	54
Массивы.....	55
Срезы	55
Работа со срезами.....	56
Оператор извлечения среза.....	58
Строки и срезы.....	58
Ассоциативные массивы	59
Проверка наличия в ассоциативном массиве.....	60
Указатели.....	61
Управляющие структуры	62
Забавный цикл for.....	63
Универсальная инструкция for.....	63
Обход в цикле элементов массивов и срезов	64
Обход в цикле элементов ассоциативных массивов.....	65
Инструкция if.....	65
Инструкция switch.....	66
Обработка ошибок	67
Создание ошибки.....	68
Необычные особенности функций: переменное число параметров и замыкания	69
Функции	69
Несколько возвращаемых значений	69
Рекурсия.....	70
Отложенные вычисления	70
Указатели как параметры	72
Функции с переменным числом аргументов	73
Передача срезов в параметре с переменным числом значений.....	74
Анонимные функции и замыкания	74
Структуры, методы и интерфейсы	75
Структуры	76
Методы	77
Интерфейсы.....	78
Проверка типа.....	79
Пустой интерфейс	79
Композиция путем встраивания типов.....	80
Встраивание интерфейсов	80
Встраивание структур.....	81
Продвижение.....	81
Прямой доступ к встроенным полям	81
Самое интересное: конкуренция.....	82

Сопрограммы	82
Каналы	83
Блокировка канала	83
Буферизация каналов	84
Закрытие каналов	84
Прием значений из канала в цикле	85
select	85
Реализация тайм-аутов для каналов	86
Итоги	87
Глава 4. Шаблоны программирования облачных приложений	88
Пакет context	89
Что может дать контекст	90
Создание контекста	91
Определение крайних сроков и тайм-аутов контекста	91
Определение значений в контексте запроса	92
Использование контекста	92
Структура этой главы	93
Шаблоны стабильности	94
Circuit Breaker (Размыкатель цепи)	94
Применимость	94
Реализация	95
Пример кода	96
Debounce (Антидребезг)	97
Применимость	97
Компоненты	98
Реализация	98
Пример кода	99
Retry (Повтор)	101
Применимость	101
Компоненты	102
Реализация	102
Пример кода	102
Throttle (Дроссельная заслонка)	104
Применимость	104
Компоненты	105
Реализация	105
Пример кода	106
Timeout (Тайм-аут)	107
Применимость	107
Компоненты	107
Реализация	108
Пример кода	108
Шаблоны конкуренции	110
Fan-In (Мультиплексор)	110
Применимость	110
Компоненты	110
Реализация	110

Пример кода	111
Fan-Out (Демультимплексор)	112
Применимость	112
Компоненты	112
Реализация	113
Пример кода	113
Future (В будущем).....	114
Применимость	115
Компоненты	116
Реализация	116
Пример кода	116
Sharding (Сегментирование)	118
Применимость	118
Компоненты	119
Реализация	119
Пример кода	121
Итоги.....	124
Глава 5. Конструирование облачной службы	125
Давайте создадим службу!.....	125
Что такое хранилище пар ключ/значение?	126
Требования.....	126
Что такое идемпотентность, и почему это важно?.....	126
Конечная цель	128
Итерация 0: базовая функциональность	128
Наш суперпростой API	129
Итерация 1: монолит	130
Создание HTTP-сервера с использованием net/http.....	131
Создание HTTP-сервера с использованием gorilla/mux	132
Создание минимальной службы	133
Инициализация проекта с помощью модулей Go.....	133
Переменные в путях URI	134
Множество сопоставлений.....	135
Создание службы RESTful	135
Методы RESTful.....	136
Реализация функции создания	136
Реализация функции чтения	138
Добавление в структуру данных поддержки использования в конкурентном окружении	140
Интеграция мьютекса чтения/записи в приложение	141
Итерация 2: долговременное хранение ресурса	142
Что такое журнал транзакций?	143
Формат журнала транзакций.....	143
Интерфейс регистратора транзакций	144
Сохранение состояния в журнале транзакций.....	144
Создание прототипа регистратора транзакций	145
Определение типа события.....	146

Реализация FileTransactionLogger	148
Создание экземпляра FileTransactionLogger	149
Добавление записей в конец журнала транзакций	150
Использование bufio.Scanner для воспроизведения транзакций из журнала	151
Интерфейс регистратора транзакций (еще раз)	153
Инициализация FileTransactionLogger в веб-службе	153
Интеграция FileTransactionLogger в веб-службу	155
Будущие улучшения	155
Сохранение состояния во внешней базе данных	155
Работа с базами данных в Go	156
Импортирование драйвера базы данных	157
Реализация PostgresTransactionLogger	157
Создание экземпляра PostgresTransactionLogger	158
Выполнение SQL-запроса INSERT с помощью db.Exec	160
Использование db.Query для воспроизведения транзакций из журнала	161
Инициализация PostgresTransactionLogger в веб-службе	162
Будущие улучшения	163
Итерация 3: реализация безопасности транспортного уровня	163
Transport Layer Security	164
Сертификаты, центры сертификации и доверие	164
Закрытый ключ и файлы сертификатов	165
Формат Privacy Enhanced Mail (PEM)	165
Защита веб-службы с помощью HTTPS	166
В заключение о транспортном уровне	167
Контейнеризация хранилища пар ключ/значение	168
Основы Docker	169
Dockerfile	169
Сборка образа контейнера	170
Запуск образа контейнера	171
Проверка запущенного образа контейнера	172
Отправка запроса в опубликованный порт контейнера	173
Запуск нескольких контейнеров	174
Остановка и удаление контейнеров	174
Сборка контейнера для службы хранилища пар ключ/значение	175
Итерация 1: добавление двоичного файла в пустой образ	176
Итерация 2: многоэтапная сборка	178
Сохранение данных контейнера вовне	179
Итоги	180
Часть III. ОБЛАЧНЫЕ АТТРИБУТЫ	182
Глава 6. Все дело в надежности	183
В чем суть облачных вычислений?	184
Все дело в надежности	184

Что такое надежность, и почему она так важна?	185
Надежность обеспечивается не только операторами	187
Достижение надежности	188
Предотвращение неисправностей	190
Рекомендуемые практики программирования	190
Особенности языка	190
Масштабируемость	191
Слабая связанность	191
Отказоустойчивость	192
Устранение неисправностей	192
Проверка и тестирование	193
Управляемость	194
Прогнозирование неисправностей	194
Непреходящая актуальность методологии «Двенадцать факторов»	194
I. Кодовая база	195
II. Зависимости	196
III. Конфигурация	196
IV. Сторонние службы	198
V. Сборка, выпуск, выполнение	199
VI. Процессы	200
VII. Изоляция данных	200
VIII. Масштабируемость	201
IX. Живучесть	202
X. Сходство окружений разработки/эксплуатации	202
XI. Журналирование	203
XII. Задачи администрирования	204
Итого	205
Глава 7. Масштабируемость	206
Что такое масштабируемость?	207
Различные формы масштабирования	208
Четыре основных узких места	209
С состоянием и без состояния	211
Состояние приложения и состояние ресурса	211
Преимущества отсутствия состояния	212
Отложенное масштабирование: эффективность	213
Эффективное кэширование с использованием кеша LRU	213
Эффективная синхронизация	217
Разделяйте память, общаясь	217
Уменьшение простоев на блокировках с помощью буферизованных каналов	219
Уменьшение простоев на блокировках с помощью сегментирования	221
Утечки памяти могут вызвать... фатальную ошибку исчерпания памяти во время выполнения	222
Утечки сопрограмм	222
Вечно тикающие таймеры	223
В заключение об эффективности	225

Архитектуры служб.....	225
Архитектура монолитной системы.....	226
Архитектура системы микросервисов.....	227
Бессерверные архитектуры.....	229
Достоинства и недостатки бессерверных вычислений.....	229
Бессерверные службы.....	231
Итоги.....	233
Глава 8. Слабая связанность.....	234
Тесная связанность.....	235
Множество форм тесной связанности.....	236
Хрупкие протоколы обмена.....	236
Общие зависимости.....	237
Общий момент времени.....	237
Фиксированные адреса.....	238
Взаимодействия между службами.....	238
Шаблон обмена сообщениями запрос/ответ.....	239
Распространенные реализации шаблона запрос/ответ.....	240
Отправка HTTP-запросов с использованием net/http.....	240
Вызов удаленных процедур с использованием gRPC.....	244
Определение интерфейса с использованием протокола буферов.....	245
Установка компилятора протокола буферов.....	246
Определение структуры сообщения.....	247
Структура сообщений для взаимодействий с хранилищем пар ключ/значение.....	248
Определение методов службы.....	249
Компиляция протокола буферов.....	250
Реализация службы gRPC.....	251
Реализация клиента gRPC.....	253
Слабое связывание локальных ресурсов с помощью плагинов.....	255
Подключение плагинов с помощью пакета plugin.....	255
Словарь плагинов.....	256
Пример плагина.....	257
Интерфейс Sayer.....	257
Код плагина.....	258
Сборка плагинов.....	258
Использование плагинов Go.....	259
Запуск примера.....	261
Система плагинов HashiCorp для Go, доступных через RPC.....	261
Еще один пример плагина.....	262
Общий код.....	263
Реализация плагина.....	265
Процесс-потребитель.....	266
Гексагональная архитектура.....	269
Архитектура.....	269
Реализация гексагональной службы.....	270
Реорганизация компонентов.....	271

Наш первый разъем	272
Основное приложение	272
Адаптеры TransactionLogger	273
Порт FrontEnd	274
Все вместе	276
Итоги	277
Глава 9. Устойчивость	279
Почему устойчивость важна	280
Что подразумевается под сбоем системы?	281
Обеспечение устойчивости	282
Каскадные сбои	282
Предотвращение перегрузки	284
Дросселирование	284
Сброс нагрузки	288
Постепенное ухудшение качества обслуживания	289
Повтори еще раз: повторные запросы	289
Алгоритмы увеличения задержки	291
Размыкание цепи	294
Тайм-ауты	295
Использование контекста Context для реализации тайм-аутов на стороне службы	296
Прерывание ожидания обработки клиентских запросов HTTP/REST	298
Прерывание ожидания обработки клиентских запросов gRPC	299
Идемпотентность	301
Как сделать службу идемпотентной?	302
А как насчет скалярных операций?	303
Избыточность служб	304
Проектирование избыточности	305
Автоматическое масштабирование	307
Проверка работоспособности	308
Что подразумевается под «работоспособностью» экземпляра?	309
Три типа проверок работоспособности	309
Проверка жизнеспособности	310
Поверхностная проверка работоспособности	310
Глубокая проверка работоспособности	312
Открытие при отказе	313
Итоги	314
Глава 10. Управляемость	315
Что такое управляемость, и почему она важна?	316
Настройка приложения	317
Рекомендуемые приемы организации конфигураций	318
Настройка с использованием переменных окружения	319
Настройка с использованием аргументов командной строки	320
Стандартный пакет flag	320

Парсер командной строки Cobra	322
Настройка с использованием файлов	326
Наша структура конфигурационных данных	326
Формат JSON	327
Формат YAML	332
Наблюдение за изменениями в конфигурационных файлах	335
Viper: швейцарский армейский нож конфигурационных пакетов	340
Явно устанавливаемые значения в Viper	341
Работа с флагами командной строки в Viper	341
Работа с переменными окружения в Viper	342
Работа с конфигурационными файлами в Viper	342
Использование удаленных хранилищ пар ключ/значение в Viper	344
Значения по умолчанию в Viper	345
Управление функциональными возможностями с помощью флагов	345
Разработка флага для управления функциональной возможностью	346
Итерация 0: начальная реализация	347
Итерация 1: жестко запрограммированный флаг	347
Итерация 2: настраиваемый флаг	348
Итерация 3: динамический флаг	349
Динамические флаги как функции	350
Реализация функции динамического флага	350
Поиск функции флага	351
Функция маршрутизации	352
Итоги	353
Глава 11. Наблюдаемость	354
Что такое наблюдаемость?	355
Зачем нужна наблюдаемость?	355
Чем наблюдаемость отличается от «традиционного» мониторинга?	356
«Три столпа наблюдаемости»	357
OpenTelemetry	358
Компоненты OpenTelemetry	359
Трассировка	360
Концепции трассировки	361
Трассировка с использованием OpenTelemetry	362
Создание экспортеров трассировки	364
Создание провайдера трассировки	366
Настройка глобального провайдера трассировки	367
Получение экземпляра трассировщика	367
Начальная и конечная операции	367
Установка метаданных операции	369
Автоматическое инструментирование	370
Собираем все вместе: трассировка	373
API-службы вычисления чисел Фибоначчи	374
Функция-обработчик службы вычисления чисел Фибоначчи	375
Функция main службы	376
Запуск служб	377

Вывод консольного экспортера	377
Просмотр результатов в Jaeger	378
Метрики.....	379
Два способа передачи метрик: принудительная и по запросу.....	381
Принудительная отправка метрик	382
Передача метрик по запросу	382
Какой подход лучше?	383
Метрики в OpenTelemetry.....	384
Создание экспортеров метрик	385
Установка глобального провайдера метрик.....	386
Экспортирование конечной точки метрик.....	386
Получение экземпляра Meter	388
Инструменты метрик.....	388
Собираем все вместе: метрики.....	394
Запуск служб.....	394
Вывод конечной точки метрик.....	395
Просмотр результатов в Prometheus	396
Журналирование	397
Рекомендуемые методы журналирования	397
Интерпретируйте журналы как потоки событий	398
Структурируйте события для последующего анализа.....	398
Лучше меньше, да лучше.....	400
Динамически фильтруйте журналируемые данные	400
Журналирование с использованием стандартного пакета log.....	401
Специальные функции журналирования	402
Журналирование в нестандартный объект записи	402
Флаги журналирования	403
Пакет журналирования Zap.....	403
Создание регистратора Zap	405
Журналирование с использованием Zap	405
Динамическая фильтрация журналируемых данных в Zap.....	407
Итоги.....	409
Предметный указатель.....	410

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.

Скачивание исходного кода примеров

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

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, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и O'Reilly очень серьезно относятся 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 электронной почты dmkpress@gmail.com.

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

Об авторе

Мэтью А. Титмус – ветеран индустрии разработки программного обеспечения. Научившись создавать виртуальные миры в LPC, он получил удивительно востребованное образование в области молекулярной биологии, создал инструменты анализа терабайтных наборов данных для лаборатории физики высоких энергий, с нуля написал фреймворк для разработки веб-приложений, применил методы распределенных вычислений для анализа ракового генома, а также в числе первых разрабатывал методы машинного обучения на связанных данных.

Он был одним из первых сторонников облачных технологий в целом и языка Go в частности. Последние четыре года специализируется на переносе монолитных приложений в контейнерный облачный мир, помогая компаниям осваивать новые способы разработки, развертывания и управления своими службами. Он увлечен решением задач повышения качества промышленных систем и потратил много времени на обдумывание и реализацию стратегий наблюдения за распределенными системами и управления ими.

Мэтью живет на Лонг-Айленде с самой терпеливой женщиной в мире, на которой ему посчастливилось жениться, и самым очаровательным мальчиком в мире, от которого ему посчастливилось услышать «папа».

Об иллюстрации на обложке

На обложке «Облачный Go» изображено животное из семейства туко-туко (*Ctenomyidae*). Эти неотропические грызуны обитают в южной части Южной Америки.

Название «туко-туко» относится к широкому кругу видов. Эти грызуны имеют плотное тело с мощными короткими лапами и хорошо развитыми когтями. У них большая голова, но маленькие уши, и хотя до 90 % времени они проводят под землей, они имеют относительно большие глаза, по сравнению с другими норными грызунами. Цвет и текстура шерсти туко-туко варьируются в зависимости от вида, но в целом шерсть довольно густая. Хвост короткий и почти без шерсти.

Туко-туко роют системы туннелей, часто весьма обширные и сложные, в песчаной и/или суглинистой почве. В этих системах туннелей имеются отдельные камеры для гнездования и хранения пищи. В ходе эволюции туко-туко претерпели различные морфологические изменения, которые помогают им прекрасно чувствовать себя под землей, и развили хорошее обоняние, помогающее им ориентироваться в туннелях. При рытье нор используют как когти, так и рыло.

Рацион туко-туко состоит в основном из корней, стеблей и трав. В настоящее время считаются сельскохозяйственными вредителями, но во времена до появления европейцев в Южной Америке они были важным источником пищи для коренных народов, особенно на Огненной Земле. Современный охранный статус туко-туко зависит от вида и географического региона. Одним видам присвоена категория «вызывающий наименьшие опасения», тогда как другие считаются «находящимися под угрозой исчезновения». Многие животные, изображенные на обложках книг издательства O'Reilly, находятся под угрозой вымирания; все они очень важны для биосферы.

Иллюстрацию для обложки нарисовал Карен Монтгомери (Karen Montgomery) на основе черно-белой гравюры из энциклопедии «English Cyclopaedia Natural History». Текст на обложке набран шрифтами Gilroy Semibold и Guardian Sans. Текст книги набран шрифтом Adobe Minion Pro; текст заголовков – шрифтом Adobe Myriad Condensed; а фрагменты программного кода – шрифтом Ubuntu Mono, созданным Далтоном Маагом (Dalton Maag).

Предисловие

ЭТО ВОЛШЕБНОЕ ВРЕМЯ ДЛЯ ИНЖЕНЕРОВ

У нас есть Docker для создания контейнеров и Kubernetes для управления ими. Prometheus помогает нам следить за ними. Consul позволяет обнаруживать их. Jaeger дает возможность организовать взаимодействия между ними. Это лишь несколько примеров, в действительности круг возможностей гораздо шире, и все эти возможности поддерживают новое поколение технологий: все они «облачные», и все они написаны на Go.

Термин «облачный» кажется двусмысленным и отдает рекламной шумихой, но на самом деле он имеет довольно конкретное определение. Согласно Cloud Native Computing Foundation, подразделению известного фонда Linux Foundation, облачное приложение – это приложение, способное масштабироваться синхронно с изменением нагрузки, устойчивое к неопределенности окружения и управляемое в условиях нестабильности и постоянно меняющихся требований. Иначе говоря, облачные приложения создаются для работы в жесткой и неопределенной вселенной.

На основе опыта, накопленного за годы разработки облачного программного обеспечения, около десяти лет назад был создан Go – первый язык программирования, спроектированный специально для разработки облачных приложений. Во многом его появление было обусловлено тем, что типичные серверные языки, использовавшиеся в то время, просто не подходили для создания распределенных приложений, которые производит Google.

С тех пор Go занял лидирующие позиции в облачной разработке и используется повсюду: от Docker до Harbour, от Kubernetes до Consul, от InfluxDB до CockroachDB. Десять из пятнадцати сертифицированных проектов Cloud Native Computing Foundation и 42 из 62¹ его проектов в целом написаны в основном или полностью на Go. И с каждым днем их становится все больше.

КОМУ АДРЕСОВАНА ЭТА КНИГА

Эта книга адресована опытным разработчикам, особенно инженерам веб-приложений и инженерам по надежности. Многие из них уже использовали Go для создания веб-сервисов, но не знали некоторых тонкостей разработки в облачных окружениях или не имели четкого представления о том, что такое «облачные приложения», и впоследствии обнаруживали, что их сервисы сложно развертывать, ими сложно управлять или наблюдать за ними. Таким

¹ Включая проекты CNCF из категорий Sandbox, Incubating и Graduated, по состоянию на февраль 2021 года.

читателям эта книга не только поможет заложить прочный фундамент для создания собственных облачных сервисов, но также покажет, в чем преимущества этих методов, и представит конкретные примеры, способствующие пониманию этой довольно абстрактной темы.

Предполагается, что многие читатели хорошо знакомы с другими языками программирования, но их привлекает репутация Go как языка облачной разработки. Таким читателям эта книга предложит передовой опыт использования Go в качестве специализированного языка разработки для облачных окружений и поможет им решить собственные проблемы управления и развертывания облачных приложений.

ПОЧЕМУ Я НАПИСАЛ ЭТУ КНИГУ

Способы проектирования, конструирования и развертывания приложений меняются со временем. Требования к масштабированию вынуждают разработчиков размещать свои сервисы на десятках и сотнях серверов: отрасль постепенно становится «облачной». Но при этом возникает множество новых проблем: как разрабатывать, развертывать или управлять сервисом, действующим на десятках, сотнях или даже тысячах серверов? К сожалению, существующие книги об облачных вычислениях сосредоточены на абстрактных принципах проектирования и содержат лишь элементарные примеры, если вообще содержат. Эта книга призвана удовлетворить потребность в демонстрации практической реализации сложных принципов проектирования облачных вычислений.

СОГЛАШЕНИЯ

В этой книге используются следующие соглашения по оформлению:

Курсив

Используется для обозначения новых терминов, имен файлов и расширений.

Моноширинный шрифт

Применяется для оформления листингов программ и программных элементов внутри обычного текста, таких как имена переменных и функций, баз данных, типов данных, переменных окружения, инструкций и ключевых слов.

Моноширинный жирный

Обозначает команды или другой текст, который должен вводиться пользователем.

Моноширинный курсив

Обозначает текст, который должен замещаться фактическими значениями, вводимыми пользователем или определяемыми из контекста.

- ✓ Так выделяются советы и предложения.
- i Так обозначаются примечания общего характера.
- ! Так обозначаются предупреждения и предостережения.

ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО КОДА ПРИМЕРОВ

Вспомогательные материалы (примеры кода, упражнения и т. д.) доступны для загрузки по адресу: <https://github.com/cloud-native-go/examples>.

Данная книга призвана оказать вам помощь в решении ваших задач. В общем случае все примеры кода из этой книги вы можете использовать в своих программах и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Например, если вы разрабатываете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения примеров из этой книги вам необходимо получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, получение разрешения не требуется. Но при включении существенных объемов программного кода примеров из этой книги в вашу документацию вам необходимо будет получить разрешение издательства.

Мы приветствуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN. Например: «Мэтью А. Титмус. Облачный Go. М.: ДМК Пресс, 2021. 978-5-97060-965-1» или «*Cloud Native Go* by Matthew A. Titmus (O'Reilly). Copyright 2021 Matthew A. Titmus, 978-1-492-07633-9».

За получением разрешения на использование значительных объемов программного кода примеров из этой книги обращайтесь по адресу permissions@oreilly.com.

БЛАГОДАРНОСТИ

В первую очередь я хочу поблагодарить жену и сына. Вы – главная причина всех моих успехов, достигнутых после того, как вы вошли в мою жизнь. Вы – мои путеводные звезды, которые помогают мне не сбиться с пути и заставляют смотреть в небо.

Моему отцу, которого мы недавно потеряли. Ты был ближе всех к людям эпохи Возрождения и при этом умудрялся быть самым добрым и скромным человеком, которого я когда-либо знал. Я все еще хочу быть похожим на тебя, когда вырасту.

Мэри. Ты чувствуешь его отсутствие острее других. Мы – одна семья, и мы всегда будем семьей, даже если я буду звонить тебе не так часто, как следовало бы. Папа гордился бы твоей силой и грацией.

Саре. Меня всегда поражала твоя сила духа и стойкость. Твой острый ум сделал тебя моим самым верным союзником и самым жестким противником с тех самых пор, как только научились говорить. Не говори Натану, но ты моя любимая сестра.

Натану. Если каждый из нас унаследовал что-то одно от отца, то ты, безусловно, получил его сердце. Я нечасто говорю это, но я очень горжусь тобой и твоими достижениями. Не говори Саре, но ты мой любимый брат.

Маме. Ты сильная, умная, яркая и необычная. Спасибо, что научила меня всегда делать то, что действительно нужно делать, независимо от того, что думают люди. Оставайся необычной и не забывай кормить цыплят.

Альберту. У тебя огромное сердце и бездонное терпение. Спасибо, что присоединился к нашей семье; мы все выиграли от этого.

Всем другим членам нашей семьи. Я не могу видеться с вами так часто, как хотелось бы, и я очень скучаю по вам, но я всегда ощущаю вас рядом, когда вы мне нужны. Спасибо, что праздновали со мной победы и поддерживали меня в поражениях.

Уолту и Альваро, с которыми я не смогу расстаться, даже поменяв работу. Спасибо за вашу восторженную поддержку в моих начинаниях и за ваш абсолютный реализм. Вы оба делаете меня лучше. Кроме того, спасибо за то, что познакомили меня с серией книг «Gradle» Уилла Уайта (Will Wight) и за последовавшую за этим пагубную зависимость.

Моим друзьям «Jeff Classic», «New Jeff», Алексу (Alex), Маркану (Markan), Приянке (Priyanka), Сэму (Sam), Оуэну (Owen), Мэтту М., Мариусу (Matt M., Marius), Питеру (Peter), Рохиту (Rohit) и коллегам из Flatiron Health. Спасибо, что позволили мне отвлечься на эту книгу, и за поддержку, что выступили в качестве советчиков, первых читателей рукописи и критиков, а также за то, что воодушевили меня и были моими помощниками.

Всем моим друзьям из CoffeeOps в Нью-Йорке и во всем мире. Вы любезно позволили мне отразить ваши мысли и бросить вам вызов, а вы приняли этот вызов. Эта книга определенно выиграла от вашего участия.

Лиз Фон-Джонс (Liz Fong-Jones), известному эксперту в области наблюдений и оракулу. Ваши указания, замечания и образцы кода были неоценимы, и без вашей щедрости написать эту книгу было бы намного труднее, а результат был бы намного хуже.

Моим техническим обозревателям Ли Атчисону (Lee Atchison), Альваро Атьензе (Alvaro Atienza), Дэвиду Никпонски (David Nicponski), Натали Пистунович (Natalie Pistunovich) и Джеймсу Куигли (James Quigley). Спасибо за терпение, которое вы проявили, чтобы прочитать каждое написанное мной слово (даже сноски). Эта книга получилась намного лучше благодаря вашей зоркости и упорному труду.

Наконец, спасибо всей команде редакторов и художников O'Reilly Media, с которыми мне посчастливилось работать, особенно Амелии Блевинс (Amelia Blevins), Дэнни Эльфанбаум (Danny Elfanbaum) и Зану Маккуэйду (Zan McQuade). 2020 год был сложным, но ваши доброта, терпение и поддержка помогли мне пройти через него.

Часть **I**



ОБЛАЧНОЕ ОКРУЖЕНИЕ

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru