Моему мужу, Глену.
В тот день, когда я встретила тебя, вся моя жизнь изменилась

Моему сыну, Максу.
В тот день, когда ты родился, вся моя жизнь изменилась

Содержание

Предисловие

На протяжении шести лет я имел честь работать с Николь Форсгрен и Джезом Хамблом над отчетом о состоянии DevOps (State of DevOps Report), в котором собраны данные более чем 30 000 респондентов. Одним из величайших откровений для меня стала важность архитектуры программного обеспечения: у высокопроизводительных команд были архитектуры, позволяющие разработчикам быстро и независимо разрабатывать, тестировать и развертывать программное обеспечение для клиентов, делая это безопасно и надежно.

Несколько десятилетий назад мы бы пошутили, сказав, что разработчики программного обеспечения были экспертами только в использовании Visio, создании диаграмм UML и генерации слайдов PowerPoint, на которые никто никогда не смотрел. Если когда-то так и было, то сейчас это точно не так. В наши дни компании одерживают победы и проигрывают на рынке благодаря программному обеспечению, которое они создают. И ничто не влияет на повседневную работу разработчиков больше, чем архитектура, в которой они должны работать.

Эта книга заполняет пробел, охватывая теорию и практику. В сущности, я думаю, что только очень небольшое число людей могло бы написать ее. Корнелия Дэвис обладает уникальной квалификацией. На протяжении нескольких лет, будучи аспирантом, она изучала языки программирования, развивая любовь к функциональному программированию и неизменяемости. В течение нескольких десятков лет она работала с крупными программными системами и помогала крупным компаниям, занимающимся разработкой программного обеспечения, достигать величия.

За последние пять лет я много раз обращался к ней за помощью и советами, часто по таким темам, как CQRS и Event Sourcing, LISP и Clojure (мой любимый язык программирования), опасности императивного программирования и состояния, и даже таким простым вещам, как рекурсия.

Корнелия не просто так начинает с шаблонов, что и делает эту книгу настолько полезной для чтения. Она начинает с основных принципов, а затем доказывает их обоснованность с помощью аргументации, иногда с помощью логики, а иногда с помощью блок-схем. Ее не устраивает одна лишь теория, поэтому затем она реализует эти шаблоны в Java Spring, итерация за итерацией, включая туда то, что вы узнали.

Я нашел эту книгу интересной и познавательной и узнал невероятное количество тем, о которых раньше у меня было лишь поверхностное представление. Теперь я полон решимости реализовать ее примеры в Clojure, желая доказать, что я могу применить эти знания на практике.

Я подозреваю, что вы объедините концепции, которые приведут вас в восторг и, возможно, даже поразят вас. Для меня одной из этих концепций была необходимость централизовать межсекторальные задачи либо с помощью аспектно-ориентированного программирования, либо sidecar-контейнеров Kubernetes или инъекций Spring Retry.

Я надеюсь, что вы найдете эту книгу полезной для чтения, как и я!

Джин Ким,
исследователь и один из авторов книг
The Phoenix Project, *The DevOps Handbook*
и *Accelerate*

От автора

Я начинала свою карьеру в области обработки изображений. Я работала с инфракрасными изображениями в отделе ракетных систем компании Hughes Aircraft, занимаясь такими вещами, как выделение границ и межкадровая корреляция (часть из этого можно найти в приложениях вашего мобильного телефона сегодня – все это было аж в 80-х!).

Одним из вычислений, которые мы часто выполняем при обработке изображений, является среднеквадратическое отклонение. Я никогда не стеснялась задавать вопросы, и один из вопросов, которые я часто задавала в то время, касался этого среднеквадратического отклонения. Коллега неизменно писал следующее:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}.$$

Но я знала формулу среднеквадратического отклонения. Черт возьми, за три месяца я писала ее уже, наверное, полдюжины раз. Я спрашивала: «Что дает нам знание среднеквадратического отклонения в этом контексте?» Среднеквадратическое отклонение используется для определения того, что является «нормальным», чтобы мы могли искать выбросы. Если я рассчитываю стандартное отклонение, а затем нахожу нечто, выходящее за рамки нормы, это признак того, что мой датчик неисправен и мне нужно выбросить кадр с изображением, или это показывает действия потенциального противника?

Какое все это имеет отношение к облачной среде? Никакого. Но это имеет отношение к шаблонам. Дело в том, что я знала схему – расчет среднеквадратического отклонения, – но из-за недостатка опыта в то время я боролась с тем, когда и зачем ее применять.

В этой книге я научу вас шаблонам для облачных приложений – и да, я покажу вам множество «формул», но гораздо больше времени я трачу на *контекст* – когда и для чего применять эти шаблоны. На самом деле шаблоны, как правило, не так уж и сложны (например, повтор запроса, описанный в главе 9, является простой концепцией, которую легко реализовать). Но выбрать, когда применять шаблон и как именно это сделать, может быть непросто. Существует так много понимания контекста, в котором вы будете применять эти шаблоны, и, честно говоря, этот контекст может быть сложным.

Так что же это за контекст? По сути, это одна из распределенных систем. Когда я начинала свою карьеру более 30 лет назад, я знала мало людей, которые работали над распределенными системами, и я не посещала занятия по распределенным системам в колледже. Да, были люди, которые работали в этой области, но, честно говоря, она была довольно нишевой.

Сегодня подавляющее большинство программного обеспечения является распределенной системой. Некоторые части вашего программного обеспечения работают в браузере, а другие – на сервере или, осмелюсь сказать, целой куче серверов. Эти серверы могут работать в вашем корпоративном центре обработки

данных, или они могут находиться в центре обработки темных данных в Прайневилле, штат Орегон, или же и там, и там. И все эти фрагменты взаимодействуют друг с другом по сети, возможно, через интернет, и, вероятно, данные вашего программного обеспечения также являются широко распределенными. Говоря проще, ПО для облачной среды – это распределенная система. Кроме того, все постоянно меняется – могут случаться проблемы с серверами, в сетях часто бывают простои, пусть даже кратковременные, а устройства хранения могут выходить из строя без предупреждения – однако ожидается, что ваше программное обеспечение будет работать. Это довольно сложный контекст.

Но его можно полностью подчинить себе! Цель этой книги – помочь вам понять этот контекст и предоставить вам инструменты для того, чтобы стать опытным архитектором и разработчиком программного обеспечения для облачных сред.

Никогда прежде я не была более интеллектуально стимулирована, чем сейчас. Во многом это связано с тем, что технологический ландшафт существенно меняется, и в центре внимания находятся облачные технологии. Мне очень нравится то, чем я зарабатываю на жизнь, и я хочу, чтобы все, особенно вы, получали удовольствие от написания программного обеспечения так же, как и я. Вот почему я и написала эту книгу: я хочу поделиться с вами сумасшедшими классными проблемами, над которыми мы работаем, и помочь вам на пути к решению этих проблем. Для меня большая честь иметь возможность сыграть даже небольшую роль в вашем пути к облачным технологиям.

Благодарности

Мое путешествие по облачным технологиям началось всерьез в 2012 году, когда мой начальник Том Магуайр попросил меня заняться моделью PaaS (Platform as a Service – Платформа как услуга). Будучи членами группы по архитектуре в офисе технического директора EMC, изучение новых технологий не было для нас чем-то новым, но, боже, у нас получилось! Я всегда буду благодарна Тому за этот импульс и за предоставленную мне возможность.

К началу 2013 года я уже знала достаточно, чтобы понять, что этим я буду заниматься в обозримом будущем, и с созданием компании Pivotal Software у меня было место для этой работы. В первую очередь я хочу поблагодарить Элизабет Хендриксон за то, что она пригласила меня на вечеринку Cloud Foundry, – даже когда я еще работала в EMC, – и за то, что познакомила меня с Джеймсом Уоттерсом. Я часто говорю, что лучший шаг в моей карьере – работа на Джеймса. Я благодарю его за те многочисленные возможности, которые он предоставил мне, за то, что он доверился мне и позволил мне максимально реализовать себя, за бесчисленные разговоры с высокой пропускной способностью, в ходе которых мы все вместе познавали облачную среду, и за то, что мы так подружились за последние шесть лет.

Я благодарна за то, что являюсь частью Pivotal с момента ее создания, где я проходила обучение наряду со многими яркими, преданными и отзывчивыми коллегами. Я хотела бы поблагодарить Элизабет Хендриксон, Джошуа МакКенти, Эндрю Клэй-Шафера, Скотта Яру, Феррана Роденаса, Мэтта Стайна, Рагвендера Арни и многих других (пожалуйста, простите меня, если я кого не упомянула) за то, что помогли мне учиться и за то, что разделили со мной шесть лучших лет моей жизни! Я также хотела бы поблагодарить Pivotal, в частности Иана Эндрюса и Келли Холл, за спонсирование мини-книги *Cloud-Native Foundations*.

Я столькому научилась у своих коллег, больше, чем могу себе представить. Спасибо каждому из вас. Но я бы хотела выделить Джина Кима. Я вспоминаю тот вечер, когда мы встретились (и еще раз благодарю Элизабет Хендриксон за ту роль, которую она сыграла, чтобы эта встреча стала возможной), и сразу же поняла, что мы будем сотрудничать на протяжении долгого времени. Я благодарю Джина за возможность поработать с ним на саммите DevOps Enterprise Summit, благодаря которому я познакомилась с большим количеством новаторов, работающих в самых разных компаниях. Я благодарю его за ободряющие и расширяющие сознание беседы и за то, что он написал предисловие к этой книге.

Конечно, я благодарю издательство Manning Publications за возможность написать эту книгу, и прежде всего Майка Стивенса, который помог мне перейти от праздного любопытства к тому, чтобы сделать реальный шаг в написании книги. Я очень благодарна своему редактору по развитию Кристине Тейлор. Она взяла начинающего автора, у которого на старте была мешанина идей из 20 глав, и раннюю главу длиной около 70 страниц, и помогла мне создать книгу, которая имеет структуру и реальную сюжетную линию. Она работала со мной более двух с половиной лет, подбадривая меня, когда я была в отчаянии, и поздравляла меня, когда

я была на пике своих достижений. Я также благодарю производственную команду, в том числе Шарон Уилки, Дейрдру Хиам, Нила Кролл, Кэрол Шилдс и Николь Берд, которые несут ответственность за, что нам удалось выйти на финишную прямую. И спасибо моим рецензентам – Бачиру Тихани, Карлосу Роберто Варгасу Монтеро, Дэвиду Шмитцу, Дониёру Улмасову, Грегору Зуровски, Джареду Дункану, Джону Гатри, Хорхе Иезекиилю Бо, Кельвину Джонсону, Кенту Р. Шпилнер, Лонни Сметана, Луису Карлосу Санчесу Гонсалесу, Марку Миллеру, Питеру Полу Селларсу, Равишу Шарма, Сергею Евсикову, Серхио Мартинесу, Шанкеру Джанакираману, Стефану Хеллвегеру, Вину Оо и Зородзайи Мукуя. Ваш отзыв оказал заметное влияние на конечный результат.

И больше всего я благодарю своего мужа Глена и сына Макса за их терпение, поддержку и неизменную веру в меня. Они, как и все остальные, являются теми двумя людьми, которые сделали это возможным не только благодаря тому, что оказывали мне поддержку в течение последних трех лет, но и помогали мне заложить основу за несколько десятилетий до этого. От всей души благодарю вас обоих. И за то, что ты, Макс, полюбил компьютеры так же сильно, как и я, и позволил мне стать зрителем в этой поездке, – это двойная шоколадная глазурь на торте «Death by chocolate» – спасибо!

Об этой книге

Кому стоит прочитать эту книгу

Переход в «облако» – это скорее о том, как вы разрабатываете свои приложения, нежели о том, где вы их развертываете. Эта книга представляет собой руководство по разработке надежных приложений, которые процветают в динамичном, распределенном, виртуальном мире облака. В ней представлена ментальная модель облачных приложений, а также шаблоны, методы и инструменты, поддерживающие их конструкцию. Здесь вы найдете реалистичные примеры и советы экспертов для работы с приложениями, данными, сервисами, маршрутизацией, и много чего еще.

По сути, это книга об архитектуре, в которой содержатся примеры кода для поддержания обсуждений, связанных с проектированием. Вы увидите, что я часто ссылаюсь на различия между шаблонами, которые я здесь описываю, и тем, как мы могли что-то делать в прошлом. Однако наличия опыта или даже знания шаблонов предшествующей эры не требуется. Поскольку я рассматриваю не только сами шаблоны, но и их мотивы и нюансы контекста, в котором они применяются, они могут оказаться очень полезными для вас, независимо от того, сколько лет вы занимаетесь разработкой программного обеспечения.

И хотя в этой книге приведено много примеров, содержащих код, это не книга по программированию. Она не научит вас программировать, если вы пока еще не знаете основ. Примеры кода написаны на Java, но с каким бы языком вы ни работали прежде, у вас не должно возникнуть проблем при чтении этой книги. Знание основ взаимодействия типа «клиент /служба», особенно через протокол HTTP, также полезно, но не обязательно.

Как устроена эта книга: дорожная карта

Эта книга состоит из 12 глав, разделенных на две части.

Часть I определяет облачный контекст и представляет характеристики среды, в которой вы будете развертывать свое программное обеспечение.

- Глава 1 дает определение термина *cloud-native* и отличает ее от облака. Он представляет мысленную модель, вокруг которой можно создать шаблоны, которые появятся позже: объектами этой модели являются *приложения/службы*, *взаимодействия* между службами и *данные*.
- Глава 2 посвящена облачным операциям – шаблонам и методам, используемым для поддержания работоспособности программного обеспечения для облачной среды в рабочем окружении во время неизбежных сбоев, которые обрушиваются на него.
- Глава 3 знакомит вас с облачной платформой, средой разработки и выполнения, которая обеспечивает поддержку и даже реализацию многих шаблонов, представленных во второй части книги. Хотя важно понимать все последующие шаблоны, вам не нужно реализовывать их все самостоятельно.

Во второй части подробно рассматриваются сами шаблоны.

- Глава 4 посвящена облачному *взаимодействию*, а также касается *данных*, знакомя вас с событийно-ориентированным обменом данными в качестве альтернативы привычному стилю «запрос/ответ». Хотя последнее практически повсеместно распространено в большинстве программных продуктов, событийно-ориентированный подход часто дает значительные преимущества сильно распределенному облачному программному обеспечению, и при изучении шаблонов важно учитывать оба протокола.
- Глава 5 посвящена облачным *приложениям/службам* и их связи с *данными*. В ней рассказывается, как развертывать приложения в качестве избыточных экземпляров часто в большом масштабе, для чего и как делать их не сохраняющими состояние, как привязать их к специальной службе с фиксацией состояния.
- В главе 6 рассказывается о том, как можно последовательно поддерживать конфигурацию приложений при развертывании большого числа экземпляров в широко распределенной инфраструктуре, а также говорится о правильном применении конфигурации приложений, когда среда, в которой они работают, постоянно меняется.
- Глава 7 охватывает жизненный цикл приложения и многочисленные способы обновления без остановки, включая последовательные обновления и сине-зеленые обновления.
- Глава 8 посвящена *взаимодействию* в облачной среде. Она фокусируется на том, как приложения могут находить нужные им сервисы (обнаружение сервисов), даже когда те постоянно перемещаются, и на том, как запросы в конечном итоге попадают в нужные сервисы (динамическая маршрутизация).
- Глава 9 сосредоточена на взаимодействии на стороне клиента. После объяснения необходимости избыточности взаимодействия и знакомства с повторными отправками запроса (при которых запросы повторяются, если они изначально были неудачными) в главе рассматриваются проблемы, которые могут возникнуть в результате неправильного применения повторных отправок, и способы избежать этих проблем.
- Глава 10 посвящена взаимодействию на стороне сервера. Даже если клиенты, инициирующие взаимодействие, делают это ответственно, служба все равно должна защищать себя от неправильного использования и от перегруженности трафиком. В этой главе рассматриваются шлюзы API и предохранители.
- Глава 11 посвящена *приложениям* и *взаимодействию*. В ней рассматриваются средства наблюдения за поведением и производительностью распределенной системы, составляющей ваше программное обеспечение.
- Глава 12 посвящена *данным* и имеет существенное влияние на *взаимодействие* между службами, составляющими ваше облачное программное обеспечение. В ней автор рассказывает о шаблонах, использующихся для разбиения того, что когда-то было монолитной базой данных, на распределенную структуру данных, в конечном итоге возвращаясь к событийно-ориентированным шаблонам, описанным в начале второй части книги.

О коде

Эта книга содержит много примеров исходного кода, как в пронумерованных листингах, так и внутри обычного текста. В обоих случаях исходный код форматируется с помощью шрифта фиксированной ширины, `вот так`, чтобы отделить его от обычного текста. Иногда код также выделяется **жирным шрифтом**, чтобы выделить фрагменты, на которые вам следует обратить внимание.

Во многих случаях исходный код был переформатирован; мы добавили разрывы строк и переработали отступы, чтобы обеспечить доступное пространство для страниц в книге. В редких случаях даже этого было недостаточно, и листинги содержат маркеры продолжения строки (➥). Кроме того, комментарии в исходном коде часто удалялись из листингов, когда описание кода приводилось в тексте. Многие листинги снабжены аннотациями, которые используются для выделения важных понятий.

Код, содержащийся в примерах этой книги, доступен для скачивания с вебсайта издательства Manning по адресу **https://www.manning.com/books/cloud-native-patterns** и на GitHub на странице **https://github.com/cdavisafc/cloudnative-abundantsunshine**.

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**.

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.

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.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты **dmkpress@gmail.com** со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Об авторе

Корнелия Дэвис – вице-президент по технологиям в компании Pivotal, где она работает над технологической стратегией для Pivotal и клиентов компании. В настоящее время она работает над тем, чтобы объединить различные модели облачных вычислений IaaS, aPaaS, CaaS и FaaS в комплексное предложение, позволяющее ИТ-организациям функционировать на самом высоком уровне.

Будучи ветераном отрасли с почти тридцатилетним опытом в обработке изображений, научной визуализации, работе с распределенными системами и архитектурами веб-приложений, а также облачными платформами, Корнелия имеет степень бакалавра и магистра компьютерных наук от Калифорнийского государственного университета, в Нортридже. Она также изучала теорию алгоритмов и языки программирования в университете Индианы.

Учитель в глубине души, Корнелия провела последние тридцать лет, занимаясь созданием передового программного обеспечения и его разработкой.

В свободное от работы время она занимается йогой или готовит на кухне.

Об иллюстрации на обложке

Рисунок на обложке книги озаглавлен «Облачение русской повитухи в 1764 году». Иллюстрация взята из опубликованной между 1757 и 1772 годом в Лондоне книги Томаса Джефферис «Коллекция платьев разных народов, древних и современных» (четыре тома). На титульном листе указано, что это медные гравюры ручной работы, украшенные гуммиарабиком.

Томаса Джефферис (1719–1771) называли «географом короля Георга III». Он был английским картографом, ведущим создателем карт своего времени. Томас гравировал и печатал карты для правительственных и других государственных учреждений и выпускал обширный спектр коммерческих карт и атласов, особенно касающихся Северной Америки. Его работа в качестве картографа пробудила интерес к местному дресс-коду, блистательно представленному в этой коллекции. Он был принят в тех землях, которые исследовал и наносил на карту. Увлечение далекими землями и путешествия ради удовольствия были относительно новым явлением в конце XVIII века, и такие коллекции, как эта, были популярны, знакомя как туристов, так и путешественников, сидящих в креслах, с жителями других стран.

Разнообразие рисунков в этом издании Джефферис ярко свидетельствует об уникальности и индивидуальности народов мира около 200 лет назад. С тех пор дресс-код изменился, а богатое в ту пору разнообразие в зависимости от региона и страны исчезло. Сейчас часто трудно отличить жителей одного континента от другого. Возможно, пытаясь взглянуть на это с оптимизмом, мы обменяли культурное и визуальное разнообразие на более разнообразную личную жизнь – или на более разнообразную и интересную интеллектуальную и техническую жизнь.

В то время когда трудно отличить одну компьютерную книгу от другой, издательство Manning празднует изобретательность и инициативу компьютерного бизнеса с помощью обложек книг, основанных на богатом разнообразии жизни регионов двухвековой давности, которое ожило благодаря рисункам Джефферис.

Часть I

КОНТЕКСТ ОБЛАЧНОЙ СРЕДЫ

Возможно, это звучит как клише, первая часть книги подготавливает основу для дальнейшей работы. Полагаю, что можно было бы сразу перейти к шаблонам, о которых, я уверена, вам не терпится узнать (обнаружение служб, предохранители и т. д.), но я хочу, чтобы вы поняли эти шаблоны на очень глубоком уровне, поэтому эти первые главы необходимы. Понимание контекста, в котором будут работать ваши приложения, инфраструктуры, а также более человеческих элементов позволит вам применять шаблоны наиболее эффективным образом. Ожидания ваших клиентов от ваших цифровых продуктов (постоянное развитие и нулевое время простоя) и то, как вы и ваши коллеги разрабатываете эти продукты (наделенные полномочиями команды и отсутствием инцидентов), имеют отношение к шаблонам проектирования, о которых вы, возможно, даже и не догадывались.

Одна из главных вещей, которые я делаю в первой главе, – это определение понятия *cloud-native*, проводя различие между ним и термином *cloud* (внимание: спойлер! Последний термин касается вопроса *где*, а предыдущий – отвечает на вопрос *как*, что действительно интересно). Я также устанавливаю ментальную модель, вокруг которой организована вторая часть книги.

Вторая глава посвящена работе с приложениями для облачной среды. Я слышу, что некоторые из вас думают: «Я – разработчик, мне не нужно об этом беспокоиться», но, пожалуйста, забудьте на мгновение о своем недоверии. Операционные методы, отвечающие требованиям некоторых ваших клиентов, тотчас же транслируются на требования к вашему программному обеспечению.

И наконец, в третьей главе я расскажу о платформах, которые удовлетворяют потребности разработки и эксплуатации. Хотя многие из шаблонов, которые я рассматриваю во второй части книги, абсолютно необходимы для создания высококачественного программного обеспечения, они не должны быть реализованы вами в полной мере; правильная платформа может оказать вам большую помощь.

Так что если у вас есть соблазн пропустить первую часть, не делайте этого. Я обещаю, что вложенные сюда инвестиции окупятся позже.

Глава 1

Вы продолжаете использовать это слово: определение понятия «cloud-native»

Это не вина Amazon. В воскресенье 20 сентября 2015 года в платформе Amazon Web Services (AWS) произошел серьезный сбой. При растущем числе компаний, работающих с критически важными рабочими нагрузками на AWS, даже с основными сервисами, ориентированными на клиентов, сбой в работе AWS может привести к далеко идущим последующим системным сбоям. В этом случае Netflix, Airbnb, Nest, IMDb и другие испытали простой, что отразилось на их клиентах и в конечном итоге на их бизнесе. Основное отключение длилось около пяти часов (или более, в зависимости от того, как считать), что привело к еще более длительным сбоям для клиентов AWS, которых это затронуло, прежде чем их системы восстановились.

Если вы компания Nest, вы платите AWS, потому хотите сосредоточиться на создании эффективного результата для своих клиентов, а не на проблемах инфраструктуры. В качестве части данной сделки AWS отвечает за поддержание своих систем и за то, чтобы вы также поддерживали функционирование своей компании. Если у AWS возникают простои, можно легко обвинить в этом Amazon.

Но вы ошибаетесь. Компания Amazon не виновата в сбое.

Подождите! Не отбрасывайте эту книгу в сторону. Пожалуйста, выслушайте меня. Мое утверждение раскрывает суть вопроса и объясняет цели книги.

Во-первых, позвольте мне прояснить кое-что. Я не утверждаю, что Amazon и другие облачные провайдеры не несут ответственности за нормальное функционирование своих систем; очевидно, что это так. И если провайдер не соответствует определенным уровням обслуживания, его клиенты могут найти альтернативу, и они это сделают. Поставщики услуг обычно предоставляют соглашение об уровне предоставления услуги (SLA). Amazon, например, предоставляет 99,95 % гарантии бесперебойной работы для большинства своих услуг.

Я говорю о том, что приложения, которые вы запускаете в конкретной инфраструктуре, могут быть более стабильными, чем сама инфраструктура. Как такое возможно? Это, друзья мои, как раз то, чему научит вас эта книга.

Давайте на минутку вернемся к перебоям в работе AWS, произошедшим 20 сентября. Netflix, одна из множества компаний, затронутых перебоями, является то-