
Оглавление

Предисловие от издательства	12
Предисловие	13
Для кого создана эта книга	13
Почему мы написали эту книгу	13
Навигация по этой книге	14
Условные обозначения, используемые в этой книге	15
Использование примеров кода.....	15
Благодарности	16
Об авторах	17
Выходные данные (колофон).....	18
Глава 1. Сервисная сетка: вводный курс	19
Основные функции сетки	20
Безопасность	22
Надежность	23
Наблюдаемость	24
Как на самом деле работают сетки?	27
Так зачем же нам это нужно?	28
Заключение	29
Глава 2. Введение в Linkerd	30
Откуда взялась Linkerd?	30
Linkerd1	30
Linkerd2	30
Прокси Linkerd	32
Архитектура Linkerd	32
mTLS и сертификаты	33
Центры сертификации	34
Плоскость управления Linkerd	35
Расширения Linkerd	35
Linkerd Viz	35
Linkerd Multicluster	38
Linkerd Jaeger	39
Linkerd CNI	39
Linkerd SMI	40
Заключение	40

Глава 3. Развертывание Linkerd	41
Соображения.....	41
Версии Linkerd	41
Stable.....	41
Edge.....	42
Рабочие нагрузки, поды и сервисы	42
Сертификаты TLS	42
Linkerd Viz	43
Развертывание Linkerd.....	44
Необходимый инструментарий.....	44
Подготовка кластера Kubernetes	45
Установка Linkerd с помощью CLI	45
Команда linkerd install и сертификаты	45
Установка Linkerd с помощью Helm	46
Генерация сертификатов Linkerd	47
Установка Helm	48
Настройка Linkerd.....	49
Сети кластера.....	50
Ресурсы плоскости управления Linkerd	50
Непрозрачные и пропускаемые порты.....	50
Заключение	51
Глава 4. Добавление рабочих нагрузок в сетку	52
Рабочие нагрузки в сравнении с сервисами.....	52
Что означает добавление рабочей нагрузки в сетку?	53
Внедрение отдельных рабочих нагрузок	54
Добавление всех рабочих нагрузок в пространстве имен	55
Значения linkerd.io/inject	55
Почему вы можете решить не добавлять рабочую нагрузку в сетку?	55
Другие параметры настройки прокси.....	56
Определение протокола.....	56
Когда определение протокола не сработало	58
Непрозрачные порты в сравнении с пропускаемыми портами	58
Настройка определения протоколов.....	59
Непрозрачные порты по умолчанию.....	60
Ограничения ресурсов Kubernetes	60
Заключение	61
Глава 5. Входящий трафик и Linkerd	62
Контроллеры входа и Linkerd.....	64
Контроллер входа – это просто еще одна рабочая нагрузка в сетке.....	65
Linkerd (преимущественно) невидима	67
Используйте незашифрованный текст в кластере	67
Маршрут к сервисам, а не к конечным точкам	68
Ingress-режим	70

Конкретные примеры контроллеров входного трафика	72
Emissary-ingress	72
NGINX	72
Envoy Gateway	72
Заключение	73
Глава 6. CLI Linkerd	74
Установка CLI	74
Обновление CLI	74
Установка определенной версии	75
Альтернативные способы установки	75
Использование CLI	75
Некоторые команды	78
linkerd version.....	78
linkerd check	79
linkerd check --pre	79
linkerd check --проху	80
Проверки расширений Linkerd	80
Дополнительные опции для linkerd check	80
linkerd inject	81
Внедрение в режиме ingress	81
Добавление в ручном режиме	81
Добавление отладочного сайдкара	82
linkerd identify	82
linkerd diagnostics	83
Сбор метрик.....	83
Проверка конечных точек.....	84
Диагностика политик.....	85
Заключение	88
Глава 7. mTLS, Linkerd, и сертификаты	89
Безопасные коммуникации	90
TLS и mTLS	90
mTLS и сертификаты	92
Linkerd и mTLS	93
Сертификаты и Linkerd	93
Точка доверия Linkerd	95
Выпускающий удостоверение Linkerd	95
Сертификаты рабочей нагрузки Linkerd	96
Срок действия и ротация сертификатов.....	97
Управление сертификатами в Linkerd	99
Автоматическое управление сертификатами с помощью cert-manager	99
Установка cert-manager	99
Настройка cert-manager для Linkerd.....	101
Установка Linkerd с помощью cert-manager	104
Заключение	105

Глава 8. Политики Linkerd: обзор и серверная политика..... 106

Обзор политики Linkerd	106
Политика Linkerd по умолчанию	107
Ресурсы политики Linkerd	109
Server	109
HTTPRoute.....	110
MeshTLSAuthentication	110
NetworkAuthentication.....	110
AuthorizationPolicy	111
Политика на уровне сервера в сравнении с политикой на уровне маршрута	111
Политика на уровне сервера с приложением emoji voto	111
Настройка политики по умолчанию	111
Политика кластера по умолчанию	112
Политика по умолчанию для пространства имен.....	113
Настройка динамической политики	114
Изоляция пространства имен.....	115
Допуск Linkerd Viz	119
Блокировка по порту и идентификатору	121
Заключение	127

Глава 9. Политика Linkerd на уровне маршрутов..... 128

Политика на уровне маршрутов.....	128
Пример приложения booksapp	129
Установка booksapp	130
Настройка политики booksapp	131
Политика инфраструктуры	131
Доступ только для чтения	133
Дополнительные техники	139
Разрешение доступа на запись.....	140
Разрешение записи в books	142
Повторное включение генератора трафика	143
Заклучение	144

Глава 10. Наблюдение за вашей платформой**с помощью Linkerd 146**

Зачем это нужно?.....	146
Чем может помочь Linkerd?	146
Наблюдаемость в Linkerd	147
Настройка кластера	147
Тар	148
Профили сервисов	149
Настройка маршрутов для emoji voto	150
Создание маршрутов для booksapp.....	152
Топология.....	152

Linkerd Viz	153
Аудиторские списки и логи доступа.....	155
Ведение логов доступа: хорошее, плохое и ужасное.....	155
Хорошее	155
Плохое	156
Ужасное	156
Включение логов доступа	156
Заключение	156

Глава 11. Обеспечение надежности с помощью Linkerd 157

Балансировка нагрузки	157
Повторные запросы.....	158
Бюджеты повторных попыток.....	159
Настройка повторных попыток.....	159
Настройка бюджета	162
Тайм-ауты	163
Настройка тайм-аутов.....	163
Перенаправление трафика	166
Перенаправление трафика, Gateway API и расширение Linkerd SMI	166
Настройка вашей среды	166
Маршрутизация на основе веса (Canary).....	168
Service в сравнении с Service: ClusterIP, endpoint и HTTPRoute	170
Маршрутизация на основе заголовков (A/B Testing)	171
Подведение итогов по перенаправлению трафика	173
Автоматическое отключение.....	173
Включение автоматического отключения.....	173
Настройка автоматического отключения.....	174
Заключение	175

Глава 12. Мультикластерная коммуникация с помощью Linkerd 176

Типы мультикластерных установок	176
Мультикластер на основе шлюзов.....	176
Мультикластер Pod-to-Pod	177
Шлюзы в сравнении с Pod-to-Pod.....	178
Мультикластерные сертификаты	179
Определение межкластерных сервисов	179
Настройка мультикластера	180
Продолжаем использовать систему на основе шлюзов.....	183
Продолжаем использовать систему Pod-to-Pod	184
Мультикластерные проблемы	185
Развертывание и подключение приложения	186
Проверка трафика	188
Политики в мультикластерных средах	189
Заключение	189

Глава 13. Linkerd CNI в сравнении с контейнерами Init 190

Kubernetes без Linkerd	190
Узлы, поды и многое другое	190
Сетевое взаимодействие в Kubernetes	192
Роль пакетного фильтра	194
Сетевой интерфейс контейнера	196
Процесс запуска пода Kubernetes	196
Kubernetes и Linkerd	197
Способ с использованием контейнера init	197
Способ с использованием Linkerd CNI	198
Гонки и упорядочивание	199
Упорядочивание контейнеров	199
Упорядочивание плагинов CNI	200
Заключение	201

Глава 14. Развертывание Linkerd для промышленной эксплуатации..... 202

Ресурсы сообщества Linkerd	202
Получение помощи	202
Ответственное раскрытие информации	203
Совместимость с Kubernetes	203
Запуск производственного проекта с Linkerd	203
Stable или Edge?	203
Подготовка рабочей среды	203
Настройка Linkerd для повышенной доступности	204
Что дает режим HA?	205
Установка режима высокой доступности с помощью Helm	206
Установка высокой доступности с помощью CLI	207
Мониторинг Linkerd	208
Состояние и срок действия сертификата	208
Плоскость управления	208
Плоскость данных	209
Сбор метрик	209
Linkerd Viz для производственного использования	209
Доступ к логам Linkerd	211
Обновление Linkerd	212
Обновление с помощью Helm	213
Обновление с помощью CLI	213
Контрольный лист готовности	214
Заключение	215

Глава 15. Отладка Linkerd..... 216

Диагностика проблем плоскости данных	216
«Типичные» неполадки плоскости данных Linkerd	216
Поды не запускаются	216
Нерегулярные ошибки прокси	217
Настройка уровней логов прокси	220

Отладка плоскости управления Linkerd.....	220
Плоскость управления Linkerd и ее доступность	221
Ядро плоскости управления	221
Контроллер идентификации	221
Контроллер назначения.....	222
Прокси-инжектор	223
Расширения Linkerd	223
Заключение	224
Предметный указатель	225

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 последующие 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

Сервисным сеткам не мешает в некоторой степени реабилитация их сложившейся репутации.

В настоящее время некоторые специалисты по облачным технологиям представляют себе сервисные сетки как нечто пугающее и сложное, чего следует избегать до самого крайнего случая – вроде спасения погибающего приложения. Авторы хотели бы изменить это мнение: сервисные сетки – это невероятно эффективные инструменты, позволяющие значительно упростить разработку и эксплуатацию изначально облачных приложений по сравнению с другими вариантами.

И конечно же, мы считаем, что Linkerd – это лучшая в мире сервисная сетка для облегчения работы специалистов.

Так что если вы уже рвете на себе волосы в тщетных попытках разобраться с неправильно работающим приложением на основе одних только логов, либо потратили месяцы на безуспешный запуск какой-либо иной сетки и уже просто хотите, чтобы все работало, или же пытаетесь объяснить другому разработчику, почему ему действительно не следует беспокоиться о кодировании повторных передач и mTLS в своем микросервисе..., вы попали по адресу. Мы рады, что вы с нами.

Для кого создана эта книга

Эта книга предназначена для всех тех, кто полагает, что создание, запуск и отладку микросервисных приложений можно упростить, и кто надеется, что Linkerd поможет в этом. Мы считаем, что книга будет полезна всем, кто интересуется Linkerd как самостоятельным продуктом, однако Linkerd, как и любые другие информационные технологии, является средством, а не целью. Данная книга как раз и отражает эту концепцию.

Помимо этого, не имеет значения, являетесь ли вы разработчиком приложений, оператором кластера, инженером платформы или кем-либо еще; в этой книге вы найдете именно то, что поможет вам максимально эффективно использовать Linkerd. Наша цель – предоставить вам все необходимое для запуска и работы Linkerd, чтобы вы могли эффективно справляться со своими задачами.

Чтобы извлечь максимальную пользу из этой книги, вам понадобятся базовые знания о Kubernetes, общее представление о работе контейнеров и командной строке Unix. Некоторое знакомство с Prometheus, Helm, Jaeger и т. д. также не мешает, хотя и не является обязательным.

Почему мы написали эту книгу

Мы оба много лет проработали в сфере облачных технологий, а до этого еще дольше занимались разработкой программного обеспечения. На протяжении

всего этого времени одной из самых актуальных проблем была подготовка специалистов: даже самые передовые технологии не приносят пользы, пока пользователи не научатся правильно их применять на практике.

К настоящему моменту сервисные сетки достаточно хорошо распространены и освоены, но, как и раньше, каждый месяц появляются новички, которым нужно разобраться в последних и самых значительных изменениях в этой области, а также все больше тех, кто переходит в совершенно новый для них мир облачных технологий. Мы написали эту книгу и будем регулярно ее обновлять в помощь всем этим людям.

НАВИГАЦИЯ ПО ЭТОЙ КНИГЕ

Глава 1 «Сервисная сетка: вводный курс» представляет собой введение в тему сервисных сеток: что они из себя представляют, в чем могут помочь и почему вам может понадобиться их использовать. Эту главу просто необходимо прочитать тем, кто еще незнаком с сервисными сетками.

Глава 2 «Введение в Linkerd» подробно рассказывает об архитектуре и истории Linkerd. Если вы уже знакомы с Linkerd, эта глава будет для вас по большей части напоминанием уже известного.

Глава 3 «Развертывание Linkerd» и глава 4 «Добавление рабочих нагрузок в сетку» посвящены запуску Linkerd в кластере и настройке вашего приложения для работы с Linkerd. Эти две главы охватывают основные аспекты фактического использования Linkerd. Глава 5 «Ingress и Linkerd» продолжает обсуждение проблем Ingress, способов их решения и взаимодействия Linkerd с контроллерами Ingress.

Глава 6 «CLI Linkerd» посвящена интерфейсу уровня вызовов (CLI) Linkerd, который можно использовать для управления и анализа развертывания Linkerd.

Глава 7 «mTLS, Linkerd и сертификаты» подробно рассматривает протокол mTLS Linkerd и то, как он использует сертификаты X.509. Глава 8 «Политики Linkerd: обзор и серверная политика» и глава 9 «Политика Linkerd на уровне маршрутов» продолжают тему, описывая возможности использования идентификаторов mTLS в Linkerd для обеспечения соблюдения политики в вашем кластере.

Глава 10 «Наблюдение за вашей платформой с помощью Linkerd» посвящена механизмам обеспечения наблюдаемости приложений Linkerd. Глава 11 «Обеспечение надежности с помощью Linkerd», в свою очередь, рассказывает об использовании Linkerd для повышения надежности вашего приложения, а глава 12 «Мультикластерная коммуникация с помощью Linkerd» посвящена развертыванию сетки Linkerd на нескольких кластерах Kubernetes.

В главе 13 «Linkerd CNI в сравнении с контейнерами Init» рассматривается сложная тема принципов взаимодействия Linkerd с низкоуровневой сетевой конфигурацией вашего кластера. Эта тема может оказаться вполне актуальной, если вы планируете использовать Linkerd в производственной среде, что является темой главы 14 «Развертывание Linkerd для промышленной эксплуатации».

Наконец, в главе 15 «Отладка Linkerd» рассказывается о методах устранения неполадок в самой программе Linkerd в том случае, если вы столкнетесь с некорректной работой (хотя мы надеемся, что этого не произойдет!).

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ, ИСПОЛЬЗУЕМЫЕ В ЭТОЙ КНИГЕ

В этой книге используются следующие типографские нормы:

Курсив

Обозначает новые термины, URL-адреса, адреса электронной почты, имена и расширения файлов.

Моноширинный шрифт

Используется в листингах программ, а также внутри абзацев для обозначения таких элементов программы, как имена переменных или функций, базы данных, типы данных, переменные среды, операторы и ключевые слова.

Моноширинный курсив

Обозначает текст для замены на значения, вводимые пользователем, или на значения из контекста.



Этот элемент обозначает общее примечание.



Этот элемент обозначает предупреждение или оповещение о потенциальной опасности.

ИСПОЛЬЗОВАНИЕ ПРИМЕРОВ КОДА

Дополнительные материалы (примеры кода, упражнения и т. д.) доступны для скачивания по адресу <https://oreil.ly/linkerd-code>.

Если у вас есть технические вопросы или возникли проблемы с использованием примеров кода, пожалуйста, отправьте электронное письмо по адресу support@oreilly.com.

Эта книга призвана помочь вам в вашей работе. В общем случае, если в этой книге приводятся примеры кода, вы можете использовать их в своих программах и документации. Вам не нужно обращаться к нам за разрешением, если вы не воспроизводите значительную часть кода. Например, для написания программы, использующей несколько фрагментов кода из этой книги, разрешение не требуется. Для продажи или распространения примеров из книг O'Reilly разрешение требуется. Для ответа на вопрос с цитированием этой книги и примером кода разрешение не требуется. Для включения значительного количества примеров кода из этой книги в документацию вашего продукта разрешение требуется.

Мы ценим, но, как правило, не требуем указания авторства. Указание авторства обычно включает название, автора, издателя и ISBN. Например, «Linkerd: Up and Running» Джейсона Моргана и Флинна (O'Reilly). Copyright 2024 Джейсон Морган и Кевин Худ, 978-1-098-14231-5.

Если вы считаете, что использование вами примеров кода выходит за рамки добросовестного использования или указанного выше разрешения, свяжитесь с нами по адресу permissions@oreilly.com.

БЛАГОДАРНОСТИ

Огромное спасибо всем замечательным людям, которые помогли нам в создании этой книги, в том числе (но не только!):

- нашему редактору Анжеле Руфино (Angela Rufino);
- техническим рецензентам Дэниэлу Брайанту (Daniel Bryant), Бену Мушко (Ben Muschko) и Свапнилу Шевате (Swapnil Shevate), которые предоставили нам ценные отзывы и благодаря которым книга стала намного лучше;
- неизвестным героям из издательства O'Reilly, которые подготовили все к публикации;
- и наконец, что не менее важно, разработчикам Linkerd и замечательным людям из Buoyant, которые создали то, о чем мы пишем.

От Флинна огромная благодарность SC и RAN за то, что терпели его в течение года, который понадобился для создания этой книги. Большое, огромное спасибо.

Об авторах

Джейсон Морган (Jason Morgan) – практикующий специалист по DevOps, который неоднократно оказывал помощь многим организациям в освоении облачных технологий. Джейсон поддерживает работу команд по внедрению изначально облачных решений, чтобы они смогли предоставлять услуги своим клиентам и всегда были готовы к быстрому развитию. Джейсон выступал с докладами и написал ряд статей, он активно участвует в работе Cloud Native Computing Foundation (CNCF).

Флинн (Flynn) – технический евангелист в компании Buoyant, который занимается распространением информации о Linkerd, Kubernetes и в целом о разработке облачных приложений. Он также является автором и разработчиком API-шлюза Emissary-ingress, который входит в число проектов CNCF. Его карьера в сфере вычислений насчитывает более 40 лет и охватывает весь спектр от запуска с нуля на новом аппаратном оборудовании до распределенных приложений, с общим акцентом на коммуникациях и безопасности.

Выходные данные (колофон)

Животное на обложке книги «Linkerd: запуск и работа» – это омар европейский, или омар обыкновенный (*Homarus gammarus*).

Это ракообразное существо серо-голубого окраса обитает в восточной части Атлантического океана, от Норвегии на севере до Марокко, а также на протяжении большей части Средиземного моря и вдоль восточного побережья Черного моря. Хотя ареалы обитания европейского омара и американского омара не пересекаются, эти виды являются близкими родственниками.

Как и их американские сородичи, европейские омары являются всеядными животными, которые по ночам прочесывают морское дно в поисках беспозвоночных, таких как крабы, морские звезды, морские ежи и даже другие омары. Они используют свою большую клешню-дробилку, чтобы разбивать панцири и раковины, а меньшую клешню-разрыватель – чтобы рвать пищу на куски. Интересно, что «зубы» (или зубообразные структуры) омаров находятся в их желудках!

Омар играет важную роль в истории кулинарии на протяжении тысячелетий. Фрагмент клешни омара обыкновенного был найден среди кухонных отходов, обнаруженных при раскопках деревни VII века до н. э. недалеко от Сардинии, а в римской кулинарной книге Apicius (V век н. э.) есть рецепт крокетов из омара. Однако омары не всегда были дорогостоящим деликатесом. Когда первые европейские поселенцы прибыли в Северную Америку, омаров было так много, что, по некоторым данным, они вымывались на берег «кучами высотой в два фута». Из-за такого изобилия омары были дешевы и поэтому считались пищей для бедняков.

Сегодня омар европейский по-прежнему остается достаточно распространенным видом, с устойчивой популяцией и широким географическим ареалом. В связи с этим Международный союз охраны природы и природных ресурсов (МСОП) включил его в категорию видов, вызывающих наименьшую озабоченность с точки зрения охраны природы. Многие животные, изображенные на обложках книг O'Reilly, находятся под угрозой исчезновения; все они имеют большое значение для нашей планеты.

Иллюстрация на обложке выполнена Карен Монтгомери (Karen Montgomery) на основе старинной гравюры из книги «Животные» (Animals) издательства Dover. Дизайн серии разработали Эди Фридман (Edie Freedman), Элли Волкхаузен (Ellie Volckhausen) и Карен Монтгомери (Karen Montgomery). Шрифты на обложке – Gilroy Semibold и Guardian Sans. Шрифт текста – Adobe Minion Pro; шрифт заголовков – Adobe Myriad Condensed; шрифт кода – Dalton Maag Ubuntu Mono.

Сервисная сетка: ВВОДНЫЙ курс

Linkerd – это самая первая *сервисная сетка* (*service mesh*; в профильной литературе также иногда встречается перевод «сетка сервисов» или «сетка служб». – *Прим. перев.*). Более того, благодаря именно этому проекту был придуман и введен в обиход термин «сервисная сетка». Это решение было создано в 2015 году компанией Buoyant, Inc., о чем мы подробнее поговорим в главе 2, и все это время проект фокусировался на том, чтобы процесс развертывания и эксплуатации воистину замечательного облачного программного обеспечения был простым и удобным.

Но что же на самом деле представляет из себя сервисная сетка? Для начала обратимся к определению из словаря Фонда облачных вычислений (Cloud Native Computing Foundation, CNCF, дочерняя компания Linux Foundation. – *Прим. перев.*):

В сфере микросервисов приложения делятся на несколько небольших сервисов, которые взаимодействуют друг с другом через сеть. Как и в случае с Wi-Fi, компьютерные сети по своей сути небезопасны, их легко взломать, и они зачастую работают медленно. Сервисная сетка решает эту проблему благодаря управлению трафиком (то есть коммуникацией) между сервисами и добавлению единых функций надежности, наблюдаемости и безопасности во всех сервисах.

Мир облачных технологий – это вычисления в огромных масштабах, от крошечных кластеров для разработки на вашем ноутбуке до гигантских инфраструктур, которыми располагают Google и Amazon. Лучше всего это работает, когда приложения используют архитектуру микросервисов, но такая архитектура по своей сути более уязвима, чем целостная архитектура.

По сути, сервисная сетка скрывает эту уязвимость от разработчика приложения и фактически от самого приложения. Для этого в них реализовано несколько функций, критически важных для построения надежных приложений, которые перенесены из приложения в инфраструктуру. Это позволяет разработчикам приложений сосредоточиться на уникальных особенностях своих приложений, а не тратить все свое время на обеспечение критически важных функций, одинаковых для всех приложений.

В этой главе мы рассмотрим общие принципы работы сервисных сетей, их принципы функционирования и причины, по которым они так важны. В процессе мы предоставим вам необходимую базовую информацию для более подробного обсуждения Linkerd в остальной части книги.

ОСНОВНЫЕ ФУНКЦИИ СЕТКИ

Ключевые функции, предоставляемые сервисными сетками, можно разделить на три обширные категории: *безопасность (security)*, *надежность (reliability)* и *наблюдаемость (observability)*. Рассматривая эти три категории, мы будем сравнивать их работу в типичном монолитном приложении и в приложении на основе микросервисов.

Безусловно, термин «монолит» может иметь несколько разных значений. На рис. 1.1 показана схема «типичного» монолитного приложения, которое будет рассматриваться в этой главе.

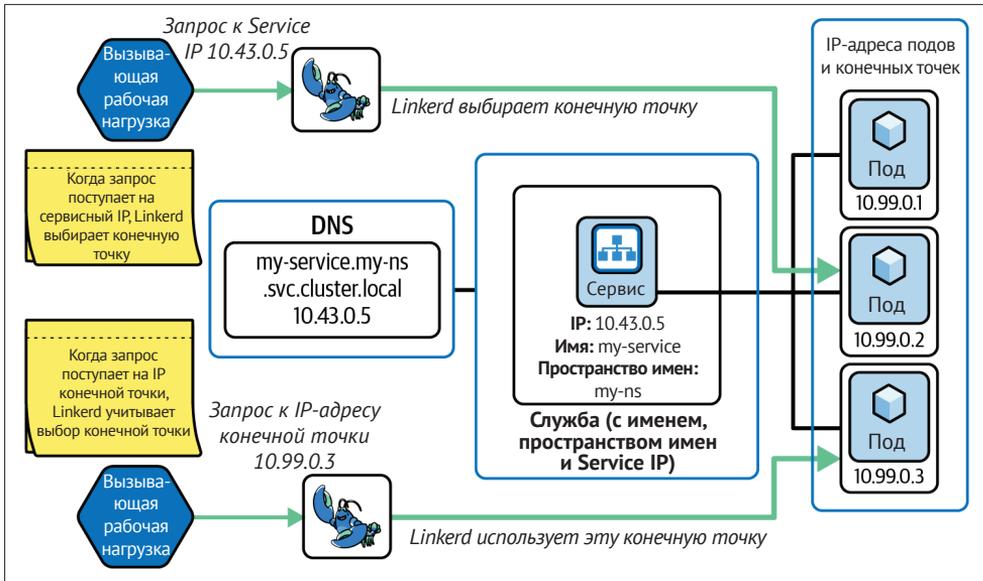


Рисунок 1.1. Монолитное приложение

Монолит представляет собой отдельный процесс в операционной системе, и это означает, что он может использовать все механизмы защиты, предлагаемые операционной системой; другие процессы не видят ничего внутри монолита и, конечно же, *определенно* не могут ничего в нем изменить. Обмен данными между различными частями монолита обычно осуществляется посредством вызовов функций в пределах единого пространства памяти монолита, поэтому у других процессов нет возможности видеть или изменять эти данные. Конечно, одна область монолита может изменять память, используемую другими частями, – на самом деле это огромный источник ошибок! Однако чаще всего это просто ошибки, а не атаки.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru