

УДК 004.042Kubernetes  
ББК 32.372  
В96

**Вьяс Дж., Лав К.**

В96 Kubernetes изнутри / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2023. – 378 с.: ил.

**ISBN 978-5-93700-153-5**

В этой книге подробно рассказывается о настройке и управлении платформой Kubernetes, а также о том, как быстро и эффективно устранять неполадки. Исследуется внутреннее устройство Kubernetes – от управления iptables до настройки динамически масштабируемых кластеров, реагирующих на изменение нагрузки. Советы профессионалов помогут вам поддерживать работоспособность ваших приложений. Особое внимание уделяется теме безопасности.

Книга адресована разработчикам и администраторам Kubernetes со средним уровнем подготовки.

УДК 004.042Kubernetes  
ББК 32.372

Copyright © DMK Press 2022. Authorized translation of the English edition © 2022 Manning Publications. This translation is published and sold by permission of Manning Publications, the owner of all rights to publish and sell the same.

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

ISBN 978-1-6172-9755-7 (англ.)  
ISBN 978-5-93700-153-5 (рус.)

© Manning Publications, 2022  
© Перевод, оформление, издание, ДМК Пресс, 2022

# Содержание

Оглавление .....	6
Предисловие .....	14
Благодарности .....	15
О книге .....	17
Об авторах .....	21
Об иллюстрации на обложке .....	23

<b>1 Почему появился Kubernetes .....</b>	<b>24</b>
1.1 Предварительный обзор некоторых основных терминов .....	25
1.2 Проблема дрейфа инфраструктуры и Kubernetes .....	26
1.3 Контейнеры и образы .....	27
1.4 Базовая основа Kubernetes .....	29
1.4.1 Все инфраструктурные правила в Kubernetes определяются в обычных файлах YAML .....	31
1.5 Возможности Kubernetes .....	32
1.6 Компоненты и архитектура Kubernetes .....	34
1.6.1 Kubernetes API .....	35
1.6.2 Пример первый: интернет-магазин .....	37
1.6.3 Пример второй: онлайн-решение для благотворительности .....	37
1.7 Когда не стоит использовать Kubernetes .....	38
Итоги .....	38

<b>2 Зачем нужны модули Pod? .....</b>	<b>40</b>
2.1 Пример веб-приложения .....	42
2.1.1 Инфраструктура нашего веб-приложения .....	44
2.1.2 Эксплуатационные требования .....	45
2.2 Что такое Pod? .....	46
2.2.1 Пространства имен в Linux .....	47
2.2.2 Kubernetes, инфраструктура и Pod .....	49
2.2.3 Объект Node .....	51
2.2.4 Наше веб-приложение и плоскость управления .....	55
2.3 Создание веб-приложения с помощью kubectl .....	56
2.3.1 Сервер Kubernetes API: kube-apiserver .....	57
2.3.2 Планировщик Kubernetes: kube-scheduler .....	58

2.3.3	Контроллеры инфраструктуры .....	59
2.4	Масштабирование, высокодоступные приложения и плоскость управления .....	63
2.4.1	Автоматическое масштабирование .....	65
2.4.2	Управление затратами .....	66
Итоги	.....	67

<b>3</b>	<b>Создание модулей Pod</b> .....	68
3.1	Общий обзор примитивов Kubernetes .....	71
3.2	Что такое примитивы Linux? .....	72
3.2.1	Примитивы Linux – это инструменты управления ресурсами .....	73
3.2.2	Все сущее является файлом (или файловым дескриптором) .....	74
3.2.3	Файлы можно комбинировать .....	75
3.2.4	Настройка kind .....	76
3.3	Использование примитивов Linux в Kubernetes .....	78
3.3.1	Предварительные условия для запуска модуля Pod .....	78
3.3.2	Запуск простого модуля Pod .....	79
3.3.3	Исследование зависимостей модуля Pod от Linux .....	81
3.4	Создание модуля Pod с нуля .....	86
3.4.1	Создание изолированного процесса с помощью chroot .....	87
3.4.2	Использование mount для передачи данных процессам .....	89
3.4.3	Защита процесса с помощью unshare .....	91
3.4.4	Создание сетевого пространства имен .....	92
3.4.5	Проверка работоспособности процесса .....	93
3.4.6	Ограничение потребления процессора с помощью cgroups .....	94
3.4.7	Создание раздела resources .....	95
3.5	Использование модуля Pod в реальном мире .....	96
3.5.1	Проблема сети .....	97
3.5.2	Как kube-проху реализует сервисы Kubernetes с помощью iptables ....	98
3.5.3	Использование модуля kube-dns .....	98
3.5.4	Другие проблемы .....	100
Итоги	.....	102

<b>4</b>	<b>Использование контрольных групп для управления процессами в модулях Pod</b> .....	103
4.1	Модули Pod простаивают до завершения подготовительных операций .....	104
4.2	Процессы и потоки в Linux .....	106
4.2.1	Процессы systemd и init .....	108
4.2.2	Контрольные группы для процессов .....	109
4.2.3	Реализация контрольных групп для обычного модуля Pod .....	112
4.3	Тестирование контрольных групп .....	114
4.4	Как kubelet управляет контрольными группами .....	115
4.5	Как kubelet управляет ресурсами .....	116
4.5.1	Почему ОС не может использовать подкачку в Kubernetes? .....	117
4.5.2	Хак: настройка приоритета «для бедных» .....	118
4.5.3	Хак: настройка HugePages с помощью контейнеров инициализации .....	119
4.5.4	Классы QoS: почему они важны и как они работают .....	120
4.5.5	Создание классов QoS путем настройки ресурсов .....	121

4.6	Мониторинг ядра Linux с помощью Prometheus, cAdvisor и сервера API.....	122
4.6.1	Публикация метрик обходится недорого и имеет большую ценность .....	124
4.6.2	Почему Prometheus? .....	125
4.6.3	Создание локального сервиса мониторинга Prometheus .....	126
4.6.4	Исследование простоя в Prometheus.....	129
Итоги	.....	131

## 5 Интерфейсы CNI и настройка сети в модулях Pod..... 132

5.1	Зачем нужны программно-определяемые сети в Kubernetes .....	134
5.2	Реализация Kubernetes SDN на стороне сервиса: kube-proxy.....	136
5.2.1	Плоскость данных в kube-proxy .....	138
5.2.2	Подробнее о NodePort .....	140
5.3	Провайдеры CNI .....	141
5.4	Два плагина CNI: Calico и Antrea .....	143
5.4.1	Архитектура плагинов CNI .....	143
5.4.2	Давайте поэкспериментируем с некоторыми CNI .....	144
5.4.3	Установка провайдера CNI Calico .....	146
5.4.4	Организация сети в Kubernetes с OVS и Antrea.....	149
5.4.5	Замечание о провайдерах CNI и kube-proxy в разных ОС .....	152
Итоги	.....	153

## 6 Устранение проблем в крупномасштабных сетях..... 154

6.1	Sonobuoy: инструмент подтверждения работоспособности кластера.....	155
6.1.1	Трассировка движения данных модулей Pod в кластере .....	156
6.1.2	Настройка кластера с CNI-провайдером Antrea .....	157
6.2	Исследование особенностей маршрутизации в разных провайдерах CNI с помощью команд <code>arp</code> и <code>ip</code> .....	158
6.2.1	Что такое IP-туннель и почему его используют провайдеры CNI? .....	159
6.2.2	Сколько пакетов проходит через сетевые интерфейсы CNI?.....	160
6.2.3	Маршруты .....	161
6.2.4	Инструменты для CNI: Open vSwitch (OVS).....	163
6.2.5	Трассировка движения данных активных контейнеров с помощью <code>tcpdump</code> .....	164
6.3	kube-proxy и iptables.....	166
6.3.1	iptables-save и diff.....	166
6.3.2	Как сетевые политики изменяют правила CNI .....	167
6.3.3	Как реализуются политики?.....	170
6.4	Входные контроллеры.....	172
6.4.1	Настройка Contour и кластера kind для изучения входных контроллеров .....	173
6.4.2	Настройка простого модуля Pod с веб-сервером .....	174
Итоги	.....	178

<b>7</b>	<b>Хранилища в модулях Pod и CSI</b>	179
7.1	Небольшое отступление: виртуальная файловая система (VFS) в Linux	181
7.2	Три вида хранилищ для Kubernetes	182
7.3	Создание PVC в кластере kind	184
7.4	Интерфейс контейнерного хранилища (CSI)	188
7.4.1	Проблема внутреннего провайдера	189
7.4.2	CSI как спецификация, работающая внутри Kubernetes	191
7.4.3	CSI: как работает драйвер хранилища	193
7.4.4	Привязка точек монтирования	193
7.5	Краткий обзор действующих драйверов CSI	194
7.5.1	Контроллер	194
7.5.2	Интерфейс узла	195
7.5.3	CSI в операционных системах, отличных от Linux	196
	Итоги	196
<b>8</b>	<b>Реализация и моделирование хранилищ</b>	198
8.1	Микрокосм в экосистеме Kubernetes: динамическое хранилище	199
8.1.1	Оперативное управление хранилищем: динамическое выделение ресурсов	200
8.1.2	Локальное хранилище в сравнении с emptyDir	201
8.1.3	Тома PersistentVolume	203
8.1.4	Интерфейс контейнерного хранилища (CSI)	204
8.2	Динамическая подготовка выигрывает от CSI, но не зависит от него	205
8.2.1	Классы хранилищ (StorageClass)	206
8.2.2	Вернемся к центрам обработки данных	207
8.3	Варианты организации хранилищ в Kubernetes	209
8.3.1	Секреты: эфемерная передача файлов	209
8.4	Как выглядит типичный провайдер динамического хранилища?	212
8.5	hostPath для управления системой и/или доступа к данным	214
8.5.1	hostPath, CSI и CNI: канонический вариант использования	214
8.5.2	Cassandra: пример реального хранилища в Kubernetes	217
8.5.3	Дополнительные возможности и модель хранения в Kubernetes	218
8.6	Дополнительная литература	219
	Итоги	220
<b>9</b>	<b>Запуск модулей Pod: как работает kubelet</b>	221
9.1	kubelet и узел	222
9.2	Основы kubelet	223
9.2.1	Среда выполнения контейнеров: стандарты и соглашения	224
9.2.2	Конфигурационные параметры и API агента kubelet	225
9.3	Создание модуля Pod и его мониторинг	228
9.3.1	Запуск kubelet	229
9.3.2	После запуска: жизненный цикл узла	230
9.3.3	Механизм аренды и блокировки в etcd, эволюция аренды узла	230
9.3.4	Управление жизненным циклом Pod в kubelet	231
9.3.5	CRI, контейнеры и образы: как они связаны	233
9.3.6	kubelet не запускает контейнеры: это делает CRI	233
9.3.7	Приостановленный контейнер: момент истины	235

9.4	Интерфейс времени выполнения контейнеров (CRI).....	235
9.4.1	Сообщаем Kubernetes, где находится среда выполнения контейнеров .....	235
9.4.2	Процедуры CRI .....	236
9.4.3	Абстракция kubelet вокруг CRI: GenericRuntimeManager .....	236
9.4.4	Как вызывается CRI? .....	237
9.5	Интерфейсы kubelet .....	237
9.5.1	Внутренний интерфейс среды выполнения .....	237
9.5.2	Как kubelet извлекает образы: интерфейс ImageService .....	239
9.5.3	Передача ImagePullSecret в kubelet .....	240
9.6	Дополнительная литература.....	241
Итоги	.....	241

<b>10</b>	<b>DNS в Kubernetes .....</b>	<b>243</b>
10.1	Краткое введение в DNS (и CoreDNS).....	243
10.1.1	NXDOMAIN, записи А и записи CNAME .....	244
10.1.2	Модулям Pod нужен внутренний DNS .....	246
10.2	Почему StatefulSet, а не Deployment?.....	248
10.2.1	DNS и автономные сервисы .....	248
10.2.2	Постоянные записи DNS в StatefulSet .....	250
10.2.3	Развертывание с несколькими пространствами имен для изучения свойств модуля DNS .....	250
10.3	Файл resolv.conf .....	252
10.3.1	Краткое примечание о маршрутизации.....	252
10.3.2	CoreDNS: вышестоящий сервер имен для ClusterFirst DNS.....	254
10.3.3	Разбор конфигурации плагина CoreDNS.....	255
Итоги	.....	256

<b>11</b>	<b>Плоскость управления .....</b>	<b>257</b>
11.1	Плоскость управления .....	258
11.2	Особенности сервера API .....	259
11.2.1	Объекты API и пользовательские ресурсы .....	259
11.2.2	Определения пользовательских ресурсов (CRD) .....	261
11.2.3	Планировщик .....	261
11.2.4	Краткий обзор фреймворка планирования .....	267
11.3	Диспетчер контроллеров .....	267
11.3.1	Хранилище.....	268
11.3.2	Учетные данные сервисов и токены .....	269
11.4	Облачные диспетчеры контроллеров Kubernetes (CCM) .....	269
11.5	Дополнительная литература.....	271
Итоги	.....	271

<b>12</b>	<b>etcd и плоскость управления .....</b>	<b>272</b>
12.1	Заметки для нетерпеливых .....	273
12.1.1	Мониторинг производительности etcd с помощью Prometheus .....	274
12.1.2	Когда нужно настраивать etcd .....	278
12.1.3	Пример: быстрая проверка работоспособности etcd .....	280
12.1.4	etcd v3 и v2 .....	280
12.2	etcd как хранилище данных .....	281
12.2.1	Можно ли запустить Kubernetes в других базах данных? .....	281

12.2.2	Строгая согласованность .....	283
12.2.3	Согласованность в etcd обеспечивают операции fsync .....	283
12.3	Обзор интерфейса Kubernetes с etcd .....	285
12.4	Задача etcd – надежное хранение фактов.....	285
12.4.1	Журнал упреждающей записи etcd .....	287
12.4.2	Влияние на Kubernetes .....	287
12.5	Теорема CAP.....	287
12.6	Балансировка нагрузки на уровне клиента и etcd .....	289
12.6.1	Ограничения по размеру: о чем (не) следует беспокоиться .....	289
12.7	Шифрование хранимых данных в etcd .....	291
12.8	Производительность и отказоустойчивость etcd в глобальном масштабе .....	292
12.9	Интервал отправки контрольных сообщений в высокораспределенной etcd .....	292
12.10	Настройка клиента etcd в кластере kind .....	293
12.10.1	Запуск etcd в окружении, отличном от Linux.....	294
Итоги	.....	295

## 13 Безопасность контейнеров и модулей Pod.....296

13.1	Радиус взрыва .....	297
13.1.1	Уязвимости .....	298
13.1.2	Вторжение .....	298
13.2	Безопасность контейнера .....	298
13.2.1	Планирование обновления контейнеров и пользовательского программного обеспечения .....	299
13.2.2	Контроль контейнеров .....	299
13.2.3	Пользователи в контейнерах – не запускайте ПО от имени root .....	300
13.2.4	Используйте наименьшие возможные контейнеры.....	300
13.2.5	Происхождение контейнера .....	301
13.2.6	Линтеры для контейнеров .....	302
13.3	Безопасность модулей Pod.....	302
13.3.1	Контекст безопасности .....	303
13.3.2	Расширение привилегий и возможностей .....	305
13.3.3	Политики безопасности Pod (PSP) .....	307
13.3.4	Не внедряйте автоматически токен учетной записи сервиса .....	309
13.3.5	Модули Pod с привилегиями root .....	309
13.3.6	Граница безопасности .....	310
Итоги	.....	311

## 14 Безопасность узлов и Kubernetes .....312

14.1	Безопасность узла.....	312
14.1.1	Сертификаты TLS.....	313
14.1.2	Неизменяемые ОС и применение исправлений на узлах .....	314
14.1.3	Изолированные среды выполнения контейнеров .....	315
14.1.4	Атаки на ресурсы .....	316
14.1.5	Единицы измерения потребления процессора .....	317
14.1.6	Единицы измерения объема памяти .....	317
14.1.7	Единицы измерения объема хранилища .....	318
14.1.8	Сети хостов и модулей Pod .....	318
14.1.9	Пример модуля Pod .....	319

14.2	Безопасность сервера API .....	320
14.2.1	Управление доступом на основе ролей (RBAC) .....	320
14.2.2	Определение RBAC API .....	321
14.2.3	Ресурсы и подресурсы .....	323
14.2.4	Субъекты и RBAC .....	325
14.2.5	Отладка RBAC .....	326
14.3	Authn, Authz и Secret .....	326
14.3.1	Учетные записи сервисов IAM: защита облачных API .....	327
14.3.2	Доступ к облачным ресурсам .....	328
14.3.3	Частные серверы API .....	329
14.4	Безопасность сети .....	329
14.4.1	Сетевые политики .....	330
14.4.2	Балансировщики нагрузки .....	334
14.4.3	Агент открытой политики (OPA) .....	335
14.4.4	Коллективная аренда .....	338
14.5	Советы по Kubernetes .....	341
Итоги	.....	341

<b>15</b>	<b>Установка приложений .....</b>	<b>343</b>
15.1	Размышления о приложениях в Kubernetes .....	344
15.1.1	Масштаб приложения влияет на выбор инструментов .....	345
15.2	Приложения на основе микросервисов обычно требуют тысячи строк определения конфигурации .....	345
15.3	Переосмысление установки приложения Guestbook в реальных условиях .....	346
15.4	Установка набора инструментов Carvel .....	347
15.4.1	Часть 1: разделение ресурсов на отдельные файлы .....	347
15.4.2	Часть 2: исправление файлов приложения с помощью ytt .....	349
15.4.3	Часть 3: развертывание приложения Guestbook и управление им .....	351
15.4.4	Часть 4: создание оператора kapp для упаковки приложения и управления им .....	355
15.5	И снова об операторах Kubernetes .....	359
15.6	Tanzu Community Edition: пример комплексного набора инструментов Carvel .....	362
Итоги	.....	363
	Предметный указатель .....	365