

*Моим родителям,
которые всегда ставили потребности своих детей
выше собственных*

Оглавление

Вступительное слово от компании ITSumma	19
Описанные/рассмотренные в книге ресурсы Kubernetes	20
Благодарности	22
Предисловие	23
Признательности	24
Об этой книге	25
Кто должен читать эту книгу	25
Как организована эта книга: дорожная карта	25
О коде	27
Книжный форум	28
Другие интернет-ресурсы	28
Об авторе	30
Об иллюстрации на обложке	31
Глава 1. Знакомство с Kubernetes.....	32
1.1 Объяснение необходимости системы наподобие Kubernetes.....	34
1.1.1 Переход от монолитных приложений к микросервисам	34
1.1.2 Обеспечение консистентного окружения для приложений	37
1.1.3 Переход к непрерывной доставке: DevOps и NoOps.....	38
1.2 Знакомство с контейнерными технологиями.....	40
1.2.1 Что такое контейнеры.....	40
1.2.2 Знакомство с контейнерной платформой Docker.....	44
1.2.3 Знакомство с rkt – альтернативой Docker.....	48
1.3 Первое знакомство с Kubernetes	49
1.3.1 Истоки	49
1.3.2 Взгляд на Kubernetes с вершины горы.....	50
1.3.3 Архитектура кластера Kubernetes	52
1.3.4 Запуск приложения в Kubernetes	53
1.3.5 Преимущества использования Kubernetes	55
1.4 Резюме	58
Глава 2. Первые шаги с Docker и Kubernetes	59
2.1 Создание, запуск и совместное использование образа контейнера	59
2.1.1 Установка Docker и запуск контейнера Hello World	60
2.1.2 Создание простейшего приложения Node.js	62
2.1.3 Создание файла Dockerfile для образа	63

2.1.4	Создание образа контейнера.....	64
2.1.5	Запуск образа контейнера	66
2.1.6	Исследование работающего контейнера изнутри	67
2.1.7	Остановка и удаление контейнера	69
2.1.8	Отправка образа в хранилище образов	70
2.2	Настройка кластера Kubernetes.....	71
2.2.1	Запуск локального одноузлового кластера Kubernetes с помощью Minikube	72
2.2.2	Использование кластера Kubernetes, предоставляемого как сервис с Google Kubernetes Engine	74
2.2.3	Настройка псевдонима и автозавершение в командной строке для kubectl.....	77
2.3	Запуск первого приложения на Kubernetes.....	78
2.3.1	Развертывание приложения Node.js	78
2.3.2	Доступ к веб-приложению.....	81
2.3.3	Логические части вашей системы.....	83
2.3.4	Горизонтальное масштабирование приложения.....	85
2.3.5	Исследование узлов, на которых запущено приложение.....	88
2.3.6	Знакомство с панелью управления Kubernetes.....	89
2.4	Резюме	91
Глава 3. Модули: запуск контейнеров в Kubernetes		92
3.1	Знакомство с модулями.....	92
3.1.1	Зачем нужны модули	93
3.1.2	Общее представление о модулях	94
3.1.3	Правильная организация контейнеров между модулями	96
3.2	Создание модулей из дескрипторов YAML или JSON.....	98
3.2.1	Исследование дескриптора YAML существующего модуля.....	99
3.2.2	Создание простого дескриптора YAML для модуля	101
3.2.3	Использование команды kubectl create для создания модуля	103
3.2.4	Просмотр журналов приложений	104
3.2.5	Отправка запросов в модуль	105
3.3	Организация модулей с помощью меток.....	106
3.3.1	Знакомство с метками	106
3.3.2	Указание меток при создании модуля.....	108
3.3.3	Изменение меток существующих модулей.....	109
3.4	Перечисление подмножеств модулей посредством селекторов меток.....	110
3.4.1	Вывод списка модулей с помощью селектора меток.....	110
3.4.2	Использование нескольких условий в селекторе меток.....	111
3.5	Использование меток и селекторов для ограничения планирования модулей.....	112
3.5.1	Использование меток для классификации рабочих узлов.....	113
3.5.2	Приписывание модулей к определенным узлам.....	113

3.5.3 Планирование размещения на один конкретный узел.....	114
3.6 Аннотирование модулей	114
3.6.1 Поиск аннотаций объекта.....	115
3.6.2 Добавление и изменение аннотаций.....	115
3.7 Использование пространств имен для группирования ресурсов.....	116
3.7.1 Необходимость пространств имен	116
3.7.2 Обнаружение других пространств имен и их модулей.....	117
3.7.3 Создание пространства имен	118
3.7.4 Управление объектами в других пространствах имен.....	119
3.7.5 Изоляция, обеспечиваемая пространствами имен	119
3.8 Остановка и удаление модулей.....	120
3.8.1 Удаление модуля по имени	120
3.8.2 Удаление модулей с помощью селекторов меток.....	120
3.8.3 Удаление модулей путем удаления всего пространства имен.....	121
3.8.4 Удаление всех модулей в пространстве имен при сохранении пространства имен.....	121
3.8.5 Удаление (почти) всех ресурсов в пространстве имен	122
3.9 Резюме	123

Глава 4. Контроллер репликации и другие контроллеры: развертывание управляемых модулей.....	124
4.1 Поддержание модулей в здоровом состоянии.....	125
4.1.1 Знакомство с проверками живости	126
4.1.2 Создание проверки живости на основе HTTP.....	126
4.1.3 Просмотр проверки живости в действии	127
4.1.4 Настройка дополнительных свойств проверки живости	129
4.1.5 Создание эффективных проверок живости	130
4.2 Знакомство с контроллерами репликации	132
4.2.1 Работа контроллера репликации	133
4.2.2 Создание контроллера репликации.....	135
4.2.3 Просмотр контроллера репликации в действии.....	136
4.2.4 Перемещение модулей в область и из области действия контроллера репликации.....	140
4.2.5 Изменение шаблона модуля	143
4.2.6 Горизонтальное масштабирование модулей	144
4.2.7 Удаление контроллера репликации.....	146
4.3 Использование набора реплик вместо контроллера репликации.....	147
4.3.1 Сравнение набора реплик с контроллером репликации	148
4.3.2 Определение набора реплик	148
4.3.3 Создание и исследование набора реплик.....	149
4.3.4 Использование более выразительных селекторов меток для набора реплик.....	150
4.3.5 Подведение итогов относительно наборов реплик	151

4.4	Запуск ровно одного модуля на каждом узле с помощью набора демонов (DaemonSet).....	151
4.4.1	Использование набора демонов для запуска модуля на каждом узле	152
4.4.2	Использование набора демонов для запуска модуля только на определенных узлах	153
4.5	Запуск модулей, выполняющих одну заканчиваемую задачу	156
4.5.1	Знакомство с ресурсом Job	156
4.5.2	Определение ресурса Job	157
4.5.3	Просмотр того, как задание управляет модулем	158
4.5.4	Запуск нескольких экземпляров модуля в задании	159
4.5.5	Ограничение времени, отпускаемого на завершение модуля задания... ..	161
4.6	Планирование выполнения заданий периодически или единоразово в будущем.....	161
4.6.1	Создание ресурса CronJob.....	161
4.6.2	Общие сведения о выполнении запланированных заданий	163
4.7	Резюме	163

Глава 5. Службы: обеспечение клиентов возможностью обнаруживать модули и обмениваться с ними информацией.....165

5.1	Знакомство со службами	166
5.1.1	Создание служб	167
5.1.2	Обнаружение служб.....	174
5.2	Подключение к службам, находящимся за пределами кластера.....	178
5.2.1	Знакомство с конечными точками служб.....	178
5.2.2	Настройка конечных точек службы вручную	179
5.2.3	Создание псевдонима для внешней службы.....	180
5.3	Предоставление внешним клиентам доступа к службам.....	181
5.3.1	Использование службы NodePort	182
5.3.2	Обеспечение доступа к службе через внешнюю подсистему балансировки нагрузки.....	186
5.3.3	Особенности внешних подключений	188
5.4	Обеспечение доступа к службам извне через ресурс Ingress.....	190
5.4.1	Создание ресурса Ingress	192
5.4.2	Доступ к службе через Ingress	193
5.4.3	Обеспечение доступа ко множеству служб через один и тот же Ingress....	194
5.4.4	Настройка Ingress для обработки трафика TLS	195
5.5	Сигналы о готовности модуля к приему подключений	198
5.5.1	Знакомство с проверкой готовности	198
5.5.2	Добавление в модуль проверки готовности	200
5.5.3	Что должны делать реальные проверки готовности	202
5.6	Использование служб без обозначенной точки входа (Headless-сервисов) для обнаружения индивидуальных модулей.....	203
5.6.1	Создание службы без обозначенной точки входа.....	204

5.6.2 Обнаружение модулей через DNS	204
5.6.3 Обнаружение всех модулей – даже тех, которые не готовы	206
5.7 Устранение неполадок в службах	206
5.8 Резюме	207

Глава 6. Тома: подключение дискового хранилища к контейнерам.....209

6.1 Знакомство с томами	210
6.1.1 Объяснение томов на примере	210
6.1.2 Знакомство с типами томов	212
6.2 Использование томов для обмена данными между контейнерами....	213
6.2.1 Использование тома emptyDir.....	213
6.2.2 Использование репозитория Git в качестве отправной точки для тома	217
6.3 Доступ к файлам в файловой системе рабочего узла	220
6.3.1 Знакомство с томом hostPath	220
6.3.2 Исследование системных модулей с томами hostPath.....	221
6.4 Использование постоянного хранилища	222
6.4.1 Использование постоянного диска GCE Persistent Disk в томе модуля ...	223
6.4.2 Использование томов других типов с базовым постоянным хранилищем.....	226
6.5 Отделение модулей от базовой технологии хранения	228
6.5.1 Знакомство с томами PersistentVolume и заявками PersistentVolumeClaim.....	229
6.5.2 Создание ресурса PersistentVolume.....	230
6.5.3 подача заявки на PersistentVolume путем создания ресурса PersistentVolumeClaim.....	231
6.5.4 Использование заявки PersistentVolumeClaim в модуле	234
6.5.5 Преимущества использования томов PersistentVolume и заявок.....	235
6.5.6 Повторное использование постоянных томов.....	236
6.6 Динамическое резервирование томов PersistentVolume	238
6.6.1 Определение доступных типов хранилища с помощью ресурсов StorageClass.....	238
6.6.2 Запрос на класс хранилища в заявке PersistentVolumeClaim.....	239
6.6.3 Динамическое резервирование без указания класса хранилища	241
6.7 Резюме	244

Глава 7. Словари конфигурации (ConfigMap) и секреты (Secret):

настройка приложений	246
7.1 Конфигурирование контейнерных приложений	246
7.2 Передача в контейнеры аргументов командной строки.....	248
7.2.1 Определение команды и аргументов в Docker	248
7.2.2 Переопределение команды и аргументов в Kubernetes	250
7.3 Настройка переменных среды для контейнера	252
7.3.1 Указание переменных среды в определении контейнера.....	253

7.3.2 Ссылка на другие переменные среды в значении переменной.....	254
7.3.3 Отрицательная сторона жесткого кодирования переменных среды	254
7.4 Отсоединение конфигурации с помощью словаря конфигурации ConfigMap.....	255
7.4.1 Знакомство со словарями конфигурации.....	255
7.4.2 Создание словаря конфигурации	256
7.4.3 Передача записи словаря конфигурации в контейнер в качестве переменной среды	258
7.4.4 Одновременная передача всех записей словаря конфигурации как переменных среды	260
7.4.5 Передача записи словаря конфигурации в качестве аргумента командной строки.....	261
7.4.6 Использование тома configMap для обеспечения доступа к записям словаря конфигурации в виде файлов.....	262
7.4.7 Обновление конфигурации приложения без перезапуска приложения	269
7.5 Использование секретов для передачи чувствительных данных в контейнерах.....	272
7.5.1 Знакомство с секретами.....	272
7.5.2 Знакомство с секретом default-token	273
7.5.3 Создание секрета.....	275
7.5.4 Сравнение словарей конфигурации и секретов	275
7.5.5 Использование секрета в модуле	277
7.5.6 Секреты для выгрузки образов.....	282
7.6 Резюме.....	283

Глава 8. Доступ к метаданным модуля и другим ресурсам из приложений

из приложений	285
8.1 Передача метаданных через нисходящий API.....	285
8.1.1 Доступные для использования метаданные	286
8.1.2 Предоставление доступа к метаданным через переменные среды	287
8.1.3 Передача метаданных через файлы в том downwardAPI	290
8.2 Обмен с сервером API Kubernetes	294
8.2.1 Исследование REST API Kubernetes.....	295
8.2.2 Обмен с сервером API изнутри модуля	300
8.2.3 Упрощение взаимодействия сервера API с контейнерами-посредниками	306
8.2.4 Использование клиентских библиотек для обмена с сервером API.....	308
8.3 Резюме	312

Глава 9. Развертывания: декларативное обновление приложений

9.1 Обновление приложений, работающих в модулях.....	314
9.1.1 Удаление старых модулей и замена их новыми	315
9.1.2 Запуск новых модулей, а затем удаление старых.....	315

9.2	Выполнение автоматического плавного обновления с помощью контроллера репликации	317
9.2.1	Запуск первоначальной версии приложения.....	317
9.2.2	Выполнение плавного обновления с помощью kubectl	319
9.2.3	Почему плавное обновление kubectl rolling-update устарело.....	324
9.3	Использование развертываний для декларативного обновления приложений.....	325
9.3.1	Создание развертывания.....	326
9.3.2	Обновление с помощью развертывания	329
9.3.3	Откат развертывания.....	333
9.3.4	Управление скоростью выкладки	336
9.3.5	Приостановка процесса выкладки.....	339
9.3.6	Блокировка раскруток плохих версий	340
9.4	Резюме	346

Глава 10. Ресурсы StatefulSet: развертывание реплицируемых приложений с внутренним состоянием	347
10.1 Репликация модулей с внутренним состоянием	347
10.1.1 Запуск множества реплик с отдельным хранилищем для каждой	348
10.1.2 Обеспечение стабильной долговременной идентификации для каждого модуля.....	350
10.2 Набор модулей с внутренним состоянием	351
10.2.1 Сопоставление наборов модулей с внутренним состоянием и наборов реплик	351
10.2.2 Обеспечение стабильной сетевой идентичности	353
10.2.3 Обеспечение стабильного выделенного хранилища для каждого экземпляра с внутренним состоянием.....	355
10.2.4 Гарантии набора StatefulSet	358
10.3 Использование набора StatefulSet	358
10.3.1 Создание приложения и образа контейнера.....	359
10.3.2 Развертывание приложения посредством набора StatefulSet	360
10.3.3 Исследование своих модулей	365
10.4 Обнаружение соседей в наборе StatefulSet	369
10.4.1 Реализация обнаружения соседей посредством DNS	371
10.4.2 Обновление набора StatefulSet.....	373
10.4.3 Опробование кластеризованного хранилища данных.....	374
10.5 Как наборы StatefulSet справляются с аварийными сбоями узлов ...	375
10.5.1 Симулирование отключения узла от сети	375
10.5.2 Удаление модуля вручную	377
10.6 Резюме	379
Глава 11. Внутреннее устройство Kubernetes	380
11.1 Архитектура.....	380

11.1.1	Распределенная природа компонентов Kubernetes.....	381
11.1.2	Как Kubernetes использует хранилище etcd	384
11.1.3	Что делает сервер API.....	388
11.1.4	Как сервер API уведомляет клиентов об изменениях ресурсов.....	390
11.1.5	Планировщик	391
11.1.6	Знакомство с контроллерами, работающими в менеджере контроллеров.....	394
11.1.7	Что делает агент Kubelet	399
11.1.8	Роль служебного сетевого прокси системы Kubernetes	401
11.1.9	Знакомство с надстройками Kubernetes	402
11.1.10	Все воедино	404
11.2	Взаимодействие контроллеров	404
11.2.1	Какие компоненты задействованы.....	404
11.2.2	Цепь событий.....	404
11.2.3	Наблюдение за событиями кластера.....	406
11.3	Что такое запущенный модуль.....	408
11.4	Интермодульное сетевое взаимодействие.....	409
11.4.1	Как должна выглядеть сеть	409
11.4.2	Более детальное рассмотрение работы сетевого взаимодействия.....	411
11.4.3	Знакомство с контейнерным сетевым интерфейсом	413
11.5	Как реализованы службы.....	414
11.5.1	Введение в kube-proxu.....	414
11.5.2	Как kube-proxu использует правила iptables	414
11.6	Запуск высокодоступных кластеров	416
11.6.1	Обеспечение высокой доступности приложений	416
11.6.2	Обеспечение высокой доступности компонентов плоскости управления Kubernetes	417
11.7	Резюме	421
Глава 12.	Защита сервера API Kubernetes	422
12.1	Аутентификация.....	422
12.1.1	Пользователи и группы.....	423
12.1.2	Знакомство с учетными записями службы	424
12.1.3	Создание учетных записей ServiceAccount.....	425
12.1.4	Назначение модулю учетной записи службы.....	428
12.2	Защита кластера с помощью управления ролевым доступом.....	430
12.2.1	Знакомство с плагином авторизации RBAC	430
12.2.2	Знакомство с ресурсами RBAC.....	432
12.2.3	Использование ролей и привязок ролей	435
12.2.4	Применение кластерных ролей (ClusterRole) и привязок кластерных ролей (ClusterRoleBinding)	439
12.2.5	Кластерные роли и привязки кластерных ролей, существующие по умолчанию	450

12.2.6 Предоставление разумных авторизационных разрешений	453
12.3 Резюме	454
Глава 13. Защита узлов кластера и сети	455
13.1 Использование в модуле пространств имен хоста	455
13.1.1 Использование в модуле сетевого пространства имен узла	456
13.1.2 Привязка к порту хоста без использования сетевого пространства имен хоста.....	457
13.1.3 Использование пространств имен PID и IPC узла.....	459
13.2 Конфигурирование контекста безопасности контейнера.....	460
13.2.1 Выполнение контейнера от имени конкретного пользователя.....	462
13.2.2 Недопущение работы контейнера в качестве root.....	462
13.2.3 Выполнение модулей в привилегированном режиме	463
13.2.4 Добавление отдельных функциональных возможностей ядра в контейнер.....	465
13.2.5 Удаление функциональных возможностей из контейнера.....	467
13.2.6 Запрет записи процессами в файловую систему контейнера.....	468
13.2.7 Совместное использование томов, когда контейнеры запущены под разными пользователями.....	469
13.3 Ограничение использования функциональности, связанной с безопасностью в модулях	472
13.3.1 Знакомство с ресурсами PodSecurityPolicy	472
13.3.2 Политики runAsUser, fsGroup и supplementalGroups	475
13.3.3 Конфигурирование разрешенных, стандартных и запрещенных возможностей	477
13.3.4 Ограничение типов томов, которые модули могут использовать.....	479
13.3.5 Назначение разных политик PodSecurityPolicy разным пользователям и группам	479
13.4 Изоляция сети модулей	483
13.4.1 Активация изоляции сети в пространстве имен.....	484
13.4.2 Разрешение подключения к серверному модулю только некоторых модулей в пространстве имен	484
13.4.3 Изоляция сети между пространствами имен Kubernetes.....	485
13.4.4 Изоляция с использованием обозначения CIDR	486
13.4.5 Лимитирование исходящего трафика набора модулей.....	487
13.5 Резюме	488
Глава 14. Управление вычислительными ресурсами модулей	489
14.1 Запрос на ресурсы для контейнеров модуля.....	489
14.1.1 Создание модулей с ресурсными запросами	490
14.1.2 Как ресурсные запросы влияют на назначение модуля узлу.....	491
14.1.3 Как запросы на ЦП влияют на совместное использование процессорного времени.....	496
14.1.4 Определяем и запрашиваем настраиваемые ресурсы	497

14.2	Лимитирование ресурсов, доступных контейнеру	498
14.2.1	Установка жесткого лимита на объем ресурсов, которые может использовать контейнер	498
14.2.2	Превышение лимитов	500
14.2.3	Как приложения в контейнерах видят лимиты	501
14.3	Классы QoS модулей	503
14.3.1	Определение класса QoS для модуля	503
14.3.2	Какой процесс уничтожается при нехватке памяти	506
14.4	Установка стандартных запросов и лимитов для модулей в расчете на пространство имен	508
14.4.1	Знакомство с ресурсом LimitRange	508
14.4.2	Создание объекта LimitRange	509
14.4.3	Обеспечение лимитов	511
14.4.4	Применение стандартных ресурсных запросов и лимитов	511
14.5	Лимитирование общего объема ресурсов, доступного в пространстве имен	512
14.5.1	Объект ResourceQuota	513
14.5.2	Указание квоты для постоянного хранилища	515
14.5.3	Лимитирование количества создаваемых объектов	515
14.5.4	Указание квот для конкретных состояний модулей и/или классов QoS	517
14.6	Мониторинг потребления ресурсов модуля	518
14.6.1	Сбор и извлечение фактических данных о потреблении ресурсов	518
14.6.2	Хранение и анализ исторической статистики потребления ресурсов	521
14.7	Резюме	524

Глава 15. Автоматическое масштабирование модулей

и узлов кластера	526	
15.1	Горизонтальное автомасштабирование модуля	527
15.1.1	Процесс автомасштабирования	527
15.1.2	Масштабирование на основе задействованности ЦП	531
15.1.3	Масштабирование на основе потребления памяти	538
15.1.4	Масштабирование на основе других, а также настраиваемых метрик	539
15.1.5	Определение метрик, подходящих для автомасштабирования	542
15.1.6	Уменьшение масштаба до нуля реплик	542
15.2	Вертикальное автомасштабирование модуля	543
15.2.1	Автоматическое конфигурирование ресурсных запросов	543
15.2.2	Модификация ресурсных запросов во время работы модуля	543
15.3	Горизонтальное масштабирование узлов кластера	544
15.3.1	Знакомство с кластерным автопреобразователем масштаба	544
15.3.2	Активация кластерного автопреобразователя масштаба	546

15.3.3 Ограничение прерывания службы во время уменьшения масштаба кластера	547
15.4 Резюме	548
Глава 16. Продвинутое назначение модулей узлам	550
16.1 Использование ограничений и допусков для отделения модулей от определенных узлов	550
16.1.1 Знакомство с ограничениями и допусками	551
16.1.2 Добавление в узел индивидуально настроенных ограничений	553
16.1.3 Добавление в модули допусков	554
16.1.4 Для чего можно использовать ограничения и допуски	555
16.2 Использование сходства узлов для привлечения модулей к определенным узлам	556
16.2.1 Указание жестких правил сходства узлов	557
16.2.2 Приоритизация узлов при назначении модуля	559
16.3 Совместное размещение модулей с использованием сходства и антисходства модулей	563
16.3.1 Использование межмодульного сходства для развертывания модулей на одном узле	563
16.3.2 Развертывание модулей в одной стойке, зоне доступности или географическом регионе	566
16.3.3 Выражение предпочтений сходства модулей вместо жестких требований	567
16.3.4 Назначение модулей на удалении друг от друга с помощью антисходства модулей	569
16.4 Резюме	571
Глава 17. Рекомендации по разработке приложений	572
17.1 Соединение всего вместе	572
17.2 Жизненный цикл модуля	574
17.2.1 Приложения должны ожидать, что они могут быть удалены и перемещены	574
17.2.2 Переназначение мертвых или частично мертвых модулей	577
17.2.3 Запуск модулей в определенном порядке	579
17.2.4 Добавление обработчиков жизненного цикла	581
17.2.5 Выключение модуля	586
17.3 Обеспечение правильной обработки всех клиентских запросов	589
17.3.1 Предотвращение прерывания клиентских подключений при запуске модуля	590
17.3.2 Предотвращение прерванных подключений при выключении модуля	590
17.4 Упрощение запуска приложений и управления ими в Kubernetes	595
17.4.1 Создание управляемых образов контейнеров	595

17.4.2 Правильное тегирование образов и рациональное использование политики imagePullPolicy	596
17.4.3 Использование многомерных меток вместо одномерных.....	596
17.4.4 Описание каждого ресурса с помощью аннотаций	597
17.4.5 Предоставление информации о причинах прекращения процесса	597
17.4.6 Работа с журналами приложений.....	599
17.5 Рекомендации по разработке и тестированию	601
17.5.1 Запуск приложений за пределами Kubernetes во время разработки...	601
17.5.2 Использование Minikube в разработке.....	603
17.5.3 Версионирование и автоматическое развертывание ресурсных манифестов.....	604
17.5.4 Знакомство с Ksonnet как альтернативой написанию манифестов YAML/JSON.....	605
17.5.5 Использование непрерывной интеграции и непрерывной доставки (CI/CD).....	606
17.6 Резюме.....	606
Глава 18. Расширение системы Kubernetes.....	608
18.1 Определение своих собственных объектов API	608
18.1.1 Знакомство с определениями CustomResourceDefinition.....	609
18.1.2 Автоматизация пользовательских ресурсов с помощью пользовательских контроллеров	613
18.1.3 Валидация пользовательских объектов.....	618
18.1.4 Предоставление пользовательского сервера API для пользовательских объектов	619
18.2 Расширение Kubernetes с помощью каталога служб Kubernetes (Kubernetes Service Catalog).....	621
18.2.1 Знакомство с каталогом служб.....	621
18.2.2 Знакомство с сервером API каталога служб и менеджером контроллеров.....	623
18.2.3 Знакомство с брокерами служб и API OpenServiceBroker	623
18.2.4 Резервирование и использование службы	625
18.2.5 Отвязывание и дерезервирование.....	628
18.2.6 Что дает каталог служб.....	629
18.3 Платформы, построенные поверх Kubernetes	629
18.3.1 Контейнерная платформа Red Hat OpenShift	629
18.3.2 Deis Workflow и Helm.....	633
18.4 Резюме	636
Приложение А. Использование kubectl с несколькими кластерами.....	637
А.1 Переключение между Minikube и Google Kubernetes Engine	637
А.2 Использование инструмента kubectl со множеством кластеров или пространств имен	638
А.2.1 Настройка расположения файла kubeconfig	638

A.2.2 Содержимое файла kubeconfig	638
A.2.3 Вывод списка, добавление и изменение записей в файле kubeconfig	640
A.2.4 Использование инструмента kubectl с разными кластерами, пользователями и контекстами	641
A.2.5 Переключение между контекстами	642
A.2.6 Перечисление контекстов и кластеров	642
A.2.7 Удаление контекстов и кластеров	642

Приложение В. Настройка многоузлового кластера

с помощью kubeadm	643
V.1 Настройка ОС и необходимых пакетов	643
V.1.1 Создание виртуальной машины	643
V.1.2 Настройка сетевого адаптера для виртуальной машины	644
V.1.3 Установка операционной системы	645
V.1.4 Установка Docker и Kubernetes	648
V.1.5 Клонирование виртуальной машины	650
V.2 Конфигурирование ведущего узла с помощью kubeadm	652
V.2.1 Как kubeadm запускает компоненты	653
V.3 Настройка рабочих узлов с помощью kubeadm	654
V.3.1 Настройка контейнерной сети	655
V.4 Использование кластера с локальной машины	656

Приложение С. Использование других контейнерных сред

выполнения	657
C.1 Замена Docker на rkt	657
C.1.1 Настройка Kubernetes для использования rkt	657
C.1.2 Опробирование платформы rkt с Minikube	658
C.2 Использование других контейнерных сред выполнения посредством CRI	660
C.2.1 Знакомство с контейнерной средой выполнения CRI-O	661
C.2.2 Запуск приложений на виртуальных машинах вместо контейнеров ..	661

Приложение D. Кластерная федерация

D.1 Знакомство с кластерной федерацией Kubernetes	662
D.2 Архитектура	663
D.3 Федеративные объекты API	663
D.3.1 Знакомство с федеративными версиями ресурсов Kubernetes	663
D.3.2 Что делают федеративные ресурсы	664

Предметный указатель

667

Вступительное слово от компании ITSumma

Любимой кофейной кружке придется подвинуться – эта книга займет центральное место на вашем столе и станет главным навигатором в контейнерных технологиях. Мы очень рады, что книга выходит на русском языке. После своего триумфального появления в 2014 году Kubernetes основательно утвердился в качестве лидирующей программы для оркестровки контейнеров в мире. Но в России до сих пор не было ресурса, который бы дал о нём исчерпывающую информацию.

Данная книга дает пошаговое разъяснение принципов работы контейнеризации и устройства модулей в Kubernetes. Вы узнаете все о создании объектов верхнего уровня, развертывании кластера Kubernetes на собственной рабочей машине и построении федеративного кластера в нескольких дата-центрах.

Автор книги – инженер-программист с двадцатилетним стажем Марко Лукша. Последние пять лет он курировал новые разработки Kubernetes для компании Red Hat и изучил все тонкости создания приложений в данной среде. Сложные рабочие схемы Лукша преподносит в очень занимательной и последовательной манере. Вы узнаете о трех способах обеспечить доступ к службам извне и разберете варианты перехода к процессу обновления с нулевым временем простоя. А также сравните размещение в Kubernetes классических контейнеризированных микросервисов и систем, использующих внутреннее состояние.

Нас приятно удивило, что Лукша уделяет особое внимание извечному спору сисадминов и разработчиков и в том числе рассказывает о практике NoOps, которая разрешает конфликты двух лагерей. Отдельное спасибо автору за то, что он детально анализирует задачи обеспечения безопасности в Kubernetes – крайне актуальную и малоисследованную на сегодня задачу.

Наглядные схемы и «разбор полетов» в книге создают впечатление индивидуального мастер-класса. Автор предвосхищает большинство вопросов, заранее объясняя, в каком направлении двигаться, если что-то пойдет не так. Те, кто только начал знакомство с платформой, смогут уверенно перейти к практической работе в Kubernetes. А «бывалые» специалисты узнают причины и пути решения для ранее безвыходных ситуаций.

Эта книга – отличный компас в увлекательном изучении облачных технологий, которые с каждым годом набирают все больше последователей. Надеемся увидеть вас в их числе.

Ваши ITSumma

Описанные/рассмотренные в книге ресурсы Kubernetes

	Ресурс (сокр.) [версия API]	Описание	Раздел
	Namespace* (ns) [v1]	Позволяет организовывать ресурсы в непере- крывающиеся группы (например, для каждого потребителя ресурсов)	3.7
Развертывающие рабочие нагрузки	Pod (po) [v1]	Основная развертываемая единица, содержа- щая один или более процессов в расположен- ных рядом контейнерах	3.1
	ReplicaSet (rs) [apps/v1beta2**]	Поддерживает одну или несколько реплик модуля	4.3
	ReplicationController (rc) [v1]	Более старый, менее мощный эквивалент ре- сурса ReplicaSet	4.2
	Job [batch/v1]	Запускает модули, выполняющие завершаемую задачу	4.5
	CronJob [batch/v1beta1]	Запускает назначаемое задание один раз или периодически	4.6
	DaemonSet (ds) [apps/v1beta2**]	Запускает одну реплику модуля в расчете на узел (на всех узлах или только на тех, которые соответствуют селектору узлов)	4.4
	StatefulSet (sts) [apps/v1beta1**]	Запускает модули, имеющие внутреннее состо- яние, со стабильной идентичностью	10.2
	Deployment (deploy) [apps/v1beta1**]	Декларативное развертывание и обновление модулей	9.3
Службы	Service (svc) [v1]	Предоставляет доступ к одному или нескольким модулям на одной и стабильной паре IP-адреса и порта	5.1
	Endpoints (ep) [v1]	Определяет, к каким модулям (или другим сер- верам) предоставляется доступ через службу	5.2.1
	Ingress (ing) [extensions/v1beta1]	Предоставляет внешним клиентам доступ к одной или нескольким службам через один до- ступный извне IP-адрес	5.4
Конфигурация	ConfigMap (cm) [v1]	Словарь в формате «ключ-значение» для хране- ния незащищенных параметров конфигурации приложений и предоставления им доступа к ним	7.4
	Secret [v1]	Как и словарь конфигурации ConfigMap, но для конфиденциальных данных	7.5
Хранение	PersistentVolume* (pv) [v1]	Указывает на постоянное хранилище, которое можно смонтировать в модуль посредством за- явки PersistentVolumeClaim	6.5
	PersistentVolumeClaim (pvc) [v1]	Запрос и заявка на PersistentVolume	6.5
	StorageClass* (sc) [storage.k8s.io/v1]	Определяет тип динамически резервируемого хранилища, заявляемого в PersistentVolumeClaim	6.6

(Окончание)

	Ресурс (сокр.) [версия API]	Описание	Раздел
Масштабирование	HorizontalPodAutoscaler (hpa) [autoscaling/v2beta1**]	Автоматически масштабирует количество реплик модулей на основе использования ЦП или другой метрики	15.1
	PodDisruptionBudget (pdb) [policy/v1beta1]	Определяет минимальное количество модулей, которые должны оставаться запущенными при эвакуации узлов	15.3.3
Ресурсы	LimitRange (limits) [v1]	Определяет мин, макс, ограничения и запросы по умолчанию (default) для модулей в пространстве имен	14.4
	ResourceQuota (quota) [v1]	Определяет объем вычислительных ресурсов, доступных для модулей в пространстве имен	14.5
Состояние кластера	Node* (net)[v1]	Представляет рабочий узел Kubernetes	2.2.2
	Cluster* [federation/v1beta1]	Кластер Kubernetes (используемый в федерации кластеров)	Прил. D
	ComponentStatus* (cs) [v1]	Статус компонента контрольной панели	11.1.1
	Event (ev) [v1]	Отчет о том, что произошло в кластере	11.2.3
Безопасность	ServiceAccount (sa) [v1]	Учетная запись, используемая приложениями, запущенными в модулях	12.1.2
	Role [rbac.authorization.k8s.io/v1]	Определяет, какие действия субъект может выполнять с какими ресурсами (в расчете на пространство имен)	12.2.3
	ClusterRole* [rbac.authorization.k8s.io/v1]	Как Role, но для ресурсов уровня кластера или для предоставления доступа к ресурсам во всех пространствах имен	12.2.4
	RoleBinding [rbac.authorization.k8s.io/v1]	Определяет, кто может выполнять действия, определенные в Role или ClusterRole (в пространстве имен)	12.2.3
	ClusterRoleBinding* [rbac.authorization.k8s.io/v1]	Как RoleBinding, но по всем пространствам имен	12.2.4
	PodSecurityPolicy* (psp) [extensions/v1beta1]	Ресурс уровня кластера, который определяет, какие чувствительные для безопасности особенности могут использовать модули	13.3.1
	NetworkPolicy (netpol) [networking.k8s.io/v1]	Изолирует сеть между модулями, указывая, какие модули могут подключаться друг к другу	13.4
CertificateSigningRequest* (csr) [certificates.k8s.io/v1beta1]	Запрос на подписание сертификата открытого ключа	5.4.4	
Расш.	CustomResourceDefinition* (crd) [apiextensions.k8s.io/v1beta1]	Определяет настраиваемый ресурс, позволяющий пользователям создавать экземпляры настраиваемого ресурса	18.1

* Ресурс уровня кластера (без пространства имен).

** Также в других версиях API; указанная версия использована в этой книге.

Благодарности

Издательство «ДМК Пресс» благодарит компанию «ITSumma» и ее генерального директора Евгения Потапова за неоценимую помощь в подготовке этой книги.

Без них вы, возможно, не так хорошо поняли бы Kubernetes, но теперь мы полностью уверены, что все термины стоят на своих местах.

Предисловие

После работы в Red Hat в течение нескольких лет, в конце 2014 года я был назначен в недавно созданную команду под названием Cloud Enablement. Нашей задачей было вывести линейку продуктов компании промежуточного уровня на OpenShift Container Platform, которая затем разрабатывалась поверх платформы Kubernetes. В то время Kubernetes все еще находилась в зачаточном состоянии – версия 1.0 еще даже не была выпущена.

Наша команда должна была узнать все входы и выходы Kubernetes, чтобы быстро установить правильное направление для нашего программного обеспечения и воспользоваться всем, что система Kubernetes должна была предложить. Когда мы сталкивались с проблемой, нам было трудно сказать, делаем ли мы что-то неправильно или просто попали на одну из ранних ошибок Kubernetes.

С тех пор и Kubernetes, и мое понимание этой платформы прошли долгий путь. Когда я впервые начал ее использовать, большинство людей даже не слышало о Kubernetes. Теперь об этой платформе знает практически каждый инженер-программист, и она стала одним из самых быстрорастущих и широко распространенных способов запуска приложений как в облаке, так и в локальных центрах обработки данных.

В первый месяц работы с Kubernetes я написал двухсоставной блог-пост о том, как запускать кластер сервера приложений Jboss WildFly в OpenShift/Kubernetes. В то время я совсем не мог себе представить, что простой пост в блоге в конечном итоге приведет к тому, что представители издательства Manning свяжутся со мной по поводу того, хочу ли я написать книгу о Kubernetes. Конечно, я не мог сказать «нет» такому предложению, хотя я был уверен, что они также обратились к другим специалистам и в конечном итоге выбрали кого-то другого.

И все же вот мы здесь. После более чем полутора лет написания и исследования книга закончена. Это было потрясающее путешествие. Написание книги о технологии является идеальным способом узнать ее гораздо подробнее, чем если вы узнаете ее как простой пользователь. По мере того как во время процесса написания книги мои знания о Kubernetes расширялись, а сама платформа Kubernetes эволюционировала, я постоянно возвращался к предыдущим, уже написанным главам и добавлял дополнительную информацию. Я перфекционист, поэтому никогда не буду абсолютно доволен книгой, но я рад узнать, что многие читатели программы раннего доступа Manning (MEAP) считают ее отличным путеводителем по Kubernetes.

Моя цель состоит в том, чтобы помочь читателю понять саму технологию и научить их использовать ее инструментарий для эффективной и действенной разработки и развертывания приложений в кластерах Kubernetes. В книге я не уделяю особого внимания тому, как на самом деле создавать и поддерживать надлежащий высокодоступный кластер Kubernetes, но последняя часть должна дать читателям очень четкое понимание того, из чего состоит такой кластер, и должна позволить им легко разбираться в дополнительных ресурсах, которые касаются этой темы.

Очень надеюсь, что вам понравится читать эту книгу и что она научит вас получать максимальную отдачу от удивительной системы под названием Kubernetes.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru