Р·СЕРВИС

Метод развертывания Kubernetes

Руководство по развертыванию платформы Р-Сервис в Kubernetes с использованием Helm.

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

Система Р-Сервис поддерживает установку в кластер Kubernetes[1], используя менеджер пакетов Helm[2].

Helm-чарт[3] устанавливает Р-Сервис в Kubernetes как набор приложений и рабочих процессов:

- r-service-app – основное приложение Ruby on Rails[4], Nginx – вспомогательный контейнер (sidecar), Clacks – компонент для опроса почтового сервера по протоколу IMAP[5], Delayed Job – система фоновых задач (воркеры[6]).
- r-service-faye – сервис реального времени Faye[7].
- r-service-pdf – сервис генерации PDF[8].
- r-service-sneakers – воркер для задач RabbitMQ[9]/Faye.
- Опциональные зависимости: реляционная база данных «PostgreSQL», система полнотекстового поиска и аналитики «OpenSearch», хранилище и брокер задач «Redis» для очередей, «Redis» для кэша, распределённый кэш «Memcached», брокер сообщений «RabbitMQ».

[1] Kubernetes – программная платформа с открытым исходным кодом, предназначенная для автоматизации развертывания, масштабирования и управления контейнеризированными приложениями.

[2] Helm - менеджер пакетов для платформы Kubernetes.

[3] Чарт - стандартный пакет, который содержит все необходимые шаблоны конфигураций и настройки для развертывания приложения.

[4] Ruby on Rails – серверная платформа с открытым исходным кодом.

[5] IMAP – сетевой протокол, который позволяет почтовому приложению получать доступ к письмам прямо на сервере.

[6] Воркеры – фоновые процессы или потоки, которые выполняют ресурсоемкие или длительные задачи, не замедляя работу основного интерфейса.

[7] Faye – масштабируемая система обмена сообщениями.

[8] PDF – формат электронных документов.

[9] RabbitMQ – брокер сообщений, который принимает, хранит и пересылает сообщения между различными компонентами системы.

2 ТРЕБОВАНИЯ

- Kubernetes 1.24+.
- Helm 3.2.0+.
- Доступ к реестру с образами «harbor.ops.r-service.tech/r-service-v1».
- Официальный стандарт «Gateway API CRD», если используются его встроенные стандартные ресурсы «HTTPRoute[10]».
- Расширение «Argo Rollouts CRD», если включается «канареечный релиз» («canary rollout») через «r-service-app.app.rollout.enabled=true».

Для среды «Прод»[11] (Prod) рекомендуется использовать внешние управляемые сервисы:

- PostgreSQL 17+.
- OpenSearch с плагинами[12] «analysis-icu» и «analysis-phonenumber».
- Redis, Memcached и RabbitMQ вне кластера или в отдельном отказоустойчивом контуре.
- S3[13]-совместимое хранилище с поддержкой загрузки файлов через HTTP POST[14] для пользовательских файлов. Если используется локальное хранилище в Kubernetes, настройте «r-service-app.storage.pvc».

Встроенные зависимости чарта удобны для стендов разработки, демонстрации (демо) и быстрого старта, но не являются рекомендуемым вариантом для среды «Прод».

[10] HTTPRoute – объект маршрутизации в Kubernetes.
[11] Прод – основная среда, где функционирует программное обеспечение или веб-сервис.
[12] Плагин – независимый программный модуль, который динамически подключается к базовой платформе для расширения или изменения ее возможностей.
[13] S3 – стандарт облачного объектного хранения.
[14] HTTP POST – метод, предназначенный для отправки данных на сервер.

3 УСТАНОВКА

1. Создайте «пространство имен» («namespace»):

```
kubectl create namespace r-service
```

2. Заполните значения в своем конфигурационном файле «values.yaml».

3. Установите чарт из реестра OCI[15]:

```
helm install r-service \
  oci://harbor.ops.r-service.tech/r-service-helm/r-service-
helm \
  --namespace r-service \
  -f values.yaml
```

Обновление существующего релиза:

```
helm upgrade r-service \
  oci://harbor.ops.r-service.tech/r-service-helm/r-service-
helm \
  --namespace r-service \
  -f values.yaml
```

[15] OCI – проект, создающий единые отраслевые стандарты для упаковки и запуска приложений в контейнерах.

4 МИНИМАЛЬНЫЙ КОНФИГУРАЦИОННЫЙ ФАЙЛ «VALUES.YAML»

Ниже пример минимальной «Прод»-конфигурации с внешними сервисами и заранее созданными секретами Kubernetes. Имена хостов[16], домены, адреса почтовых ящиков и названия секретов нужно заменить на значения заказчика.

```
r-service-app:
  domain: service.example.ru
  ssl: true

  httproute:
    enabled: true
    parentRefs:
      - name: public-gateway
        namespace: gateway
        sectionName: https

  database:
    writer:
      host: postgres-writer.example.ru
      port: 5432
      existingSecret:
        name: r-service-postgres-writer
        usernameKey: username
        passwordKey: password
        databaseKey: database
    reader:
      host: postgres-reader.example.ru
      port: 5432
      existingSecret:
        name: r-service-postgres-reader
        usernameKey: username
        passwordKey: password
        databaseKey: database
    ssl:
      mode: require

  redisCache:
    host: redis-cache.example.ru
    port: "6379"
```

[16] Хост – любое устройство, подключенное к сети и имеющее свой уникальный адрес.

```
  existingSecret:
    name: r-service-redis-cache
    passwordKey: password

memcached:
  host: memcached.example.ru
  port: "11211"

rabbitmq:
  host: rabbitmq.example.ru
  port: "5672"
  existingSecret:
    name: r-service-rabbitmq
    usernameKey: username
    passwordKey: password

opensearch:
  urls: https://opensearch.example.ru:9200
  existingSecret:
    name: r-service-opensearch
    usernameKey: username
    passwordKey: password

smtp:
  host: smtp.example.ru
  port: "587"
  domain: service.example.ru
  authentication: login
  existingSecret:
    name: r-service-smtp
    usernameKey: username
    passwordKey: password

imap:
  host: imap.example.ru
  port: "993"
  enableSsl: true
  mailbox: support@example.ru
  archivebox: archive@example.ru
  existingSecret:
    name: r-service-imap
```


```
      usernameKey: username
      passwordKey: password

  mailboxes:
    errors: errors@example.ru
    info: info@example.ru
    debug: debug@example.ru
    noreply: noreply@example.ru
    support: support@example.ru
    bounced: bounced@example.ru

  secrets:
    secret:
      create: false
    existingSecret:
      name: r-service-base
      cookieSecretTokenKey: cookieSecretToken
      otpSecretKey: otpSecret
      otpSaltKey: otpSalt
      storageLocalSecretKey: storageLocalSecret
      memcachedSecretKey: memcachedSecret
      fayeSecretKey: fayeSecret
      jwtGlobalCertKey: jwtGlobalCert

r-service-faye:
  redis:
    host: redis-cache.example.ru
    port: "6379"
    existingSecret:
      name: r-service-redis-cache
      passwordKey: password
  faye:
    existingSecret:
      name: r-service-base
      secretTokenKey: fayeSecret
  httproute:
    enabled: true
    parentRefs:
      - name: public-gateway
        namespace: gateway
        sectionName: https
```


```
r-service-sneakers:
  faye:
    publishUrl: https://realtime.service.example.ru/faye
    existingSecret:
      name: r-service-base
      secretTokenKey: fayeSecret
  rabbitmq:
    existingSecret:
      name: r-service-rabbitmq-url
      urlKey: url
```

Если секреты создаются чартом, вместо «existingSecret.name» включите «secret.create=true» и передайте значения в «secret.*». Для среды «Прод» предпочтительнее создавать секрет отдельно через корпоративный механизм управления секретами.

5 ОСНОВНЫЕ ПАРАМЕТРЫ ЧАРТА

Данной список не является полным – рекомендуется обратиться к схеме значений («values.schema.json») для полной документации. За более подробным описанием назначения параметров необходимо обратиться к основному руководству системного администратора.

5.1 Обязательные параметры

Параметр	Назначение
r-service-app.domain	Публичный домен приложения
r-service-app.database.writer.host	Host PostgreSQL writer[17]
r-service-app.database.reader.host	Host PostgreSQL reader[18]
r-service-app.smtp.host	SMTP[19] хост
r-service-app.smtp.port	SMTP порт
r-service-app.smtp.domain	Домен, передаваемый в команде «HELO» («SMTP HELO/domain»)
r-service-app.smtp.authentication	Тип SMTP-аутентификации, по умолчанию «login»[20]
r-service-app.mailboxes.*	Служебные адреса электронной почты: сообщения об ошибках (errors), информационные сообщения (info), технические сообщения для отладки (debug), ящик «без ответа» (noreply), поддержка (support), «вернувшиеся» сообщения (bounced).
r-service-app.pdf.url	URL[21] сервиса PDF, по умолчанию http://r-service-pdf

Учетные данные (credentials) PostgreSQL, учетные данные (credentials) SMTP, основные секреты приложения (base application secrets), учетные данные (credentials) OpenSearch, учетные данные (credentials) RabbitMQ и пароль (password) Redis задаются через «existingSecret» или через «secret.create=true».

[17] Host PostgreSQL writer – основной узел (экземпляр), который обладает полными правами на чтение и запись данных.

[18] Host PostgreSQL reader – выделенный узел базы данных, который используется только для чтения информации.

[19] SMTP – сетевой протокол для отправки и маршрутизации электронной почты.

[20] Метод аутентификации с передачей логина и пароля в два этапа.

[21] URL – уникальный адрес любого ресурса в интернете.

5.2 Главный чарт (Root chart)

Параметр	Назначение
postgres.enabled	Устанавливать схему зависимостей (dependency chart) PostgreSQL
opensearch.enabled	Устанавливать схему зависимостей (dependency chart) OpenSearch
redis-queue.enabled	Устанавливать Redis для очередей
redis-cache.enabled	Устанавливать Redis для кэша/Faye
memcached.enabled	Устанавливать схему зависимостей (dependency chart) Memcached
rabbitmq.enabled	Устанавливать схему зависимостей (dependency chart) RabbitMQ

5.3 r-service-app

Параметр	По умолчанию	Описание
r-service-app.app.replicaCount	1	Количество под[22] основного веб-приложения.
r-service-app.clacks.enabled	истина (true)	Включить периодический опрос IMAP (IMAP polling workload).
r-service-app.delayedJob.*.enabled	истина (true)	Включить отдельные очереди «Delayed Job»: периодически, срочно, быстро, нормально, уведомления, медленно.
r-service-app.delayedJobSingle.enabled	ложь (false)	Использовать один воркер для всех очередей вместо набора отдельных воркеров.
r-service-app.migrationJob.enabled	истина (true)	Рендерить[23] Helm hook job[24] для настройки базы данных (БД).
r-service-app.esImportAllJob.enabled	истина (true)	Рендерить Helm hook job для импорта индексов OpenSearch.
r-service-app.storage.pvc.enabled	ложь (false)	Подключить PVC[25] для защищенного хранилища (protected storage).
r-service-app.httproute.enabled	истина (true)	Создавать HTTPRoute для веб-приложения.
r-service-app.app.rollout.enabled	ложь (false)	Использовать Argo Rollouts[26] вместо объекта развертывания (Deployment) для веб-приложения.
r-service-app.extraEnvs	[]	Дополнительные переменные окружения (env-переменные) для компонентов «app», «jobs» и «workers».

[22] Под – минимальная и неделимая единица развертывания.
[23] Рендерить – преобразовывать, превращать абстрактное в визуальное.
[24] Helm hook job – инструмент для автоматизации развертывания в Kubernetes.
[25] PVC – запрос на использование постоянного тома в Kubernetes.
[26] Argo Rollouts – контроллер для продвинутого развертывания.

5.4 r-service-faye

Параметр	По умолчанию	Описание
r-service-faye.enabled	истина (true)	Включить рабочую нагрузку в реальном времени (realtime workload).
r-service-faye.replicaCount	1	Количество под для Faye.
r-service-faye.redis.host	""	Redis хост для Faye.
r-service-faye.faye.existingSecret.name	""	Секрет с секретным токеном[27] (Secret с secretToken).
r-service-faye.httproute.enabled	истина (true)	Создавать HTTPRoute для /faye.

5.5 r-service-pdf

Параметр	По умолчанию	Описание
r-service-pdf.enabled	истина (true)	Включить генератор PDF.
r-service-pdf.replicaCount	1	Количество под для генератора PDF.

5.6 r-service-sneakers

Параметр	По умолчанию	Описание
r-service-sneakers.enabled	истина (true)	Включить воркер Sneakers[28].
r-service-sneakers.replicaCount	1	Количество под для Sneakers.
r-service-sneakers.faye.publishUrl	""	Адрес публикации для сервиса реального времени (Publish URL realtime-сервиса).
r-service-sneakers.rabbitmq.existingSecret.name	""	Секрет с адресом подключения к RabbitMQ (Secret с RabbitMQ connection URL).

[27] Токен – уникальный цифровой ключ или маркер.
[28] Sneakers – библиотека для работы с очередями сообщений.

6 INGRESS[29] И GATEWAY API[30]

Чарт создает HTTPRoute, а не классический Kubernetes Ingress.

Для веб-приложения настройте:

```
r-service-app:
  httproute:
    enabled: true
    parentRefs:
      - name: public-gateway
        namespace: gateway
        sectionName: https
```

Для сервиса реального времени настройте аналогичный parentRefs[31] в «r-service-faye.httproute». Маршрут Faye обслуживает путь /faye.

Если в кластере используется другой ingress-контроллер или маршрутизация создается внешней «инфраструктурой как кодом» (IaC), отключите генерацию маршрута (route):

```
r-service-app:
  httproute:
    enabled: false
r-service-faye:
  httproute:
    enabled: false
```

[29] Ingress – объект, управляющий входящим сетевым трафиком в Kubernetes.

[30] Gateway api – стандартизированный набор интерфейсов для управления входящим и внутренним сетевым трафиком в Kubernetes.

[31] ParentRefs – ссылки на родительские объекты маршрутизации.

7 СЕКРЕТЫ

Чарт поддерживает два режима:

- «existingSecret» – использовать заранее созданный секрет Kubernetes.
- «secret.create=true» – создать секрет из значений Helm.

Для среды «Прод» рекомендуется режим «existingSecret», чтобы не хранить чувствительные данные в файле «values.yaml» и истории «Helm release»[32].

Базовый секрет приложения должен содержать ключи:

- cookieSecretToken
- otpSecret
- otpSalt
- storageLocalSecret
- memcachedSecret
- fayeSecret
- jwtGlobalCert

Если оставить «r-service-app.secrets.secret.create=true», чарт сгенерирует базовые значения автоматически и при последующих обновлениях попытается переиспользовать уже существующий секрет релиза.

[32] Helm release – установленный чарт Helm с конкретными значениями.

8 ХРАНИЛИЩЕ ФАЙЛОВ

По умолчанию «r-service-app.storage.pvc.enabled=false».

ВАЖНО! Необходимо использование запроса на выделение постоянного хранилища (RWM PVC).

Для подключения существующего PVC:

```
r-service-app:
  storage:
    pvc:
      enabled: true
      claimName: r-service-storage
```

Для создания PVC чартом:

```
r-service-app:
  storage:
    pvc:
      enabled: true
      create: true
      claimName: r-service-storage
      spec:
        accessModes:
          - ReadWriteMany
        resources:
          requests:
            storage: 100Gi
```

Для среды «Прод» предпочтительно использовать отказоустойчивое внешнее объектное хранилище.

9 МАСШТАБИРОВАНИЕ И ОТКАЗОУСТОЙЧИВОСТЬ

Рекомендуемые настройки для среды «Прод»:

- Увеличить количество реплик (replicaCount) минимум до 2 для «r-service-app.app», «r-service-faye», критичных воркеров Delayed Job и «r-service-pdf», если нагрузка требует параллельной обработки.
- Включить «podDisruptionBudget.enabled=true» для рабочих нагрузок (workloads) с несколькими репликами.
- Настроить «resources.requests» и «resources.limits» по фактической нагрузке.
- Разнести поды по узлам через механизмы привязки пода к узлам кластера (affinity, nodeSelector, tolerations) или ограничением на распределение подов по топологии (topologySpreadConstraints).
- Не использовать встроенный PostgreSQL/OpenSearch/RabbitMQ/Redis для «Прод».

10 КАНАРЕЕЧНОЕ РАЗВЕРТЫВАНИЕ (CANARY ROLLOUT)

Для веб-приложения можно включить Argo Rollouts:

```
r-service-app:
  app:
    rollout:
      enabled: true
      steps:
        - setWeight: 10
        - pause: {}
        - setWeight: 50
        - pause: {}
        - setWeight: 100
```

Перед включением проверьте, что в кластере установлены Argo Rollouts CRD[33] и настроена маршрутизация трафика Gateway API.

[33] CRD – механизм расширения Kubernetes.