

Функциональное описание Q.Runner

Exported on Jul 08, 2019

1	Возможности	4
1.1	Реестр сервисов и балансировка нагрузки.....	4
1.2	Устойчивость	4
1.3	Управление секретами и конфигурацией.....	4
1.4	Масштабирование	4
1.5	Автоматическое развертывание и откат.....	4
2	Состав Q.Runner	5
2.1	Master компоненты	5
2.1.1	qrun-apiserver	5
2.1.2	etcd.....	5
2.1.3	qrun-scheduler	5
2.1.4	qrun-controller-manager.....	5
2.2	Компоненты рабочих узлов	5
2.2.1	qrunlet	5
2.2.2	qrun-proxu	6
2.2.3	Среда исполнения Контейнера	6

Q.Runner - это программное обеспечение для автоматизации развертывания, масштабирования и управления контейнеризированными приложениями. Поддерживает основные технологии контейнеризации.

Q.Runner представляет из себя:

- Контейнерная платформа;
- Платформа микросервисов;
- Портативная облачная платформа.

Благодаря современным веб-службам пользователи ожидают, что приложения будут доступны круглосуточно, а разработчики ожидают развертывания новых версий этих приложений несколько раз в день.

Контейнеризация помогает пакетировать программное обеспечение для достижения этих целей, позволяя выпускать и обновлять приложения простым и быстрым способом без простоев.

Q.Runner помогает вам убедиться, что эти контейнеризованные приложения работают где и когда выхотите, и помогает им находить ресурсы и инструменты, необходимые для их работы.

Q.Runner представляет контейнерно - ориентированную среду управления. Он организует вычислительную, сетевую инфраструктуру и инфраструктуру хранения данных от пользовательских рабочих процессов.

Это обеспечивает предоставление платформы как услуги (PaaS), с гибкостью инфраструктуры как услуги (SaaS), и обеспечивает переносимость приложений между поставщиками инфраструктуры.

Почему Q.Runner является платформой?

Несмотря на то, что Q.Runner предоставляет множество встроенных функций и возможностей, всегда может возникнуть потребность в новых сценариях.

Рабочие процессы для конкретных приложений можно упростить, чтобы ускорить разработку. Специальная оркестровка, которой изначально было достаточно, может потребовать улучшения при работе под значительной нагрузкой.

Именно поэтому Q.Runner был так же разработан в качестве платформы для построения экосистемы компонентов и инструментов, призванных упростить развертывание, масштабирование и управление приложениями.

Так же, платформа предоставляет API, которое доступно пользователям и разработчикам. Это и дает возможность создавать свои контроллеры, к примеру, такие как планировщики, которые доступны из командной строки.

1 Возможности

1.1 Реестр сервисов и балансировка нагрузки

Не требуется изменять приложение для использования механизма поиска новых сервисов
Всем модулям предоставляются свои IP адреса и одно DNS имя для набора модулей, и может перераспределять нагрузку между ними.

1.2 Устойчивость

Аварийные контейнеры перезапускаются, заменяет контейнеры когда узлы останавливаются, изолирует контейнеры, которые не реагируют на обращения и запрещает обращения к ним.

1.3 Управление секретами и конфигурацией

Изменяйте секреты и конфигурацию не пересобирая образ

1.4 Масштабирование

Масштабируйте свое приложение с помощью командной строки, из интерфейса или автоматически.

1.5 Автоматическое развертывание и откат

Платформа постепенно развертывает ваше приложение, отслеживая его состояние и выполняет откат в случае если установленное приложение вызывает сбой.

2 Состав Q.Runner

2.1 Master компоненты

Отвечают за управление кластером. Они обнаруживают и реагируют на события кластера. Эти компоненты могут быть запущены на любом компьютере кластера, однако для простоты настройки, обычно, запускают все основные компоненты на одном компьютере (ведущем устройстве), на котором не запускают пользовательские контейнеры с приложениями.

2.1.1 qrun-apiserver

Компонент на ведущем устройстве, предоставляющий Q.Runner. Предоставляет внешний интерфейс для панели управления Q.Runner. Предоставляет возможность для горизонтального масштабирования, то есть он масштабируется путем развертывания большего количества экземпляров.

2.1.2 etcd

Согласованное и высокодоступное хранилище ключевых значений, используемое в качестве резервного хранилища Q.Runner для всех данных кластера.

2.1.3 qrun-scheduler

Компонент на ведущем устройстве, который наблюдает за вновь созданными модулями (pod) без назначенного узла и выбирает узел для их запуска. К факторам, принимаемым во внимание при принятии решений о планировании, относятся индивидуальные и коллективные потребности в ресурсах, аппаратные/программные/политические ограничения, спецификации сходства и анти-сходства, локальность данных, помехи между рабочими нагрузками и крайние сроки.

2.1.4 qrun-controller-manager

Компонент на ведущем устройстве, который запускает контроллеры. Логически каждый контроллер является отдельным процессом, но для снижения сложности все они компилируются в один двоичный файл и запускаются в одном процессе. Они включают:

- Контроллер узла: ответственный за уведомление и ответ, когда узлы аварийно завершаются.
- Контроллер репликации: отвечает за поддержание правильного количества модулей для каждого объекта контроллера репликации в системе.
- Контроллер конечных точек: заполняет объект конечных точек (т. е. соединяет службы и модули).
- Контроллер учетных записей и токенов безопасности: создание учетных записей по умолчанию и токенов безопасности API для новых пространств имен.

2.2 Компоненты рабочих узлов

Компоненты узла, выполняющиеся на каждом узле, поддерживают запущенные модули и обеспечивает среду выполнения Q.Runner.

2.2.1 qrunlet

Агент, который выполняется на каждом узле кластера. Это гарантирует, что контейнеры работают в модуле. Qrunlet принимает набор спецификаций (PodSpecs), которые предоставляются через различные механизмы и гарантирует, что контейнеры, описанные в спецификации будут

запущены и поддерживаться в работоспособном состоянии. Qrunlet не управляет контейнерами, которые не были созданы Q.Runner.

2.2.2 qrun-проxy

Сетевой прокси агент, который работает на каждом узле кластера.

Позволяет производить переадресацию запросов. Компонент позволяет производить переадресацию как потоков TCP/UDP, так и циклическую переадресацию TCP/UDP.

2.2.3 Среда исполнения Контейнера

Программное обеспечение, которое отвечает за запуск контейнеров. Q.Runner поддерживает несколько вариантов среды исполнения контейнеров: Q.Porter, containerd, cri-o, rktlet.