

Афанасьев Г.И.

кандидат технических наук

доцент кафедры «Системы обработки информации и управления»

Московский государственный технический университет им. Н.Э. Баумана

Россия, г. Москва

Абулкасимов М.М.

старший преподаватель кафедры «Системы обработки информации и

управления»

Московский государственный технический университет им. Н.Э. Баумана

Россия, г. Москва

СОЗДАНИЕ КЛАСТЕРА POSTGRESQL В СРЕДЕ KUBERNETES

Аннотация: Описывается один из путей развёртывания базы данных в кластере. Описан механизм развёртывания, приведены инструкции по установке необходимых дополнительных решений. Описаны методы масштабирования и проверки кластера.

Ключевые слова: докер, кранчи дата, кубернетес, постгресскюел, кластер

Abstract: The article touches upon the issue of the actual problem of deployment of the database in a cluster. The deployment mechanism is described, instructions on installations of necessary additional decisions are provided. Scaling techniques and checks of a cluster are described.

Keywords: docker,crunchy data, Kubernetes, PostgreSQL, cluster.

Введение

Методы контейнеризации (Docker), позволяют развернуть систему любой сложности за очень короткое время. Данная статья нацелена на то, что бы показать развертывание кластера базы данных PostgreSQL. В качестве управляющего Docker

контейнерами будет служить Kubernetes.

Kubernetes является проектом с открытым исходным кодом, предназначенным для управления кластером контейнеров Linux как единой системой. Kubernetes управляет и запускает контейнеры Docker на большом количестве хостов, а так же обеспечивает совместное размещение и репликацию большого количества контейнеров. Проект был начат Google и теперь поддерживается многими компаниями, среди которых Microsoft, RedHat, IBM и Docker.

Для работы потребуется:

- Компьютер с ОС на базе Linux, в качестве управляющей машины
- Минимум 1 сервер с установленным Kubernetes и Docker (процесс установки описываться не будет, т.к. зависит от выбранной ОС и ее релизом; инструкцию по установке можно найти на официальном сайте Kubernetes).
- Скачанный репозиторий `crunchy-containers` от Crunchy Data
- Установленный диспетчер пакетов Helm для Kubernetes

Установка Crunchy-containers

Crunchy-containers – специальная сборка Docker контейнеров, которые позволяют быстрое развертывание PostgreSQL, включая администрирование и мониторинг. Данные сборки позволяют запустить кластерные решения.

Для установки необходимо клонировать репозиторий на локальный компьютер:

```
git clone https://github.com/CrunchyData/crunchy-containers.git
```

В качестве примера используются стандартные Chart файлы из директории:

```
cd crunchy-containers/examples/kubehelm/crunchy-postgresql cluster/templates
```

Установка Helm

Для установки Helm, необходимо скачать образ с официального github репозитория и затем установить его:

```
wget https://kubernetes-helm.storage.googleapis.com/helm-v2.1.3-linux-  
amd64.tar.gz  
tar -xzf helm-v2.1.3-linux-amd64.tar.gz  
cd linux-amd64  
sudo cp helm /usr/bin
```

Далее необходимо перейти в директорию проекта и создать проект helm:

```
helm init  
cd crunchy-containers/examples/kubehelm  
helm install ./crunchy-postgres
```

Архитектура кластера

Для создания кластера потребуется в среде Kubernetes развернуть:

- Мастер сервис Postgres (Master Service)
- Реплика сервис Postgres (Replica Service)
- Мастер базы данных Postgres (Postgres Master Database Container)
- Контроллер репликации базы данных Postgres (Replication Controller)

(Рисунок 1)

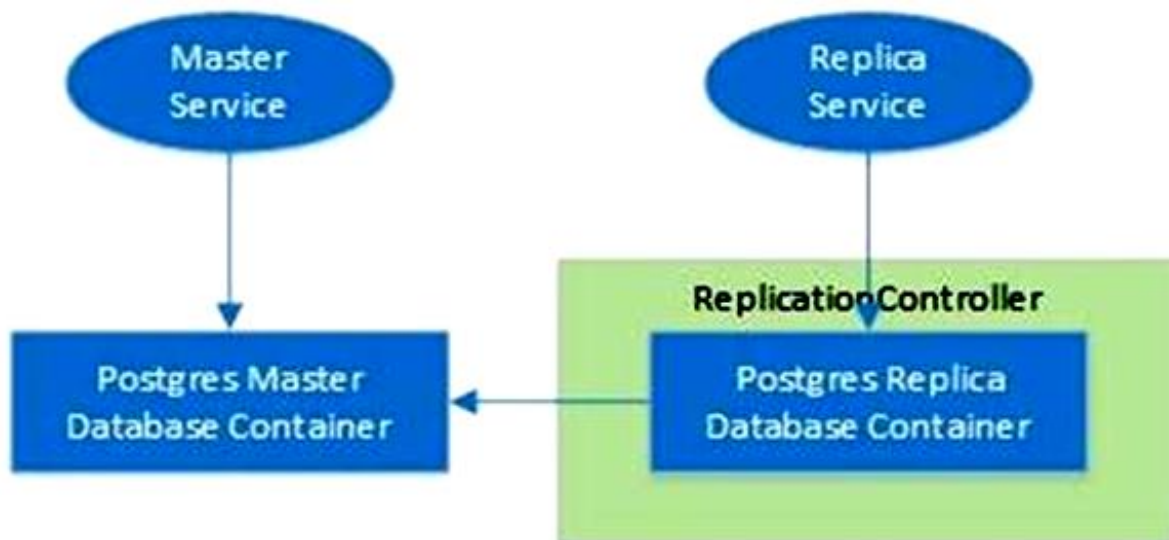


Рис. 1 Создание кластера

В данном примере создается мастер контейнер с базой данных с возможностью чтения и записи, а так же одиночной реплики мастера с возможностью только чтения. Масштабирование реплики осуществляется с помощью контроллера репликации и может изменяться динамически.

Основные файлы

- **values.yaml** – этот файл содержит значения, которые можно использовать в базе данных шаблонов, что позволяет указать их в качестве паролей базы данных
- **templates/master-pod.yaml** – определение pod'а базы данных Мастер Postgres. Этот файл вызывает создание одиночного pod'а мастера Postgres
- **templates/master-service.yaml** - база данных Мастер Postgres имеет сервис, созданный для работы в качестве прокси. Этот файл вызывает создание одиночного сервиса для прокси-вызовов базы данных Мастер
- **templates/replica-rc.yaml** - этим файлом определяется база данных Реплика Postgres. Файл вызывает создание контроллера репликации, который позволяет по

требованию масштабировать контейнеры реплики Postgres

- **templates/replica-service.yaml** - этот файл вызывает создание прокси сервиса для контейнера(-ов) базы данных Реплика.

Тестирование

После установки Helm Charts должны появиться следующие сервисы:

```
kubectl get servicesNAME CLUSTER-IP EXTERNAL-IP PORT(S) AGE
crunchy -master 10.0.0.171 <none> 5432/TPC 1h
crunchy-replica 10.0.0.31 <none> 5432/TCP 1h
kubernetes 10.0.0.1 <none> 433/TCP 1h
```

Через некоторое время реплика начнёт копировать данные из мастера. Для проверки репликации можно выполнить команду:

```
psql -h crunchy-master -U postgres postgres -c 'table pg_stat_replication'
```

При выводе не пустых строк, будет следовать что репликация работает. Для большей гарантии стоит добавить несколько тестовых данных в базу данных. Для этого создаётся таблица **foo** и записывается туда значение:

```
psql -h crunchy-master -U postgres postgres -c 'create table foo (id int)'
```

```
psql -h crunchy-master -U postgres postgres -c 'insert into foo values (1)'
```

Затем убедиться что данные реплицируются на машину подчиненного:

```
psql -h crunchy-replica -U posygres -c'table foo'
```

Масштабирование

После проверки работы репликации, можно проверить функцию масштабирования, для этого необходимо выполнить:

```
kubectl scale rc crunchy-replica --replicas=2
```

При этом в течении минуты создастся новая база данных и данные начнут реплицироваться из мастера. В зависимости от количества данных процесс репликации может занять внушительное время.

Заключение

Проекты Kubernetes Helm и Charts обеспечивают усовершенствованный способ упаковки сложных приложений и их развертывание в кластере Kubernetes. Развертывание кластеров PostgreSQL иногда может оказаться сложной задачей, но с помощью Helm и Charts задача значительно упрощается.

Использованные источники

1. Документация Docker. [Электронный ресурс]. Режим доступа: <https://docs.docker.com> (дата обращения: 27.11.2017).
2. Документация Crunchy Data. [Электронный ресурс]. Режим доступа: <http://crunchydata.com/documentation/> (дата обращения: 27.11.2017).
3. Документация PostgreSQL. [Электронный ресурс]. Режим доступа: <https://www.postgresql.org/docs> (дата обращения: 27.11.2017).
4. Jeff Nickoloff. Docker in Action. USA, Manning 2016, 304 p..