



Следующая статья из единого цикла, посвященному современной системе оркестрации контейнеров. Сегодня расскажу, что такое Helm и как с его помощью можно быстро и легко разворачивать приложения в кластере Kubernetes. Статья будет поверхностная, без углубления в сам инструмент. Я просто покажу на примерах, как с ним работать.

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Курс стоящий, все подробности читайте по ссылке. Есть бесплатные курсы.

Содержание:

- 1 Цели статьи
- 2 Что такое Helm и зачем нужен
- 3 Установка Helm 3
- 4 Установка приложений через Helm 3
- 5 Редактирование Helm чартов
- 6 Установка WordPress в Kubernetes
- 7 Заключение

Цели статьи

1. Рассказать, что такое Helm и для чего нужен.
2. Установить Helm 3 и показать, как работать с репозиториями.
3. На конкретных примерах выполнить установку приложений в кластер Kubernetes через Helm 3.
4. Показать, как редактировать готовые чарты для кастомизации установки.



Что такое Helm и зачем нужен

Если вы уже работаете с Kubernetes или читали мои предыдущие статьи по этой теме, то представляете себе, какие тонны yaml файлов приходится писать, чтобы запустить реальное приложение в кластере. Необходимо держать в голове и учитывать кучу абстракций, взаимосвязанных через метки, имена и т.д.

Helm призван упростить это. По своей сути он пытается работать как пакетный менеджер для запуска приложений в кластере. С его помощью можно создавать единые шаблоны для описания приложений, и helm будет дальше готовить и запускать все остальное сам.

Установка Helm 3

Установить Helm можно различными способами, но самый простой и быстрый — скачать и использовать готовый бинарник. Не так давно увидел свет свежий релиз Helm 3. Мы будем использовать именно его. В связи с этим многие приложения из репозитория могут не устанавливаться или устанавливаться с ошибками, так как еще не поддерживают 3-ю версию. Проще было бы пока использовать 2-ю, но мне не хочется разбирать старую версию. Надо учиться работать с новой. Переход все равно неизбежен.

Идем на страницу загрузок и копируем ссылку на последнюю стабильную версию. Качаем, распаковываем и устанавливаем в систему. В моем случае я ставлю на master-1, так как все управление кластером веду с него.

```
# wget https://get.helm.sh/helm-v3.0.0-linux-amd64.tar.gz
# tar xzvf helm-v3.0.0-linux-amd64.tar.gz
# mv linux-amd64/helm /usr/local/bin/helm
```

Проверяем установленную версию.

```
# helm version
```



После установки самого helm, надо подключить репозиторий. По-умолчанию ни один из них не подключен. Поставим самый популярный дефолтный



репозиторий.

```
# helm repo add stable https://kubernetes-charts.storage.googleapis.com/
```

Проверяем.

```
# helm search repo stable
```

Вы должны увидеть список всех чартов (charts), которые могут быть установлены из этого репозитория. По сути это набор приложений, которые может развернуть в kubernetes helm.

Обновить репозиторий можно командой.

```
# helm repo update
```

Все как в привычных пакетных менеджерах. Рекомендуется обновлять репозиторий перед установкой, чтобы забрать самую свежую версию приложения, которое будет развернуто в кластере с помощью Helm 3.

Искать в репозиториях helm можно ключом search (вот это да :)

```
# helm search hub wordpress
```



Установка приложений через Helm 3

Для примера давайте поставим интересную панельку для управления и установки чартов helm через браузер. Называется она Kubeapps. Для ее установки подключим репозиторий разработчика.

```
# helm repo add bitnami https://charts.bitnami.com/bitnami
```



```
# helm repo update
```

Kubeapps использует отдельный namespace, создадим его.

```
# kubectl create namespace kubeapps
```

Ставим приложение Kubeapps через Helm.

```
# helm install kubeapps --namespace kubeapps bitnami/kubeapps
```

Наблюдать за созданием подом можно с помощью команды.

```
# kubectl get pods -w --namespace kubeapps
```



После установки получите инструкцию по пробросу порта в нужный pod для доступа к dashboard. Но это актуально, если у вас машина, с которой вы управляете кластером — полноценная рабочая станция, на которой вы можете открыть браузер и зайти в панель. У меня это не так. Я буду использовать ingress для доступа к панели.

Chart Kubeapps поддерживает конфигурацию с ingress controller, но мне не захотелось с ним разбираться, выгружая шаблон и правя его. Вместо этого я просто быстренько набросал конфиг *ingress-kubeapps.yaml* для запуска.

```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-kubeapps
  namespace: kubeapps
spec:
  rules:
```



```
- host: kubeapps.cluster.local
  http:
    paths:
      - backend:
          serviceName: kubeapps
          servicePort: 80
        path: /
  tls: []
```

Применяю его и проверяю.

```
# kubectl apply -f ingress-kubeapps.yaml
# kubectl get ingress --namespace kubeapps
```

NAME	HOSTS	ADDRESS	PORTS	AGE
ingress-kubeapps	kubeapps.cluster.local	10.1.4.39	80	11m

Теперь на своей локальной машине добавляю запись в host.

```
10.1.4.39 kubeapps.cluster.local
```

Захожу в панель по адресу kubeapps.cluster.local.



Дальше нам нужно получить токен для авторизации и доступа в панель. Делаем это в консоли.

```
# kubectl create serviceaccount kubeapps-operator
# kubectl create clusterrolebinding kubeapps-operator --clusterrole=cluster-admin --serviceaccount=default:kubeapps-operator
# kubectl get secret $(kubectl get serviceaccount kubeapps-operator -o jsonpath='{.secrets[].name}') -o
jsonpath='{.data.token}' | base64 --decode
```



Получившийся токен копируем и вставляем в окно логина на Dashboard.

В момент написания статьи Kubeapps у меня не заработал без ошибок. Я погуглил и узнал, что установка через Helm 3 еще не поддерживается, но будет со дня на день, поэтому я не стал убирать эту информацию из статьи.

Редактирование Helm чартов

Для примера возьмем еще одну панельку, но намного проще — **kube-ops-view**. Она показывает состояние кластера Kubernetes. Найдем ее в репозитории.

```
# helm search hub kube-ops-view
URL                                CHART VERSION    APP VERSION      DESCRIPTION
https://hub.helm.sh/charts/incubator/kube-ops-view  0.1.0            Kubernetes Operational View
- read-only system ...
https://hub.helm.sh/charts/stable/kube-ops-view    1.1.1            19.9.0          Kubernetes Operational View
- read-only system ...
```

Выгрузим ее манифест в отдельный файл.

```
# helm inspect values stable/kube-ops-view > kube-ops-view.yaml
```

Открываем yaml файл и видим весь состав чарта. По сути это просто описание нескольких абстракций Kubernetes, которые нужны для работы панели. По умолчанию панель устанавливается без поддержки ingress и не включен rbac. Без rbac состояние кластера она не сможет отображать. Давайте это исправим. Добавляем в секцию ingress и rbac параметры.

```
ingress:
  enabled: true
  path: /
  hostname: kou.cluster.local
```



```
rbac:  
  create: true
```

Устанавливаем чарт, используя измененный конфиг.

```
# helm install ops-view --namespace kube-system stable/kube-ops-view -f kube-ops-view.yaml
```



Переходим по адресу `kou.cluster.local` и наблюдаем визуально состояние кластера. Показаны ноды и запущенные в них поды.



Панель неказистая и странная, но для учебных целей подходит. Зато очень маленькая и простая. Можно быстро визуально оценить весь кластер.

Установка WordPress в Kubernetes

Скачать чарт для изучения или изменения можно следующей командой на примере чарта с установкой wordpress.

```
# helm fetch bitnami/wordpress --untar
```

В директорию `wordpress` будет скачан полностью `chart`, который вы можете изучить. О структуре helm чартов я рассказывать не буду. Есть хорошая статья на хабре на эту тему. Давайте отредактируем манифест чарта и установим `wordpress` в кластер Kubernetes с помощью Helm.

```
# helm inspect values bitnami/wordpress > wordpress.yaml
```

В манифесте вы можете изменить данные пользователя `wordpress`, который будет создан автоматически, а так же некоторые другие параметры. Меня тут больше всего интересуют настройки `ingress`, который по-умолчанию не используется. Я это изменяю, включаю `ingress` и назначаю имя для сайта — `kuber.serveradmin.ru`. Оно нам пригодится в других статьях. Например, при получении сертификатов от `let's encrypt`. Можно тут же включить `certManager`, но это не тема данной статьи, поэтому не буду этого делать.



```
ingress:  
  enabled: true  
  hostname: kuber.serveradmin.ru
```

Так же в манифесте указаны PVC для самого сайта и базы mariadb с характеристиками ReadWriteOnce, 10Gi и 8Gi. Подробно о pvc и работе с дисками я рассказывал в статье по работе с volumes в kubernetes. Если у вас нет provisioner, который автоматически создаст PV, то сделайте это сами. Если не хотите, то просто отключите эти запросы. Для теста можно и без них обойтись.

Устанавливаем wordpress в kubernetes через helm 3 в отдельный namespace.

```
# kubectl create namespace wordpress  
# helm install wordpress --namespace wordpress bitnami/wordpress -f wordpress.yaml
```

Наблюдать в реальном времени за созданием подов можно с помощью команды.

```
# kubectl get pod -w -o wide --namespace wordpress
```



Убедитесь, что у вас работает настройка ingress для wordpress.

```
# kubectl get ingress -o wide --namespace wordpress  
NAME          HOSTS          ADDRESS      PORTS    AGE  
wordpress    kuber.serveradmin.ru  10.1.4.39    80      54s
```

Добавляйте этот адрес и имя сайта в hosts и идите проверять работу блога.



Пароль пользователя wordpress хранится в secret. Инструкция о том, как его посмотреть будет дана в консоли после установки.



```
# kubectl get secret --namespace wordpress wordpress -o jsonpath="{.data.wordpress-password}" | base64 --decode
```

После того, как наиграетесь, удалить установленный через helm софт можно командой.

```
# helm uninstall wordpress --namespace wordpress
```

Не забывайте указывать namespace. Я периодически забываю и не понимаю сразу, почему команда helm не отработывает. Например, посмотреть список установленных чартов можно командой.

```
# helm ls --namespace wordpress
NAME                NAMESPACE    REVISION    UPDATED                               STATUS    CHART
APP VERSION
wordpress           wordpress    1           2019-11-25 17:06:11.3688785 +0300 MSK  deployed
wordpress-8.0.1    5.3.0
```

Указание namespace обязательно. Без него будет выведена пустота.

Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

На этом заканчиваю обзорную статью про Helm. Это мощный и достаточно сложный инструмент, если писать чарты к нему самостоятельно. Я постарался просто показать, как с ним работать и быстро устанавливать программы в Kubernetes. С помощью Helm этот процесс значительно упрощается. Можно в несколько команд, к примеру, на всем кластере развернуть мониторинг. Об этом я расскажу в следующих статьях более подробно.



Онлайн курсы по Mikrotik

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую пройти курсы по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Помимо официальной программы, в курсах будут лабораторные работы, в которых вы на практике сможете проверить и закрепить полученные знания. Все подробности на сайте . Стоимость обучения весьма демократична, хорошая возможность получить новые знания в актуальной на сегодняшний день предметной области. Особенности курсов:

- Знания, ориентированные на практику;
- Реальные ситуации и задачи;
- Лучшее из международных программ.

Помогла статья? Есть возможность отблагодарить автора