K8S 53: Kubernetes. Helm. Dependencies

Описание:

В Helm существует механизм для подключения других чартов как зависимостей по аналогии с тем, как пакеты в дистрибутивах Linux могут требовать установки других пакетов.
Существует 2 метода определения зависимостей:

1. Через chart.yaml - актуально для версии Helm 3.
2. Через requirements.yaml - актуально для версий Helm 2 и 3.

Разница в методах определения появилась с приходом Helm версии 3. Определяется то, где должны храниться зависимости, тем, какой apiVersion выставлен в chart.yaml в вашем Helm Chart. Первая версия актуальна со времен Helm 2 и хранила зависимости именно в requirements.yaml. Helm 3 поддерживает работу с apiVersion=1, поэтому логика работы должна быть одинакова с любой версией Helm.
Давайте рассмотрим, как выглядит определение зависимостей:

```
dependencies:
  - name: kube-state-metrics
    version: 2.4.*
    repository: https://kubernetes-charts.storage.googleapis.com/
    alias: kubestatemetrics
    condition: kubeStateMetrics.enabled
    tags:
    - monitoring
```

Как видно, зависимостей может быть несколько, так как это массив. Теперь пройдемся по полям зависимости:

1. name - имя зависимости;
2. version - версия чарта, который должен быть получен;
3. repository - адрес репозитория, из которого нужно запросить архив с чартом;
4. condition - указание boolean поля в Values, которое должно быть в значении true для того, чтобы чарт был установлен. Может содержать несколько значений, разделенных запятой. Если хотя бы одно из значений отсутствует или равняется false, то зависимость не установится;
5. tags - схоже с condition, но требует наличия boolean значения с именем тега в блоке tags в values.

Важное замечание - если по condition или по tags чарт должен быть установлен, то он будет установлен, даже в случае разночтения значений (скажем, tag - false, но связанный condition - true).
При помощи Values в вашем чарте можно устанавливать значения и для зависимых чартов.
Так, предположим, у нас есть чарт с именем sub1, у которого следующий values.yaml:

```
sub1_password: qwerty
subblock:
  enabled: true
```

Для того, чтобы установить значения для subchart в values.yaml нашего чарта, требуется описать его следующим образом:

```
db_password: ytrewq

sub1:
  sub1_password: ytrewq
  subblock:
    enabled: false
```

Раз уж мы начали разбираться, зачем нужны зависимости, стоит рассказать, зачем в tempates/_helpers.tpl у нас присутствуют шаблоны для CHART.name, CHART.fullname и так далее. Присутствуют они для того, чтобы в случае множественной установки одного и того же чарта с разными значениями у нас создавались не захардкорженные значения, а значения, использующие значение, связанные именно с релизом, а не чартом (скажем, имена Deployment). В противном же случае у нас установка той же базы данных приводила бы к созданию манифестов с одинаковыми именами, из-за чего установка проваливалась бы.
Хорошо, зависимости мы описали, но как ими пользоваться? Есть явный и неявный путь.

- Явный - используя команды, связанные с helm dependency - они позволяют создать requirements.lock файлы (аналогично package.lock в NodeJS приложениях, что позволяет жестко закрепить версии зависимостей, обновить зависимости и вывести список зависимостей);
- Неявный - просто установив чарт. С этим все просто - при установке Helm просто сам скачает и положит зависимости в директорию charts, поскольку без них просто не сможет запустить приложение с зависимостями.

Полезные ссылки:

- Charts (official docs)
- Dependencies (official docs)
- helm-dependency example (github)

Задание:

1. В репозитории helm-task (из предыдущего задания) создайте новую ветку с именем k8s-52, в которой выполните следующие изменения.
2. Для вашего chart добавьте в зависимость чарт postgresql из (официального репозитория)[https://github.com/helm/charts/tree/master/stable/].
3. Через values.yaml определите для него следующие значения:

- версия базы - 11,
- пароль администратора - rmbK8sPostgresPassword,
- пользователя базы - gocalc,
- пароль пользователя - gocalcPassword,
- имя базы данных - gocalcDB.

1. В values.yaml определите строку для подключения к PostgreSQL в ключе postgres_uri в формате postgres://${POSTGRES_USER}:${POSTGRES_PASSWORD}@${POSTGRES_HOST}/${POSTGRES_DATABASE}.
2. В определении Deployment используйте ключ postgres_uri для установки переменной окружения приложения gocalc.
3. Обновите ваш chart в кластере (команду и вывод сохраните).
4. На проверку отправить ссылку на репозиторий с обновленным чартом и все сохраненные команды и выводы.