K8S 55: Kubernetes. Helm. Secrets

Описание:

Теперь, когда вы уже знаете о плагинах, пришло время поговорить об одном из наиболее незаменимых на реальных проектах плагине - helm secrets.
Values для реального проекта могут содержать, а фактически почти всегда содержат, какие-либо приватные данные, как, например, пароли, ключи авторизации либо иное. DevOps-практики и организация CI/CD подразумевает то, что инфраструктурные данные, как, например, helm-charts, содержатся в сетевых репозиториях (имплементация системы контроля версий). Да, в конечном итоге, согласно best practices мы будем использовать k8s Secret сущности для хранения подобных данных непосредственно в инфраструктуре, но как же предыдущие этапы жизненного цикла нашего чарта? Хранить все данные в открытом виде в системах контроля версий?
Эту дилемму решает модуль helm secrets.
Данный плагин обладает весьма широкими возможностями, однако сосредоточимся на сути. Плагин использует систему шифрования данных на основании ключа (строго говоря, там есть возможность использования PGP, AWS KMS, GCP KMS, etc), благодаря чему мы можем шифровать чувствительные данные, хранимые в нашем чарте. При этом важно помнить, как работает блок values в helm - по дефолту используются данные из values.yaml, который находится в корневом каталоге чарта. При использовании дополнительных файлов с переменными (которые мы можем указать через параметр --values в строке запуска), данные между дополнительным файлом и оригинальным values.yaml сливаются (merge). Другими словами, при наличии неконфликтующих значений они сливаются, если значения конфликтуют, то наибольший приоритет имеют те параметры, которые передаются через командную строку.
Таким образом, мы можем вынести все чувствительные данные в отдельный файл, secrets.yaml, зашифровать его, используя модуль, и спокойно хранить все данные в системах контроля версий. Даже при компроментации доступа к системе контроля версий, у вас остается еще один рубеж безопасности, ведь для использования полученных данных необходимо также иметь ключ шифрования, которым зашифрованы все важные данные в вашем проекте.

Полезные ссылки:

- Helm Docs: plugins

Задание:

1. Загрузите и установите плагин https://github.com/futuresimple/helm-secrets.

2. Сгенерируйте gpg-key и на его основе зашифруйте ключи postgres_uri, postgresqlPostgresPassword и postgresqlPassword из values для чарта, который вы подготовили ранее, а из values их удалите.
3. Обновите ваш релиз chart'a с помощью команды helm upgrade.
4. Предоставьте gpg-key и чарт с зашифрованным значением переменных на проверку.