OTUS

Методическое пособие
по выполнению домашнего
задания курса
Инфраструктурная платформа на основе Kubernetes

Хранилище секретов для приложения. Vault.

Содержание

1. Введение

Секреты

Любые чувствительные и конфиденциальные данные (например пароли, ключи API, сертификаты и т.д), обычно используемые для доступа к информационным системам.

Hashicorp Vault

Инструмент с открытым исходным кодом, который обеспечивает безопасный и надежный способ хранения и распространения секретов, таких как ключи API, токены доступа и пароли.

В задании вы установите vault в ваш кластер в HA режиме и с Hashicorp Colnsul в качестве бэкенда.

External Secrets Operator

Это оператор для kubernetes, который автоматически синхронизирует секреты из внешних API и создает их в kubernetes. Если секрет во внешнем API изменяется, контроллер обновляет секреты.

В задании вы установите сам оператор, и настроим получение секретов из vault

2. Цели домашнего задания

1) Установить в кластер hashicorp vault в HA режиме и научиться его конфигурировать.
2) Понимать как работает хранилище секретов, как создавать секреты в нем, роли и политики доступа
3) Установить External Secret Operator и настроить его для получения секретов, хранящихся в vault

3. Описание домашнего задания

Подготовка к выполнению домашнего задания

- Создайте branch kubernetes-vault - данное домашнее задание будет выполняться в этой ветке.
- Создайте папку kubernetes-vault - все файлы, которые у вас получается во время выполнения данного ДЗ необходимо поместить в эту папку.

Рекомендуемые источники

- Документация YC по установке и настройке Managed Kubernetes
- Установка consul https://github.com/hashicorp/consul-k8s
- Установка vault https://github.com/hashicorp/vault-helm.git
- Инициализация и распечатывания кластера
- Работа с KV-v1 Secrets Engine - https://developer.hashicorp.com/vault/docs/secrets/kv/kv-v1
- Настройка авторизации kubernetes
- Установка и конфигурирование External Secrets Operator

4. Пошаговая инструкция выполнения домашнего задания

- Данное задание будет выполняться в managed k8s в Yandex cloud
- Разверните managed Kubernetes cluster в Yandex cloud любым удобным вам способом. Создайте 3 ноды для кластера
- В namespace consul установите consul из helm-чарта https://github.com/hashicorp/consul-k8s.git с параметрами 3 реплики для сервера. Приложите команду установки чарта и файл с переменными к результатам ДЗ.
- В namespace vault установите hashicorp vault из helm-чарта https://github.com/hashicorp/vault-helm.git
    - Сконфигурируйте установку для использования ранее установленного consul в HA режиме
    - Приложите команду установки чарта и файл с переменными к результатам ДЗ.
- Выполните инициализацию vault и распечатайте с помощью полученного unseal key все поды хранилища
- Создайте хранилище секретов otus/ с Secret Engine KV, а в нем секрет otus/cred, содержащий username='otus' password='asajkjkahs'
- В namespace vault создайте serviceAccount с именем vault-auth и ClusterRoleBinding для него с ролью system:auth-delegator. Приложите получившиеся манифесты к результатам ДЗ
- В Vault включите авторизацию auth/kubernetes и сконфигурируйте ее используя токен и сертификат ранее созданного ServiceAccount
- Создайте и примените политику otus-policy для секретов /otus/cred с capabilities = [“read”, “list”]. Файл .hcl с политикой приложите к результатам ДЗ

4. Пошаговая инструкция выполнения домашнего задания

- Создайте роль auth/kubernetes/role/otus в vault с использованием ServiceAccount vault-auth из namespace Vault и политикой otus-policy
- Установите External Secrets Operator из helm-чарта в namespace vault. Команду установки чарта и файл с переменными, если вы их используете приложите к результатам ДЗ
- Создайте и примените манифест crd объекта SecretStore в namespace vault, сконфигурированный для доступа к KV секретам Vault с использованием ранее созданной роли otus и сервис аккаунта vault-auth. Убедитесь, что созданный SecretStore успешно подключился к vault. Получившийся манифест приложите к результатам ДЗ.
- Создайте и примените манифест crd объекта ExternalSecret с следующими параметрами:
  - ns – vault
  - SecretStore – созданный на прошлом шаге
  - Target.name = otus-cred
  - Получает значения KV секрета /otus/cred из vault и отображает их в два ключа – username и password соответственно
- Убедитесь, что после применения ExternalSecret будет создан Secret в ns vault с именем otus-cred и хранящий в себе 2 ключа username и password, со значениями, которые были сохранены ранее в vault. Добавьте манифест объекта ExternalSecret к результатам ДЗ.

5. Сдача задания

- Добавьте все получившиеся файлы в ветку **kubernetes-logging**
- Создайте Pull Request к ветке master
- Заполните описание PR по шаблону
- **Не мерджите** PR самостоятельно
- Если у вас возникли вопросы при выполнении ДЗ и необходима консультация преподавателей – добавьте к PR метку **Review Required**
- В личном кабинете Otus сдайте ДЗ на проверку, указав ссылку на Pull Request

6. Критерий оценивания

- 0 баллов – задание не выполнено или выполнено не полностью
- 1 балл – все задания выполнены полностью

7. Рекомендуемые источники

- Документация YC по установке и настройке Managed Kubernetes
- Установка consul https://github.com/hashicorp/consul-k8s
- Установка vault https://github.com/hashicorp/vault-helm.git
- Инициализация и распечатывания кластера
- Работа с KV-v1 Secrets Engine - https://developer.hashicorp.com/vault/docs/secrets/kv/kv-v1
- Настройка авторизации kubernetes
- Установка и конфигурирование External Secrets Operator