Настройка Federation в k8s кластере

Цели практики

Целью данной практической работы является демонстрация процесса настройки и работы Prometheus в режиме Federation. В рамках данного урока в Kubernetes кластер будет установлен и настроен Prometheus Server таким образом, что он будет собирать данные с уже установленного Prometheus сервера.

Установка Prometheus будет производиться с помощью Helm Chart от prometheus community. Для удобства все остальные компоненты будут отключены.

С чем будем работать

- Helm chart prometheus community - необходим для установки Prometheus server в кластер

Где будем выполнять практику

ВНИМАНИЕ: Перед тем, как работать со стендом, его необходимо запустить из Личного кабинета

- Подключаемся по SSH к **sbox.slurm.io**:

  ```
  ssh s<номер вашего логина>@sbox.slurm.io
  ```

- Подключаемся к своему кластеру Kubernetes:

  ```
  ssh master-1.s<ваш номер логина>
  ```

- Становимся root:

  ```
  sudo -s
  ```

Практика

Подготовительная работа

1. Клонируем манифесты

Для выполнения практической части потребуются манифесты, их необходимо склонировать с Gitlab. Сделать это можно, выполнив команды:

```
cd ~
git clone git@gitlab.slurm.io:edu/lmk8s.git
```

2. Проверяем репо prometheus-community:

```
helm repo list | grep prometheus
```

Репозиторий уже был добавлен автоматически при создании стенда.

3. Создаем namespace

В кластере уже установлен Prometheus server в namespace: `monitoring`. В рамках данного урока Prometheus будет устанавливаться также в отдельный namespace: `prom-federation`. Для его создания необходимо выполнить команду:

```
kubectl create ns prom-federation
```

4. Создание secret для basic auth

Также как и основной Prometheus, новый сервер будет доступен через Ingress и доступ к нему будет ограничен basic auth. Для этого необходимо создать Secret, который содержит данные для basic auth. Для этого в подготовленном манифесте: `~/lmk8s/4.prometheus_advanced/4.1.prometheus_federation/basic-auth-secret.yml` необходимо подставить ваши данные аутентификации. Это можно сделать, выполнив команду:

```
sed -i "s/<basic auth>/$(k get secrets -n monitoring basic-auth -o jsonpath='{.data.auth}')/g" ~/lmk8s/4.prometheus_advanced/4.1.prometheus_federation/basic-auth-secret.yml
```

В результате выполнения данной команды будет получены данные, необходимые для аутентификации, из уже созданного Secret, для основного Prometheus и подставлены в манифест. Далее этот манифест необходимо применить в кластер, выполнив команду:

```
kubectl apply -f ~/lmk8s/4.prometheus_advanced/4.1.prometheus_federation/basic-auth-secret.yml -n prom-federation
```

5. Подготовка values

Для установки Prometheus будет использоваться уже подготовленные значения values:

```
~/lmk8s/4.prometheus_advanced/4.1.prometheus_federation/federation.yml
```

Но в них необходимо заменить адрес Ingress для Prometheus, это можно сделать выполнив команду:

```
sed -i 's/<student name>/Ваш номер студента, например s00000/g' ~/lmk8s/4.prometheus_advanced/4.1.prometheus_federation/federation.yml
```

В данных values уже настроен сбор данных по federation. Все сводится к этим строкам:

```
- job_name: federation
        scrape_interval: 30s

        metrics_path: '/federate'

        params:
          match[]:
            - '{job=~".+"}'
        static_configs:
          - targets:
            - prometheus-server.monitoring.svc:80
```

Тут надо обратить внимание на поля:

- match - является обязательным полем, и в нем указывается фильтр по labels, какие метрики мы хотим получать. Через этот параметр может быть ограничен набор метрик, которые забираются с нижестоящего Prometheus.
- targets - в нем указано обращение к Prometheus через имя сервиса.

Установка Prometheus в кластер

Теперь все готово для установки, и достаточно выполнить команду:

```
helm upgrade -i prom-federation -n prom-federation -f
~/lmk8s/4.prometheus_advanced/4.1.prometheus_federation/federation.yml
prometheus-community/prometheus
```

После установки необходимо проверить, что все работает корректно:

- Открываем в браузере в анонимном режиме URL `http://federation.<номер студента>.edu.slurm.io`
- Выполняем запрос: `node_cpu_seconds_total`

В результате выполнения данного запроса вам должны быть выведены данные по всем node в кластере.