УДК 004

SSL/TLS для приложения в Kubernetes

Еровлева Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье будут рассмотрены работы по добавлению SSL/TLS для защиты приложения развернутом на Kubernetes. Будет произведена настройка и создание собственного сертификата SSL, а также компонентов необходимых для подключения сертификатов. Итоговым результатом будет являться настроенное приложение с подключенными сертификатами SSL/TLS.
Ключевые слова: Kubernetes, Java, SSL

SSL/TLS for application in Kubernetes

Erovleva Regina Viktorovna
Sholom-Aleichem Priamursky State University
Student

Erovlev Pavel Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article will cover the work of adding SSL / TLS to secure an application deployed on Kubernetes. You will configure and create your own SSL certificate, as well as the components necessary for connecting certificates. The final result will be a configured application with connected SSL/TLS certificates.
Keywords: Kubernetes, Java, SSL

SSL/TLS – это криптографические протоколы, которые обеспечивают защиту при передачи данных. У абсолютно всех значимых компаний на серверах, где лежит их сайт установлены данные сертификаты, так как они подтверждают, что данные пользователей под защитой.

Kubernetes – это открытое ПО для автоматического развертывания контейнеризированных приложений.

Цель данной статьи – настроить SSL сертификат в кластере Kubernetes.

Д.П. Белов и И.А. Пелевин разработали методику для обеспечения отказоустойчивости микросервисных архитектур на базе Kubernetes и Docker [1]. С.П. Хиков разработал модель эффективного выбора средств сканирования изображений в Kubernetes [2]. М.С. Ивченко, В.Г. Тарасов рассмотрели Kubernetes для построения облачной платформы для запуска удаленных учебных сервисов [3]. О.О. Сергеева и А.Р. Белозерова провели сравнительный анализ потребительский сведений об аппаратной виртуализации на основе Kubernetes и Docker [4]. В своей работе А.А. Артамонова провела сравнительный анализ систем для автоматической контейнеризации приложений: Kubernetes и Docker [5].

Сначала нужно создать собственный самозаверяющий сертификат в «OpenSSL», а затем добавить его в качестве секрета в «Kubernetes» (рис.1).

```
1  openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout k8s-ssl.test.key -out k8s-ssl.test.crt
2  kubectl create secret tls k8s-ssl.test --key="k8s-ssl.test.key" --cert="k8s-ssl.test.crt"
```

Рисунок 1 – Добавление сертификата

В рисунке выше производится создание сертификата, который будет действовать 365 дней, и определяем домен «k8s-ssl.test» в сертификат. Если необходимо изменить имя или количество дней, в течение которых будет действителен сертификат, то это можно сделать перед его созданием.

После создания сертификата, создадим секрет в Kubernetes с именем «k8s-ssl.test» и загрузим файлы сертификата в секрет Kubernetes.

Важно, чтобы в «Ingress» был включен TLS и выбран, какой домен и секретное имя будет выбрано.

В правилах также устанавливаем домен «k8s-ssl.test» и указываем имя службы, которое должно использоваться для этого пути, и номер порта (рис.2).

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tls-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  tls:
  - hosts:
    - k8s-ssl.test
    secretName: k8s-ssl.test
  rules:
  - host: k8s-ssl.test
    http:
      paths:
      - path: /(.*)
        pathType: Prefix
        backend:
          service:
            name: k8s-ssl-web1
            port:
              number: 8080
```

Рисунок 2 – Настройка правил

Далее нужно создать службу в «Kubernetes», чтобы разрешить входящий трафик через порт 80, который использует протокол HTTP, и порт 443, который использует протокол HTTPS, для этого сгруппируем его в селектор с именем «TLSIngress», чтобы знать, что порт попадет в эту область (рис.3).

```yaml
apiVersion: v1
kind: Service
metadata:
  name: k8s-ssl
spec:
  selector:
    ingress: TLSIngress
  ports:
    - name: http
      protocol: TCP
      port: 80

    - name: https
      protocol: TCP
      port: 443
```

Рисунок 3 – Настройка службы

Развертывание добавит метки «ingress: TLSIngress» и «service: k8s-ssl-web1», поэтому, когда служба использует селектор, она будет соответствовать развертыванию в Kubernetes.

Изменим «containerPort», чтобы объяснить, что он работает на порту 8080, а не на порте 80 по умолчанию, поскольку NGINX использует его по умолчанию (рис.4).

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: k8s-tls
spec:
  selector:
    matchLabels:
      ingress: TLSIngress
  replicas: 3
  template:
    metadata:
      labels:
        ingress: TLSIngress
        service: k8s-ssl-web1
    spec:
      containers:
        - name: k8s-web1
          image: nginx
          ports:
            - containerPort: 8080
              protocol: TCP
```

Рисунок 4 – Настройка службы

Теперь при подключении к адресу «k8s-ssl.test» имя адреса будет защищено протоколом https.

В данной статье был рассмотрен процесс настройки и подключения сертификатов SSL в приложение на Kubernetes.

Библиографический список

1. Белов Д.П., Пелевин И.А. Реализация отказоустойчивости в системе оркестрации микросервисной архитектуры kubernetes // Вестник УРФО. безопасность в информационной сфере. 2019. №2(32). С. 5-11.
2. Хиков С.П. Модель выбора эффективного средства сканирования изображений контейнеров в инфраструктуре kubernetes // Инновационные технологии: теория, инструменты, практика. 2019. №1. С. 249-253.
3. Ивченко М.С., Тарасов В.Г. Использование kubernetes для построения облачной платформы для удаленного запуска учебных сервисов // Информационные технологии в науке, промышленности и образовании. 2020. С.107-109.
4. Сергеева О.О., Белозерова А.Р. Kubernetes с docker на локальной машине // Вестник димитровградского инженерно-технологического института. 2020. №1(21). С. 44-53.
5. Артамонова А.А. Сравнительный анализ двух систем управления контейнерами: docker swarm и kubernetes // Синергия наук. 2018. №23. С. 1173-1182.