УДК 004

Аутентификация и авторизация пользователей в Apache Kafka

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлева Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается как реализовать аутентификацию и авторизацию пользователей в Apache kafka. Для реализации будет использоваться распределенная платформа потоковой передачи событий Apache kafka.
Ключевые слова: Пользователь, Kafka, Безопасность

User Authentication and Authorization in Apache Kafka

Erovlev Pavel Andreevich
Sholom-Aleichem Priamursky State University
Student

Erovleva Regina Victorovna
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses how to implement user authentication and authorization in Apache kafka. The distributed event streaming platform Apache kafka will be used for implementation.
Keywords: User, Kafka, Security

1 Введение

1.1 Актуальность

Apache Kafka — это распределенная платформа потоковой передачи событий с открытым исходным кодом, используемая тысячами компаний для высокопроизводительных конвейеров данных, потоковой аналитики, интеграции данных и критически важных приложений.

Тысячи организаций используют Kafka, от интернет-гигантов до производителей автомобилей и фондовых бирж.

1.2 Обзор исследований

В.А. Сухомлин описал кратко в своей статье принцип работа комплексной программы дополнительного образования, ориентированную на подготовку разработчиков Java enterprise приложений [1]. В.Л. Радишевский и А.Д. Кульневич описали принцип работы распределенного брокера сообщений Kafka для высокоскоростной передачи и агрегации данных [2]. Ю.А. Флёров и Л.Л. Вышинский разработали программу для организации взаимодействия с Web-клиентами в программных комплексах [3]. А.И. Куреленко разработал программу предназначенную для генерации, приема и передачи типизированных сообщений для программного брокера Apache Kafka [4]. М.А. Потовиченко, М.В. Привалов и С.В. Корнев рассмотрели разработку программного продукта, который обеспечивает учет данных посещения занятий студентов, а также защиту их работ [5].

1.3 Цель исследования

Цель исследования – продемонстрировать возможности аутентификации и авторизации Kafka.

2 Материалы и методы

Работа будет происходить с использованием распределенной платформой потоковой передачи событий Apache kafka.

3 Результаты и обсуждения

Две встроенные функции безопасности Apache Kafka — это контроль доступа пользователей и шифрование данных. Хотя производственный кластер Kafka обычно предоставляет обе эти функции, они не обязательно требуются в средах разработки, тестирования или экспериментов. Фактически, в некоторых производственных средах эти функции не нужны, например, когда кластер находится за брандмауэром. Среда реализации кластера и другие соображения играют роль при принятии решения о том, какие функции безопасности необходимо реализовать в кластере Kafka.

Kafka обеспечивает аутентификацию и авторизацию с помощью списков управления доступом Kafka и через несколько интерфейсов.

Kafka использует SASL для аутентификации. В настоящее время он поддерживает множество механизмов, включая «PLAIN», «SCRAM», «OAUTH» и «GSSAPI», что позволяет администратору подключать собственные реализации. Аутентификацию можно включить между брокерами, между клиентами и брокерами, а также между брокерами и «ZooKeeper». Это позволяет ограничить доступ только тем сторонам, у которых есть необходимые секреты.

Kafka управляет и обеспечивает авторизацию через ACL авторизатор. Авторизатор реализует определенный интерфейс и является подключаемым. Kafka предоставляет реализацию авторизатора по умолчанию «AclAuthorize», которая хранит «ACL» в «ZooKeeper». Имя класса авторизатора указывается в

конфигурации брокера «authorizer.class.name». Если такой конфигурации не существует, то всем разрешен доступ к любому ресурсу.

Чтобы включить аутентификацию и авторизацию клиентов в кластере Kafka, как брокеры, так и клиенты должны быть правильно настроены. Брокеры должны знать действительные учетные данные, а клиенты должны предоставлять действительные учетные данные для правильного выполнения базовых команд.

Чтобы включить аутентификацию и авторизацию на стороне брокера, необходимо выполнить два шага на каждом брокере:

- Настроить действительные учетные данные.
- Настроить правильный протокол безопасности и реализацию авторизатора.

```
KafkaServer {
    org.apache.kafka.common.security.plain.PlainLoginModule required
    username="admin"
    password="admin"
    user_admin="admin"
    user_alice="alice"
    user_alex="alex"
    user_pavel="pavel";
};
```

Рисунок 1 – Файл JAAS

В файле JAAS для сущности определяется следующее:

- Пользовательский модуль входа, который используется для аутентификации пользователя;
- admin/admin— имя пользователя и пароль для связи между брокерами, учетные данные, которые брокер использует для подключения к другим брокерам в кластере;
- admin/admin, alice/alice, alex/alex и pavel/pavel в качестве учетных данных пользователя клиента.

Действительные имя пользователя и пароль предоставляются в следующем формате: user_username="password". Если строка user_admin="admin" будет удалена из этого файла, брокер не сможет аутентифицировать и авторизовать admin пользователя. В этом случае только admin пользователь может подключаться к другим брокерам.

Далее определим принятый протокол и авторизатор ACL, используемый брокером, добавив следующую конфигурацию в файл свойств брокера.

```
authorizer.class.name=kafka.security.authorizer.AclAuthorizer
listeners=SASL_PLAINTEXT://:9092
security.inter.broker.protocol= SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=PLAIN
sasl.enabled.mechanisms=PLAIN
```

Рисунок 2 – Файл свойств брокера

Другая конфигурация, которую можно добавить, предназначена для суперпользователей Kafka, пользователей с полным доступом ко всем API. Эта конфигурация снижает нагрузку на определение ACL для каждого API пользователя, который должен иметь полный доступ к API. Создадим admin суперпользователя со следующей конфигурацией.

```
super.users=User:admin
```

Рисунок 3 – Добавление суперпользователя

Этот модифицированный файл свойств называется «sasl-server.properties».

Когда брокер работает с этой конфигурацией безопасности, только аутентифицированные и авторизованные клиенты могут подключаться к нему и использовать его.

Теперь укажем протокол брокера, а также учетные данные для использования на стороне клиента. Следующая конфигурация помещается в соответствующий файл конфигурации, предоставленный конкретному клиенту.

```
security.protocol=SASL_PLAINTEXT
sasl.mechanism=PLAIN
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="alice" password="alice";
```

Рисунок 4 – Конфигурация alice.properties

Если запуститm команду с конфигурацией, то они не будут работать должным образом.

```
$ bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic
test --producer.config config/alice.properties
>message1
[2017-10-24 16:20:52,259] WARN [Producer clientId=console-producer] Error
while fetching metadata with correlation id 1 :
{test=TOPIC_AUTHORIZATION_FAILED} (org.apache.kafka.clients.NetworkClient)
[2017-10-24 16:20:52,260] ERROR Error when sending message to topic test
with key: null, value: 1 bytes with error:
(org.apache.kafka.clients.producer.internals.ErrorLoggingCallback)
org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized
to access topics: [test]
>
```

Рисунок 5 – Не авторизованный пользователь

Конфигурация безопасности по-прежнему не дает конкретных разрешений пользователям Kafka. Эти разрешения определяются с помощью команды ACL. Чтобы проверить существующие ACL, запустим команду.

```
./bin/kafka-acls.sh --bootstrap-server localhost:9092 --command-config
./config/admin.properties --list
```

Рисунок 6 – Добавление разрешений пользователям

После выдачи всех разрешений и добавлении пользователей в свойства kafka у пользователя Alice должны быть разрешения создавать топики с темой test.

```
Principal alice is Allowed Operation Write From Host * On Topic test.
Principal alice is Allowed Operation Create From Host * On Topic test.
```

Рисунок 7 – Создание темы

```
Adding ACLs for resource `ResourcePattern(resourceType=TOPIC, name=test,
patternType=LITERAL)`:
     (principal=User:alice, host=*, operation=WRITE, permissionType=ALLOW)
    (principal=User:alice, host=*, operation=CREATE, permissionType=ALLOW)

Current ACLs for resource `ResourcePattern(resourceType=TOPIC, name=test,
patternType=LITERAL)`:
     (principal=User:alice, host=*, operation=CREATE,
permissionType=ALLOW)
    (principal=User:alice, host=*, operation=WRITE, permissionType=ALLOW)
```

Рисунок 8 – Вывод при создании темы

В результате предоставления разрешения Alice теперь может создать тему «test».

Выводы

Kafka предоставляет средства для принудительной аутентификации и авторизации пользователей для доступа к различным ресурсам и операциям. Аутентификация осуществляется через SASL с несколькими поддерживаемыми механизмами. Авторизация выполняется с использованием его списков ACL и подключаемых объектов авторизации.

В данной статье показано, как можно настроить аутентификацию и авторизацию в кластере Kafka.

Библиографический список

1. Сухомлин В.А. Подготовка разработчиков корпоративных java-приложений в режиме дистанционного обучения // Информатика, управляющие системы, математическое и компьютерное моделирование. 2013. № 9. С. 175-180.
2. Радишевский В.Л., Кульневич А.Д. Распределенный брокер сообщений

kafka для высокоскоростной передачи и агрегации данных // Молодёжь и современные информационные технологии. 2018. № 1. С. 284-285.
3. Флёров Ю.А. и Вышинский Л.Л. Программа организации интерфейса web-серверов с java-приложениями // Аллея науки. 2016. № 7. С. 58-61.
4. Куреленко А.И. Генератор сообщений для программного брокера сообщений apache kafka // Тихоокеанский государственный университет. 2017. С. 171-176.
5. Потовиченко М.А., Привалов М.В., Корнев С.В. Компьютеризированная подсистема учета текущей успеваемости студента в условиях вуза // Информатика, управляющие системы, математическое и компьютерное моделирование. 2019. № 2. С. 71-75.