

Konfigurieren Sie ein Kafka-Ziel für einen offenen Datenstrom

Veröffentlicht: 2025-03-28

Sie können Daten auf einem ExtraHop-System auf jeden Kafka-Server exportieren, um sie langfristig zu archivieren und mit anderen Quellen zu vergleichen.

1. Loggen Sie sich in die Administrationseinstellungen des ExtraHop-Systems ein über `https://<extrahop-hostname-or-IP-address>/admin`.
Wiederholen Sie diese Schritte für jeden Sensor in Ihrer Umgebung.
2. In der Konfiguration des Systems Abschnitt, klicken **Datenströme öffnen**.
3. Klicken Sie **Ziel hinzufügen**.
4. Aus dem Typ des Ziels Drop-down-Menü, wählen **Kafka**.
5. In der Name Feld, geben Sie einen Namen ein, um das Ziel zu identifizieren.
6. Aus dem Kompression Wählen Sie im Dropdownmenü eine der folgenden Komprimierungsmethoden aus, die auf die übertragenen Daten angewendet werden sollen:
 - **Keine**
 - **GZIP**
 - **Bissig**
7. Aus dem Partitionsstrategie Wählen Sie im Dropdownmenü eine der folgenden Partitionierungsmethoden aus, die auf die übertragenen Daten angewendet werden:
 - **Standard (Hash-Schlüssel)**
 - **Manuell**
 - **Zufällig**
 - **Round Robin**
8. Optional: Konfigurieren Sie die SASL/SCRAM-Authentifizierung.
 - a) Aus dem Authentifizierung Drop-down-Menü, wählen **SASL/SCRAM**.
 - b) In der Nutzernamen Feld, geben Sie den Namen des SASL/SCRAM-Benutzers ein.
 - c) In der Passwort Feld, geben Sie das Passwort des SASL/SCRAM-Benutzers ein.
 - d) Aus dem **Hashing-Algorithmus** Wählen Sie im Dropdownmenü den Hashing-Algorithmus für die SASL-Authentifizierung aus.
9. Aus dem **Protokoll** Wählen Sie im Dropdownmenü eines der folgenden Protokolle aus, über das Daten übertragen werden sollen:
 - **TCP**
 - **TLS**
10. Optional: Wenn Sie das ausgewählt haben **TLS** Protokoll, geben Sie die Zertifikatsoptionen an.
 - a) Wenn der Kafka-Server eine Client-Authentifizierung erfordert, in der **Client-Zertifikat** Feld, geben Sie ein TLS-Client-Zertifikat an, das an den Server gesendet werden soll.
 - b) Wenn Sie ein Client-Zertifikat angegeben haben, in der **Kundenschlüssel** Feld, geben Sie den privaten Schlüssel des Zertifikats an.
 - c) Wenn Sie das Zertifikat des Kafka-Servers nicht verifizieren möchten, wählen Sie **Serverzertifikatsüberprüfung überspringen**.
 - d) Wenn Sie das Zertifikat des Kafka-Servers verifizieren möchten, das Zertifikat jedoch nicht von einer gültigen Zertifizierungsstelle (CA) signiert wurde, finden Sie in der **CA-Zertifikate (optional)** Feld, geben Sie vertrauenswürdige Zertifikate im PEM-Format an, mit denen das Serverzertifikat überprüft werden soll. Wenn diese Option nicht angegeben ist, wird das Serverzertifikat mit der integrierten Liste gültiger CA-Zertifikate validiert.

11. Geben Sie mindestens einen Kafka-Broker an, der in einem Kafka-Cluster auch als Knoten bezeichnet wird und übertragene Daten empfangen kann.



Hinweis Sie können mehrere Broker hinzufügen, die Teil desselben Kafka-Clusters sind, um die Konnektivität sicherzustellen, falls ein einzelner Broker nicht verfügbar ist. Alle Broker müssen Teil desselben Cluster sein.

- a) In der Gastgeber Feld, geben Sie den Hostnamen oder die IP-Adresse des Kafka-Brokers ein.
 - b) In der Hafen In diesem Feld geben Sie die Portnummer des Kafka-Brokers ein.
 - c) Klicken Sie auf das Plus (+) Symbol.
12. Optional: Klicken Sie **Testen** um eine Verbindung zwischen dem ExtraHop-System und dem Remote-Kafka-Server herzustellen und eine Testnachricht an den Server zu senden.
Das Dialogfeld zeigt eine Meldung an, die angibt, ob die Verbindung erfolgreich war oder fehlgeschlagen ist.



Hinweis Wenn der Test fehlschlägt, überprüfen Sie die Protokolle auf Ihrem Kafka-Server auf detailliertere Informationen zum Fehler, bearbeiten Sie dann die Zielkonfiguration und testen Sie die Verbindung erneut.

13. Klicken Sie **Speichern**.

Nächste Schritte

Erstellen Sie einen Auslöser, der angibt, welche Kafka-Nachrichtendaten gesendet werden sollen, und der die Übertragung der Daten an das Ziel initiiert. Weitere Informationen finden Sie in der [Remote.Kafka](#) Klasse in der [ExtraHop Trigger API-Referenz](#).