


# Configure a Kafka target for an open data stream

Published: 2024-04-01

You can export data on an ExtraHop system to any Kafka server for long-term archiving and comparison with other sources.

1. Log in to the Administration settings on the ExtraHop system through `https://<extrahop-hostname-or-IP-address>/admin`.  
Repeat these steps on each sensor in your environment.
2. In the System Configuration section, click **Open Data Streams**.
3. Click **Add Target**.
4. From the Target Type drop-down menu, select **Kafka**.
5. In the Name field, type a name to identify the target.
6. From the Compression drop-down list, select one of the following compression methods that will be applied to the transmitted data:
  - **None**
  - **GZIP**
  - **Snappy**
7. From the Partition strategy drop-down list, select one of the following partitioning methods that will be applied to the transmitted data:
  - **Default (Hash Key)**
  - **Manual**
  - **Random**
  - **Round Robin**
8. From the Protocol drop-down menu, select one of the following protocols over which to transmit data:
  - **TCP**
  - **SSL/TLS**
9. Optional: If you selected the **SSL/TLS** protocol, specify certificate options.
  - a) If the Kafka server requires client authentication, specify a TLS client certificate to send to the server in the **Client certificate** field.
  - b) If you specified a client certificate, specify the private key of the certificate in the **Client key** field.
  - c) If you do not want to verify the certificate of the Kafka server, select **Skip server certificate verification**.
  - d) If you want to verify the certificate of the Kafka server, but the certificate has not been signed by a valid Certificate Authority (CA), specify trusted certificates to verify the server certificate with in the **CA certificates (optional)** field. Specify the certificates in PEM format. If this option is not specified, the server certificate is validated with the built-in list of valid CA certificates.
10. Specify at least one Kafka broker, also referred to as a node in a Kafka cluster, that can receive transmitted data.
 



**Note:** You can add multiple brokers that are part of the same Kafka cluster to ensure connectivity in case a single broker is unavailable. All brokers must be part of the same cluster.

  - a) In the Host field, type the hostname or IP address of the Kafka broker.
  - b) In the Port field, type the port number of the Kafka broker.
  - c) Click the plus (+) icon.

11. Optional: Click **Test** to establish a connection between the ExtraHop system and the remote Kafka server and send a test message to the server.  
The dialog box displays a message that indicates whether the connection succeeded or failed. If the test fails, edit the target configuration and test the connection again.
12. Click **Save**.

#### Next steps

Create a trigger that specifies what Kafka message data to send and initiates the transmission of data to the target. For more information, see the [Remote.Kafka](#) class in the [ExtraHop Trigger API Reference](#).