

# Connecting Kafka Streams to Confluent Cloud

You can connect Kafka Streams to your Confluent Platform Apache Kafka® cluster in Confluent Cloud.

## Prerequisites

- [Confluent Platform](#)

To connect Streams to Confluent Cloud, update your [existing Streams configs](#) with the properties described here.

1. Create a `java.util.Properties` instance.
2. Configure your streams application. Kafka and Kafka Streams configuration options must be configured in the `java.util.Properties` instance before using Streams. In this example you must configure the Confluent Cloud broker endpoints (`StreamsConfig.BOOTSTRAP_SERVERS_CONFIG`) and SASL config (`SASL_JAAS_CONFIG`)

```
import java.util.Properties;
import org.apache.kafka.clients.producer.ProducerConfig;
import org.apache.kafka.common.config.SaslConfigs;
import org.apache.kafka.streams.StreamsConfig;

Properties props = new Properties();
// Comma-separated list of the Confluent Cloud broker endpoints. For example:
// r0.great-app.confluent.aws.prod.cloud:9092,r1.great-app.confluent.aws.prod.cloud:9093,r2.great-app.confluent.aws.prod.cloud:9094
props.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "<broker-endpoint1, broker-endpoint2, broker-endpoint3>");
props.put(StreamsConfig.REPLICATION_FACTOR_CONFIG, 3);
props.put(StreamsConfig.SECURITY_PROTOCOL_CONFIG, "SASL_SSL");
props.put(SaslConfigs.SASL_MECHANISM, "PLAIN");
props.put(SaslConfigs.SASL_JAAS_CONFIG, "org.apache.kafka.common.security.plain.PlainLoginModule \
username=<api-key>\" password=<api-secret>\"");

// Recommended performance/resilience settings
props.put(StreamsConfig.producerPrefix(ProducerConfig.RETRIES_CONFIG), 2147483647);
props.put("producer.confluent.batch.expiry.ms", 9223372036854775807);
props.put(StreamsConfig.producerPrefix(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG), 300000);
props.put(StreamsConfig.producerPrefix(ProducerConfig.MAX_BLOCK_MS_CONFIG), 9223372036854775807);

// Any further settings
props.put(... , ...);
```

3. (Optional) Add configs for Confluent Cloud Schema Registry to your streams application per the example in [java\\_streams.delta](#) on GitHub at [ccloud/examples/template\\_delta\\_configs](#).

```
// Confluent Schema Registry for Java
props.put("basic.auth.credentials.source", "USER_INFO");
props.put("schema.registry.basic.auth.user.info", "<SCHEMA_REGISTRY_API_KEY>:<SCHEMA_REGISTRY_API_SECRET>");
props.put("schema.registry.url", "https://<SCHEMA_REGISTRY_ENDPOINT>");
```

- For more information, see [Configuring a Streams Application](#).
- For example configs for all Confluent Cloud components, see [ccloud/examples/template\\_delta\\_configs](#) on GitHub.
- For the Confluent Cloud demo, which leverages the example code, see [ccloud/examples/](#).

© Copyright 2019, Confluent, Inc. [Privacy Policy](#) | [Terms & Conditions](#). Apache, Apache Kafka, Kafka and the Kafka logo are trademarks of the [Apache Software Foundation](#). All other trademarks, servicemarks, and copyrights are the property of their respective owners.

[Please report any inaccuracies on this page or suggest an edit.](#)

## Rate this page



