CONFLUENT

Kafka Connect / Connectors

Nathan Nam, Senior Product Manager

6/11/2020

Confluent 한국지사, 한국영업대표 최영주
(dchey@confluent.io)

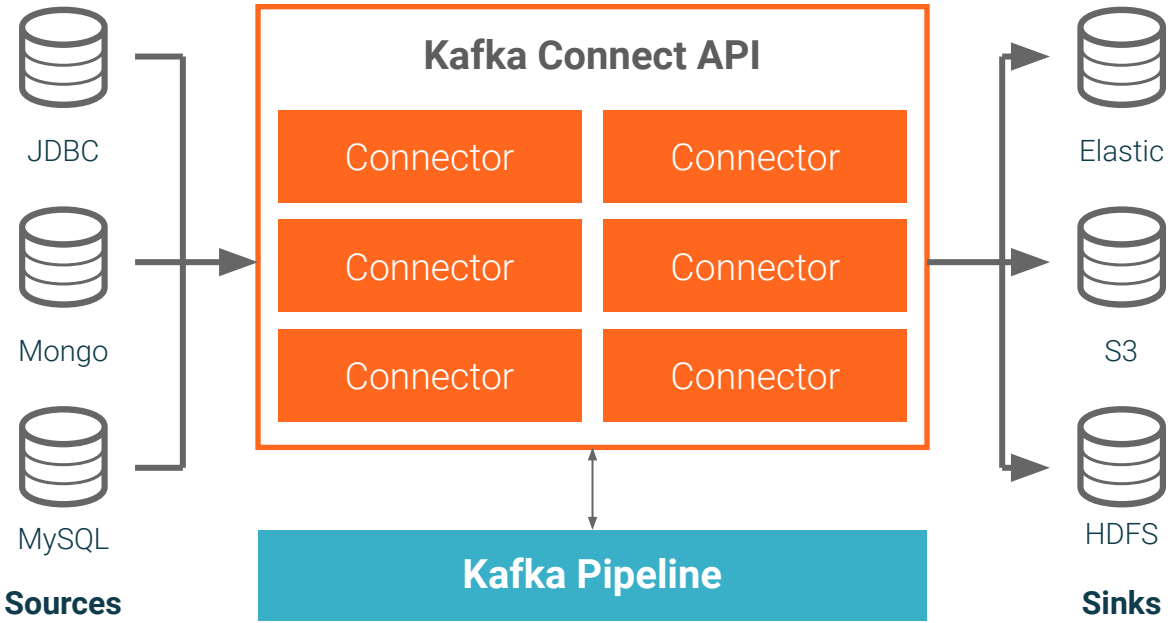
About me

- Nathan Nam, Senior Product Manager at Confluent, responsible for Kafka Connect and connectors. About 2 years at Confluent
- Previously at MuleSoft/Salesforce, Samsung Electronics
- Holds an MBA from Tuck School of Business at Dartmouth and an Master of Information and Data Science from UC Berkeley.

Kafka Connect

Apache Kafka Connect API: Import and Export Data In & Out of Kafka

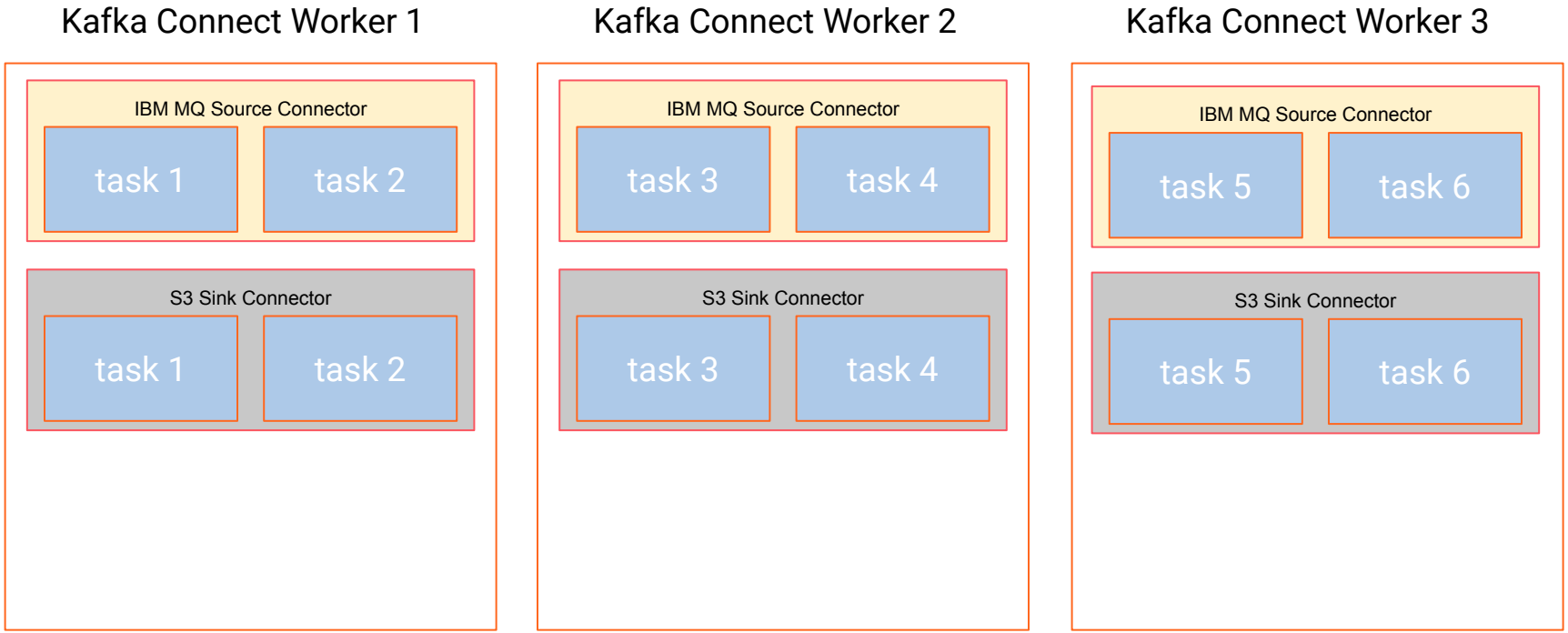
- Fault tolerant
- Manage hundreds of data sources and sinks
- Preserves data schema
- Integrated within Confluent Control Center



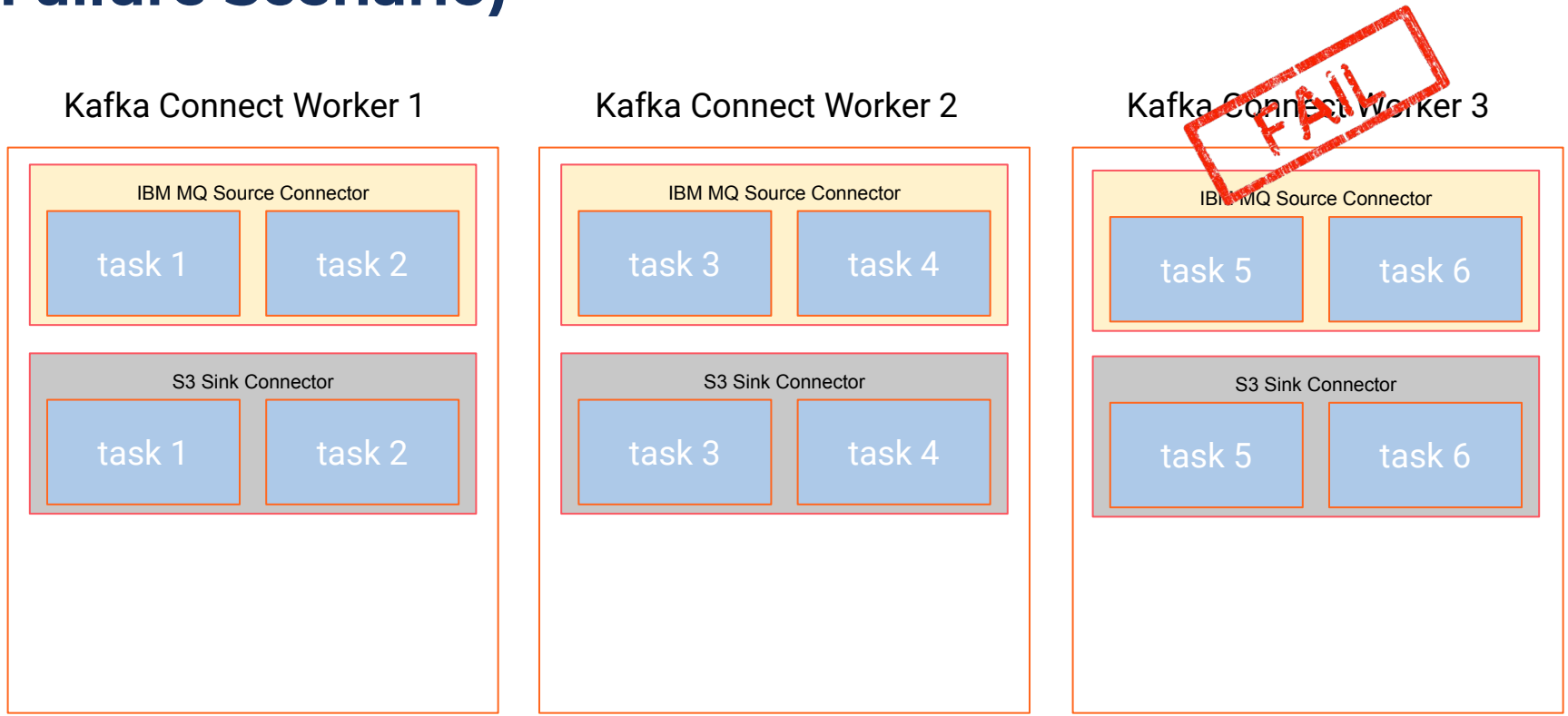
Kafka Connect Concepts

- **Connector** - the high level abstraction that coordinates data streaming by managing tasks. (Amazon S3 Sink Connector, IBM MQ Source Connector)
- **Task** - the implementation of how data is copied to or from Kafka
- **Worker** - the running processes that execute connectors and tasks. (JVM)
- **Cluster** - a group of Kafka Connect Workers
- **Converter** - the code used to translate data between Connect and the system sending or receiving data (AVRO, Protobuf, JSON Schema, JSON, Bytes, and others)
- **Transform** - a simple logic to alter each message produced by or sent to a connector (ExtractField, ExtractTopic, and others)

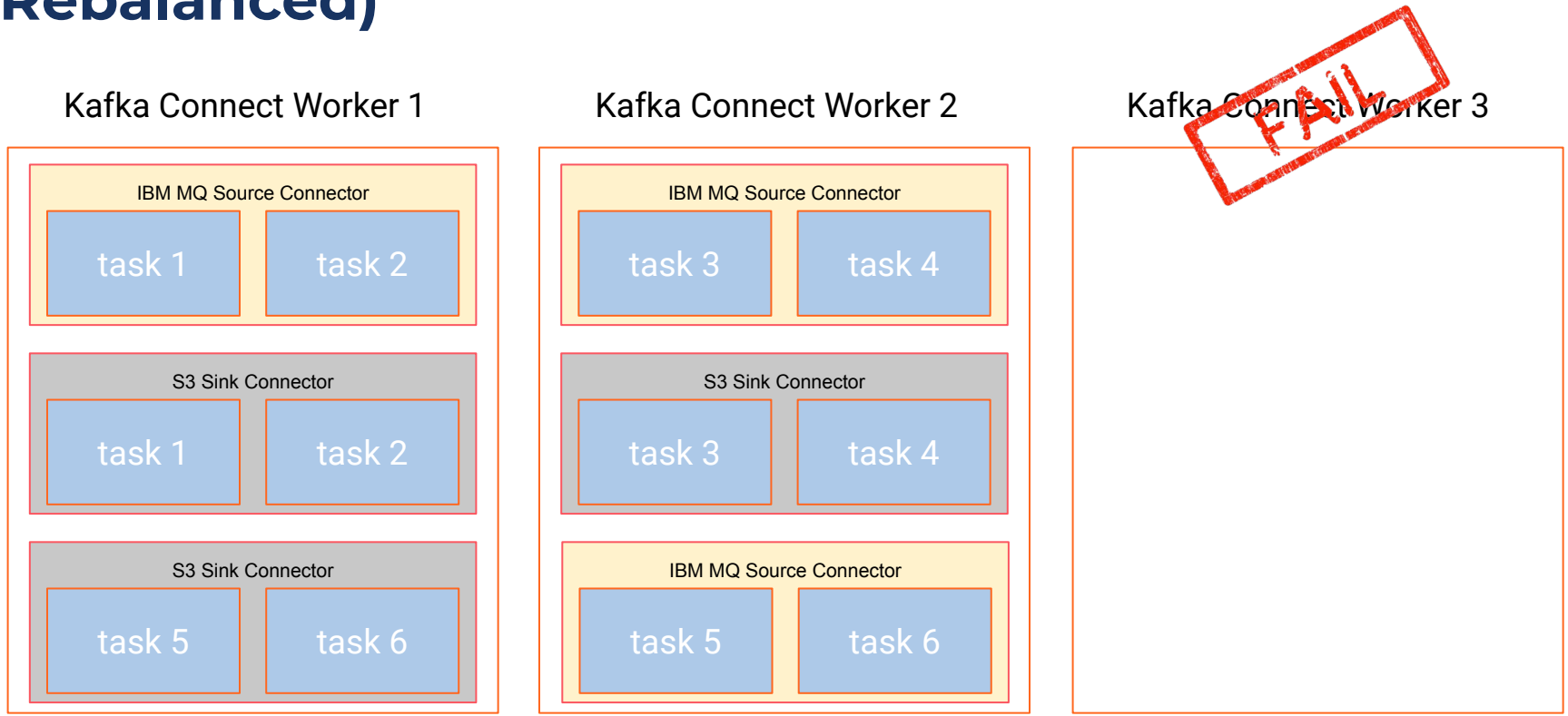
Kafka Connect: Workers, Connectors and Tasks



Kafka Connect: Workers, Connectors and Tasks (Failure Scenario)

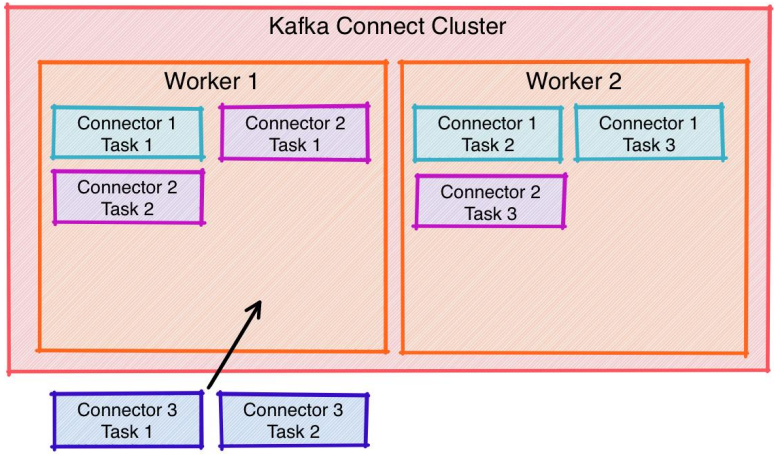


Kafka Connect: Workers, Connectors and Tasks (Rebalanced)



KIP-415: Incremental Cooperative Rebalancing in Kafka Connect

- What triggers a rebalance?
  a. Configuration of a connector changed(added, reconfigured, deleted)
  b. A worker joined/failed
- When a rebalance happened in Apache Kafka 2.2 or earlier, it stops all tasks in a Connect cluster and restarts them. This can be a hard stop for users who run multiple connectors in a Connect cluster.
- With KIP-415, a rebalance happens more gracefully. It stops only the tasks that need to move between workers (if any), leaving the rest running on their assigned worker



A new connector won't stop the existing tasks in a Kafka Connect Cluster.

KIP-449: Add connector contexts to Connect worker logs

- In addition to the previous rebalancing behavior, logs are another challenge for users to run multiple connectors in a Kafka Connect cluster.
- With KIP-449, The SLF4J API includes "Mapped Diagnostic Contexts" (MDC) that allow injection of a series of parameters that can be included in every log message written using that thread, regardless of how the SLF4J Logger instance was obtained.
    a. Makes it easier to identify which connector faces issues in Connect log.

[2019-04-02 17:01:38,451] INFO **[local-file-source|task-0] [Producer clientId=producer-1]** Cluster ID: BmIDH-sOQ0OshXX877eSrw (org.apache.kafka.clients.Metadata:274)
[2019-04-02 17:01:38,481] INFO **[local-file-sink|task-0] [Consumer clientId=consumer-1, groupId=connect-local-file-sink]** Successfully joined group with generation 5 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:456)

KIP-495: Dynamically adjust log levels in Connect

What

Enables operators to change a log level (Info, Trace, Debug) dynamically (KIP-495)

Why

- It is often required to change a log level to understand the root cause of a problem, but changing the log level means “restart connect clusters”
- Restarting connect clusters won’t be acceptable in production for many customers.
- Doing so makes it difficult to reproduce the same error.

For whom

- Developers/Operators

Key features

- Introduces an /admin/loggers endpoint to the Connect worker that can be used to get/modify the log levels of any named loggers in the Connect worker without restarting the Connect worker.

KIP-558: Track a connector's active topics

What

KIP-558 enables developers/operators/applications to easily identify topics used by connectors. easily.

Why

- During runtime it's not easy to know the topics a sink connector reads records from when a regex is used for topic selection. For a source connector, bookkeeping of active topics requires some sort of external tracking by the connector or its user.

For whom

- Developers/operators

Key features

- ```
  $ curl -s 'http://localhost:8083/connector/a-source-connector/topics'
  {"a-source-connector":{"topics":["foo","bar","baz"]}}
  ```
  ```

KIP-521: Store Connect logs in a file by default

What

Enables redirection of Connect's log4j messages to a file by default (KIP-521)

Why

- There have been lots of back-and-forth to get the right logs from customers

For whom

- Developers/Operators

Key features

- Instructs the logging framework to append log lines into a file as well as the console (standard output).

KIP-475: New metrics for connectors

What

Measure the number of tasks on a connector (KIP-475)

Why

- Increases visibility of connectors with the number of tasks and the status of tasks

For whom

- Operators

Key features

- JMX metrics will have the following metrics
  - `connector-total-task-count, connector-running-task-count, connector-paused-task-count, connector-failed-task-count, connector-unassigned-task-count, connector-destroyed-task-count`

AK 2.6

KIP-158: Kafka Connect should allow source connectors to set topic-specific settings for new topics

- Some Kafka clusters disable auto topic creation via **auto.create.topics.enable=false**, and in these cases users running source connectors must manually pre-create the necessary topics. That can be challenging for source connectors that write to lots of topics.
- With KIP-158, Kafka Connect will optionally create any topics used by source connectors using topic setting rules declared in the source connector's configuration.

KIP-585: Filter and Conditional SMTs

- Allows defining predicates that determine whether SMTs should be applied to records. This enables more complex use cases where a particular SMT should be applied selectively to a subset of records.
- Custom predicates can be used, but three useful predicates are provided:
  a. TopicNameMatches -- used to apply an SMT based on topic names
  b. HasHeaderKey -- used to apply an SMT when a record has a specific header
  c. RecordIsTombstone -- used to apply an SMT based on whether a record is a tombstone

KIP-606: Add Metadata Context to Metrics Reporter

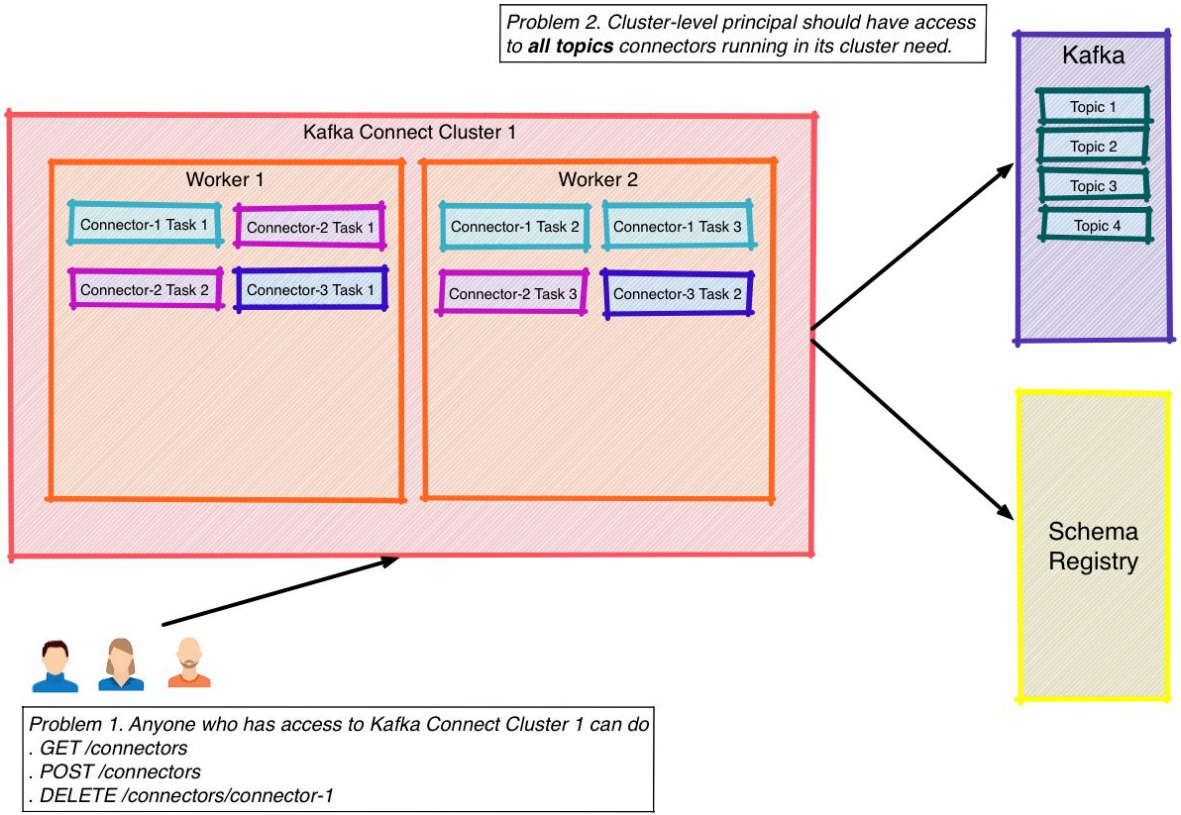
- The current metrics reporter interface does not give much context about the system exposing those metrics. For instance, it is currently difficult to infer which component or library is exposing those metrics: a broker, a connect worker, or a client library running within any of those components.
- KIP-606 passes additional metadata to metrics reporters so they can properly infer the context in which metrics are registered and which component exposes them.
- Also adds new client configurations, so any application can propagate metrics context metadata from the parent application down to the metric reporter instantiated by the client.

KIP-610: Error Reporting in Sink Connectors

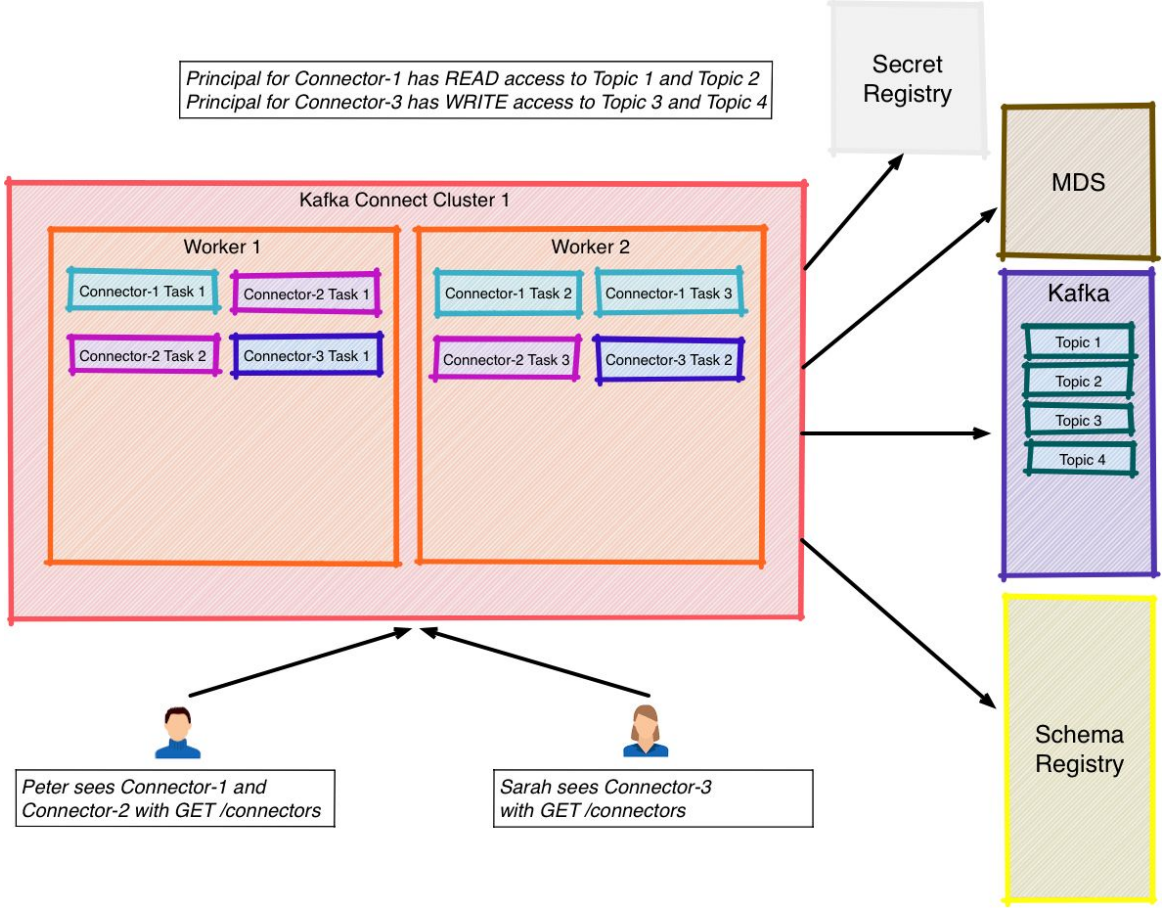
- Previously, any errors in (de)serialization and SMTs could be written to a Dead Letter Queue (DLQ) rather than failing the task. But until now a sink connector that received a problem record could either ignore the record or fail the task, but could not send it to the DLQ.
- KIP-610 allows sink connectors to report individual records as being problematic, and they will be sent to the DLQ.

Confluent Platform

Connectors and Connect before RBAC

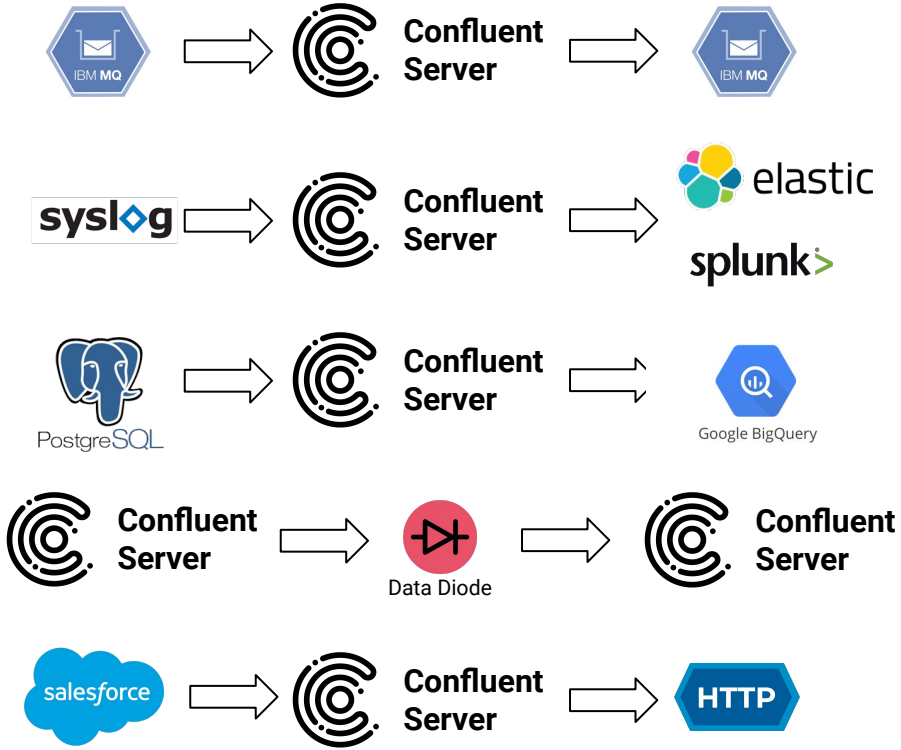


Connectors and Connect with RBAC



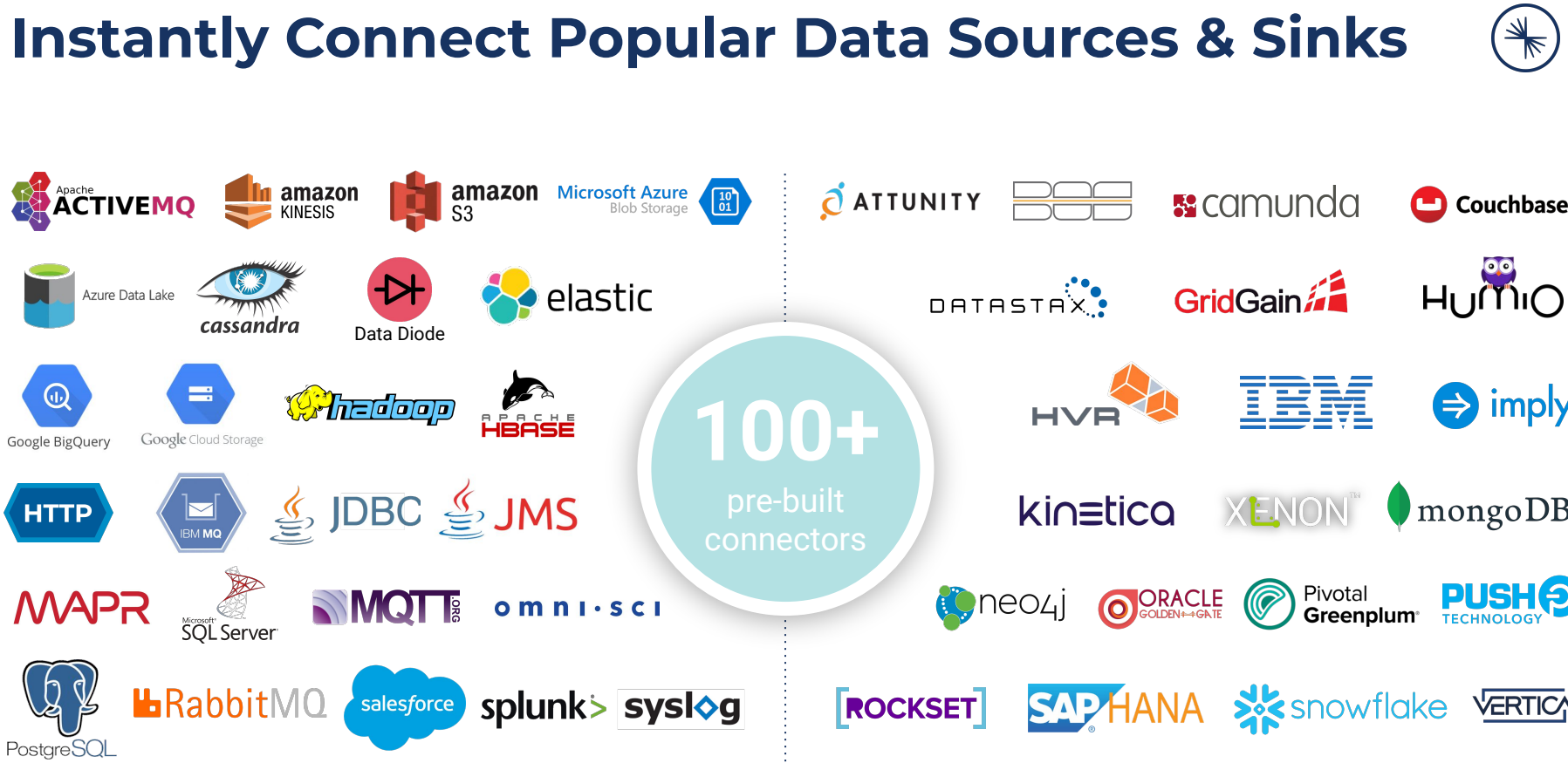
Connectors

Increase adoption velocity / Reduce TCO



Level of difficulty	Time to build/test/release (months)	Ongoing maintenance (month per year)	TCO in 3 years (months)
High	6	1.2	8.4
Medium	3	0.6	4.2
Low	1	0.2	1.4

Instantly Connect Popular Data Sources & Sinks



80+ Confluent Supported

20+ Partner Supported, Confluent Verified

JDBC vs. Debezium CDC

JDBC vs. Debezium CDC

	JDBC	Debezium CDC
Pro	• No additional setup	• Not much additional burden on database • Captures deleted records
Con	• Not able to capture deleted records • A burden on database increase as JDBC Source connector needs to fetch records from more tables.	• CDC requires additional setups in the DBMS
Notes	• Work with cloud provider's managed RDBM services (e.g. Amazon RDS for MySQL, Google Cloud SQL for MySQL, and Azure Database for MySQL)	• Debezium CDC for MySQL works with Amazon RDS for MySQL, however, it has not been validated whether it works with Google Cloud SQL for MySQL and Azure Database for MySQL • Debezium CDC for PostgreSQL works with Amazon RDS for PostgreSQL, however, it has not been validated whether it works with Google Cloud SQL for PostgreSQL and Azure Database for PostgreSQL

JDBC

Log a query for JDBC Source

What

Log a query JDBC Source executes

Why

- It is difficult to debug a query created by JDBC Source since a query is not logged in Connect log.
- This improvement reduces our support cost and improves customer's experience of JDBC Source.

For whom

- Developers

Key features

- A query is logged at the TRACE level.

The latest version of Postgres JDBC driver

What

Package the latest version, 42.2.10, of Postgres JDBC driver

Why

- Installing a JDBC driver properly is not as intuitive as it could be. Packaging the latest version removes an additional step for developers.

For whom

- Developers

Key features

- The latest version, 42.2.10, of Postgres JDBC driver comes with CP 5.5.

Key enhancements

- Confluent Platform
  - Enables retries when a connection fails. (PR-715)
  - Adds an option to start an initial query with a specific timestamp Using the `timestamp.initial` configuration property in JDBC Source (PR-429)
  - Supports a suffix query config (`query.suffix`) for DB2 (PR-730)

- Confluent Cloud
  - MySQL Source (Preview)
  - PostgreSQL Source (Preview)
  - Microsoft SQL Source (Preview)
  - Oracle Database Source (Preview)

S3 Sink

Key enhancements

- Confluent Platform
  - Adds AWS IAM Assume Role credentials provider to enable cross-account authorization.
  - AWS credentials per connector: Introduces configuration properties to easily define key and secret credentials per connector instance.
  - Allows setting a compression level for gzip in Json and ByteArray formats.
  - Upgrades AWS SDK to 1.11.725.

- Available on Confluent Cloud

Elasticsearch Sink

Add retries for ElasticSearch Sink

What

Add retries for creating an index.

Why

- Sometimes, it takes long time to create an index in ElasticSearch, resulting in timeout/connector failure.
- Connector will behave intelligently with an exponential backoff.

For whom

- Operators

Key features

- Extend a logic to cover “create index” timeout in addition to “write records” timeout.

Key enhancements

- Confluent Platform
  - Introduced an option to control HTTP-level compression (CP 5.5)
  - Improved throughput by skipping index-exists check. (CP 5.3.2)
  - Elasticsearch 7 support (CP 5.3.1)
  - Document upsert support (CP 5.3.0)

- Confluent Cloud
  - Fully-managed Elasticsearch Service Sink in preview (late Q2)

Other Enhancements

Key enhancements

- HDFS v2 Sink
  - Support ORC in HDFS Connector
  - Support for changing a topic directory dynamically for HDFS Sink Connector
- Google BigQuery Sink
  - Adds time-based partitioning in Kafka records
- Google Cloud Storage Sink / Azure Blob Storage Sink
  - Supports Parquet file format
- Azure Data Lake Storage Gen2 Sink
  - SAS token support
- RabbitMQ Source
  - TLS support has been added.

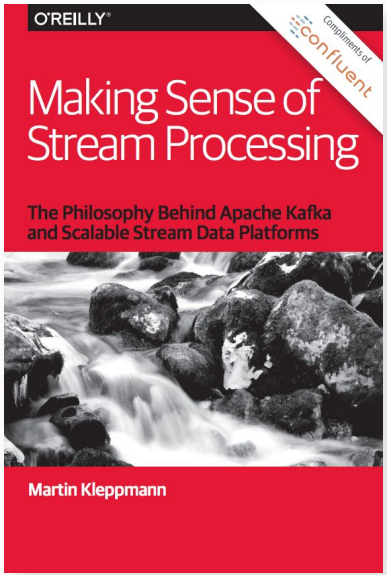
Resources

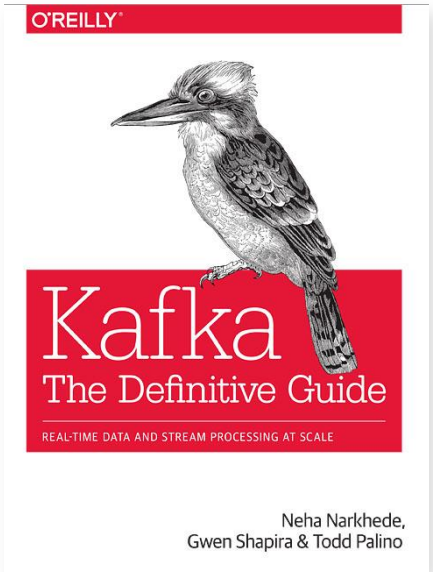
- Kafka Connect
- Confluent Hub
  - Supported Connectors
  - Preview Connectors
- Connecting Kafka Connect to Confluent Cloud
- Kafka Connect Transformations
- How to Use Single Message Transforms in Kafka Connect
- Kafka Connect Deep Dive – Converters and Serialization Explained
- From Apache Kafka to Amazon S3: Exactly Once
- Incremental Cooperative Rebalancing in Apache Kafka: Why Stop the World When You Can Change It?

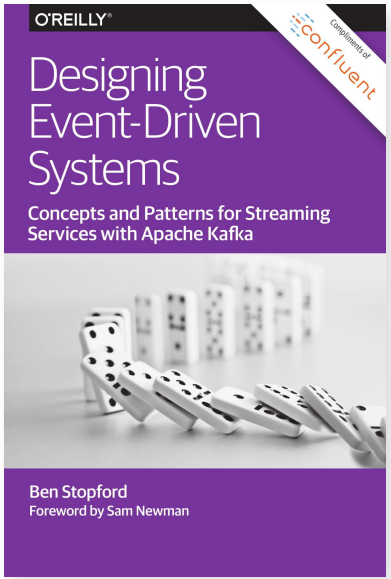
Register Now

https://events.kafka-summit.org/2020

Download your Apache Kafka and Stream Processing O'Reilly Book Bundle







Download at:

https://www.confluent.io/apache-kafka-stream-processing-book-bundle/

Appendix

Securing Internal Connect REST Endpoints

What

Secures interworker communications (KIP-507)

Why

- Prevents arbitrary rewrite of task configurations, which is a significant security vulnerability that could lead to leaking of topic data, writing arbitrary data to topics, and other serious problems.

For whom

- Operators

Key features

- Adds new sessioned option for connect.protocol configuration by default, which enables this security feature.

Extend Connect Converter to support headers

What

Enables a converter to use information in headers. (KIP-440)

Why

- Makes Kafka Connect consistent with Producer, Consumer or Stream that relies on headers for serialization / deserialization.

For whom

- Developers

Key features

- Add new default methods to the existing converter interface
  - default byte[] fromConnectData(String topic, **Headers headers**, Schema schema, Object value)
  - default SchemaAndValue toConnectData(String topic, **Headers headers**, byte[] value)

SerDe Improvements for Connect Decimal type in JSON

What

Supports decimals as they are in JSON Converter. (KIP-481)

Why

- Most JSON data that utilizes precise decimal data represents it as a decimal number. Connect, on the other hand, only supports a binary BASE64 string encoding.
  - 10.2345 with BASE64 -> D3J5
  - 10.2345 with NUMERIC -> 10.2345

For whom

- Developers

Key features

- Add a new configuration, **decimal.format,** to specify either BASE64 or NUMERIC.

KIP-131: Add access to OffsetStorageReader from SourceConnector

- Allow SourceConnector implementations to also read committed source offsets the same way that SourceTask objects can. This allows source connectors to use these offsets when they determine task configurations.

KIP-577: Allow HTTP Response Headers to be Configured for Kafka Connect

- Kafka users have reported that their security scanners have identified some missing HTTP headers from Connect REST API response.
- KIP-577 adds a new configuration property to customize HTTP response headers for all Connect REST API responses. The following is an example.

```
# [action] [header name]: [header value]

response.http.headers.config="add Cache-Control: no-cache, no-store,
must-revalidate", add X-XSS-Protection: 1; mode=block, add
Strict-Transport-Security: max-age=31536000; includeSubDomains, add
X-Content-Type-Options: nosniff
```

KIP-605: Expand Connect Worker Internal Topic Settings

- KIP-605 allows Connect to create internal topics using Kafka broker defaults for the replication factor and number of partitions, and allows setting other topic settings for internal topics. For example:

```
# Use the broker's default replication factor for these topics
config.storage.replication.factor=-1
offset.storage.replication.factor=-1
status.storage.replication.factor=-1
# Override any broker default min ISR to always use 3
config.storage.min.insync.replicas=3
offset.storage.min.insync.replicas=3
status.storage.min.insync.replicas=3
```

Other KIPs

KIP-454: Expansion of the ConnectClusterState interface
Provides filters that rewrite/validate the connector requests to enforce additional constraints on the connector configurations

KIP-458: Connector Client Config Override Policy
Allows an administrator to define/implement policy around what can be overridden in connector configurations. As a result, each connector can use a different principal so that a user could control ACL at a fine-grained level.

KIP-465: Add Consolidated Connector Endpoint to Connect REST API
Addes a new query param (`?expand=(status|info)`) to get information about all their running connectors in one API call.