Connector Lifecycle

info This document describes the lifecycle of a connector implemented in Go using theConnector SDK , which abstracts away some parts of the Connector Protocol. If you are implementing a connector in a different language you should refer to theConnector Protocol . Connector plugins that are implemented using theConnector SDK can implement 2 interfaces:Source for source connectors andDestination for destination connectors. Below we will describe the order of the method calls in sources and destinations, as well as when lifecycle events are triggered. For more information on how to implement a connector see theConnector SDK godocs , theConnector Template or theguide for building a connector .

Lifecycle

The lifecycle of the connector generally looks like this:

1. Theconnector plugin is loaded
2. The usercreates the connector
3. The usersstarts and stops the pipeline
4. The userdeletes the connector

Connector plugin is loaded

Conduit loads all connector plugins it discovers in theconnector path while starting up. Each plugin is started in a separate process to ensure it is a valid binary and to obtain the connector specifications. These specifications are then stored in Conduit's memory and exposed through the API. This helps the UI display the available connector plugins and their corresponding parameters that can be configured.

During the loading of connector specifications, theSource.Parameters andDestination.Parameters methods are invoked to retrieve the parameters. These methods are called only once, and no other methods are triggered before the connector process is stopped.

Connector is created

Upon creating a connector (POST /v1/connectors), Conduit validates the connector configuration. It starts an instance of the connector plugin and invokes theConfigure method to validate the configuration. If the method returns no errors, the configuration is deemed valid and the connector creation process continues, otherwise, the creation of the connector is aborted. After that Conduit signals the plugin to stop by invoking theTeardown method.

It's worth noting that the methods mentioned above are called either onSource orDestination , depending on the type of the connector.

Pipeline is started and stopped

Starting (POST /v1/pipelines/{id}/start) and stopping (POST /v1/pipelines/{id}/stop) a pipeline triggers the start and stop of all connectors on that pipeline. This means that a connector plugin process is started for each connector, and it is kept running until the pipeline is stopped.

When the connector plugin starts, Conduit calls its methods in a specific order. Note that the methods mentioned here are called either onSource orDestination , depending on the type of the connector.

- Configure
- is first called to validate and store the connector configuration.
- Then Conduit triggers any applicable lifecycle events, specificallyLifecycleOnCreated
- orLifecycleOnUpdated
- . Seelifecycle events
- for more detailed information on when
- each event is triggered.
- Open
- is called to allow the connector to prepare for reading or writing
- records (e.g. open connections).
- At this point the called methods depend on the type of the connector:* For a Source
  - ,
  - Conduit will start continuously callingRead
  - to fetch new records andAck
  - to signal acknowledgments for successfully processed records.

- For aDestination
- ,
- Conduit will continuously callWrite
- with records that should be written
- to the destination resource.

When the pipeline is stopped and Conduit stops reading/writing records, the last method called isTeardown . This signals that the connector plugin process will be terminated and provides a chance to clean up any open resources (e.g. close connections).

Connector is deleted

When a user deletes a connector (DELETE /v1/connectors/{id}), Conduit starts the connector plugin and triggers theOn Deleted event. After that it callsTeardown and stops the connector plugin process.

The specific methods called depend on whether the connector is aSource orDestination .

Lifecycle Events

Connector Lifecycle Events are additional integration points that let connector plugins execute certain operations when the connector is created, updated, or deleted.

These events are useful for connectors that need to initialize resources only once in their lifecycle and clean them up when the connector is deleted. For example, the Postgres connector needs to create a logical replication slot when the pipeline is started for the first time. This slot should persist between pipeline restarts but should be deleted when the pipeline is deleted. Lifecycle events can be used to implement this behavior.

The methods mentioned below are called either onSource orDestination , depending on the type of the connector.

On Created

When this event is triggered, Conduit invokes the methodLifecycleOnCreated . This method can be used to perform initialization tasks that only need to happen once in the lifetime of a connector, such as creating a bucket or preparing a replication slot. Any resources created during this method are owned by the connector and should be cleaned up during theOn Deleted lifecycle event.

The On Created event is triggered the first time the pipeline is started after the connector is created. The connector is considered "successfully started" only after this event is successfully triggered (i.e.LifecycleOnCreated returns no error). After that, the On Created event won't be triggered again, and instead,On Updated orOn Deleted events will be triggered when applicable.

Edge cases explained:

- If a connector is created and deleted without being started in between, the
- event won't be triggered.
- If a connector is created and updated without being started in between, the
- On Created event will be triggered at the next start with the latest
- configuration.

On Updated

When this event is triggered, Conduit calls the methodLifecycleOnUpdated . This method can be used to update anything that was initialized inOn Created , in case the configuration change affects it.

This event is triggered the first time the pipeline is started after the connector plugin configuration was updated, provided that the connector was started at least once before and has already received theOn Created event. The event will only be triggered if the connector plugin configuration is different than the last time the connector was started.

Edge cases explained:

- If a connector is created and updated without being started in between, the
- On Updated event willnot
- be triggered. Instead it will trigger theOn Created
- event.

- If the connector plugin configuration is updated multiple times, and the
- connector ends up with the same configuration as the last time it was started,
- then no lifecycle event will be triggered.
- If the connector plugin configuration is updated and changed multiple times,
- the On Updated event will be triggered only with the newest configuration.

On Deleted

When this event is triggered, Conduit calls the methodLifecycleOnDeleted . This method can be used to perform cleanup operations for any resources that were initialized during theOn Created event.

The On Deleted event is triggered when the connector gets deleted (DELETE /v1/connectors/{id}), but only if the connector was previously started and received theOn Created event.

Edge cases explained:

- If a connector is created and deleted without ever being started, the event
- won't be triggered.
- If a connector is updated and deleted without being started in between, the
- On Deleted event will receive the configuration that was active the last time
- the connector successfully started. Edit this page Previous Output Formats Next Kafka Connect Connectors with Conduit