

Provisioning

This document describes how provisioning of pipeline configuration files works in Conduit. To see how a pipeline configuration file is structured check the [specifications](#) .

Conduit provisions pipelines at startup

Conduit provisions all pipeline configuration files found in the [pipelines directory](#) when it starts. After that it does not monitor the configuration files for changes. If you want to apply changes to one of the configurations you need to restart Conduit for the changes to take effect.

Pipelines Directory

When Conduit starts, it will search for all files ending with .yaml or .yml in folder ./pipelines and its sub-folders. It will parse the configuration files and provision the pipelines defined in them.

If you want Conduit to search a different directory, or want to provision a single file, you can specify the CLI flag `pipelines.path` and point to your file or directory:

`./conduit -pipelines.path /path/to/my-directory` If the directory does not exist, Conduit will fail with the error "pipelines.path" config value is invalid .

Errors are logged and ignored

Conduit will try to provision all pipelines regardless of the errors it may encounter while going through the configuration files. Invalid configuration files will be ignored (e.g. files that miss a mandatory field or have an invalid configuration value) and won't impact the provisioning of valid configuration files. Conduit will log a warning for every error it encounters during this process.

Pipelines need unique IDs

You need to be careful when choosing IDs for your pipelines, connectors and processors. Each pipeline needs to get a unique pipeline ID, otherwise all pipelines with the same ID will be ignored. If a pipeline contains connectors or processors with the same ID, they will be ignored as well.

Note : Connector IDs and processor IDs will be prefixed with the parent ID. In case of a connector this means the pipeline ID will be attached as a prefix to the connector ID (e.g. `pipelineID:connectorID`). Similarly, a pipeline processor's ID will be prefixed with the pipeline ID (e.g. `pipelineID:processorID`) and a connector processor's ID will be prefixed with the connector ID (e.g. `pipelineID:connectorID:processorID`). This ensures that connector and processor IDs are globally unique as long as the pipeline ID is unique.

Environment variables

Conduit injects environment variables in pipeline configuration files. To specify a value using an environment variable, wrap the name of the variable in `{}` (e.g. `{MY_ENV_VAR}`). Any value in the configuration file can be injected using an environment variable.

This can be useful for injecting secrets into a pipeline without specifying it explicitly in the pipeline configuration file. You need to make sure that the environment variable is set when running Conduit.

Pipelines Immutability

Pipelines provisioned by configuration files are immutable , they can not be changed using the API. Any updates on a provisioned pipeline have to be done through the configuration file.

You can still stop or start the pipeline through the UI or API.

Updating a provisioned pipeline

Pipelines provisioned by configuration files can only be updated through configuration files. Conduit has to be restarted for the changes to take effect.

Generally, updates will preserve the state of the pipeline. This means that even after an update Conduit will resume the processing at the last record that was successfully processed. However, updates of some fields will cause the state to be lost, as it can't be guaranteed that the state is still applicable to the new configuration. In that case the pipeline will start

processing records from the beginning, as if it's running for the first time.

Here is a full list of fields that will cause the connector to start from the beginning if they are updated:

- [pipeline.id](#)
- - The entire pipeline will be recreated and all source connectors will start from the beginning.
- [connector.id](#)
- - The updated connector will start from the beginning (only source connectors).
- [connector.type](#)
- -
- The updated connector will start from the beginning (only source connectors).
- [connector.plugin](#)
- -
- The updated connector will start from the beginning (only source connectors).

Deleting a provisioned pipeline

Pipelines provisioned by configuration files can be deleted by removing them from the configuration file or deleting the file entirely. Next time Conduit starts it will delete the pipeline. [Edit this page](#) [Previous](#) [Getting Started](#) [Next](#) [Specifications](#)