Cloudera Data Flow

Migration Tool

Date published: 2021-04-06
Date modified: 2025-09-30

https://docs.cloudera.com/

Legal Notice

Contents

Cloudera Data Flow NiFi 2 migrations

Cloudera Data Flow enables semi-automatic migrations of flows from NiFi 1.x to NiFi 2.x, using predefined transformation rules and logic. [Technical Preview]. Learn about the migration process.

Note: This capability is currently in Technical Preview. Do not use it in production environments.

You can migrate NiFi 1.x flow definitions from the Cloudera Data Flow Catalog. If the flow definition you want to migrate is not there, you must add it first.

1. You select a NiFi 1.x flow definition from the Cloudera Data Flow Catalog and start the migration, providing the required inputs to the Migration Tool.

   Migrating a flow does not modify the source flow or the associated NiFi instances. The source flow is referenced as the starting point (NiFi 1.x) of a migration and a copy of that flow is migrated to NiFi 2.x. The source flow remains in the catalog as is.

   Migration always occurs between two specific NiFi 1 and NiFi 2 versions. The Migration Tool calculates the minimum compatible NiFi version where the created flow draft can be opened in Flow Designer and only allows you to select target workspaces that meet this requirement. If you have no access to any such workspace, you cannot proceed with the migration.
2. The Migration Tool checks the NiFi 1.x flow to identify the requirements that need to be addressed in order for it to run in NiFi 2.x. The Tool then automatically addresses any variances or anomalies that it can handle on its own. In the event that it is unable to automate a change, the Tool flags it as something that requires manual review or manual intervention.

   The tool proceeds to create a draft of the migrated flow in Flow Designer. Using this draft, you will be able to make any necessary manual changes to the flow that the tool was not able to automatically complete on its own.

   Some of the major transformations are:

   - Parameter providers are converted to parameters.
   - All controller services are moved to the root process group. Names of the moved controller services are appended with the name of their original process group and a random hash.
   - All parameters are moved to the root process group. If there are several parameters with identical names, they are renamed in the parameterContextName.parameter format, where parameterContextName is the name of the original parameter context in the source flow definition.
   - Variables, not supported in NiFi 2.x, are converted to parameters.
   - Sensitive parameter values are not migrated.
3. If the source flow had multiple parameter contexts in different process groups, you are asked to match parameter contexts to shared parameter groups or merge them into root level draft parameters in an intermediate step.
4. The created flow draft opens in Flow Designer where you address issues that require manual intervention. In Flow Designer, unresolved issues are listed in the **Migration Report** pane. In the downloadable tar.gz package you find them in the migration_report.json artifact.

   Tip:

   If you prefer not to address open issues in the Flow Designer, you may download the tar.gz package and edit the flow definition in the tool of your choice.
5. Once you have resolved all outstanding issues, publish the flow draft to the Catalog as a NiFi 2.x flow.

Running a NiFi 2 migration

You can migrate NiFi 1.x flow definitions in the Catalog and open them in Flow Designer.

Before you begin

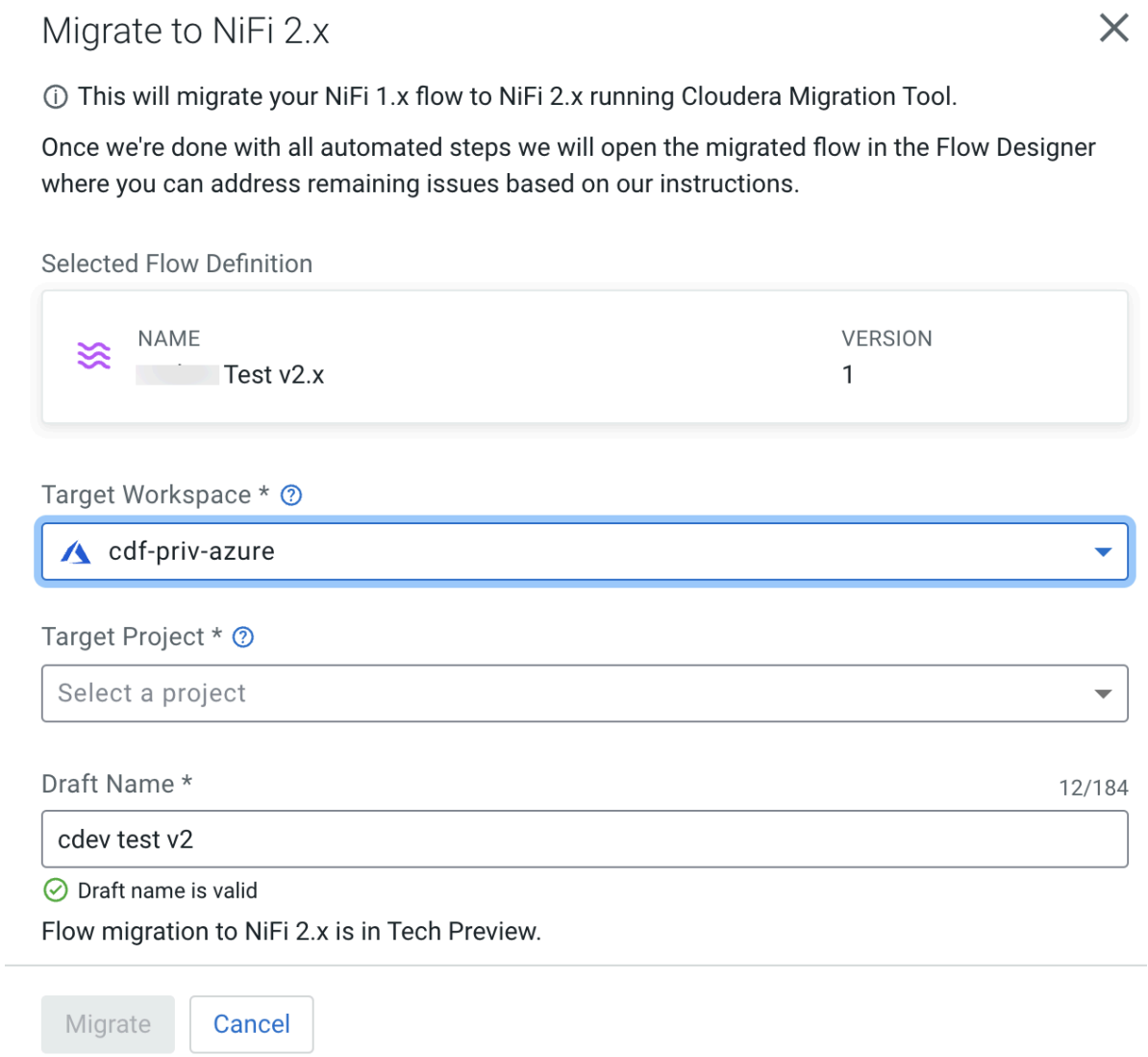
- You have been assigned at least the DFCatalogViewer role granting you access to the Catalog.
- You have been assigned at least the DFCollectionUser role granting you access to the collection where the flow definition resides.
- You have been assigned the DFFlowDeveloper role to edit the migrated flow draft in Flow Designer. You do not need this role if your migrated flow cannot be opened in Flow Designer.
- You have been granted the DFCatalogPublisher role to either publish the NiFi 2.x draft from Flow Designer, or to import it to the Catalog as a NiFi 2.x flow definition.

Procedure

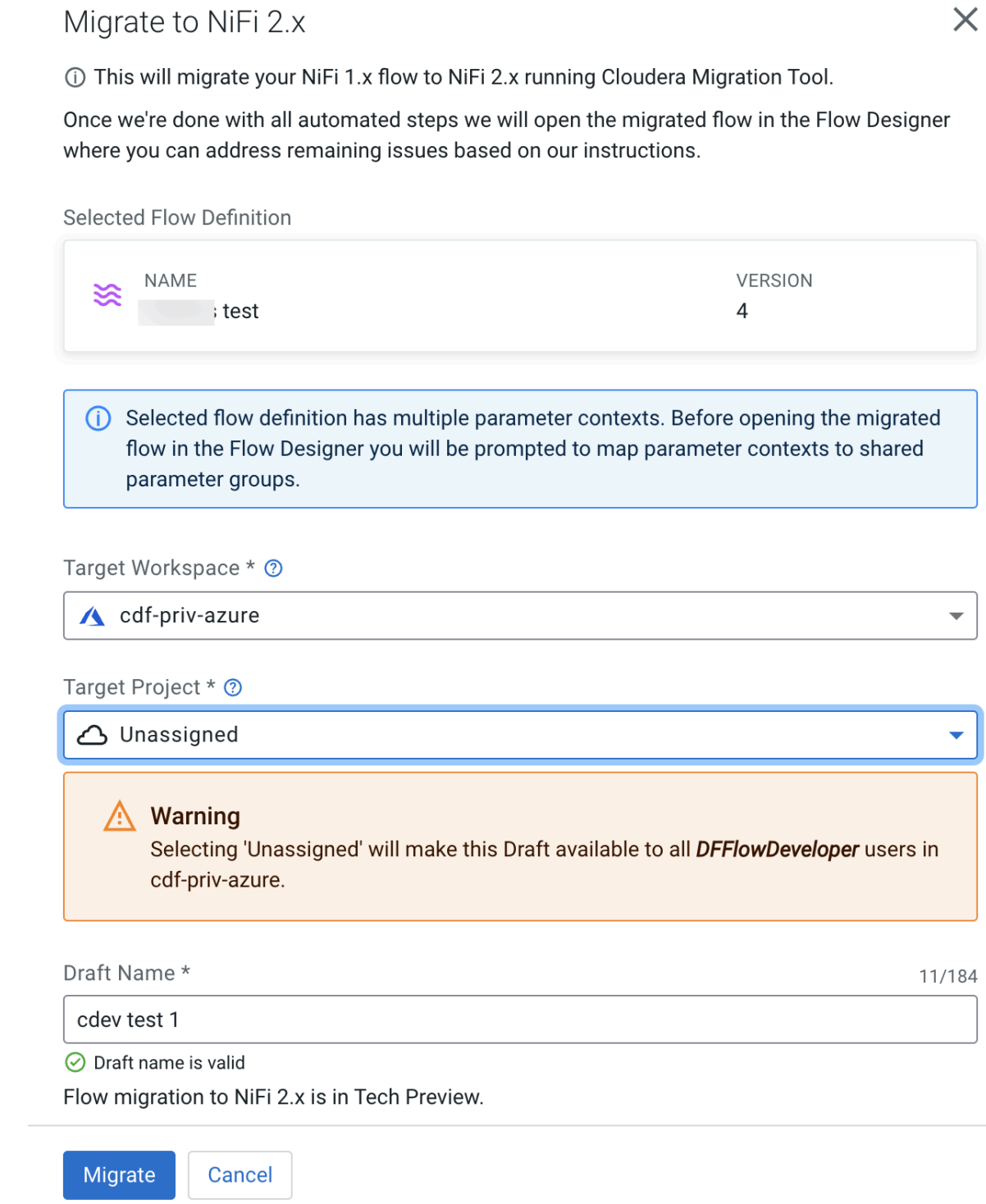
1.

Go to Data Flow Catalog .

2. Select the flow that you want to migrate, then select Actions.
   The Migrate to NiFi 2.x option is available for NiFi 1.x flows.
   Depending on the selected flow definition, you see one of the following options:
   - The NiFi 2.x flow can be opened in Flow Designer as a flow draft where you can manually address any possible remaining issues.



   - The NiFi 2.x flow can be opened in Flow Designer as a flow draft where you can manually address any possible remaining issues. You will be asked to match parameter contexts to shared parameter groups before opening the flow in Flow Designer.


Migrate to NiFi 2.x
This will migrate your NiFi 1.x flow to NiFi 2.x running Cloudera Migration Tool.
Once we're done with all automated steps we will open the migrated flow in the Flow Designer where you can address remaining issues based on our instructions.
Selected Flow Definition
NAME
test
VERSION
4
Selected flow definition has multiple parameter contexts. Before opening the migrated flow in the Flow Designer you will be prompted to map parameter contexts to shared parameter groups.
Target Workspace *
cdf-priv-azure
Target Project *
Unassigned
Warning
Selecting 'Unassigned' will make this Draft available to all DFFlowDeveloper users in cdf-priv-azure.
Draft Name *
11/184
cdev test 1
Draft name is valid
Flow migration to NiFi 2.x is in Tech Preview.
Migrate
Cancel

3. Provide the required inputs.

   Target Workspace

   Select the workspace where you want the migration to take place.

   Target Project

   Select the project where you want the NiFi 2.x flow draft to be created.

   Draft Name

   Specify a draft name that is unique in the selected workspace.

4. Click Migrate.

What to do next

- Proceed to Matching parameter contexts to shared parameter groups if prompted to do so during the migration process.
- Proceed to review the flow draft, paying close attention to the migration report that the tool provides as part of this migration process. The key items you need to either change or inspect will be surfaced. For a full list of resolved and unresolved issues, you can download the full migration report via the download option.

Related reference

Migration report

Matching parameter contexts to shared parameter groups

If the migrated NiFi 1.x flow had multiple parameter contexts, you can match them with shared parameter groups before opening the migrated NiFi 2.x flow in Flow Designer.

About this task

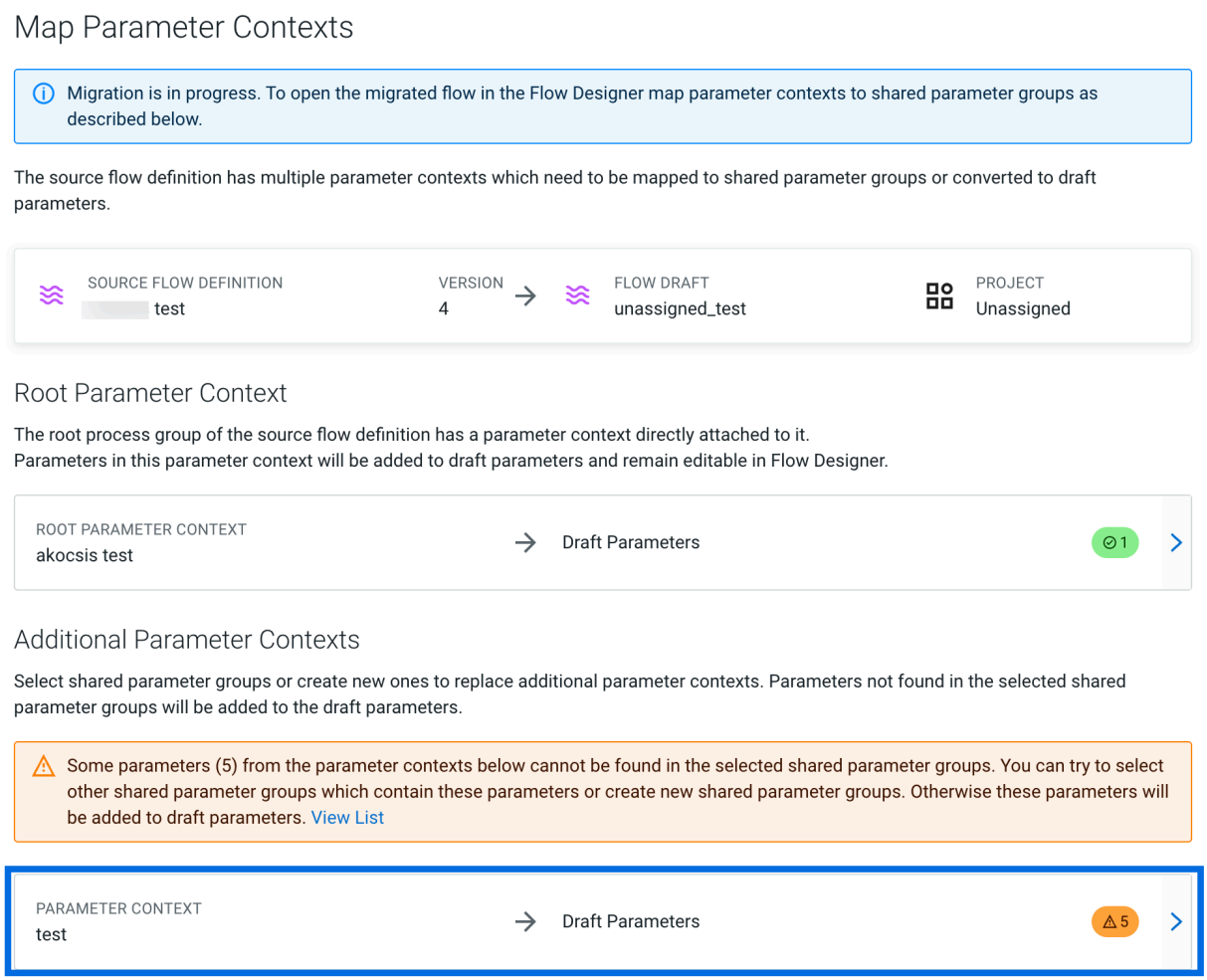
In a previous step, the Migration Tool threw this message:

> ⓘ Selected flow definition has multiple parameter contexts. Before opening the migrated flow in the Flow Designer you will be prompted to map parameter contexts to shared parameter groups.

At this point, you can either start matching parameter contexts, click Create Migrated Flow Draft to skip matching and create the flow draft with draft parameters, or click Cancel to abort migration and discard all changes.

Procedure

- Click on the PARAMETER CONTEXT you want to interact with.



- From the Select a Shared Parameter Group drop-down select the group you want to match with the selected parameter context.

  - A green badge with an icon shows the number of matching parameters in the parameter context.
  - An orange badge with an icon shows the number of parameters that have no match in the parameter context.

  Only shared parameters that match a context parameter are imported to your NiFi 2.x flow draft.
- When you are done with matching, click Create Migrated Flow Draft.

Related reference

Migration report

Related Information

Shared parameter groups

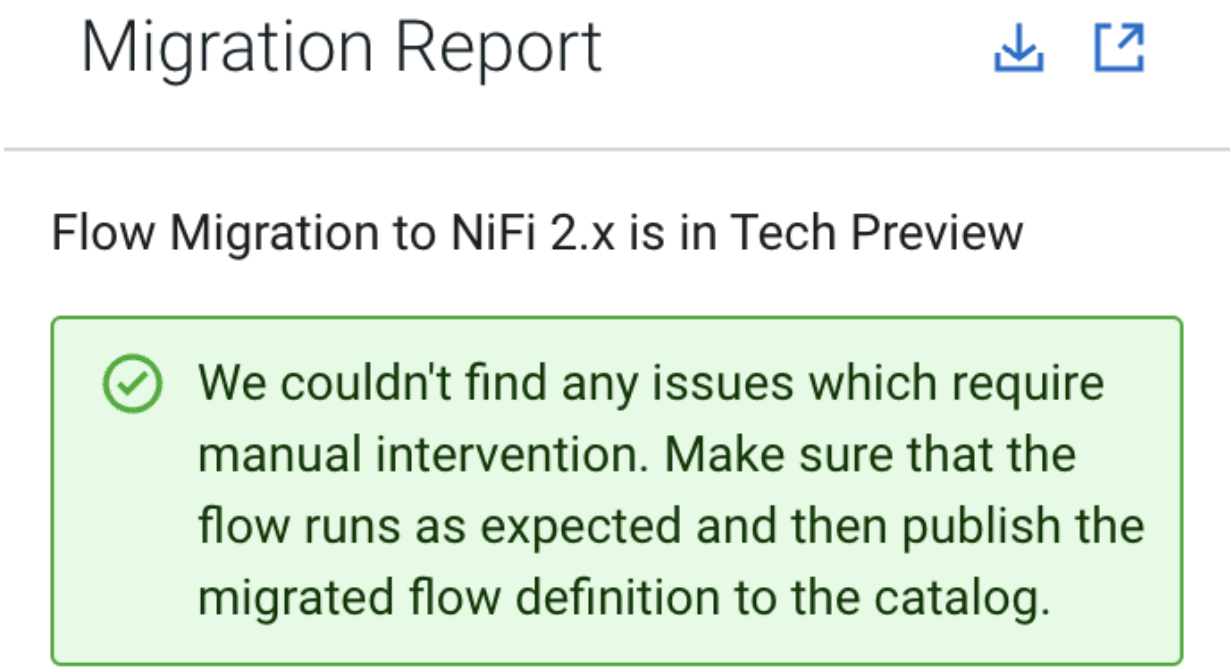
Validating the migrated flow in Flow Designer

After a migration is complete, the migrated flow is opened in Flow Designer as a NiFi 2.x flow draft. Perform required manual cleanup actions, validate that the draft works as intended, and then publish the draft to the Catalog as a NiFi 2.x flow definition.

About this task

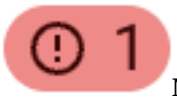
Procedure

- If the migration concludes without any remaining issues, you can proceed to verifying that the NiFi 2.x flow works as intended, for example, by starting a test session. After successful testing, publish the flow draft to the Catalog as a NiFi 2.x flow definition.



- If there are unresolved issues after the migration, they are listed in the Migration report. These are items that require some level of attention, or possible manual intervention.

  These issues fall into one of the following types:

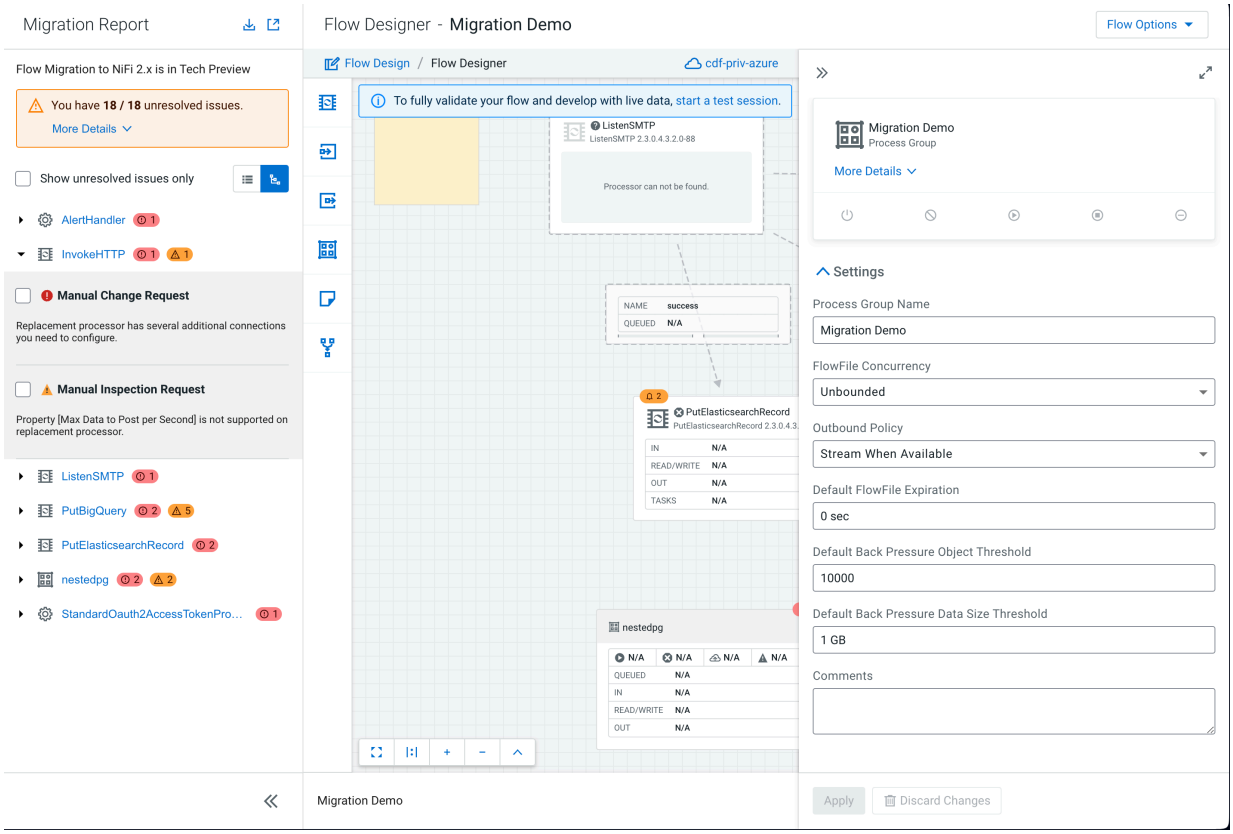
  Manual Change Request

  Describes changes that cannot be applied automatically and require human intervention. For example, deprecated components without replacements or decisions requiring domain expertise.

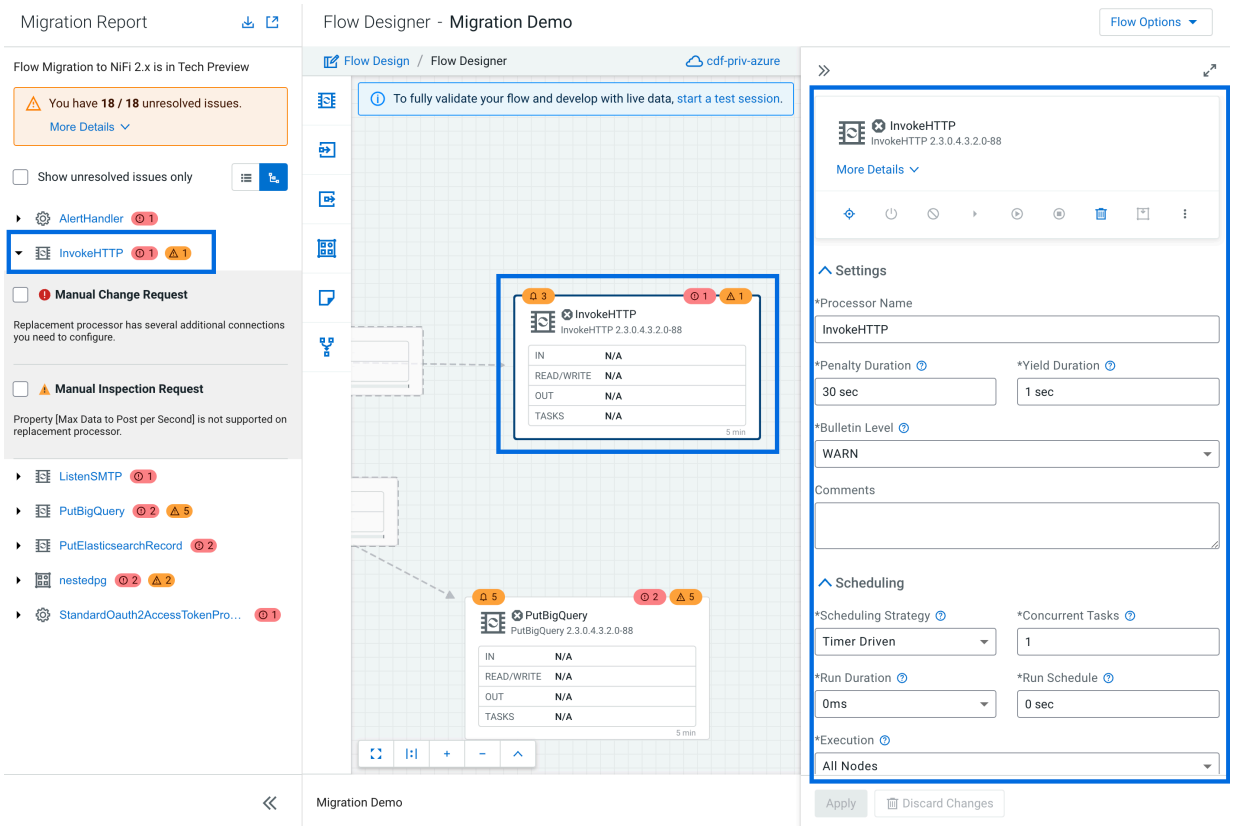
  Manual Inspection Request

  Highlights changes that should be explicitly reviewed to ensure correctness. These changes may or may not require adjustments.



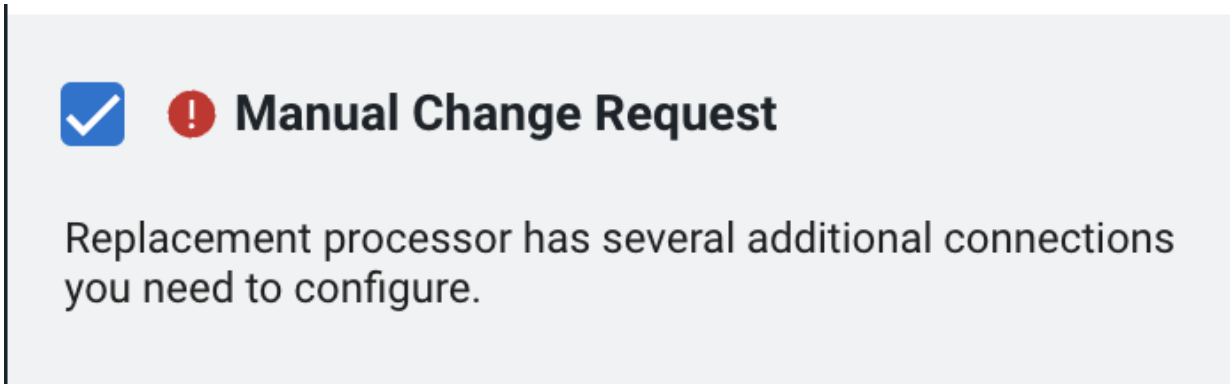
Tip: If you prefer to edit your flow draft in an external tool, you can download the migration report by clicking download.

- Selecting an issue in the migration report highlights the associated component on the **Canvas** and also opens it for editing in the **Configuration pane**.



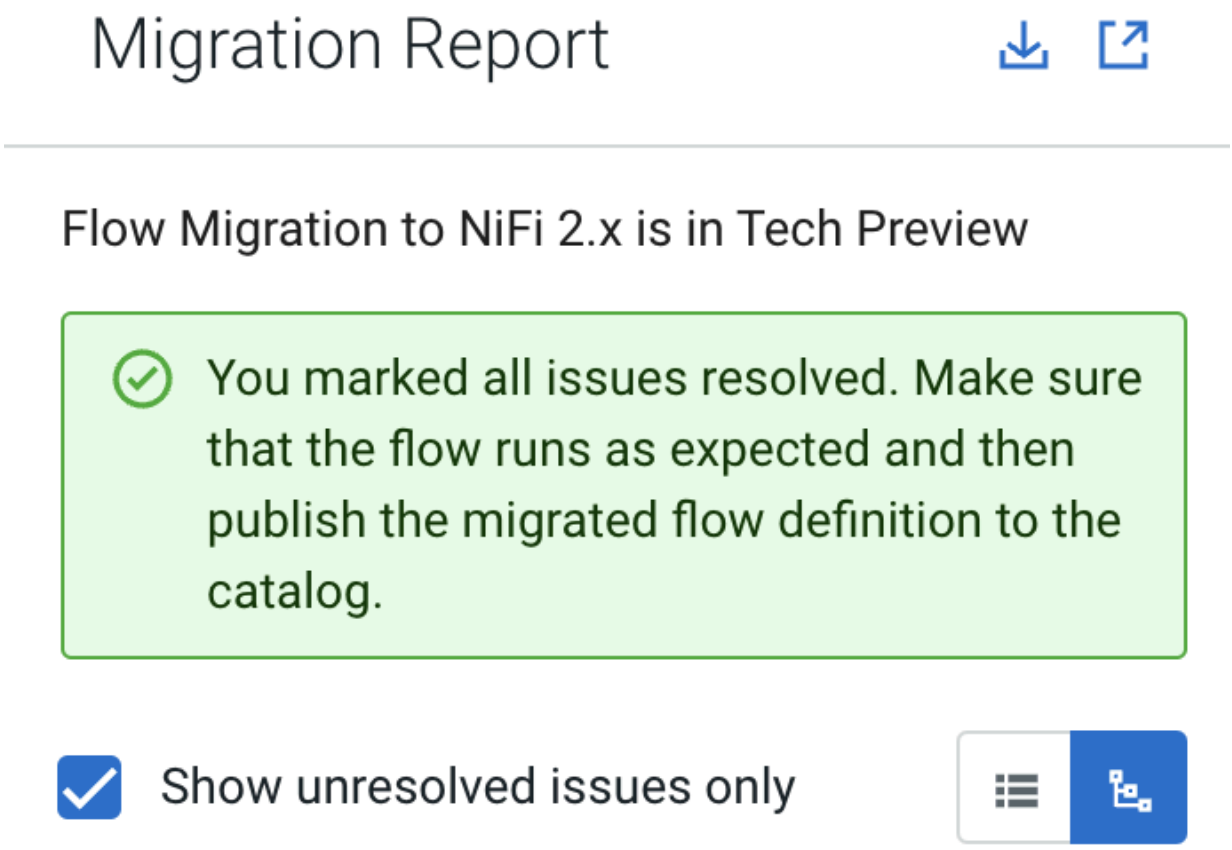
Inspect the component and make the necessary modifications.

- Once you address an issue, you can mark it resolved by selecting the checkbox next to it.



Important: The sole purpose of the checkboxes is to track your progress. Interacting with them has no effect on the flow draft itself.

- After resolving all outstanding issues, test that the NiFi 2.x flow works as intended, for example, by starting a test session.



Editing a migrated flow outside of Flow Designer

If you want to edit the migrated NiFi 2.x flow definition outside Flow Designer, you can download it with its associated migration report and resolve open issues in an external tool.

Procedure

1. **Tip:** If you prefer to edit your flow draft in an external tool, you can download the migration report by clicking download.

   Once the migration is complete, click [Download] in Flow Designer to retrieve the TAR.GZ package containing the flow_definition.json and migration_report.json files.
2. Open the flow_definition.json and migration_report.json files.

3. In the migration_report.json file look for manual-change-request and manual-inspection-request entries. and modify the flow_definition.json accordingly.

   Manual Change Request

   Describes changes that cannot be applied automatically and require human intervention. For example, deprecated components without replacements or decisions requiring domain expertise.

   Manual Inspection Request

   Highlights changes that should be explicitly reviewed to ensure correctness. These changes may or may not require adjustments.

4. After addressing all open issues, import the flow definition to Cloudera Data Flow Catalog.

Related Information

Migration report

Importing flow definitions from Apache NiFi

Migration report

This reference contains information about migration_report.json. This log file provides a comprehensive record of the migration process, capturing detailed information about changes, events, and required interventions.

Summary

- File name: migration_report.json
- Serves a business-level record of changes and the context of the migration process.
- Provides a detailed log of the activities during migration in JSON format that follows a structured schema: it begins with metadata and consists of a list of entries.
- Designed for human readability and automated filtering, but not for fully automatic processing.

Metadata

- Provides context for understanding the log entries.
- Metadata fields:
  - recordingStarts: (number) UNIX epoch timestamp indicating when logging began, the approximate start of the stage run (not suitable for performance measurements)
  - recordingEnds: (number) UNIX epoch timestamp indicating when logging concluded, the approximate end of the stage run (not suitable for performance measurements)
  - stage: (string) Identifies the stage that is run. Possible values are "STAGE_1" or "STAGE_2"

Entry

- The migration report contains a list of entries, each representing a distinct event using different entry types.
- Each entry serves to provide insight into the actions performed during the migration process.

Entry properties

- sequence: A unique number within the log file, that represents the entry's sequence. It can be used to order and reference entries.
- type: A string denoting the purpose of the entry. For more information, see Entry types below.
- subject: A string representing the entity involved in the entry, such as a component, group, or other relevant entity.

Note: In the case of component migration, the entry’s subject refers to the triggering component. This is important when a component is replaced, as the log will reference the original component’s ID (which may have been deleted at this point) rather than the new component's ID.

- message: A string containing the main content or description of the entry.
- context: A JSON object with additional key-value pairs, providing further details about the entry (for example: applied rules, affected properties).

Entry types

- manual-change-request
  - Subject: Usually the ID of a component requiring manual intervention
  - Message: Describes changes that cannot be applied automatically and require human intervention. For example, deprecated components without replacements or decisions requiring domain expertise.
  - Context: Not commonly used
  - Example:

```
{
      "sequence": 5,
      "type": "manual-change-request",
      "subject": "69c4f8f3-549a-4103-b8f6-f8627b689e12",
      "message": "The value [null] for property [assume-r
ole-sts-region] is not supported any more.",
      "context": {  }
    }
```

- manual-inspection-request
  - Subject: Typically the ID of a component under review.
  - Message: Highlights changes that should be explicitly reviewed to ensure correctness. These changes may or may not require adjustments.
  - Context: May include a rule key with a UUID for the applied migration rule, which helps identify errors.
- change-info
  - Subject: The component being migrated.
  - Message: Describes version differences and provides context for changes, without indicating any actual transformation.
  - Context: Includes a rule key with a UUID for the applied migration rule.
  - Example:

```
{
      "sequence": 4,
      "type": "change-info",
      "subject": "69c4f8f3-549a-4103-b8f6-f8627b689e12",
      "message": "Property [assume-role-sts-region] has bee
n changed as follows: [Allowed values have been changed. Val
ues removed: [eu-isoe-west-1]]",
      "context": {
        "rule": "6ca65082-09bd-4713-9fde-f5beb0722d1c"
      }
    }
```

- change
  - Subject: The entity (for example: a component or template) that has been modified during migration.
  - Message: Records actual changes applied to the flow during migration. These entries provide specifics on transformations in response to version differences.
  - Context: Includes a rule key with a UUID and may contain additional transformation details. While change-info describes general version differences in NiFi (for example: Processor ConsumeKafka_1_0 has been removed from NiFi), change entry details the specific transformations applied to your flow to address those changes (For example: Deprecated

processor ea9178e3-ad81-4c6f-b23c-88a01e08ddac has been replaced with processor d9730bae-4f88-430e-801d-8805f6069988).

- Example:

```
{
      "sequence": 39,
      "type": "change",
      "subject": "0a8d5eb6-6791-1faa-2303-b0adcf300df1",
      "message": "Property [Kerberos Principal] has been re
moved",
      "context": {
        "rule": "65d775f9-dc70-45c9-b4d4-e18a21f8ccba"
      }
    }
```

- control
  - Subject: Unspecified, often a Process Group
  - Message: Marks the progression of migration activities, such as the start or completion of a Process Group's migration.
  - Context: No widespread usage
  - Example:

```
{
      "sequence": 4,
      "type": "control",
      "subject": "0a8d5a99-6791-1faa-78ea-9582a48a4113",
      "message": "Group migration has started",
      "context": {
      }
    }
```