

## Cloudera DataFlow for Data Hub Release Notes

Date published: 2019-12-16

Date modified: 2025-06-27

# CLOUDERA

# Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>What's new in Cloudera DataFlow for Data Hub 7.3.1.....</b>	<b>5</b>
What's new in Flow Management with NiFi 1.....	5
What's new in Flow Management with NiFi 2.....	6
What's new in Edge Management [Technical Preview].....	10
What's new in Streams Messaging.....	10
What's new in Cloudera Streaming Analytics.....	11
 <b>Component support in Cloudera DataFlow for Data Hub 7.3.1.....</b>	 <b>12</b>
 <b>Supported NiFi extensions in Cloudera DataFlow for Data Hub 7.3.1.....</b>	 <b>13</b>
Supported NiFi extensions in Cloudera DataFlow for Data Hub 7.3.1.400.....	13
Supported NiFi processors.....	13
Supported NiFi controller services.....	20
Supported NiFi reporting tasks.....	24
Supported NiFi parameter providers.....	25
Supported NiFi flow analysis rules.....	25
Supported NiFi flow registry clients.....	26
Cloudera exclusive components.....	26
Components supported by partners.....	28
Supported NiFi extensions in Cloudera DataFlow for Data Hub 7.3.1.0.....	28
Supported NiFi processors.....	28
Supported NiFi controller services.....	35
Supported NiFi reporting tasks.....	39
Supported NiFi parameter providers.....	40
Supported NiFi flow analysis rules [Technical Preview].....	41
Supported NiFi Python components [Technical Preview].....	41
Cloudera exclusive components [Technical Preview].....	43
Components supported by partners.....	45
 <b>Unsupported features in Cloudera DataFlow for Data Hub 7.3.1.....</b>	 <b>46</b>
Unsupported Flow Management features.....	46
Unsupported Edge Management features [Technical Preview].....	46
Unsupported Streams Messaging features.....	46
Unsupported Cloudera Streaming Analytics features.....	47
 <b>Known issues In Cloudera DataFlow for Data Hub 7.3.1.....</b>	 <b>48</b>
Known issues in Flow Management.....	48
Known issues in Edge Management [Technical Preview].....	55
Known issues in Streams Messaging.....	55
Known issues in Cloudera Streaming Analytics.....	63
 <b>Deprecation notices in Cloudera DataFlow for Data Hub 7.3.1.....</b>	 <b>64</b>
Deprecation notices for Cloudera Streaming Analytics.....	65

## **Fixed issues in Cloudera DataFlow for Data Hub 7.3.1..... 65**

Fixed issues in Flow Management.....	65
Fixed issues in Edge Management [Technical Preview].....	76
Fixed Issues in Streams Messaging.....	77
Fixed Issues in Cloudera Streaming Analytics.....	78
UI fixes and improvements.....	79

## **Fixed CVEs in Cloudera DataFlow for Data Hub 7.3.1..... 79**

CVE-2021-45105 & CVE-2021-44832 remediation for Cloudera DataFlow for Data Hub.....	79
Fixed CVEs in Flow Management.....	80

## **Behavioral changes in Cloudera DataFlow for Data Hub 7.3.1..... 91**

Behavioral changes in Flow Management.....	91
Behavioral changes in Flow Management in Cloudera DataFlow for Data Hub 7.3.1.400.....	91
Behavioral changes in Flow Management in Cloudera DataFlow for Data Hub 7.3.1.0.....	96
Behavioral changes in Streams Messaging.....	101
Behavioral changes in Cloudera Streaming Analytics.....	101

## What's new in Cloudera DataFlow for Data Hub 7.3.1

Cloudera DataFlow for Data Hub 7.3.1 includes documentation for components used in Flow Management, Edge Management, Streaming Analytics, and Streams Messaging Data Hub clusters. Learn about the new features and improvements available for these components.

### What's new in Flow Management with NiFi 1

Learn about the new Flow Management features using NiFi 1 available in Cloudera Data Hub clusters in Cloudera on cloud 7.3.1.

Cloudera 7.3.1.0 and 7.3.1.400 are platform-level releases defining the base environment used to create and run different Data Hub cluster types including Flow Management. Flow Management Data Hub clusters are compatible with both NiFi 1 and NiFi 2.

#### 7.3.1.400

The following sections provide details about Cloudera Flow Management 2.2.9.400 based on **Apache NiFi 1.28.1**, with information on the most important new features, improvements, and fixes included in this release. The new version incorporates several stability and security improvements.

##### New processors

- CopyS3Object
- GetS3ObjectMetadata
- SplitExcel
- DeleteFile
- DeleteSFTP

These enhancements improve integration with S3 and provide expanded capabilities for file and data management.

##### New controller service

- HTTPRecordSink

For a comprehensive list of supported NiFi components in Cloudera 7.3.1.400 Flow Management Data Hub clusters, see [Supported NiFi extensions](#).

##### Migration tool

Cloudera provides a Flow Management Migration Tool that helps you replace deprecated processors, update configurations, and handle breaking changes automatically in your data flows. For more information, see the [Cloudera Flow Management Migration Tool documentation](#).

##### Fixed issues

For detailed information about the issues resolved in Cloudera Flow Management Data Hub 2.2.9.400, see [Fixed issues in Flow Management](#) and [Fixed CVEs in Flow Management](#)

For more information about the latest updates in Cloudera Flow Management Data Hub clusters using NiFi 2, see [What's new in Flow Management with NiFi 2](#).

#### 7.3.1.0

The following sections provide details about Cloudera Flow Management 2.2.9.0 based on **Apache NiFi 1.28.1**, with information on the most important new features, improvements, and fixes included in this release. The new version incorporates several stability and security improvements.

### Rebase on NiFi 1.28.1

This upgrade offers access to the newest NiFi features and enhancements on the 1.x branch.

important

Direct upgrades from NiFi 1.28 to NiFi 2.0 are currently not supported. To transition, you can create new Flow Management clusters using NiFi 2-based templates and migrate your existing flows to the new clusters.

Note that NiFi 2.0 in Data Hub is currently in Technical Preview. NiFi 2-based clusters are not production-ready and should not be used for critical workloads.

### New NiFi components

For a comprehensive list of supported NiFi components in Flow Management 7.3.1.0 Data Hub clusters, see [Supported NiFi extensions](#).

For more information about the latest updates in Flow Management Data Hub clusters using NiFi 2, see [What's new in Flow Management with NiFi 2](#).

## What's new in Flow Management with NiFi 2

Learn about the new Flow Management features using NiFi 2 available in Cloudera Data Hub clusters in Cloudera on cloud 7.3.1.

Cloudera 7.3.1.0 and 7.3.1.400 are platform-level releases defining the base environment used to create and run different Data Hub cluster types including Flow Management. Flow Management Data Hub clusters are compatible with both NiFi 1 and NiFi 2.

### 7.3.1.400

The following sections provide details about Cloudera Flow Management 4.2.1.400 based on **Apache NiFi 2.3.0**, with information on the most important new features, improvements, and fixes included in this release. This release includes a range of Cloudera-specific capabilities and improvements, providing an enterprise-ready foundation for building, operating, and scaling your data pipelines. Switching to NiFi 2 in your data flow development and operation is not just a technical update, it is a modernization of the whole data flow experience, offering greater performance, flexibility, and security.

### Native Python 3 support

NiFi 2 introduces a powerful Python API, enabling you to develop processors, controller services, and reporting tasks directly in Python. This allows you to embed Python-based orchestration and data manipulation into your flows without relying on external scripting.

### AI & GenAI components

NiFi 2 provides a framework for building AI-enabled processors. Cloudera extends this with production-grade GenAI components, supporting advanced AI-driven workflows.

### Flow analysis rules

A new, built-in rules engine enables real-time data quality validation within your flows. You can define rules to detect schema violations, missing fields, or implement custom business logic, all without additional coding.

### Enhanced authentication

NiFi 2 removes legacy Kerberos configuration methods and enforces the use of Kerberos UserService. Authentication and encryption are now stricter and more secure by default.

### Parameterization enhancement

Global variables have been replaced by a granular parameter framework, improving flow modularity and manageability.

**Redesigned user interface**

The new NiFi 2 UI is faster, cleaner, and more intuitive. It supports modular flow design with reusable components, speeding up development and allows for the centralization of design patterns in large projects.

**Migration tool**

Cloudera provides a Flow Management Migration Tool that helps you replace deprecated processors, update configurations, and handle breaking changes automatically in your data flows. For more information, see the [Cloudera Flow Management Migration Tool documentation](#).

**Other improvements**

NiFi 2 brings significant performance enhancements, especially important for high-volume workloads. It expands connectivity options both for source and target systems, simplifying to connect your data pipelines in hybrid environments. With richer native integration features, you can reduce reliance on custom processors. NiFi 2 also improves developer productivity with enhanced SDKs for Python and Java, enabling faster development and more efficient flow management.

**New NiFi components**

Cloudera Flow Management 4.2.1.400 introduces several new NiFi components to support broader integration and processing capabilities.

**New processors:**

- GetBoxFileCollaborators
- ExecuteSparkInteractive
- GetBoxGroupMembers
- ConsumeBoxEnterpriseEvents
- SawmillTransformRecord
- PutSolrRecord
- CaptureChangeDebeziumMongoDB
- SawmillTransformJSON
- GetSolr
- GetS3ObjectTags
- FetchBoxFileRepresentation
- PutSolrContentStream
- ListBoxFileInfo
- QuerySolr
- ListenBeats
- ConsumeBoxEvents
- FetchBoxFileInfo

**Python processors:**

- New
  - PromptClaude
  - TokenCount
  - PromptAzureOpenAI
  - PromptOpenAI
- Renamed
  - Bedrock renamed to PromptBedrock

**New controller services:**

- ClouderaEncodedSchemaReferenceReader
- ClouderaEncodedSchemaReferenceWriter

- PhoenixThickConnectionPool
- PhoenixThinConnectionPool
- ClouderaAttributeSchemaReferenceWriter
- DeveloperBoxClientService
- RESTCatalogService
- ClouderaAttributeSchemaReferenceReader
- LivySessionController
- PEMEncodedSSLContextProvider
- StandardDatabaseDialectService

**New parameter provider:**

- PropertiesFileParameterProvider

**Flow Analysis Rules:**

- New
  - RequireMergeBeforePutIceberg
  - RestrictFlowFileExpiration
- Renamed
  - RestrictBackpressure replaced with RestrictBackpressureSettings

For a comprehensive list of supported NiFi components in Cloudera 7.3.1.400 Flow Management Data Hub clusters, see [Supported NiFi extensions](#).

**Removed components**

Over 140 components have been removed and must be replaced in NiFi 2. For more information on breaking changes between NiFi 1 and 2, see [Behavioral changes in Flow Management](#).

**Readded components**

Cloudera Flow Management 4.2.1.400 restores support for several NiFi components that were deprecated in Apache NiFi 2. These components have been re-added to maintain compatibility and support key use cases.

**Processors:**

- PutSolrContentStream
- PutSolrRecord
- QuerySolr
- ExecuteSparkInteractive

**Controller service:**

- LivySessionController

**Upgrade and migration options**

There is no supported in-place upgrade path from Flow Management Data Hub clusters powered by NiFi 1 to Flow Management Data Hub clusters powered by NiFi 2.

Cloudera provides a Migration Tool that automates complex, repetitive, and error-prone manual tasks in updating flow configurations, reducing manual effort and ensuring compatibility with NiFi 2 features. This Cloudera Flow Management Migration Tool simplifies the transition by:

- Replacing removed processors
- Converting variable-based configurations to parameters
- Reconfiguring flows to use new controller services
- Converting older templates into flow definitions
- Adapting to security, data type, and API changes introduced in NiFi 2



**Important:**

To transition from Cloudera Flow Management versions powered by NiFi 1 to Cloudera Flow Management 4.2.1.400 and use the Migration Tool for migrating your flows, you must be on Cloudera Flow Management 2.2.9.400. For version details, see the [component versions](#) page. For more information on the migration tooling, see the [Cloudera Flow Management Migration Tool documentation](#).

**7.3.1.0**

The following sections provide details about Cloudera Flow Management 4.2.1.0 based on **Apache NiFi 2.0.0**, with information on the most important new features, improvements, and fixes included in this release. NiFi 2 introduces numerous changes compared to NiFi 1, including several breaking changes. See the [Behavioral changes](#) for more information about the differences. Additionally, expect further breaking changes in future releases, particularly with components being removed in favor of more efficient alternatives.

**Rebase on NiFi 2.0.0 M2**

This upgrade offers access to the newest NiFi features and enhancements on the 2.x branch.

**Important:**

Direct upgrades from NiFi 1.28 to NiFi 2.0 are not supported. To transition, you can create new Flow Management clusters using NiFi 2-based templates and migrate your existing flows to the new clusters.

Note that NiFi 2.0 in Data Hub is in Technical Preview. NiFi 2-based clusters are not production-ready and should not be used for critical workloads.

**Python processors**

One of the key features introduced in Apache NiFi 2 is native support for Python processors. This capability allows you to create custom processors using Python, enabling seamless integration of Python scripts into your dataflows. With each milestone release of NiFi 2, Python integration continues to evolve, providing developers with enhanced functionality, greater flexibility, and more powerful tools for building robust dataflows.

The below list shows the Python processors that are available in Flow Management 7.3.1.0 clusters using NiFi 2.

- Bedrock
- ChunkData
- ChunkDocument
- EmbedData
- InsertToMilvus
- LexicalQueryMilvus
- ParseDocument
- PartitionCsv
- PartitionDocx
- PartitionHtml
- PartitionPdf
- PartitionText
- PromptChatGPT
- PutChroma
- PutOpenSearchVector
- PutPinecone
- PutQdrant
- QueryChroma
- QueryOpenSearchVector

- [QueryPinecone](#)
- [QueryQdrant](#)
- [VectorQueryMilvus](#)

For a comprehensive list of supported NiFi components in Cloudera 7.3.1.0 Flow Management Data Hub clusters, see [Supported NiFi extensions](#).

For more information about the latest updates in Flow Management Data Hub clusters using NiFi 1, see [What's new in Flow Management with NiFi 1](#).

## What's new in Edge Management [Technical Preview]

Learn about the Technical Preview for Light Duty Edge Flow Management cluster definitions introduced in Cloudera DataFlow for Data Hub 7.3.1 in Cloudera on cloud.

Edge Flow Management cluster definitions provide all Cloudera Edge Management 2.0.0 functionalities. This means an improved user experience with enhanced functionalities for seamless management and integration. Edge Flow Manager integrates with Cloudera User Management, making it easier to manage users and groups. For more information about the integration between Edge Flow Manager and Cloudera User Management, see [After creating your cluster](#) and [Managing user groups using LDAP](#).

## What's new in Streams Messaging

Learn about the new Streams Messaging features in Cloudera DataFlow for Data Hub 7.3.1.

### Kafka

#### **Kafka Rolling Restart check—all partitions fully replicated**

A new broker rolling restart check option, all partitions fully replicated has been introduced. Selecting this option ensures that all partitions are in a fully synchronized state when a broker is stopped.

### Schema Registry

#### **Enable Streams Messaging Manager principal as trusted proxy user in Schema Registry**

Streams Messaging Manager usually connects to Schema Registry on behalf of an end user. For requests coming from Streams Messaging Manager, Schema Registry can now extract and authorize the end user to authorize the request.

### Streams Messaging Manager

#### **Validation for duplicate property keys in Kafka Connect connector configuration**

When validating Kafka Connect connector configurations, a warning is displayed if the configuration contains duplicate property keys. Duplicate property keys are highlighted with orange. The form can still be validated with the warnings present, but if there are duplicates, you are notified that only the value of the last occurrence is used.

#### **Search supports regular expressions**

The search component on the Topics, Brokers, Consumers, Producers page can now perform a regexp search.

#### **Visual clue when restarting on Kafka Connect**

When clicking restart on Kafka Connect tasks or connectors, a loading circle is displayed in case of synchronous calls. The loading circle disappears once a response is received. For asynchronous calls, a pop-up is displayed, stating that the task or connector is restarted.

#### **UX improvements**

- Fixed text overflow in the side panel column headers

- Listing page table headers are now sticky of the nested table headers
- Listing page table styling has been improved for readability
- Filter selector drop-downs are now styled consistently
- Sidebar menu pop-ups are no longer hidden under tables
- Class names on the Kafka Connect popup are now wrapped into the containing pop-ups
- The password field is no longer obfuscated when using a file provider as a password
- Fixed the alignment of values on the Connector metrics page
- Source and sink connectors are now separate tabs on the connector creation modal
- Fixed visual issues on the topic creation modal
- Increased consistency in element contrast and text style throughout the UI
- Active and Inactive statuses now have high contrast
- The expand icon is now consistent throughout the UI

### **Expand security-related headers set by SMM**

The following security related headers were added to Streams Messaging Manager UI endpoints:

- Referrer-Policy
- Cross-Origin-Embedder-Policy
- Cross-Origin-Opener-Policy
- Cross-Origin-Resource-Policy

### **Streams Messaging Manager uses trusted proxy authentication when connecting to Schema Registry**

You can only interact with schemas through SMM if the necessary Ranger policies are set up for Schema Registry. For SMM UI, you must have the correct permissions to check messages deserialized with Avro on Data Explorer.

### **Streams Replication Manager**


#### **The --to option in srm-control now creates the file if it does not exist**

From now on, srm-control creates the file specified with the --to option if the file does not exist.

### **Cruise Control**

#### **Cruise Control is added to Streams Messaging Manager UI**

A new page is added to Streams Messaging Manager to monitor the Kafka cluster state and rebalancing process with Cruise Control. The Cruise Control User Interface (UI) enables you to review and configure the rebalancing of Kafka clusters through dashboards and a rebalancing wizard. The available goals and anomaly detectors are based on the Cloudera Manager configurations of Cruise Control. You can access Cruise Control from Streams Messaging Manager

using the  on the navigation sidebar.

For more information about Cruise Control in Streams Messaging Manager, see [Monitoring and managing Kafka cluster rebalancing](#).

## **What's new in Cloudera Streaming Analytics**

Learn about the new Cloudera Streaming Analytics features in Cloudera DataFlow for Cloudera Data Hub 7.3.1.

The following new features are introduced in Cloudera Streaming Analytics 7.3.1:

#### **Rebase to Apache Flink 1.19.1**

Apache Flink 1.19.1 is supported in the Cloudera Streaming Analytics 7.3.1 cluster definition.

For more information on what is included in the Apache Flink 1.19.1 version, see the [Apache Flink 1.19.1 Release Announcement](#) and [Release Notes](#).

#### **Support for Python UDFs in Cloudera SQL Stream Builder**

This feature allows customers to start using Python for creating User-Defined Functions (UDFs). Cloudera recommends that customers start using Python UDFs for all new developments, and start migrating their JavaScript UDFs to Python to prepare for future upgrades, as Javascript UDFs will be removed in the future due to the deprecation of the Nashorn engine used in JDK 8 and 11.

For more information on using Python UDFs, see [Python UDFs](#). For more information on supported JDK versions, refer to the [Support Matrix](#).

### Global logging configuration for Cloudera SQL Stream Builder jobs

A new global settings view has been enabled, which currently includes log4j configuration of Flink jobs started on SSB. Users with SSB administrator rights can set a default logging configuration applied to all SSB jobs, which can be overridden at the job level.

For more information see [Adjusting logging configuration in Advanced Settings](#).

### Customizable default Kafka TrustStore configuration in Streaming SQL Console

Customizing default Kafka TrustStore configurations was added to Streaming SQL Console. Kafka TrustStore can be configured during adding Kafka as a Data Source on the UI.

## Component support in Cloudera DataFlow for Data Hub 7.3.1

Cloudera DataFlow for Data Hub 7.3.1 includes the following components.

### Flow Management clusters

Cloudera 7.3.1.0 and 7.3.1.400 are platform-level releases defining the base environment used to create and run different Data Hub cluster types including Flow Management. Flow Management Data Hub clusters are compatible with both NiFi 1 and NiFi 2. I

For the Flow Management cluster type, Cloudera provides specific NiFi-based component bundles through Cloudera Flow Management versions. These are aligned with the platform version and include a specific NiFi version used in the cluster:

#### Flow Management clusters with NiFi 1

##### Platform version: 7.3.1.400

Cloudera Flow Management version: 2.2.9.400

Components:

- Apache NiFi 1.28.1.2.2.9.400
- Apache NiFi Registry 1.28.1.2.2.9.400
- Schema Registry 0.10.0.7.3.1.400

##### Platform version: 7.3.1.0

Cloudera Flow Management version: 2.2.9.0

Components:

- Apache NiFi 1.28.1.2.2.9.0
- Apache NiFi Registry 1.28.1.2.2.9.0
- Schema Registry 0.10.0.7.3.1.0

#### Flow Management clusters with NiFi 2

##### Platform version: 7.3.1.400

Cloudera Flow Management version: 4.2.1.400

Components:

- Apache NiFi 2.3.0.4.2.1.400
- Apache NiFi Registry 2.3.0.4.2.1.400
- Schema Registry 0.10.0.7.3.1.0

**Platform version: 7.3.1.0**

Cloudera Flow Management version: 4.2.1.0

Components:

- Apache NiFi 2.0.0.4.2.1.0
- Apache NiFi Registry 2.0.0.4.2.1.0
- Schema Registry 0.10.0.7.3.1.0

**Edge Management clusters [Technical Preview]**

- Edge Flow Manager 2.2.99

**Streams Messaging clusters**

- Apache Kafka 3.4.0
- Schema Registry 0.10.0
- Streams Messaging Manager 2.3.0
- Streams Replication Manager 1.1.0
- Cruise Control 2.5.85

**Streaming Analytics clusters**

- Apache Flink 1.20.1

## Supported NiFi extensions in Cloudera DataFlow for Data Hub 7.3.1

Flow Management clusters using Apache NiFi 1 or NiFi 2 include a broad set of NiFi extensions, most of which are fully supported by Cloudera. To ensure stability and receive full support, avoid using unsupported extensions in production environments.

The following sections provide detailed information on the supported extensions available for both NiFi versions across the corresponding Cloudera Flow Management versions used in Data Hub clusters.

### Supported NiFi extensions in Cloudera DataFlow for Data Hub 7.3.1.400

Apache NiFi versions 1.28.1 and 2.3.0 used in Flow Management clusters offer a comprehensive set of extensions, with the majority fully supported by Cloudera.

The following sections provide detailed information on the supported extensions available for both NiFi versions. To ensure seamless operation and reliable support, it is recommended to avoid using unsupported extensions in production environments.

### Supported NiFi processors

Learn about the processors supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.400.

To ensure optimal performance and reliable support, it is crucial to use only supported processors and avoid deploying unsupported ones in production environments.

Additional processors are developed and tested by the community but are not officially supported by Cloudera. Processors may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.400

AttributesToCSV	GetElasticsearch	PutDropbox
AttributesToJSON	GetFile	PutDynamoDB
Base64EncodeContent	GetFTP	PutDynamoDBRecord
CalculateParquetOffsets	GetGcpVisionAnnotateFilesOperationStatus	PutElasticsearchHttp1
CalculateParquetRowGroupOffsets	GetGcpVisionAnnotateImagesOperationStatus	PutElasticsearchHttpRecord1
CalculateRecordStats	GetHBase	PutElasticsearchJson
CaptureChangeDebeziumDB2 [Technical Preview]	GetHDFS	PutElasticsearchRecord1
CaptureChangeDebeziumMySQL [Technical Preview]	GetHDFSFileInfo	PutEmail
CaptureChangeDebeziumOracle [Technical Preview]	GetHDFSSequenceFile	PutFile
CaptureChangeDebeziumPostgreSQL [Technical Preview]	GetHTMLElement	PutFTP1
CaptureChangeDebeziumSQLServer [Technical Preview]	GetHTTP	PutGCSObject
CaptureChangeMySQL	GetHubSpot	PutGoogleDrive
CompressContent1, 2	GetIgniteCache	PutGridFS
ConnectWebSocket	GetJiraIssue	PutHBaseCell
ConsumeAMQP	GetJMSQueue	PutHBaseJSON
ConsumeAzureEventHub	GetJMSTopic	PutHBaseRecord1
ConsumeElasticsearch	GetMongoRecord	PutHDFS
ConsumeEWS	GetSFTP	PutHive3QL
ConsumeGCPubSub	GetShopify	PutHive3Streaming
ConsumeGCPubSubLite	GetSNMP	PutHiveQL
ConsumeJMS	GetSnowflakeIngestStatus	PutHiveStreaming
ConsumeKafka_1_0	GetSolr	PutHTMLElement
ConsumeKafka_2_0	GetSplunk	PutIceberg [Technical Preview]
ConsumeKafka_2_6	GetSQS	PutIcebergCDC
ConsumeKafka2CDP	GetTCP	PutInfluxDB
ConsumeKafka2RecordCDP	GetTwitter	PutJiraIssue
ConsumeKafkaRecord_1_0	GetWorkdayReport	PutJMS1
ConsumeKafkaRecord_2_0	GetZendesk	PutKinesisFirehose
ConsumeKafkaRecord_2_6	HandleHttpRequest	PutKinesisStream
ConsumeKinesisStream	HandleHttpResponse	PutKudu

ConsumeMQTT1	HashAttribute	PutLambda
ConsumeTwitter	HashContent	PutMongoRecord
ConsumeWindowsEventLog	IdentifyMimeType	PutORC1
ControlRate	InvokeAWSGatewayApi	PutParquet
ConvertAvroSchema	InvokeGRPC	PutRecord
ConvertAvroToJSON	InvokeGRPC	PutRedisHashRecord [Technical Preview]
ConvertAvroToORC	InvokeHTTP	PutRiemann
ConvertAvroToParquet	InvokeScriptedProcessor	PutS3Object
ConvertCharacterSet	JoinEnrichment	PutSalesforceObject
ConvertCSVToAvro	JoltTransformJSON	PutSFTP
ConvertJSONToAvro	JoltTransformRecord	PutSmbFile
ConvertJSONToSQL	JSLTTransformJSON	PutSnowflakeInternalStage
ConvertProtobuf	JsonQueryElasticsearch	PutSNS
ConvertRecord	ListAzureBlobStorage	PutSolrContentStream
CountText [2.1.7 SP2+ only]	ListAzureBlobStorage_v12	PutSolrRecord
CreateHadoopSequenceFile	ListAzureDataLakeStorage	PutSplunk
CryptographicHashAttribute	ListBoxFile	PutSplunkHTTP
CryptographicHashContent	ListCDPObjectStore	PutSQL
DecryptContent	ListDatabaseTables	PutSQS1
DecryptContentAge	ListDropbox	PutSyslog
DecryptContentCompatibility	ListenBeats	PutTCP
DecryptContentPGP	ListenFTP	PutUDP
DeduplicateRecord	ListenGRPC*	PutWebSocket
DeleteAzureBlobStorage	ListenGRPC*	PutZendeskTicket
DeleteAzureBlobStorage_v12	ListenHTTP	QueryAirtableTable
DeleteAzureDataLakeStorage	ListenNetFlow	QueryCassandra
DeleteByQueryElasticsearch	ListenOTLP	QueryDatabaseTable1
DeleteCDPObjectStore	ListenRELP	QueryDatabaseTableRecord
DeleteDynamoDB	ListenSyslog	QueryElasticsearchHttp
DeleteGCXObject	ListenTCP	QueryRecord
DeleteGridFS	ListenTCPRecord	QuerySalesforceObject
DeleteHBaseCells	ListenTrapSNMP	QuerySolr
DeleteHBaseRow	ListenUDP	QuerySplunkIndexingStatus
DeleteHDFS	ListenUDPRecord	QueryWhois
DeleteS3Object	ListenWebSocket	RemoveRecordField
DeleteSQS	ListFile	ReplaceText
DetectDuplicate	ListFTP	ReplaceTextWithMapping
DistributeLoad	ListGCSBucket	ResizeImage1
DuplicateFlowFile	ListGoogleDrive	RetryFlowFile
EncodeContent	ListHDFS	RouteHL7

EncryptContent2	ListS3	RouteOnAttribute
EncryptContentAge	ListSFTP	RouteOnContent
EncryptContentPGP	ListSmb	RouteText
EnforceOrder	LogAttribute	SampleRecord
EvaluateJsonPath	LogMessage	ScanAccumulo
EvaluateXPath	LookupAttribute	ScanAttribute1
EvaluateXQuery	LookupRecord	ScanContent
ExecuteGroovyScript	MergeContent	ScanHBase
ExecuteInfluxDBQuery	MergeRecord1	ScriptedFilterRecord
ExecuteProcess	ModifyCompression	ScriptedPartitionRecord
ExecuteScript	ModifyHTMLElement	ScriptedTransformRecord
ExecuteSQL	MonitorActivity	ScriptedValidateRecord
ExecuteSQLRecord	MoveAzureDataLakeStorage	ScrollElasticsearchHttp
ExecuteStateless1,2	MoveHDFS	SearchElasticsearch
ExecuteStreamCommand	Notify	SegmentContent
ExtractAvroMetadata	PackageFlowFile	SelectClouderaHiveQL
ExtractGrok	PaginatedJsonQueryElasticsearch	SelectHive3QL1
ExtractHL7Attributes	ParseCEF1	SelectHiveQL
ExtractImageMetadata	ParseEvtx	SendTrapSNMP
ExtractRecordSchema	ParseSyslog	SetSNMP
ExtractText	PartitionRecord	SignContentPGP
FetchAzureBlobStorage	PostHTTP	SplitAvro
FetchAzureBlobStorage_v12	PublishAMQP	SplitContent
FetchAzureDataLakeStorage	PublishGCPubSub1	SplitJson1
FetchBoxFile	PublishGCPubSubLite1	SplitRecord1
FetchCDPObjectStore	PublishJMS1	SplitText1
FetchDistributedMapCache	PublishKafka_1_0	SplitXml
FetchDropbox	PublishKafka_2_0	StartAwsPollyJob
FetchElasticsearchHttp	PublishKafka_2_6	StartAwsTextractJob
FetchFile	PublishKafka2CDP	StartAwsTranscribeJob
FetchFTP	PublishKafka2RecordCDP	StartAwsTranslateJob
FetchGCSObject	PublishKafkaRecord_1_0	StartGcpVisionAnnotateFilesOperation
FetchGoogleDrive	PublishKafkaRecord_2_0	StartGcpVisionAnnotateImagesOperation
FetchGridFS	PublishKafkaRecord_2_6	StartSnowflakeIngest
FetchHBaseRow	PublishMQTT	TagS3Object
FetchHDFS	PublishSlack	TailFile
FetchParquet	PutAccumuloRecord1	TransformXml
FetchS3Object	PutAzureBlobStorage	TriggerClouderaHiveMetaStoreEvent
FetchSFTP	PutAzureBlobStorage_v12	TriggerHiveMetaStoreEvent
FetchSmb	PutAzureCosmosDBRecord	UnpackContent



FilterAttribute	PutAzureDataLakeStorage1	UpdateAttribute
FlattenJson	PutAzureEventHub	UpdateByQueryElasticsearch
ForkEnrichment	PutAzureQueueStorage1	UpdateClouderaHiveTable
ForkRecord	PutAzureQueueStorage_v12	UpdateCounter
GenerateFlowFile	PutBigQuery	UpdateDatabaseTable
GenerateRecord	PutBigQueryBatch	UpdateDeltaLakeTable [Technical Preview]
GenerateTableFetch	PutBigQueryStreaming	UpdateHive3Table
GeoEnrichIP	PutBoxFile	UpdateHiveTable
GeoEnrichIPRecord	PutCassandraQL1	UpdateRecord
GeohashRecord	PutCassandraRecord1	ValidateCsv
GetAsanaObject	PutCDPObjectStore	ValidateJson
GetAwsPollyJobStatus	PutClouderaHiveQL	ValidateRecord
GetAwsTextractJobStatus	PutClouderaHiveStreaming	ValidateXml
GetAwsTranscribeJobStatus	PutClouderaORC	VerifyContentMAC
GetAwsTranslateJobStatus	PutCloudWatchMetric	VerifyContentPGP
GetAzureEventHub	PutCouchbaseKey	Wait
GetAzureQueueStorage	PutDatabaseRecord1	YandexTranslate
GetAzureQueueStorage_v12	PutDistributedMapCache	
GetCouchbaseKey1		

## Footnotes

- 1 – indicates a memory-intensive processor
- 2 – indicates a CPU-intensive processor

## NiFi 2.3.0 in Cloudera Flow Management 4.2.1.400

AttributesToCSV	GetBoxGroupMembers
AttributesToJSON	GetCouchbaseKey1
CalculateParquetOffsets	GetElasticsearch
CalculateParquetRowGroupOffsets	GetFile
CalculateRecordStats	GetFTP
CaptureChangeDebeziumDB2 [Technical Preview]	GetGcpVisionAnnotateFilesOperationStatus
CaptureChangeDebeziumMongoDB [Technical Preview]	GetGcpVisionAnnotateImagesOperationStatus
CaptureChangeDebeziumMySQL [Technical Preview]	GetHBase
CaptureChangeDebeziumOracle	GetHDFS
CaptureChangeDebeziumPostgreSQL	GetHDFSFileInfo
CaptureChangeDebeziumSQLServer [Technical Preview]	GetHDFSSequenceFile
CaptureChangeMySQL	GetHubSpot
ChunkDocument	GetJiraIssue
CompressContent1, 2	GetMongoRecord
ConnectWebSocket	GetS3ObjectTags
ConsumeAMQP	GetSFTP

ConsumeAzureEventHub	GetShopify
ConsumeBoxEnterpriseEvents	GetSNMP
ConsumeBoxEvents	GetSnowflakeIngestStatus
ConsumeElasticsearch	GetSolr
ConsumeGCPubSub	GetSplunk
ConsumeGCPubSubLite	GetSQS
ConsumeJMS	GetWorkdayReport
ConsumeKafka_2_6	GetZendesk
ConsumeKafka2CDP	HandleHttpRequest
ConsumeKafka2RecordCDP	HandleHttpResponse
ConsumeKafkaRecord_2_6	IdentifyMimeType
ConsumeKinesisStream	InvokeAWSGatewayApi
ConsumeMQTT1	InvokeGRPC
ConsumePLC [Technial Preview]	InvokeHTTP
ConsumeSlack	InvokeScriptedProcessor
ConsumeTwitter	JoinEnrichment
ConsumeWindowsEventLog	JoltTransformJSON
ControlRate	JoltTransformRecord
ConvertAvroToJSON	JSLTTransformJSON
ConvertAvroToParquet	JsonQueryElasticsearch
ConvertCharacterSet	ListAzureBlobStorage_v12
ConvertJSONToSQL	ListAzureDataLakeStorage
ConvertProtobuf	ListBoxFile
ConvertRecord	ListBoxFileInfo
CopyAzureBlobStorage_v12	ListCDPObjectStore
CountText	ListDatabaseTables
CreateHadoopSequenceFile	ListDropbox
CryptographicHashContent	ListenBeats
DecryptContent	ListenFTP
DecryptContentAge	ListenGRPC
DecryptContentCompatibility	ListenHTTP
DecryptContentPGP	ListenNetFlow
DeduplicateRecord	ListenOTLP
DeleteAzureBlobStorage_v12	ListenRELP
DeleteAzureDataLakeStorage	ListenSlack
DeleteByQueryElasticsearch	ListenSyslog
DeleteCDPObjectStore	ListenTCP
DeleteDynamoDB	ListenTCPRecord
DeleteGCXObject	ListenTrapSNMP
DeleteGridFS	ListenUDP

DeleteHBaseCells	ListenUDPRecord
DeleteHBaseRow	ListenWebSocket
DeleteHDFS	ListFile
DeleteS3Object	ListFTP
DeleteSQS	ListGCSBucket
DetectDuplicate	ListGoogleDrive
DistributeLoad	ListHDFS
DuplicateFlowFile	ListS3
EncodeContent	ListSFTP
EncryptContentAge	ListSmb
EncryptContentPGP	LogAttribute
EnforceOrder	LogMessage
EvaluateJsonPath	LookupAttribute
EvaluateXPath	LookupRecord
EvaluateXQuery	MergeContent
ExecuteGroovyScript	MergeRecord1
ExecuteProcess	ModifyCompression
ExecuteScript	MonitorActivity
ExecuteSQL	MoveAzureDataLakeStorage
ExecuteSQLRecord	MoveHDFS
ExecuteStateless1, 2	Notify
ExecuteStreamCommand	PackageFlowFile
ExtractAvroMetadata	PaginatedJsonQueryElasticsearch
ExtractGrok	ParseCEF1
ExtractHL7Attributes	ParseDocument
ExtractImageMetadata	ParseEvtx
ExtractRecordSchema	ParseSyslog
ExtractText	PartitionRecord
FetchAzureBlobStorage_v12	PromptAzureOpenAI
FetchAzureDataLakeStorage	PromptBedrock
FetchBoxFile	PromptChatGPT
FetchBoxFileInfo	PromptClaude
FetchBoxFileRepresentation	PromptOpenAI
FetchCDPObjectStore	PublishAMQP
FetchDistributedMapCache	PublishGCPubSub1
FetchDropbox	PublishGCPubSubLite1
FetchFile	PublishJMS1
FetchFTP	PublishKafka_2_6
FetchGCSObject	PublishKafka2CDP
FetchGoogleDrive	PublishKafka2RecordCDP

FetchGridFS	PublishKafkaRecord_2_6
FetchHBaseRow	PublishMQTT
FetchHDFS	PublishSlack
FetchParquet	PutAccumuloRecord1
FetchPLC [Technial Preview]	PutAzureBlobStorage_v12
FetchS3Object	PutAzureCosmosDBRecord
FetchSFTP	PutAzureDataLakeStorage1
FetchSmb	PutAzureEventHub
FilterAttribute	PutAzureQueueStorage_v12
FlattenJson	PutBigQuery
ForkEnrichment	PutBoxFile
ForkRecord	PutCassandraQL1
GenerateFlowFile	PutCassandraRecord1
GenerateRecord	PutCDPObjectStore
GenerateTableFetch	PutChroma
GeoEnrichIP	PutClouderaHiveQL
GeoEnrichIPRecord	PutClouderaHiveStreaming
GeohashRecord	PutClouderaORC
GetAsanaObject	PutCloudWatchMetric
GetAwsPollyJobStatus	PutCouchbaseKey
GetAwsTextractJobStatus	PutDatabaseRecord1
GetAwsTranscribeJobStatus	PutDistributedMapCache
GetAwsTranslateJobStatus	PutDropbox
GetAzureEventHub	PutDynamoDB
GetAzureQueueStorage_v12	PutDynamoDBRecord
GetBoxFileCollaborators	

#### Footnotes

- 1 – indicates a memory-intensive processor
- 2 – indicates a CPU-intensive processor

## Supported NiFi controller services

Learn about the controller services supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.400.

To ensure optimal performance and reliable support, it is crucial to use only supported controller services and avoid deploying unsupported ones in production environments.

Additional controller services are developed and tested by the community but are not officially supported by Cloudera. Controller services may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

**NiFi 1.28.1 in Cloudera Flow Management 2.2.9.400**

AccumuloService	ElasticSearchClientServiceImpl	PrometheusRecordSink
ActionHandlerLookup	ElasticSearchLookupService	ProtobufReader
ActiveMQJMSConnectionFactoryProvider	ElasticSearchStringLookupService	RabbitMQJMSConnectionFactoryProvider
ADLSCredentialsControllerService	EmailRecordSink	ReaderLookup
ADLSCredentialsControllerServiceLookup	EmbeddedHazelcastCacheManager	RecordSetWriterLookup
ADLSIDBrokerCloudCredentialsProviderControllerService	EmbeddedHazelcastCacheManager	RecordSinkHandler
AlertHandler	ExpressionHandler	RecordSinkServiceLookup
AmazonGlueSchemaRegistry	ExternalHazelcastCacheManager	RedisConnectionPoolService
AvroReader	FreeFormTextRecordSetWriter	RedisDistributedMapCacheClientService
AvroRecordSetWriter	GCPCredentialsControllerService	RedshiftConnectionPool
AvroSchemaRegistry	GrokReader	RestLookupService
AWSCredentialsProviderControllerService	HadoopCatalogService	ScriptedActionHandler
AWSIDBrokerCloudCredentialsProviderControllerService	HadoopCatalogService	ScriptedLookupService
AzureBlobIDBrokerCloudCredentialsProviderControllerService	HBase1_1_2_ClientMapCacheClient	ScriptedReader
AzureCosmosDBClientService	HBase_1_1_2_ClientMapCacheService	ScriptedRecordSetWriter
AzureEventHubRecordSink	HBase_1_1_2_ClientService	ScriptedRecordSink
AzureServiceBusJMSConnectionFactoryProvider	HBase_1_1_2_ListLookupService	ScriptedRulesEngine
AzureStorageCredentialsControllerService	HBase_1_1_2_RecordLookupService	SimpleDatabaseLookupService
AzureStorageCredentialsControllerService_v12	HBase_2_ClientMapCacheService	SimpleKeyValueLookupService
AzureStorageCredentialsControllerServiceLookup	HBase_2_ClientService	SimpleRedisDistributedMapCacheClientService
AzureStorageCredentialsControllerServiceLookup_v12	HBase_2_RecordLookupService	SimpleScriptedLookupService
CassandraDistributedMapCache	Hive3ConnectionPool	SiteToSiteReportingRecordSink
CassandraSessionProvider	HiveCatalogService	SmbjClientProviderService
CdpCredentialsProviderControllerService	HiveConnectionPool	SnowflakeComputingConnectionPool
CdpOauth2AccessTokenProviderControllerService	HortonworksSchemaRegistry	StandardAsanaClientProviderService
CEFReader	ImpalaConnectionPool	StandardAzureCredentialsControllerService
CiscoEmblemSyslogMessageReader	IPFIXReader	StandardDropboxCredentialService
ClouderaHiveConnectionPool	IPLookupService	StandardFileResourceService
ClouderaSchemaRegistry	JASN1Reader	StandardHashiCorpVaultClientService
CMLLookupService	JiraRecordSink	StandardHttpContextMap
ConfluentSchemaRegistry	JMSConnectionFactoryProvider	StandardJsonSchemaRegistry [Technical Preview]
CouchbaseClusterService	IndiJmsConnectionFactoryProvider	StandardOauth2AccessTokenProvider
CouchbaseKeyValueLookupService	JsonConfigBasedBoxClientService	StandardPGPPrivateKeyService
CouchbaseMapCacheClient	JsonPathReader	StandardPGPPublicKeyService
CouchbaseRecordLookupService	JsonRecordSetWriter	StandardPrivateKeyService
CSVReader	JsonTreeReader	StandardProxyConfigurationService
CSVRecordLookupService	KafkaRecordSink_1_0	StandardRestrictedSSLContextService
CSVRecordSetWriter	KafkaRecordSink_2_0	StandardS3EncryptionService
DatabaseRecordLookupService	KafkaRecordSink_2_6	StandardSnowflakeIngestManagerProviderService

DatabaseRecordSink	KerberosKeytabUserService	StandardSSLContextService
DatabaseTableSchemaRegistry	KerberosPasswordUserService	StandardWebClientServiceProvider
DBCPCConnectionPool	KerberosTicketCacheUserService	Syslog5424Reader
DBCPCConnectionPoolLookup	KeytabCredentialsService	SyslogReader
DistributedMapCacheClientService	KuduLookupService	UDPEventRecordSink
DistributedMapCacheLookupService	LoggingRecordSink	VolatileSchemaCache
DistributedMapCacheServer	LogHandler	WindowsEventLogReader
DistributedSetCacheClientService	MongoDBControllerService	XMLReader
DistributedSetCacheServer	MongoDBLookupService	XMLRecordSetWriter
EasyRulesEngineProvider	ParquetReader	YamlTreeReader
EasyRulesEngineService	ParquetRecordSetWriter	ZendeskRecordSink
EBCDICRecordReader [Technical Preview]	PostgreSQLConnectionPool	

### NiFi 2.3.0 in Cloudera Flow Management 4.2.1.400

AccumuloService	IPLookupService
ActiveMQJMSConnectionFactoryProvider	JASN1Reader
ADLSCredentialsControllerService	JiraRecordSink
ADLSCredentialsControllerServiceLookup	JMSConnectionFactoryProvider
ADLSIDBrokerCloudCredentialsProviderControllerService	IndiJmsConnectionFactoryProvider
AmazonGlueSchemaRegistry	JsonConfigBasedBoxClientService
ApicurioSchemaRegistry	JsonPathReader
AvroReader	JsonRecordSetWriter
AvroRecordSetWriter	JsonTreeReader
AvroSchemaRegistry	KafkaRecordSink_2_6
AWSCredentialsProviderControllerService	KerberosKeytabUserService
AWSIDBrokerCloudCredentialsProviderControllerService	KerberosPasswordUserService
AzureBlobIDBrokerCloudCredentialsProviderControllerService	KerberosTicketCacheUserService
AzureCosmosDBClientService	KuduLookupService
AzureEventHubRecordSink	LoggingRecordSink
AzureServiceBusJMSConnectionFactoryProvider	MongoDBControllerService
AzureStorageCredentialsControllerService_v12	MongoDBLookupService
AzureStorageCredentialsControllerServiceLookup_v12	ParquetReader
CassandraDistributedMapCache	ParquetRecordSetWriter
CassandraSessionProvider	PEMEncodedSSLContextProvider
CdpCredentialsProviderControllerService	PhoenixThickConnectionPool
CdpOAuth2AccessTokenProviderControllerService	PhoenixThinConnectionPool
CEFReader	PostgreSQLConnectionPool
CiscoEmblemSyslogMessageReader	PrometheusRecordSink
ClouderaAttributeSchemaReferenceReader	ProxyPLC4XConnectionPool [Technical Preview]
ClouderaAttributeSchemaReferenceWriter	RabbitMQJMSConnectionFactoryProvider

ClouderaEncodedSchemaReferenceReader	ReaderLookup
ClouderaEncodedSchemaReferenceWriter	RecordSetWriterLookup
ClouderaHiveConnectionPool	RecordSinkServiceLookup
ClouderaSchemaRegistry	RedisConnectionPoolService
CMLLookupService	RedisDistributedMapCacheClientService
ConfluentEncodedSchemaReferenceReader	RedshiftConnectionPool
ConfluentEncodedSchemaReferenceWriter	RESTCatalogService
ConfluentSchemaRegistry	RestLookupService
CouchbaseClusterService	ScriptedLookupService
CouchbaseKeyValueLookupService	ScriptedReader
CouchbaseMapCacheClient	ScriptedRecordSetWriter
CouchbaseRecordLookupService	ScriptedRecordSink
CSVReader	SimpleDatabaseLookupService
CSVRecordLookupService	SimpleKeyValueLookupService
CSVRecordSetWriter	SimpleRedisDistributedMapCacheClientService
DatabaseRecordLookupService	SimpleScriptedLookupService
DatabaseRecordSink	SiteToSiteReportingRecordSink
DatabaseTableSchemaRegistry	SlackRecordSink
DeveloperBoxClientService	SmbjClientProviderService
DBCPCConnectionPool	SnowflakeComputingConnectionPool
DBCPCConnectionPoolLookup	StandardAsanaClientProviderService
DistributedMapCacheClientService	StandardAzureCredentialsControllerService
DistributedMapCacheLookupService	StandardDatabaseDialectService
DistributedMapCacheServer	StandardDropboxCredentialService
DistributedSetCacheClientService	StandardFileResourceService
DistributedSetCacheServer	StandardHashiCorpVaultClientService
EBCDICRecordReader [Technical Preview]	StandardHttpContextMap
ElasticSearchClientServiceImpl	StandardJsonSchemaRegistry [Technical Preview]
ElasticSearchLookupService	StandardOAuth2AccessTokenProvider
ElasticSearchStringLookupService	StandardPGPPrivateKeyService
EmailRecordSink	StandardPGPPublicKeyService
EmbeddedHazelcastCacheManager	StandardPLC4XConnectionPool [Technical Preview]
ExcelReader	StandardPrivateKeyService
ExternalHazelcastCacheManager	StandardProxyConfigurationService
FreeFormTextRecordSetWriter	StandardRestrictedSSLContextService
GCPCredentialsControllerService	StandardS3EncryptionService
GCSFileResourceService	StandardSnowflakeIngestManagerProviderService
GenericPLC4XConnectionPool [Technical Preview]	StandardSSLContextService
GrokReader	StandardWebClientServiceProvider
HadoopCatalogService	Syslog5424Reader

HadoopDBCPCConnectionPool	SyslogReader
HazelcastMapCacheClient	UDPEventRecordSink
HBase_2_ClientMapCacheService	VolatileSchemaCache
HBase_2_ClientService	WindowsEventLogReader
HBase_2_RecordLookupService	XMLReader
Hive3ConnectionPool	XMLRecordSetWriter
HiveCatalogService	YamlTreeReader
ImpalaConnectionPool	ZendeskRecordSink
IPFIXReader	

## Supported NiFi reporting tasks

Learn about the reporting tasks supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.400.

To ensure optimal performance and reliable support, it is crucial to use only supported reporting tasks and avoid deploying unsupported ones in production environments.

Additional reporting tasks are developed and tested by the community but are not officially supported by Cloudera. Reporting tasks may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.400

- AmbariReportingTask
- ControllerStatusReportingTask
- MetricsEventReportingTask
- MonitorDiskUsage
- MonitorMemory
- PrometheusReportingTask
- QueryNiFiReportingTask
- ReportLineageToAtlas
- ScriptedReportingTask
- SiteToSiteBulletinReportingTask
- SiteToSiteMetricsReportingTask
- SiteToSiteProvenanceReportingTask
- SiteToSiteStatusReportingTask

### NiFi 2.3.0 in Cloudera Flow Management 4.2.1.400

- ControllerStatusReportingTask
- MonitorDiskUsage
- MonitorMemory
- PrometheusReportingTask
- QueryNiFiReportingTask
- ReportLineageToAtlas
- ScriptedReportingTask
- SiteToSiteBulletinReportingTask
- SiteToSiteMetricsReportingTask
- SiteToSiteProvenanceReportingTask



- SiteToSiteStatusReportingTask

## Supported NiFi parameter providers

Learn about the parameter providers supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.400.

To ensure optimal performance and reliable support, it is crucial to use only supported parameter providers and avoid deploying unsupported ones in production environments.

Additional parameter providers are developed and tested by the community but are not officially supported by Cloudera. Parameter providers may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.400

- AwsSecretsManagerParameterProvider
- AzureKeyVaultSecretsParameterProvider
- CyberArkConjurParameterProvider
- DatabaseParameterProvider
- EnvironmentVariableParameterProvider
- FileParameterProvider
- GcpSecretManagerParameterProvider
- HashiCorpVaultParameterProvider

### NiFi 2.3.0 in Cloudera Flow Management 4.2.1.400

- AwsSecretsManagerParameterProvider
- AzureKeyVaultSecretsParameterProvider
- CyberArkConjurParameterProvider
- DatabaseParameterProvider
- EnvironmentVariableParameterProvider
- FileParameterProvider
- GcpSecretManagerParameterProvider
- HashiCorpVaultParameterProvider
- OnePasswordParameterProvider
- PropertiesFileParameterProvider

## Supported NiFi flow analysis rules

Review the list of flow analysis rules supported in Cloudera Flow Management 4.2.1.400 included in Cloudera DataFlow for Data Hub 7.3.1.400.

Apache NiFi 2 has introduced flow analysis rules, a feature that enhances flow validation and management by evaluating components or parts of a flow and flagging rule violations to help you optimize and maintain flow design best practices.

Flow Analysis Rules can be configured as either:

- Recommendations – informational only; violations are logged but do not impact flow execution.
- Policies – enforceable; violations must be resolved before the affected component can be used.

The following rules are supported:

- DisallowComponentType
- DisallowConsecutiveConnectionsWithRoundRobinLB

- DisallowDeadEnd
- DisallowDeprecatedProcessor
- DisallowExtractTextForFullContent
- RecommendRecordProcessor
- RequireHandleHttpResponseAfterHandleHttpRequest
- RequireMergeBeforePutIceberg
- RestrictBackpressureSettings
- RestrictComponentNaming
- RestrictConcurrentTasksVsThreadPoolSizeInProcessors
- RestrictFlowFileExpiration
- RestrictProcessorConcurrency
- RestrictSchedulingForListProcessors
- RestrictThreadPoolSize
- RestrictYieldDurationForConsumeKafkaProcessors

## Supported NiFi flow registry clients

Learn about the Apache NiFi flow registry clients supported in Flow Management Data Hub clusters using NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.400.

NiFi Flow Registry clients are components in Apache NiFi that allow it to connect to external Flow Registries, services used to store and manage versioned dataflows.

- ClouderaDataFlowRegistryClient
- ClouderaFlowLibraryFlowRegistryClient
- GitHubFlowRegistryClient
- GitLabFlowRegistryClient
- NifiRegistryFlowRegistryClient

## Cloudera exclusive components

Learn about Cloudera-exclusive components supported in Flow Management Data Hub clusters using NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.400.

Cloudera offers a set of NiFi components available only to its customers. These components provide additional functionalities and are specifically designed to enhance the Cloudera NiFi experience.

### In Flow Management clusters with NiFi 2.3.0

#### Processors

- CaptureChangeDebeziumDB2
- CaptureChangeDebeziumMongoDB
- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- ConsumeKafka2CDP
- ConsumeKafka2RecordCDP
- ConsumePLC
- ConvertProtobuf
- DeleteCDPObjectStore
- FetchCDPObjectStore
- FetchPLC
- GetJiraIssue
- InvokeGRPC

- ListCDPObjectStore
- ListenGRPC
- ListenNetFlow
- PromptAzureOpenAI
- PromptBedrock
- PromptClaude
- PromptOpenAI
- PublishKafka2CDP
- PublishKafka2RecordCDP
- PutCDPObjectStore
- PutClouderaHiveQL
- PutClouderaHiveStreaming
- PutClouderaORC
- PutIcebergCDC
- PutJiraIssue
- PutPLC
- SawmillTransformJSON
- SawmillTransformRecord
- SelectClouderaHiveQL
- TokenCount
- TriggerClouderaHiveMetaStoreEvent
- UpdateClouderaHiveTable
- UpdateDeltaLakeTable

**Controller services**

- ActiveMQJMSConnectionFactoryProvider
- ADLSIDBrokerCloudCredentialsProviderControllerService
- AWSIDBrokerCloudCredentialsProviderControllerService
- AzureBlobIDBrokerCloudCredentialsProviderControllerService
- AzureServiceBusJMSConnectionFactoryProvider
- CdpCredentialsProviderControllerService
- CdpOAuth2AccessTokenProviderControllerService
- CiscoEmblemSyslogMessageReader
- ClouderaAttributeSchemaReferenceReader
- ClouderaAttributeSchemaReferenceWriter
- ClouderaEncodedSchemaReferenceReader
- ClouderaEncodedSchemaReferenceWriter
- ClouderaHiveConnectionPool
- ClouderaSchemaRegistry
- CMLLookupService
- EBCDICRecordReader
- GenericPLC4XConnectionPool
- ImpalaConnectionPool
- IPFIXReader
- JiraRecordSink
- PhoenixThickConnectionPool
- PhoenixThinConnectionPool
- PostgreSQLConnectionPool
- ProxyPLC4XConnectionPool
- RabbitMQJMSConnectionFactoryProvider

- RedshiftConnectionPool
- RESTCatalogService
- StandardPLC4XConnectionPool

#### Parameter providers

- CyberArkConjurParameterProvider
- PropertiesFileParameterProvider

#### Flow registry clients

- ClouderaDataFlowRegistryClient
- ClouderaFlowLibraryFlowRegistryClient

## Components supported by partners

Learn about the components built, maintained, and supported by Cloudera partners, and available in Flow Management Data Hub clusters in Cloudera DataFlow for Data Hub 7.3.1.400.

Although Cloudera's Quality Engineering teams have added test coverage for these components, they are not officially supported by Cloudera. For assistance, contact the respective partners directly.

#### NiFi 1.28.1 processors supported by partners

- ConsumePulsar (1.18.0)
- ConsumePulsarRecord (1.18.0)
- PublishPulsar (1.18.0)
- PublishPulsarRecord (1.18.0)

#### NiFi 1.28.1 controller services supported by partners

- PulsarClientAthenzAuthenticationService (1.18.0)
- PulsarClientJwtAuthenticationService (1.18.0)
- PulsarClientOAuthAuthenticationService (1.18.0)
- PulsarClientTlsAuthenticationService (1.18.0)
- StandardPulsarClientService (1.18.0)

These components can be used to push data into Apache Pulsar as well as getting data out of it. In case you have issues or questions while using these components, Cloudera recommends you to reach out to your StreamNative representative team.



**Note:** In Flow Management Data Hub clusters using NiFi 2, the Pulsar components are not included. You can manually download the components from a [Maven repository](#) and add them into your cluster.

## Supported NiFi extensions in Cloudera DataFlow for Data Hub 7.3.1.0

Apache NiFi versions 1.28.1 and 2.0.0 used in Flow Management clusters offer a comprehensive set of extensions, with the majority fully supported by Cloudera.

The following sections provide detailed information on the supported extensions available for both NiFi versions. To ensure seamless operation and reliable support, it is recommended to avoid using unsupported extensions in production environments.

### Supported NiFi processors

Learn about the processors supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.0.

To ensure optimal performance and reliable support, it is crucial to use only supported processors and avoid deploying unsupported ones in production environments.

Additional processors are developed and tested by the community but are not officially supported by Cloudera. Processors may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.0

AttributesToCSV	GetElasticsearch	PutDropbox
AttributesToJSON	GetFile	PutDynamoDB
Base64EncodeContent	GetFTP	PutDynamoDBRecord
CalculateParquetOffsets	GetGcpVisionAnnotateFilesOperationStatus	PutElasticsearchHttp1
CalculateParquetRowGroupOffsets	GetGcpVisionAnnotateImagesOperationStatus	PutElasticsearchHttpRecord1
CalculateRecordStats	GetHBase	PutElasticsearchJson
CaptureChangeDebeziumDB2 [Technical Preview]	GetHDFS	PutElasticsearchRecord1
CaptureChangeDebeziumMySQL [Technical Preview]	GetHDFSFileInfo	PutEmail
CaptureChangeDebeziumOracle [Technical Preview]	GetHDFSSequenceFile	PutFile
CaptureChangeDebeziumPostgreSQL [Technical Preview]	GetHTMLElement	PutFTP1
CaptureChangeDebeziumSQLServer [Technical Preview]	GetHTTP	PutGCSObject
CaptureChangeMySQL	GetHubSpot	PutGoogleDrive
CompressContent1, 2	GetIgniteCache	PutGridFS
ConnectWebSocket	GetJiraIssue	PutHBaseCell
ConsumeAMQP	GetJMSQueue	PutHBaseJSON
ConsumeAzureEventHub	GetJMSTopic	PutHBaseRecord1
ConsumeElasticsearch	GetMongoRecord	PutHDFS
ConsumeEWS	GetSFTP	PutHive3QL
ConsumeGCPubSub	GetShopify	PutHive3Streaming
ConsumeGCPubSubLite	GetSNMP	PutHiveQL
ConsumeJMS	GetSnowflakeIngestStatus	PutHiveStreaming
ConsumeKafka_1_0	GetSolr	PutHTMLElement
ConsumeKafka_2_0	GetSplunk	PutIceberg
ConsumeKafka_2_6	GetSQS	PutIcebergCDC [Technical Preview]
ConsumeKafka2CDP	GetTCP	PutInfluxDB
ConsumeKafka2RecordCDP	GetTwitter	PutJiraIssue
ConsumeKafkaRecord_1_0	GetWorkdayReport	PutJMS1
ConsumeKafkaRecord_2_0	GetZendesk	PutKinesisFirehose
ConsumeKafkaRecord_2_6	HandleHttpRequest	PutKinesisStream
ConsumeKinesisStream	HandleHttpResponse	PutKudu
ConsumeMQTT1	HashAttribute	PutLambda
ConsumeTwitter	HashContent	PutMongoRecord

ConsumeWindowsEventLog	IdentifyMimeType	PutORC1
ControlRate	InvokeAWSGatewayApi	PutParquet
ConvertAvroSchema	InvokeGRPC	PutRecord
ConvertAvroToJSON	InvokeGRPC	PutRedisHashRecord [Technical Preview]
ConvertAvroToORC	InvokeHTTP	PutRiemann
ConvertAvroToParquet	InvokeScriptedProcessor	PutS3Object
ConvertCharacterSet	JoinEnrichment	PutSalesforceObject
ConvertCSVToAvro	JoltTransformJSON	PutSFTP
ConvertJSONToAvro	JoltTransformRecord	PutSmbFile
ConvertJSONToSQL	JSLTTransformJSON	PutSnowflakeInternalStage
ConvertProtobuf	JsonQueryElasticsearch	PutSNS
ConvertRecord	ListAzureBlobStorage	PutSolrContentStream
CreateHadoopSequenceFile	ListAzureBlobStorage_v12	PutSolrRecord
CryptographicHashAttribute	ListAzureDataLakeStorage	PutSplunk
CryptographicHashContent	ListBoxFile	PutSplunkHTTP
DecryptContent	ListCDPObjectStore	PutSQL
DecryptContentAge	ListDatabaseTables	PutSQS1
DecryptContentCompatibility	ListDropbox	PutSyslog
DecryptContentPGP	ListenBeats	PutTCP
DeduplicateRecord	ListenFTP	PutUDP
DeleteAzureBlobStorage	ListenGRPC	PutWebSocket
DeleteAzureBlobStorage_v12	ListenGRPC	PutZendeskTicket
DeleteAzureDataLakeStorage	ListenHTTP	QueryAirtableTable
DeleteByQueryElasticsearch	ListenNetFlow	QueryCassandra
DeleteCDPObjectStore	ListenOTLP	QueryDatabaseTable1
DeleteDynamoDB	ListenRELP	QueryDatabaseTableRecord
DeleteGCXObject	ListenSyslog	QueryElasticsearchHttp
DeleteGridFS	ListenTCP	QueryRecord
DeleteHBaseCells	ListenTCPRecord	QuerySalesforceObject
DeleteHBaseRow	ListenTrapSNMP	QuerySolr
DeleteHDFS	ListenUDP	QuerySplunkIndexingStatus
DeleteS3Object	ListenUDPRecord	QueryWhois
DeleteSQS	ListenWebSocket	RemoveRecordField
DetectDuplicate	ListFile	ReplaceText
DistributeLoad	ListFTP	ReplaceTextWithMapping
DuplicateFlowFile	ListGCSBucket	ResizeImage1
EncodeContent	ListGoogleDrive	RetryFlowFile
EncryptContent2	ListHDFS	RouteHL7
EncryptContentAge	ListS3	RouteOnAttribute
EncryptContentPGP	ListSFTP	RouteOnContent

EnforceOrder	ListSmb	RouteText
EvaluateJsonPath	LogAttribute	SampleRecord
EvaluateXPath	LogMessage	ScanAccumulo
EvaluateXQuery	LookupAttribute	ScanAttribute1
ExecuteGroovyScript	LookupRecord	ScanContent
ExecuteInfluxDBQuery	MergeContent	ScanHBase
ExecuteProcess	MergeRecord1	ScriptedFilterRecord
ExecuteScript	ModifyCompression	ScriptedPartitionRecord
ExecuteSQL	ModifyHTML element	ScriptedTransformRecord
ExecuteSQLRecord	MonitorActivity	ScriptedValidateRecord
ExecuteStateless12	MoveAzureDataLakeStorage	ScrollElasticsearchHttp
ExecuteStreamCommand	MoveHDFS	SearchElasticsearch
ExtractAvroMetadata	Notify	SegmentContent
ExtractGrok	PackageFlowFile	SelectClouderaHiveQL
ExtractHL7Attributes	PaginatedJsonQueryElasticsearch	SelectHive3QL1
ExtractImageMetadata	ParseCEF1	SelectHiveQL
ExtractRecordSchema	ParseEvtx	SendTrapSNMP
ExtractText	ParseSyslog	SetSNMP
FetchAzureBlobStorage	PartitionRecord	SignContentPGP
FetchAzureBlobStorage_v12	PostHTTP	SplitAvro
FetchAzureDataLakeStorage	PublishAMQP	SplitContent
FetchBoxFile	PublishGCPubSub1	SplitJson1
FetchCDPObjectStore	PublishGCPubSubLite1	SplitRecord1
FetchDistributedMapCache	PublishJMS1	SplitText1
FetchDropbox	PublishKafka_1_0	SplitXml
FetchElasticsearchHttp	PublishKafka_2_0	StartAwsPollyJob
FetchFile	PublishKafka_2_6	StartAwsTextextractJob
FetchFTP	PublishKafka2CDP	StartAwsTranscribeJob
FetchGCSObject	PublishKafka2RecordCDP	StartAwsTranslateJob
FetchGoogleDrive	PublishKafkaRecord_1_0	StartGcpVisionAnnotateFilesOperation
FetchGridFS	PublishKafkaRecord_2_0	StartGcpVisionAnnotateImagesOperation
FetchHBaseRow	PublishKafkaRecord_2_6	StartSnowflakeIngest
FetchHDFS	PublishMQTT	TagS3Object
FetchParquet	PublishSlack	TailFile
FetchS3Object	PutAccumuloRecord1	TransformXml
FetchSFTP	PutAzureBlobStorage	TriggerClouderaHiveMetaStoreEvent
FetchSmb	PutAzureBlobStorage_v12	TriggerHiveMetaStoreEvent
FilterAttribute	PutAzureCosmosDBRecord	UnpackContent
FlattenJson	PutAzureDataLakeStorage1	UpdateAttribute
ForkEnrichment	PutAzureEventHub	UpdateByQueryElasticsearch

ForkRecord	PutAzureQueueStorage1	UpdateClouderaHiveTable
GenerateFlowFile	PutAzureQueueStorage_v12	UpdateCounter
GenerateRecord	PutBigQuery	UpdateDatabaseTable
GenerateTableFetch	PutBigQueryBatch	UpdateDeltaLakeTable [Technical Preview]
GeoEnrichIP	PutBigQueryStreaming	UpdateHive3Table
GeoEnrichIPRecord	PutBoxFile	UpdateHiveTable
GeohashRecord	PutCassandraQL1	UpdateRecord
GetAsanaObject	PutCassandraRecord1	ValidateCsv
GetAwsPollyJobStatus	PutCDPObjectStore	ValidateJson
GetAwsTextractJobStatus	PutClouderaHiveQL	ValidateRecord
GetAwsTranscribeJobStatus	PutClouderaHiveStreaming	ValidateXml
GetAwsTranslateJobStatus	PutClouderaORC	VerifyContentMAC
GetAzureEventHub	PutCloudWatchMetric	VerifyContentPGP
GetAzureQueueStorage	PutCouchbaseKey	Wait
GetAzureQueueStorage_v12	PutDatabaseRecord1	YandexTranslate
GetCouchbaseKey1	PutDistributedMapCache	

## Footnotes

- 1 – indicates a memory-intensive processor
- 2 – indicates a CPU-intensive processor

**NiFi 2.0.0 in Cloudera Flow Management 4.2.1.0**

AttributesToCSV	GetElasticsearch	PutFile
AttributesToJSON	GetFile	PutFTP1
CalculateParquetOffsets	GetFTP	PutGCSObject
CalculateParquetRowGroupOffsets	GetGcpVisionAnnotateFilesOperationStatus	PutGoogleDrive
CalculateRecordStats	GetGcpVisionAnnotateImagesOperationStatus	PutGridFS
CaptureChangeDebeziumDB2 [Technical Preview]	GetHBase	PutHBaseCell
CaptureChangeDebeziumMySQL [Technical Preview]	GetHDFS	PutHBaseJSON
CaptureChangeDebeziumOracle [Technical Preview]	GetHDFSFileInfo	PutHBaseRecord1
CaptureChangeDebeziumPostgreSQL [Technical Preview]	GetHDFSSequenceFile	PutHDFS
CaptureChangeDebeziumSQLServer [Technical Preview]	GetHubSpot	PutHive3QL
CaptureChangeMySQL	GetJiraIssue	PutHive3Streaming
ChunkDocument [Technical Preview]	GetMongoRecord	PutIceberg
CompressContent1, 2	GetSFTP	PutIcebergCDC [Technical Preview]
ConnectWebSocket	GetShopify	PutJiraIssue
ConsumeAMQP	GetSNMP	PutKinesisFirehose
ConsumeAzureEventHub	GetSnowflakeIngestStatus	PutKinesisStream



ConsumeElasticsearch	GetSolr	PutKudu
ConsumeGCPubSub	GetSplunk	PutLambda
ConsumeGCPubSubLite	GetSQS	PutMongoBulkOperations
ConsumeJMS	GetWorkdayReport	PutMongoRecord
ConsumeKafka_2_6	GetZendesk	PutORC1
ConsumeKafka2CDP	HandleHttpRequest	PutParquet
ConsumeKafka2RecordCDP	HandleHttpResponse	PutPinecone [Technical Preview]
ConsumeKafkaRecord_2_6	IdentifyMimeType	PutPLC [Technical Preview]
ConsumeKinesisStream	InvokeAWSGatewayApi	PutRecord
ConsumeMQTT1	InvokeGRPC	PutRedisHashRecord [Technical Preview]
ConsumePLC [Technical Preview]	InvokeHTTP	PutS3Object
ConsumeSlack	InvokeScriptedProcessor	PutSalesforceObject
ConsumeTwitter	JoinEnrichment	PutSFTP
ConsumeWindowsEventLog	JoltTransformJSON	PutSmbFile
ControlRate	JoltTransformRecord	PutSnowflakeInternalStage
ConvertAvroToJSON	JSLTTransformJSON	PutSNS
ConvertAvroToParquet	JsonQueryElasticsearch	PutSolrContentStream
ConvertCharacterSet	ListAzureBlobStorage_v12	PutSolrRecord
ConvertJSONToSQL	ListAzureDataLakeStorage	PutSplunk
ConvertProtobuf	ListBoxFile	PutSplunkHTTP
ConvertRecord	ListCDPObjectStore	PutSQL
CopyAzureBlobStorage_v12	ListDatabaseTables	PutSQS1
CreateHadoopSequenceFile	ListDropbox	PutSyslog
CryptographicHashContent	ListenBeats	PutTCP
DecryptContent	ListenFTP	PutUDP
DecryptContentAge	ListenGRPC	PutWebSocket
DecryptContentCompatibility	ListenHTTP	PutZendeskTicket
DecryptContentPGP	ListenNetFlow	QueryAirtableTable
DeduplicateRecord	ListenOTLP	QueryCassandra
DeleteAzureBlobStorage_v12	ListenRELP	QueryChroma [Technical Preview]
DeleteAzureDataLakeStorage	ListenSlack	QueryDatabaseTable1
DeleteByQueryElasticsearch	ListenSyslog	QueryDatabaseTableRecord
DeleteCDPObjectStore	ListenTCP	QueryPinecone [Technical Preview]
DeleteDynamoDB	ListenTCPRecord	QueryRecord
DeleteGCXObject	ListenTrapSNMP	QuerySalesforceObject
DeleteGridFS	ListenUDP	QuerySolr
DeleteHBaseCells	ListenUDPRecord	QuerySplunkIndexingStatus
DeleteHBaseRow	ListenWebSocket	QueryWhois
DeleteHDFS	ListFile	RemoveRecordField
DeleteS3Object	ListFTP	RenameRecordField

DeleteSQS	ListGCSBucket	ReplaceText
DetectDuplicate	ListGoogleDrive	ReplaceTextWithMapping
DistributeLoad	ListHDFS	ResizeImage1
DuplicateFlowFile	ListS3	RetryFlowFile
EncodeContent	ListSFTP	RouteHL7
EncryptContentAge	ListSmb	RouteOnAttribute
EncryptContentPGP	LogAttribute	RouteOnContent
EnforceOrder	LogMessage	RouteText
EvaluateJsonPath	LookupAttribute	SampleRecord
EvaluateXPath	LookupRecord	ScanAccumulo
EvaluateXQuery	MergeContent	ScanAttribute1
ExecuteGroovyScript	MergeRecord1	ScanContent
ExecuteProcess	ModifyCompression	ScanHBase
ExecuteScript	MonitorActivity	ScriptedFilterRecord
ExecuteSQL	MoveAzureDataLakeStorage	ScriptedPartitionRecord
ExecuteSQLRecord	MoveHDFS	ScriptedTransformRecord
ExecuteStateless1, 2	Notify	ScriptedValidateRecord
ExecuteStreamCommand	PackageFlowFile	SearchElasticsearch
ExtractAvroMetadata	PaginatedJsonQueryElasticsearch	SegmentContent
ExtractGrok	ParseCEF1	SelectClouderaHiveQL
ExtractHL7Attributes	ParseDocument [Technical Preview]	SelectHive3QL1
ExtractImageMetadata	ParseEvtx	SendTrapSNMP
ExtractRecordSchema	ParseSyslog	SetSNMP
ExtractText	PartitionRecord	SignContentPGP
FetchAzureBlobStorage_v12	PromptChatGPT [Technical Preview]	SplitAvro
FetchAzureDataLakeStorage	PublishAMQP	SplitContent
FetchBoxFile	PublishGCPubSub1	SplitJson1
FetchCDPObjectStore	PublishGCPubSubLite1	SplitRecord1
FetchDistributedMapCache	PublishJMS1	SplitText1
FetchDropbox	PublishKafka_2_6	SplitXml
FetchFile	PublishKafka2CDP	StartAwsPollyJob
FetchFTP	PublishKafka2RecordCDP	StartAwsTextractJob
FetchGCSObject	PublishKafkaRecord_2_6	StartAwsTranscribeJob
FetchGoogleDrive	PublishMQTT	StartAwsTranslateJob
FetchGridFS	PublishSlack	StartGcpVisionAnnotateFilesOperation
FetchHBaseRow	PutAccumuloRecord1	StartGcpVisionAnnotateImagesOperation
FetchHDFS	PutAzureBlobStorage_v12	StartSnowflakeIngest
FetchParquet	PutAzureCosmosDBRecord	TagS3Object
FetchPLC [Technical Preview]	PutAzureDataLakeStorage1	TailFile
FetchS3Object	PutAzureEventHub	TransformXml

FetchSFTP	PutAzureQueueStorage_v12	TriggerClouderaHiveMetaStoreEvent
FetchSmb	PutBigQuery	TriggerHiveMetaStoreEvent
FilterAttribute	PutBoxFile	UnpackContent
FlattenJson	PutCassandraQL1	UpdateAttribute
ForkEnrichment	PutCassandraRecord1	UpdateByQueryElasticsearch
ForkRecord	PutCDPObjectStore	UpdateClouderaHiveTable
GenerateFlowFile	PutChroma [Technical Preview]	UpdateCounter
GenerateRecord	PutClouderaHiveQL	UpdateDatabaseTable
GenerateTableFetch	PutClouderaHiveStreaming	UpdateDeltaLakeTable [Technical Preview]
GeoEnrichIP	PutClouderaORC	UpdateHive3Table
GeoEnrichIPRecord	PutCloudWatchMetric	UpdateRecord
GeohashRecord	PutCouchbaseKey	ValidateCsv
GetAsanaObject	PutDatabaseRecord1	ValidateJson
GetAwsPollyJobStatus	PutDistributedMapCache	ValidateRecord
GetAwsTextractJobStatus	PutDropbox	ValidateXml
GetAwsTranscribeJobStatus	PutDynamoDB	VerifyContentMAC
GetAwsTranslateJobStatus	PutDynamoDBRecord	VerifyContentPGP
GetAzureEventHub	PutElasticsearchJson	Wait
GetAzureQueueStorage_v12	PutElasticsearchRecord1	YandexTranslate
GetCouchbaseKey1	PutEmail	

#### Footnotes

- 1 – indicates a memory-intensive processor
- 2 – indicates a CPU-intensive processor

## Supported NiFi controller services

Learn about the controller services supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.0.

To ensure optimal performance and reliable support, it is crucial to use only supported controller services and avoid deploying unsupported ones in production environments.

Additional controller services are developed and tested by the community but are not officially supported by Cloudera. Controller services may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.0

AccumuloService	ImpalaConnectionPool
ActionHandlerLookup	IPFIXReader
ActiveMQJMSConnectionFactoryProvider	IPLookupService
ADLSCredentialsControllerService	JASNIReader
ADLSCredentialsControllerServiceLookup	JiraRecordSink
ADLSIDBrokerCloudCredentialsProviderControllerService	JMSConnectionFactoryProvider

AlertHandler	JndiJmsConnectionFactoryProvider
AmazonGlueSchemaRegistry	JsonConfigBasedBoxClientService
AvroReader	JsonPathReader
AvroRecordSetWriter	JsonRecordSetWriter
AvroSchemaRegistry	JsonTreeReader
AWSCredentialsProviderControllerService	KafkaRecordSink_1_0
AWSIDBrokerCloudCredentialsProviderControllerService	KafkaRecordSink_2_0
AzureBlobIDBrokerCloudCredentialsProviderControllerService	KafkaRecordSink_2_6
AzureCosmosDBClientService	KerberosKeytabUserService
AzureEventHubRecordSink	KerberosPasswordUserService
AzureServiceBusJMSConnectionFactoryProvider	KerberosTicketCacheUserService
AzureStorageCredentialsControllerService	KeytabCredentialsService
AzureStorageCredentialsControllerService_v12	KuduLookupService
AzureStorageCredentialsControllerServiceLookup	LoggingRecordSink
AzureStorageCredentialsControllerServiceLookup_v12	LogHandler
CassandraDistributedMapCache	MongoDBControllerService
CassandraSessionProvider	MongoDBLookupService
CdpCredentialsProviderControllerService	ParquetReader
CdpOAuth2AccessTokenProviderControllerService	ParquetRecordSetWriter
CEFReader	PostgreSQLConnectionPool
CiscoEmblemSyslogMessageReader	PrometheusRecordSink
ClouderaHiveConnectionPool	ProtobufReader
ClouderaSchemaRegistry	RabbitMQJMSConnectionFactoryProvider
CMLLookupService	ReaderLookup
ConfluentSchemaRegistry	RecordSetWriterLookup
CouchbaseClusterService	RecordSinkHandler
CouchbaseKeyValueLookupService	RecordSinkServiceLookup
CouchbaseMapCacheClient	RedisConnectionPoolService
CouchbaseRecordLookupService	RedisDistributedMapCacheClientService
CSVReader	RedshiftConnectionPool
CSVRecordLookupService	RestLookupService
CSVRecordSetWriter	ScriptedActionHandler
DatabaseRecordLookupService	ScriptedLookupService
DatabaseRecordSink	ScriptedReader
DatabaseTableSchemaRegistry	ScriptedRecordSetWriter
DBCPCConnectionPool	ScriptedRecordSink
DBCPCConnectionPoolLookup	ScriptedRulesEngine
DistributedMapCacheClientService	SimpleDatabaseLookupService
DistributedMapCacheLookupService	SimpleKeyValueLookupService
DistributedMapCacheServer	SimpleRedisDistributedMapCacheClientService

DistributedSetCacheClientService	SimpleScriptedLookupService
DistributedSetCacheServer	SiteToSiteReportingRecordSink
EasyRulesEngineProvider	SmbjClientProviderService
EasyRulesEngineService	SnowflakeComputingConnectionPool
EBCDICRecordReader [Technical Preview]	StandardAsanaClientProviderService
ElasticSearchClientServiceImpl	StandardAzureCredentialsControllerService
ElasticSearchLookupService	StandardDropboxCredentialService
ElasticSearchStringLookupService	StandardFileResourceService
EmailRecordSink	StandardHashiCorpVaultClientService
EmbeddedHazelcastCacheManager	StandardHttpContextMap
ExcelReader	StandardJsonSchemaRegistry [Technical Preview]
ExpressionHandler	StandardOAuth2AccessTokenProvider
ExternalHazelcastCacheManager	StandardPGPPrivateKeyService
FreeFormTextRecordSetWriter	StandardPGPPublicKeyService
GCPCredentialsControllerService	StandardPrivateKeyService
GrokReader	StandardProxyConfigurationService
HadoopCatalogService	StandardRestrictedSSLContextService
HadoopDBCPCConnectionPool	StandardS3EncryptionService
HazelcastMapCacheClient	StandardSnowflakeIngestManagerProviderService
HBase_1_1_2_ClientMapCacheService	StandardSSLContextService
HBase_1_1_2_ClientService	StandardWebClientServiceProvider
HBase_1_1_2_ListLookupService	Syslog5424Reader
HBase_1_1_2_RecordLookupService	SyslogReader
HBase_2_ClientMapCacheService	UDPEventRecordSink
HBase_2_ClientService	VolatileSchemaCache
HBase_2_RecordLookupService	WindowsEventLogReader
Hive3ConnectionPool	XMLReader
HiveCatalogService	XMLRecordSetWriter
HiveConnectionPool	YamlTreeReader
HortonworksSchemaRegistry	ZendeskRecordSink

### NiFi 2.0.0 in Cloudera Flow Management 4.2.1.0

AccumuloService	IPLookupService
ActiveMQJMSConnectionFactoryProvider	JASN1Reader
ADLSCredentialsControllerService	JiraRecordSink
ADLSCredentialsControllerServiceLookup	JMSConnectionFactoryProvider
ADLSIDBrokerCloudCredentialsProviderControllerService	JndiJmsConnectionFactoryProvider
AmazonGlueSchemaRegistry	JsonConfigBasedBoxClientService
ApicurioSchemaRegistry	JsonPathReader
AvroReader	JsonRecordSetWriter

AvroRecordSetWriter	JsonTreeReader
AvroSchemaRegistry	KafkaRecordSink_2_6
AWSCredentialsProviderControllerService	KerberosKeytabUserService
AWSIDBrokerCloudCredentialsProviderControllerService	KerberosPasswordUserService
AzureBlobIDBrokerCloudCredentialsProviderControllerService	KerberosTicketCacheUserService
AzureCosmosDBClientService	KuduLookupService
AzureEventHubRecordSink	LoggingRecordSink
AzureServiceBusJMSConnectionFactoryProvider	MongoDBControllerService
AzureStorageCredentialsControllerService_v12	MongoDBLookupService
AzureStorageCredentialsControllerServiceLookup_v12	ParquetReader
CassandraDistributedMapCache	ParquetRecordSetWriter
CassandraSessionProvider	PostgreSQLConnectionPool
CdpCredentialsProviderControllerService	PrometheusRecordSink
CdpOAuth2AccessTokenProviderControllerService	ProxyPLC4XConnectionPool [Technical Preview]
CEFReader	RabbitMQJMSConnectionFactoryProvider
CiscoEmblemSyslogMessageReader	ReaderLookup
ClouderaHiveConnectionPool	RecordSetWriterLookup
ClouderaSchemaRegistry	RecordSinkServiceLookup
CMLLookupService	RedisConnectionPoolService
ConfluentEncodedSchemaReferenceReader	RedisDistributedMapCacheClientService
ConfluentEncodedSchemaReferenceWriter	RedshiftConnectionPool
ConfluentSchemaRegistry	RestLookupService
CouchbaseClusterService	ScriptedLookupService
CouchbaseKeyValueLookupService	ScriptedReader
CouchbaseMapCacheClient	ScriptedRecordSetWriter
CouchbaseRecordLookupService	ScriptedRecordSink
CSVReader	SimpleDatabaseLookupService
CSVRecordLookupService	SimpleKeyValueLookupService
CSVRecordSetWriter	SimpleRedisDistributedMapCacheClientService
DatabaseRecordLookupService	SimpleScriptedLookupService
DatabaseRecordSink	SiteToSiteReportingRecordSink
DatabaseTableSchemaRegistry	SlackRecordSink
DBCPCConnectionPool	SmbjClientProviderService
DBCPCConnectionPoolLookup	SnowflakeComputingConnectionPool
DistributedMapCacheClientService	StandardAsanaClientProviderService
DistributedMapCacheLookupService	StandardAzureCredentialsControllerService
DistributedMapCacheServer	StandardDropboxCredentialService
DistributedSetCacheClientService	StandardFileResourceService
DistributedSetCacheServer	StandardHashiCorpVaultClientService
EBCDICRecordReader [Technical Preview]	StandardHttpContextMap

ElasticSearchClientServiceImpl	StandardJsonSchemaRegistry [Technical Preview]
ElasticSearchLookupService	StandardOAuth2AccessTokenProvider
ElasticSearchStringLookupService	StandardPGPPrivateKeyService
EmailRecordSink	StandardPGPPublicKeyService
EmbeddedHazelcastCacheManager	StandardPLC4XConnectionPool [Technical Preview]
ExcelReader	StandardPrivateKeyService
ExternalHazelcastCacheManager	StandardProxyConfigurationService
FreeFormTextRecordSetWriter	StandardRestrictedSSLContextService
GCPCredentialsControllerService	StandardS3EncryptionService
GCSFileResourceService	StandardSnowflakeIngestManagerProviderService
GenericPLC4XConnectionPool [Technical Preview]	StandardSSLContextService
GrokReader	StandardWebClientServiceProvider
HadoopCatalogService	Syslog5424Reader
HadoopDBCPCConnectionPool	SyslogReader
HazelcastMapCacheClient	UDPEventRecordSink
HBase_2_ClientMapCacheService	VolatileSchemaCache
HBase_2_ClientService	WindowsEventLogReader
HBase_2_RecordLookupService	XMLReader
Hive3ConnectionPool	XMLRecordSetWriter
HiveCatalogService	YamlTreeReader
ImpalaConnectionPool	ZendeskRecordSink
IPFIXReader	

## Supported NiFi reporting tasks

Learn about the reporting tasks supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.0.

To ensure optimal performance and reliable support, it is crucial to use only supported reporting tasks and avoid deploying unsupported ones in production environments.

Additional reporting tasks are developed and tested by the community but are not officially supported by Cloudera. Reporting tasks may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.0

- AmbariReportingTask
- ControllerStatusReportingTask
- MetricsEventReportingTask
- MonitorDiskUsage
- MonitorMemory
- PrometheusReportingTask
- QueryNiFiReportingTask
- ReportLineageToAtlas
- ScriptedReportingTask

- SiteToSiteBulletinReportingTask
- SiteToSiteMetricsReportingTask
- SiteToSiteProvenanceReportingTask
- SiteToSiteStatusReportingTask

#### NiFi 2.0.0 in Cloudera Flow Management 4.2.1.0

- ControllerStatusReportingTask
- MonitorDiskUsage
- MonitorMemory
- PrometheusReportingTask
- QueryNiFiReportingTask
- ReportLineageToAtlas
- ScriptedReportingTask
- SiteToSiteBulletinReportingTask
- SiteToSiteMetricsReportingTask
- SiteToSiteProvenanceReportingTask
- SiteToSiteStatusReportingTask

### Supported NiFi parameter providers

Learn about the parameter providers supported in Flow Management Data Hub clusters using Apache NiFi 1 or NiFi 2 in Cloudera DataFlow for Data Hub 7.3.1.0.

To ensure optimal performance and reliable support, it is crucial to use only supported parameter providers and avoid deploying unsupported ones in production environments.

Additional parameter providers are developed and tested by the community but are not officially supported by Cloudera. Parameter providers may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

By adhering to the above guidelines, you can maintain stable and reliable workflows in your production environments.

#### NiFi 1.28.1 in Cloudera Flow Management 2.2.9.0

- AwsSecretsManagerParameterProvider
- AzureKeyVaultSecretsParameterProvider
- CyberArkConjurParameterProvider
- DatabaseParameterProvider
- EnvironmentVariableParameterProvider
- FileParameterProvider
- GcpSecretManagerParameterProvider
- HashiCorpVaultParameterProvider

#### NiFi 2.0.0 in Cloudera Flow Management 4.2.1.0

- AwsSecretsManagerParameterProvider
- AzureKeyVaultSecretsParameterProvider
- CyberArkConjurParameterProvider
- DatabaseParameterProvider
- EnvironmentVariableParameterProvider
- FileParameterProvider
- GcpSecretManagerParameterProvider
- HashiCorpVaultParameterProvider



- OnePasswordParameterProvider

## Supported NiFi flow analysis rules [Technical Preview]

Review the list of flow analysis rules supported in Cloudera Flow Management 4.2.1.0 included in Cloudera DataFlow for Data Hub 7.3.1.0.

Apache NiFi 2.0.0 introduces flow analysis rules, a new feature that enhances flow validation and management by evaluating components or parts of a flow and flagging rule violations to help optimize and maintain flow design.



### Important:

The Flow Analysis Rules feature is provided in Technical Preview. Do not use this feature in production!

Only the DisallowComponentType flow analysis rule is available for use.

## Supported NiFi Python components [Technical Preview]

Apache NiFi 2.0.0 introduces a set of NiFi components written in Python. Most of these Python components are supported by Cloudera.



### Important:

The Python API feature is provided in Technical Preview. Work is still in progress and breaking changes may occur in the upcoming release. Do not use this feature in production!

### Supported Python components:

#### Bedrock

Invokes different type of models with the given prompt via Bedrock.

#### ChunkData

Processes the output of Partition\* processors, and creates chunks from the input document in a standardized format, based on the user defined settings. The processor processes only text content, any other data - like images - are silently ignored. The output is a JSON document.

#### ChunkDocument

Divides a large text document into smaller chunks. Input is expected in the form of a FlowFile containing a JSON Lines document, where each line includes a 'text' and a 'metadata' element.

#### EmbedData

Embeds incoming data using a locally present model. The processor either embeds the whole incoming data, or specific values of an incoming JSON input. Models can be downloaded for example from huggingface.co by cloning the model's repository.

#### InsertToMilvus

Inserts or updates a vector in a Milvus collection. The input data is expected to be a float vector in JSON format (the dimension of the input must match dimension of the collection). Usually used together with EmbedData processor providing the float vector as the input for InsertToMilvus.

#### LexicalQueryMilvus

Performs a lexical search on a Milvus collection. The processor can query Milvus either by a list of IDs or by a filter. The IDs can be specified in a comma separated list in a specified attribute or in the content of the FlowFile. If the IDs are extracted from the content, the FlowFile should be in JSON format having an array of Milvus element objects. The JSON format is either the format of the output of the VectorQueryMilvus processor (list of lists) or a simple JSON list of Milvus objects. Each Milvus object is expected to have at least the primary key field specified.

#### ParseDocument

Parses incoming unstructured text documents and performs optical character recognition (OCR) to extract text from PDF and image files. The output is formatted as 'json-lines' with two keys: 'text'

and 'metadata'. The use of this processor may require significant storage space and RAM utilization due to third-party dependencies necessary for processing PDF and image files. Additionally, it is important to install Tesseract and Poppler on your system to enable the processing of PDFs or images.

**PartitionCsv**

Partitions a CSV file using the `partition_csv` function of `unstructured.io`. Properties are forwarded to `partition_csv` as parameters. The output is a JSON document in the format output by `partition_csv`.

**PartitionDocx**

Partitions DOCX data using the `partition_docx` function of `unstructured.io`. Properties are forwarded to `partition_docx` as parameters. The output is a JSON document in the format output by `partition_docx`.

**PartitionHtml**

Partitions HTML data using the `partition_html` function of `unstructured.io`. Properties are forwarded to `partition_html` as parameters. The output is a JSON document in the format output by `partition_html`.

**PartitionPdf**

Partitions a PDF file using the `partition_pdf` function of `unstructured.io`. Properties are forwarded to `partition_pdf` as parameters. The output is a JSON document in the format output by `partition_pdf`.

**PartitionText**

Partitions a text file using the `partition_text` function of `unstructured.io`. Properties are forwarded to `partition_text` as parameters. The output is a JSON document in the format output by `partition_text`.

**PromptChatGPT**

Submits a prompt to ChatGPT, writing the results either to a FlowFile attribute or to the contents of the FlowFile.

**PutChroma**

Publishes JSON data to a Chroma VectorDB. The incoming data must be in single JSON per Line format, containing two keys: 'text' and 'metadata'. The text must be a string, while metadata must be a map with string values. Any additional fields are ignored. If the collection name specified does not exist, the processor automatically creates the collection.

**PutOpenSearchVector**

Publishes JSON data to OpenSearch. The Incoming data must be in single JSON per Line format, each with two keys: 'text' and 'metadata'. The text must be a string, while metadata must be a map with strings for values. Any additional fields will be ignored.

**PutPinecone**

Creates vectors/embeddings that represent text content and sends the vectors to Pinecone. This use case assumes that the data has already been formatted in JSONL format with the text to be stored in Pinecone provided in the 'text' field.

**PutQdrant**

Publishes JSON data to Qdrant. The Incoming data must be in single JSON per Line format, each with two keys: 'text' and 'metadata'. The text must be a string, while metadata must be a map with strings for values. Any additional fields will be ignored.

**QueryChroma**

Queries a Chroma Vector Database to gather a specified number of documents that are most closely related to the given query.

**QueryOpenSearchVector**

Queries OpenSearch in order to gather a specified number of documents that are most closely related to the given query.

### **QueryPinecone**

Queries Pinecone to gather a specified number of documents that are most closely related to the given query.

### **QueryQdrant**

Queries Qdrant in order to gather a specified number of documents that are most closely related to the given query.

### **VectorQueryMilvus**

Performs a vector search in a Milvus collection. The input data is expected to be a float vector in JSON format. (the dimension of the input must match dimension of the collection). Usually used together with EmbedData processor providing the float vector as the input for VectorQueryMilvus.

While additional Python components are developed and tested by the community, they are not officially supported by Cloudera. Python components may be excluded due to various reasons, such as insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices. Do not use these unsupported Python components in your production environments.

## **Cloudera exclusive components [Technical Preview]**

Learn about Cloudera-exclusive components supported in Flow Management Data Hub clusters in Cloudera DataFlow for Data Hub 7.3.1.0.

Cloudera offers a set of NiFi components available only to its customers. These components provide additional functionalities and are specifically designed to enhance the Cloudera NiFi experience.

### **In Flow Management clusters with NiFi 1.28.1**

#### **Processors**

- CaptureChangeDebeziumDB2
- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- ConsumeKafka2CDP
- ConsumeKafka2RecordCDP
- ConvertProtobuf
- DeleteCDPObjectStore
- FetchCDPObjectStore
- GetJiraIssue
- InvokeGRPC
- ListCDPObjectStore
- ListenGRPC
- ListenNetFlow
- PublishKafka2CDP
- PublishKafka2RecordCDP
- PutCDPObjectStore
- PutClouderaHiveQL
- PutClouderaHiveStreaming
- PutClouderaORC
- PutIcebergCDC
- PutJiraIssue

- SelectClouderaHiveQL
- TriggerClouderaHiveMetaStoreEvent
- UpdateClouderaHiveTable
- UpdateDeltaLakeTable

**Controller services**

- ActiveMQJMSConnectionFactoryProvider
- ADLSIDBrokerCloudCredentialsProviderControllerService
- AWSIDBrokerCloudCredentialsProviderControllerService
- AzureBlobIDBrokerCloudCredentialsProviderControllerService
- AzureServiceBusJMSConnectionFactoryProvider
- CdpCredentialsProviderControllerService
- CdpOAuth2AccessTokenProviderControllerService
- CiscoEmblemSyslogMessageReader
- ClouderaHiveConnectionPool
- ClouderaSchemaRegistry
- CMLLookupService
- EBCDICRecordReader
- ImpalaConnectionPool
- IPFIXReader
- JiraRecordSink
- PostgreSQLConnectionPool
- RabbitMQJMSConnectionFactoryProvider
- RedshiftConnectionPool

**Parameter providers**

- CyberArkConjurParameterProvider

**In Flow Management clusters with NiFi 2.0.0****Processors**

- CaptureChangeDebeziumDB2
- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- ConsumeKafka2CDP
- ConsumeKafka2RecordCDP
- ConvertProtobuf
- DeleteCDPObjectStore
- FetchCDPObjectStore
- GetJiraIssue
- InvokeGRPC
- ListCDPObjectStore
- ListenGRPC
- ListenNetFlow
- PublishKafka2CDP
- PublishKafka2RecordCDP
- PutCDPObjectStore
- PutClouderaHiveQL
- PutClouderaHiveStreaming

- PutClouderaORC
- PutIcebergCDC
- PutJiraIssue
- SelectClouderaHiveQL
- TriggerClouderaHiveMetaStoreEvent
- UpdateClouderaHiveTable
- UpdateDeltaLakeTable

#### **Controller services**

- ActiveMQJMSConnectionFactoryProvider
- ADLSIDBrokerCloudCredentialsProviderControllerService
- AWSIDBrokerCloudCredentialsProviderControllerService
- AzureBlobIDBrokerCloudCredentialsProviderControllerService
- AzureServiceBusJMSConnectionFactoryProvider
- CdpCredentialsProviderControllerService
- CdpOAuth2AccessTokenProviderControllerService
- CiscoEmblemSyslogMessageReader
- ClouderaHiveConnectionPool
- ClouderaSchemaRegistry
- CMLLookupService
- EBCDICRecordReader
- IPFIXReader
- JiraRecordSink
- PostgreSQLConnectionPool
- RabbitMQJMSConnectionFactoryProvider
- RedshiftConnectionPool

#### **Parameter providers**

- CyberArkConjurParameterProvider

## **Components supported by partners**

Learn about the components built, maintained, and supported by Cloudera partners, and available in Flow Management Data Hub clusters in Cloudera DataFlow for Data Hub 7.3.1.0.

Although Cloudera's Quality Engineering teams have added test coverage for these components, they are not officially supported by Cloudera. For assistance, contact the respective partners directly.

#### **NiFi 1.28.1 processors supported by partners**

- ConsumePulsar (1.18.0)
- ConsumePulsarRecord (1.18.0)
- PublishPulsar (1.18.0)
- PublishPulsarRecord (1.18.0)

#### **NiFi 1.28.1 controller services supported by partners**

- PulsarClientAthenzAuthenticationService (1.18.0)
- PulsarClientJwtAuthenticationService (1.18.0)
- PulsarClientOAuthAuthenticationService (1.18.0)
- PulsarClientTlsAuthenticationService (1.18.0)
- StandardPulsarClientService (1.18.0)

These components can be used to push data into Apache Pulsar as well as getting data out of it. In case you have issues or questions while using these components, Cloudera recommends you to reach out to your StreamNative representative team.



**Note:** In Flow Management Data Hub clusters using NiFi 2, the Pulsar components are not included. You can manually download the components from a [Maven repository](#) and add them into your cluster.

## Unsupported features in Cloudera DataFlow for Data Hub 7.3.1

Some features exist within Cloudera DataFlow for Data Hub 7.3.1 components, but are not supported by Cloudera.

### Unsupported Flow Management features

Some Flow Management features exist in Cloudera DataFlow for Data Hub, but are not supported by Cloudera.

#### 7.3.1.400

There are no unsupported features in this release.

#### 7.3.1.0

##### NiFi 1.28.1

There are no unsupported features in this release.

##### NiFi 2.0.0

The following Flow Management technical preview features are available in Cloudera DataFlow for Data Hub 7.3.1.0. These features are not ready for production use. Cloudera encourages you to explore these technical preview features in non-production environments and share your feedback through the [Cloudera Community Forums](#).

- The Flow Analysis Rules engine is in Technical Preview, including all provided rules.
- The Python API feature is in Technical Preview, along with all Python extensions that use this API.
- The ability to run a Process Group using the Stateless Engine is in Technical Preview.

##### NiFi Registry

There are no unsupported features in this release.

#### Related Information

[Cloudera Community Forum](#)

### Unsupported Edge Management features [Technical Preview]

See the unsupported features listed in the [Cloudera Edge Management documentation](#).

### Unsupported Streams Messaging features

Some Streams Messaging features exist in Cloudera DataFlow for Data Hub 7.3.1, but are not supported by Cloudera.

#### Kafka

The following Kafka features are not ready for production deployment. Cloudera encourages you to explore these features in non-production environments and provide feedback on your experiences through the *Cloudera Community Forums*.

- Only Java and .Net based clients are supported. Clients developed with C, C++, Python, and other languages are currently not supported.
- The Kafka default authorizer is not supported. This includes setting ACLs and all related APIs, broker functionality, and command-line tools.
- SASL/SCRAM is only supported for delegation token based authentication. It is not supported as a standalone authentication mechanism.
- Kafka KRaft in this release of Cloudera Runtime is in technical preview and does not support the following:
  - Deployments with multiple log directories. This includes deployments that use JBOD for storage.
  - Delegation token based authentication.
  - Migrating an already running Kafka service from ZooKeeper to KRaft.
  - Atlas Integration.

### Schema Registry

There are no updates for this release.

### Streams Messaging Manager

There are no updates for this release.

### Streams Replication Manager

There are no updates for this release.

### Cruise Control

There are no updates for this release.

### Related Information

[Cloudera Community Forum](#)

[Setting up your Streams Messaging cluster](#)

## Unsupported Cloudera Streaming Analytics features

Some Cloudera Streaming Analytics features exist in Cloudera DataFlow for Cloudera Data Hub 7.3.1, but are not supported by Cloudera.

The following features are not ready for production deployment. Cloudera encourages you to explore these features in non-production environments and provide feedback on your experiences through the *Cloudera Community Forums*.

### Cloudera SQL Stream Builder

- Virtual environments for Python are not supported.

### Flink

- Apache Flink batch (DataSet) API
- GPU Resource Plugin
- SQL Client
- RAZ-enabled GCP environment
- The following features are not supported in SQL and Table API:
  - HBase Table Connector
  - Old Planner
  - Non-windowed (unbounded) joins, distinct

## Related Information

[Cloudera Community Forum](#)

# Known issues In Cloudera DataFlow for Data Hub 7.3.1

You must be aware of the known issues and limitations, the areas of impact, and workaround in Cloudera DataFlow for Data Hub 7.3.1.

## Known issues in Flow Management

Learn about the known issues and limitations in Flow Management clusters, their impact on functionality, and any available workarounds to mitigate these issues.

### 7.3.1.400

#### NiFi 1.28.1 with Cloudera Flow Management 2.2.9.400

There are no known issues in this release.

#### NiFi 2.3.0 with Cloudera Flow Management 4.2.1.400

#### NiFi service fails to start after upgrading to 7.3.1.400 from an earlier 7.3.1 version due to missing flow.json.gz file

When upgrading a Data Hub cluster from version 7.3.1.0, 7.3.1.100, 7.3.1.200, or 7.3.1.300 to 7.3.1.400 with NiFi 2, the upgrade process may fail, leaving NiFi instances in an unhealthy state and preventing the NiFi service from starting.

The issue occurs when only a flow.xml.gz file is present in the affected clusters, as the upgrade process expects a flow.json.gz file, the default format in NiFi 2.

This causes the post-upgrade validation script to fail with an error:

```
ERROR: please provide correct path to flow.json.gz file,  
current one is empty or invalid: "/hadoopfs/fsl/working-dir/  
flow.json.gz"
```

Resolve the issue using the following steps:

1. Stop the NiFi service.
2. Check your working directory.
3. If only a flow.xml.gz file is present, rename it to flow.json.gz on all NiFi nodes.
4. Start the NiFi service.



**Important:** After renaming, only flow.json.gz will be used by NiFi 2.

### 7.3.1.0

#### NiFi 1.28.1 with Cloudera Flow Management 2.2.9.0

#### CFM-4331: HBase 1.1.2 components incompatible with JDK17

HBase 1.1.2 components are not compatible with JDK 17.

To ensure full functionality and compatibility:

1. Upgrade HBase 1.1.2 components to their corresponding versions in HBase 2.
2. Upgrade your HBase servers.



If upgrading the servers is not feasible, the HBase 2 client can still interact with HBase 1 servers, but compatibility is limited. While basic functionalities work, new features introduced in the HBase 2 client are supported when interacting with an HBase 1 server.

### Unused NiFi configuration values

The following NiFi configuration values are no longer in use. They are still visible in the UI, but they are obsolete and have no effect on functionality.

- nifi.nar.hotfix.provider.file.list.identifier
- nifi.nar.hotfix.provider.location.identifier
- nifi.nar.hotfix.provider.last.modification.identifier
- nifi.nar.hotfix.provider.directory.identifier
- nifi.nar.hotfix.provider.date.time.format
- nifi.nar.hotfix.provider.proxy.user
- nifi.nar.hotfix.provider.proxy.password
- nifi.nar.hotfix.provider.proxy.server
- nifi.nar.hotfix.provider.proxy.server.port
- nifi.nar.hotfix.provider.connect.timeout
- nifi.nar.hotfix.provider.read.timeout
- nifi.nar.hotfix.provider.nar.location
- nifi.nar.hotfix.provider.poll.interval
- nifi.nar.hotfix.provider.implementation
- nifi.nar.hotfix.provider.user.name
- nifi.nar.hotfix.provider.password
- nifi.nar.hotfix.provider.base.url
- nifi.nar.hotfix.provider.required.version
- nifi.nar.hotfix.provider.enabled

### Unable to view NiFi or NiFi Registry user interface after upgrade due to authorization provider change

After upgrading Flow Management Data Hub clusters to Cloudera on cloud 7.3.1 (or 7.2.18), you may encounter an issue where the NiFi or NiFi Registry user interface is inaccessible, displaying the following error:

```
Unable to view the user interface
```

In versions prior to 7.2.18, NiFi group authorization relied on the host's SSSD configuration to synchronize groups using the SHELL user group provider. Starting in Cloudera on cloud 7.2.18, the SHELL user group provider is deprecated, and newly deployed clusters default to the LDAP user group provider. The impacted components are NiFi and NiFi Registry.



**Important:** Only newly deployed Flow Management Data Hub clusters on Cloudera on cloud 7.2.18 and higher are automatically configured to use the LDAP user group provider. Upgraded clusters from previous versions still use the SHELL user group provider, leading to potential authorization issues.

To resolve this issue in upgraded clusters, you must manually reconfigure the authorization provider to use LDAP. A script is available to automate the configuration update for both NiFi and NiFi Registry.

Follow the steps below to manually configure LDAP on your Flow Management Data Hub cluster:

1. Identify the management node of the Flow Management cluster and copy the Fully Qualified Domain Name (FQDN).
2. SSH into the management node.
3. Copy the script content provided below and save it to a file on the management node.
4. Set executable permissions on the script file: `chmod 755 [***SCRIPT_NAME***].sh`

5. Run the script passing the management node's FQDN as an argument by using the following command: `./[***SCRIPT_NAME***].sh FQDN_OF_MANAGEMENT_NODE`
6. When prompted, enter your Cloudera username and password to authorize the changes.

After completing these steps, NiFi and/or NiFi Registry will be configured to use the LDAP user group provider.

The script includes two functions, `nifi` and `nifiregistry`, which configure the LDAP user group provider for their respective services. Running the script updates NiFi and/or NiFi Registry to use LDAP, resolving the "Unable to view the user interface" error.

```
#!/bin/bash
clear
#init incoming variables
GREEN="\033[1;32m"
ORANGE="\033[38;2;255;165;0m"
RESET="\033[0m"
RED="\033[1;31m"
CM_HOST=$( 'hostname' )

#Get my auth
echo -ne "${ORANGE}User Name: " # Need to bracket this var to a
void a space in front
read -s USERNAME
echo -ne "\nEnter Password: $RESET"
read -s PASSWORD
echo #to prevent weird need to hit enter twice
AUTH="$USERNAME:$PASSWORD"

# GetMY LDAP INFO

#extract password and ldap info from cm.settings
if [[ ! -f /etc/cloudera-scm-server/cm.settings ]]; then
    echo -ne "${RED}Error: File /etc/cloudera-scm-server/cm.settin
gs does not exist." \
"\nMust be on Management node of DataHub\n"
    exit 1
fi
LDAP_URL=$(awk '/setsettings LDAP_URL/ {print $NF}' /etc/clouder
a-scm-server/cm.settings)
LDAP_BIND_DN=$(awk '/setsettings LDAP_BIND_DN/ {print $NF}' /etc/
cloudera-scm-server/cm.settings)
LDAP_BIND_PW=$(awk '/setsettings LDAP_BIND_PW/ {print $NF}' /etc/
cloudera-scm-server/cm.settings)
LDAP_USER_SEARCH_BASE=$(awk '/setsettings LDAP_USER_SEARCH_BASE/
{print $NF}' /etc/cloudera-scm-server/cm.settings)
LDAP_GROUP_SEARCH_BASE=$(awk '/setsettings LDAP_GROUP_SEARCH_B
ASE/ {print $NF}' /etc/cloudera-scm-server/cm.settings)

# Get My CM_API and if this fails it could be bad password or
host so I will ERROR
CM_API=$(curl -s -k -u "$AUTH" https://$CM_HOST:7183/api/version)
if [[ ${#CM_API} -gt 4 ]]; then # This means probably bad user or
password
    echo -ne "${RED} Error! Most likely bad credentials below is
response\n\n$CM_API $RESET"
    exit 1
fi
CM_HOST_API_URL="https://$CM_HOST:7183/api/$CM_API"
CM_CLUSTER_NAME=$(curl -s -k -u "$AUTH" -X GET "$CM_HOST_API_URL/
clusters?clusterType=any&view=SUMMARY" |
jq -r '.items[].name')
```

```

nifi ()
{
SERVICE="nifi-NIFI-BASE"

mapfile -t CM_ROLES < <(curl --header "Content-Type: applicatio
n/json" --silent --insecure --request GET \
"$CM_HOST_API_URL/clusters/$CM_CLUSTER_NAME/services/$SERVICE/
roleConfigGroups" \
-u $AUTH | jq -r '.items[].name' | grep -v "GATEWAY")

cat > .cloudera-$SERVICE.json <<- EOF
{"items":[
{"name":"nifi.ldap.url","value":"$LDAP_URL"},
{"name":"nifi.ldap.manager.dn","value":"$LDAP_BIND_DN"},
{"name":"nifi.ldap.manager.password","value":"$LDAP_BIND_P
W"},
{"name":"nifi.ldap.user.search.base","value":"$LDAP_USER_SE
ARCH_BASE"},
{"name":"xml.authorizers.userGroupProvider.ldap-user-group-prov
ider.property.Group Search Base","value":"$LDAP_GROUP_SEARCH_BAS
E"},
{"name":"nifi.ldap.enabled","value":"true"},
{"name":"xml.authorizers.userGroupProvider.shell-user-group-
provider.enabled","value":"false"},
{"name":"nifi.ldap.authentication.strategy","value":"LDAPS"},
{"name":"nifi.ldap.tls.protocol","value":"TLS"},
{"name":"nifi.ldap.tls.keystore.type","value":"jks"},
{"name":"nifi.ldap.tls.truststore.type","value":"jks"},
{"name":"nifi.ldap.tls.keystore","value":"\${nifi.security.ke
ystore}"},
{"name":"nifi.ldap.tls.truststore","value":"\${nifi.security.
truststore}"},
{"name":"xml.authorizers.userGroupProvider.ldap-user-group-pr
ovider.property.Group Object Class","value":"top"},
{"name":"xml.authorizers.userGroupProvider.ldap-user-group-p
rovider.property.User Group Name Attribute","value":"memberOf"},
{"name":"xml.authorizers.userGroupProvider.ldap-user-group-
provider.property.User Identity Attribute","value":"uid"},
{"name":"xml.authorizers.userGroupProvider.ldap-user-group-pr
ovider.property.Group Name Attribute","value":"cn"},
{"name":"xml.authorizers.userGroupProvider.composite-user-g
roup-provider.property.User Group Provider 2","value":"ldap-user-
group-provider"},
{"name":"staging/login-identity-providers.xml_role_safety_val
ve","value":"<property><name>xml.loginIdentityProviders.provider
.ldap-provider.property.TLS - Keystore Password</name><value>\${
nifi.security.keystorePasswd}</value></property><property><name>
xml.loginIdentityProviders.provider.ldap-provider.property.TLS -
Truststore Password</name><value>\${nifi.security.truststorePas
swd}</value></property>"},
{"name":"staging/authorizers.xml_role_safety_valve","value":"
<property><name>xml.authorizers.userGroupProvider.ldap-user-grou
p-provider.property.TLS - Keystore Password</name><value>\${nifi
.security.keystorePasswd}</value></property><property><name>xml.
authorizers.userGroupProvider.ldap-user-group-provider.property.
TLS - Truststore Password</name><value>\${nifi.security.truststo
rePasswd}</value></property>"}}
EOF
updateService
}
updateService ()
{

```

```

echo -ne "\n$GREEN Calling CM api to change $SERVICE config\n
$RESET"
for role in "${CM_ROLES[@]"; do
    echo -ne "$ORANGE \n Updating role $role *** $RESET\n"
    curl -s --header "Content-Type: application/json" --insecure
    --request PUT --data @.cloudera-$SERVICE.json \
    -u $AUTH "$CM_HOST_API_URL/clusters/$CM_CLUSTER_NAME/service
s/$SERVICE/roleConfigGroups/$role/config" > /dev/null
done
curl -s --header "Content-Type: application/json" --insecure
--request POST \
-u $AUTH "$CM_HOST_API_URL/clusters/$CM_CLUSTER_NAME/services/
$SERVICE/commands/restart" > /dev/null
rm -f .cloudera-$SERVICE.json
echo -ne "\n$GREEN Configured $ORANGE ldap-user-group provider
$GREEN on SERVICE: $ORANGE $SERVICE\n\n\n"
}

nifiregistry ()
{
    SERVICE="nifiregistry"
    mapfile -t CM_ROLES < <(curl --header "Content-Type: applicatio
n/json" --silent --insecure --request GET \
    "$CM_HOST_API_URL/clusters/$CM_CLUSTER_NAME/services/$SERVI
CE/roleConfigGroups" \
    -u $AUTH | jq -r '.items[] | name' | grep -v "GATEWAY")
    cat > .cloudera-$SERVICE.json <<- EOF
    {"items":[
        {"name": "nifi.registry.ldap.url", "value": "$LDAP_URL"},
        {"name": "nifi.registry.ldap.manager.dn", "value": "$LDAP_BI
ND_DN"},
        {"name": "nifi.registry.ldap.manager.password", "value": "$LDA
P_BIND_PW"},
        {"name": "nifi.registry.ldap.user.search.base", "value": "$LDA
P_USER_SEARCH_BASE"},
        {"name": "xml.authorizers.userGroupProvider.ldap-user-group-
provider.property.Group Search Base", "value": "$LDAP_GROUP_SEARC
H_BASE"},
        {"name": "nifi.registry.ldap.enabled", "value": "true"},
        {"name": "xml.authorizers.userGroupProvider.shell-user-group-
provider.enabled", "value": "false"},
        {"name": "nifi.registry.ldap.authentication.strategy", "value"
: "LDAPS"},
        {"name": "nifi.registry.ldap.tls.protocol", "value": "TLS"},
        {"name": "nifi.registry.ldap.tls.keystore.type", "value": "jks
"},
        {"name": "nifi.registry.ldap.tls.truststore.type", "value": "jk
s"},
        {"name": "nifi.registry.ldap.tls.keystore", "value": "\${nifi.
registry.security.keystore}"},
        {"name": "nifi.registry.ldap.tls.truststore", "value": "\${nifi.
registry.security.truststore}"},
        {"name": "xml.authorizers.userGroupProvider.ldap-user-group-
provider.property.Group Object Class", "value": "top"},
        {"name": "xml.authorizers.userGroupProvider.ldap-user-group-
provider.property.User Group Name Attribute", "value": "memberOf"},
        {"name": "xml.authorizers.userGroupProvider.ldap-user-group-pr
ovider.property.User Identity Attribute", "value": "uid"},
        {"name": "xml.authorizers.userGroupProvider.ldap-user-group-
provider.property.Group Name Attribute", "value": "cn"},
        {"name": "xml.authorizers.userGroupProvider.composite-user-gro
up-provider.property.User Group Provider 2", "value": "ldap-user-g
roup-provider"},
    ]}
}

```

```

    {"name": "staging/identity-providers.xml_role_safety_valve",
    "value": "<property><name>xml.loginIdentityProviders.provider.lda
p-provider.property.TLS - Keystore Password</name><value>\${nifi
.registry.security.keystorePasswd}</value></property><property><
name>xml.loginIdentityProviders.provider.ldap-provider.property.
TLS - Truststore Password</name><value>\${nifi.registry.security
.truststorePasswd}</value></property>"},
    {"name": "staging/authorizers.xml_role_safety_valve", "value"
: "<property><name>xml.authorizers.userGroupProvider.ldap-user-gr
oup-provider.property.TLS - Keystore Password</name><value>\${ni
fi.registry.security.keystorePasswd}</value></property><property
><name>xml.authorizers.userGroupProvider.ldap-user-group-provide
r.property.TLS - Truststore Password</name><value>\${nifi.regist
ry.security.truststorePasswd}</value></property>" }
  ]}
EOF
updateService
}
nifi
nifiregistry
echo -ne "$GREEN\nRestarting $ORANGE NiFi / NiFi Registry $RESET
\n"
sleep 10
exit 0

```

### PutIcebergCDC processor error: Unable to specify server's Kerberos Principal name

When using the PutIcebergCDC processor, you may encounter an error if the Hadoop Configuration Resources property specified for the Catalog Service only includes the standard Hadoop configuration files from Cloudera environment (/etc/hadoop/conf/core-site.xml, /etc/hadoop/conf/ssl-client.xml, and /etc/hive/conf/hive-site.xml). The error message states:

```
Failed to specify server's Kerberos principal name.
```

To resolve this issue, simply add the hdfs-site.xml file to the Hadoop Configuration Resources of the PutIcebergCDC processor's Catalog Service.

### Incomplete Ranger policy for NiFi metrics in Cloudera Manager

Cloudera Manager does not accurately reflect NiFi metrics for the NiFi service due to incomplete Flow NiFi access policies in Ranger. The required 'nifi' group is not included in the access policies, resulting in restricted access to the metrics data.

To ensure that Cloudera Manager accurately reflects the NiFi metrics for the NiFi service:

1. Log in to Ranger and navigate to the Flow NiFi access policies.
2. Add the 'nifi' group to the relevant access policies to ensure that Cloudera Manager can access the metrics data.
3. Confirm and save the updated policies.

### InferAvroSchema may fail when inferring schema for JSON data

In Apache NiFi 1.17, the dependency on Apache Avro has been upgraded to 1.11.0. However, the InferAvroSchema processor depends on the hadoop-libraries NAR from which the Avro version comes from, causing a NoSuchMethodError exception.



**Important:** This processor is not supported by Cloudera and its use is highly discouraged as inferring a schema from the data is not recommended in production data flows.

Having well defined schemas ensures consistent behavior, allows for proper schema versioning, and prevents downstream systems from generating errors because of unexpected schema changes.

Besides, schema inference may not always be 100% accurate and can be an expensive operation in terms of performances.

Use the ExtractRecordSchema processor with the proper Reader to infer the Avro schema for your data.

### NiFi 2.0.0 with Cloudera Flow Management 4.2.1.0



**Important:** Apache NiFi 2.0.0 is not suitable for production when using the newly released features. Specifically, there are known issues regarding resource consumption when using the Python-based extensions.

### Processors using OpenAI library may not work

When using Flow Management clusters, several processors relying on the OpenAI library are not functional due to compatibility issues caused by OpenAI API changes. The affected processors use an outdated OpenAI library version that is no longer supported. The impacted processors are:

- PutChroma
- QueryChroma
- PromptChatGPT
- PutOpenSearchVector
- QueryOpenSearchVector
- PutPinecone
- QueryPinecone
- PutQdrant
- QueryQdrant

These processors require an updated OpenAI library version (1.56.2 or later) to function correctly.

To restore functionality for the impacted processors, follow these steps:

1. Update the OpenAI library version in the associated Python code to version 1.56.2.

- a. Locate the processor's py file.

For example: `/opt/cloudera/parcels/CFM-4.0.0.0-382/NIFI-2/python/extensions/openai/PromptChatGPT.py`

- b. Find `"openai==1.9.0"` and replace it with `"openai==1.56.2"`.

2. Navigate to the NiFi work directory and delete the folder for the affected processors.

The directory path for an affected processor typically follows this structure: `/var/lib/nifi/python_artifacts/extensions/<ProcessorName>/<Version>`

For example, for the PromptChatGPT processor, the path would be: `/var/lib/nifi/python_artifacts/extensions/PROMPTCHATGPT/2.0.0.4.0.0.0-382`

So in this case, delete the entire PromptChatGPT folder, including its version folder.

3. Restart the NiFi service to apply the changes.

### Invalid Python version

Due to the invalid Python version defined for the NiFi service, the Python API based processors (such as PromptChatGPT, QueryPinecone, and so on) will remain invalid as the NiFi service will be unable to download the associated dependencies. The issue can be resolved by changing the version for the `nifi.python.command` property.

1. Go to your cluster in Cloudera Manager.
2. Select NiFi from the list of services.
3. Select Configuration.
4. Review the value defined for `nifi.python.command` property.
5. Change the value to `python3.11` if the current value is `python3.9`.
6. Click Save changes.

7. Stop the NiFi service.
8. Delete the /hadoopfs/fs4/working-dir/python\_artifacts directory from all NiFi nodes.
9. Restart the NiFi service.

#### **PutIcebergCDC processor error: Unable to specify server's Kerberos Principal name**

When using the PutIcebergCDC processor, you may encounter an error if the Hadoop Configuration Resources property specified for the Catalog Service only includes the standard Hadoop configuration files from Cloudera environment (/etc/hadoop/conf/core-site.xml, /etc/hadoop/conf/ssl-client.xml, and /etc/hive/conf/hive-site.xml). The error message states: Failed to specify server's Kerberos principal name.

To resolve this issue, simply add the hdfs-site.xml file to the Hadoop Configuration Resources of the PutIcebergCDC processor's Catalog Service.

#### **NiFi service fails to start after upgrading from 7.3.1.0 to a higher version due to missing flow.json.gz file**

When upgrading a Data Hub cluster from version 7.3.1.0 to 7.3.1.100, 7.3.1.200, or 7.3.1.300 with NiFi 2, the upgrade process may fail, leaving NiFi instances in an unhealthy state and preventing the NiFi service from starting.

The issue occurs because the upgrade process expects a flow.json.gz file (the default for NiFi 2), but the affected clusters only contain a flow.xml.gz file. This mismatch causes the post-upgrade validation script to fail with the following error:

```
ERROR: please provide correct path to flow.json.gz file,  
current one is empty or invalid: "/hadoopfs/fs1/working-dir/  
flow.json.gz"
```

Resolve the issue using the following steps:

1. Stop the NiFi service.
2. Replace the existing NiFi CSD JAR on the Cloudera Manager node with a patched version. Contact Cloudera Support to obtain the correct file.
3. Restart Cloudera Manager Services.
4. Rename the existing flow.xml.gz file to flow.json.gz on all NiFi nodes.
5. Start the NiFi service.



**Important:** After this workaround is applied, only flow.json.gz will be used by NiFi 2.

This issue was addressed in Cloudera on cloud 7.3.1.400 with Cloudera Flow Management 2.2.9.400.

## **Known issues in Edge Management [Technical Preview]**

Learn about the known issues in Edge Management clusters, the impact or changes to the functionality, and any available workaround.

For Edge Management known issues, see the [Cloudera Edge Management documentation](#).

## **Known issues in Streams Messaging**

Learn about the known issues in Streams Messaging clusters, the impact or changes to the functionality, and the workaround.

### **Kafka**

Learn about the known issues and limitations in Kafka in this release:

## Known Issues

**OPSAPS-59553: Streams Messaging Manager's bootstrap server config should be updated based on Kafka's listeners**

Streams Messaging Manager does not show any metrics for Kafka or Kafka Connect when multiple listeners are set in Kafka.

Workaround: Streams Messaging Manager cannot identify multiple listeners and still points to bootstrap server using the default broker port (9093 for SASL\_SSL). You need to override the bootstrap server URL by performing the following steps:

1. In Cloudera Manager, go to Streams Messaging Manager Configuration Streams Messaging Manager Rest Admin Server Advanced Configuration Snippet (Safety Valve) for streams-messaging-manager.yaml
2. Override bootstrap server URL (hostname:port as set in the listeners for broker) for streams-messaging-manager.yaml.
3. Save your changes.
4. Restart Streams Messaging Manager.

**The offsets.topic.replication.factor property must be less than or equal to the number of live brokers**

The offsets.topic.replication.factor broker configuration is now enforced upon auto topic creation. Internal auto topic creation will fail with a GROUP\_COORDINATOR\_NOT\_AVAILABLE error until the cluster size meets this replication factor requirement.

None

**Requests fail when sending to a nonexistent topic with auto.create.topics.enable set to true**

The first few produce requests fail when sending to a nonexistent topic with auto.create.topics.enable set to true.

Increase the number of retries in the producer configuration setting retries.

**KAFKA-2561: Performance degradation when SSL Is enabled**

In some configuration scenarios, significant performance degradation can occur when SSL is enabled. The impact varies depending on your CPU, JVM version, Kafka configuration, and message size. Consumers are typically more affected than producers.

Configure brokers and clients with ssl.secure.random.implementation = SHA1PRNG. It often reduces this degradation drastically, but its effect is CPU and JVM dependent.

**RANGER-3809: Idempotent Kafka producer fails to initialize due to an authorization failure**

Kafka producers that have idempotence enabled require the Idempotent Write permission to be set on the cluster resource in Ranger. If permission is not given, the client fails to initialize and an error similar to the following is thrown:

```
org.apache.kafka.common.KafkaException: Cannot execute transactional method because we are in an error state
    at org.apache.kafka.clients.producer.internals.TransactionManager.maybeFailWithError(TransactionManager.java:1125)
    at org.apache.kafka.clients.producer.internals.TransactionManager.maybeAddPartition(TransactionManager.java:442)
    at org.apache.kafka.clients.producer.KafkaProducer.doSend(KafkaProducer.java:1000)
    at org.apache.kafka.clients.producer.KafkaProducer.send(KafkaProducer.java:914)
    at org.apache.kafka.clients.producer.KafkaProducer.send(KafkaProducer.java:800)
    .
    .
    .
    Caused by: org.apache.kafka.common.errors.ClusterAuthorizationException: Cluster authorization failed.
```



Idempotence is enabled by default for clients in Kafka 3.0.1, 3.1.1, and any version after 3.1.1. This means that any client updated to 3.0.1, 3.1.1, or any version after 3.1.1 is affected by this issue.

This issue has two workarounds, do either of the following:

- Explicitly disable idempotence for the producers. This can be done by setting `enable.idempotence` to `false`.
- Update your policies in Ranger and ensure that producers have Idempotent Write permission on the cluster resource.

#### **CDPD-49304: AvroConverter does not support composite default values**

AvroConverter cannot handle schemas containing a STRUCT type default value.

None.

#### **DBZ-4990: The Debezium Db2 Source connector does not support schema evolution**

The Debezium Db2 Source connector does not support the evolution (updates) of schemas. In addition, schema change events are not emitted to the schema change topic if there is a change in the schema of a table that is in capture mode. For more information, see [DBZ-4990](#).

None.

#### **CFM-3532: The Stateless NiFi Source, Stateless NiFi Sink, and HDFS Stateless Sink connectors cannot use Snappy compression**

This issue only affects Stateless NiFi Source and Sink connectors if the connector is running a dataflow that uses a processor that uses Hadoop libraries and is configured to use Snappy compression. The HDFS Stateless Sink connector is only affected if the Compression Codec or Compression Codec for Parquet properties are set to SNAPPY.

If you are affected by this issue, errors similar to the following will be present in the logs.

```
Failed to write to HDFS due to java.lang.UnsatisfiedLinkError: org.apache.hadoop.util.NativeCodeLoader.buildSupportsSnappy()
```

```
Failed to write to HDFS due to java.lang.RuntimeException: native snappy library not available: this version of libhadoop was built without snappy support.
```

Download and deploy missing libraries.



**Important:** Ensure that you complete steps 1-11 on all Kafka Connect hosts. Additionally, ensure that the advanced configuration snippet in step 12 is configured for all Kafka Connect role instances.

1. Create the `/opt/nativelibs` directory.

```
mkdir /opt/nativelibs
```

2. Change the owner to kafka.

```
chown kafka:kafka /opt/nativelibs
```

3. Locate the directory containing the Hadoop native libraries and copy its contents to the directory you created.

```
cp /opt/cloudera/parcels/CDH/lib/hadoop/lib/native/* /opt/nativelibs
```

4. Verify that `libsnapy.so` was copied to the directory you created.

- Remove the following from /opt/nativelibs.

```
libhadoop.a
libhadoop.so
libhadoop.so.1.0.0
```

- Run the following command.

```
hadoop version
```

The command returns the Hadoop version running in the cluster. Note down the first three digits in the version.

- Go to <https://archive.apache.org/dist/hadoop/common/> and download the Hadoop version that matches the first three digits of the version running in the cluster.

For example, if your Hadoop version is 3.1.1.7.1.9.0-296, then you need to download Hadoop 3.1.1.

- Extract the downloaded archive.
- Copy the following libraries from the downloaded archive to /opt/nativelibs on the cluster host.

```
libhadoop.a
libhadoop.so.1.0.0
```

The libraries are located in `hadoop-***VERSION***/lib/native`.

- Create a symlink named libhadoop.so and point it to /opt/nativelibs/libhadoop.so.1.0.0.

```
ln -s /opt/nativelibs/libhadoop.so.1.0.0 /opt/nativelibs/libhadoop.so
```

- Change the owner of every entry within /opt/nativelibs to kafka.

```
chown -h kafka:kafka /opt/nativelibs/*
```

- In Cloudera Manager, go to **Kafka service Configuration**.
- Add the following key-value pair to **Kafka Connect Environment Advanced Configuration Snippet (Safety Valve)**.
  - Key: LD\_LIBRARY\_PATH
  - Value: /opt/nativelibs
- Click **Save Changes**.
- Restart the Kafka service.

#### **OPSAPS-69317: Kafka Connect Rolling Restart Check fails if SSL Client authentication is required**

The rolling restart action does not work in Kafka Connect when the `ssl.client.auth` option is set to required. The health check fails with a timeout which blocks restarting the subsequent Kafka Connect instances.

You can set `ssl.client.auth` to requested instead of required and initiate a rolling restart again. Alternatively, you can perform the rolling restart manually by restarting the Kafka Connect instances one-by-one and checking periodically whether the service endpoint is available before starting the next one.

#### **Unsupported features**

The following Kafka features are not supported in Cloudera:

- Only Java and .Net based clients are supported. Clients developed with C, C++, Python, and other languages are currently not supported.
- The Kafka default authorizer is not supported. This includes setting ACLs and all related APIs, broker functionality, and command-line tools.

- SASL/SCRAM is only supported for delegation token based authentication. It is not supported as a standalone authentication mechanism.
- Kafka KRaft in this release of Cloudera Runtime is in technical preview and does not support the following:
  - Deployments with multiple log directories. This includes deployments that use JBOD for storage.
  - Delegation token based authentication.
  - Migrating an already running Kafka service from ZooKeeper to KRaft.
  - Atlas Integration.

## Limitations

### Collection of Partition Level Metrics May Cause Cloudera Manager Performance to Degrade

If the Kafka service operates with a large number of partitions, collection of partition level metrics may cause Cloudera Manager performance to degrade.

If you are observing performance degradation and your cluster is operating with a high number of partitions, you can choose to disable the collection of partition level metrics.



**Important:** If you are using Streams Messaging Manager to monitor Kafka or Cruise Control for rebalancing Kafka partitions, be aware that both Streams Messaging Manager and Cruise Control rely on partition level metrics. If partition level metric collection is disabled, Streams Messaging Manager will not be able to display information about partitions. In addition, Cruise Control will not operate properly.

Complete the following steps to turn off the collection of partition level metrics:

1. Obtain the Kafka service name:
  - a. In Cloudera Manager, Select the Kafka service.
  - b. Select any available chart, and select Open in Chart Builder from the configuration icon drop-down.
  - c. Find \$SERVICENAME= near the top of the display.

The Kafka service name is the value of \$SERVICENAME.

2. Turn off the collection of partition level metrics:
  - a. Go to Hosts Hosts Configuration .
  - b. Find and configure the Cloudera Manager Agent Monitoring Advanced Configuration Snippet (Safety Valve) configuration property.

Enter the following to turn off the collection of partition level metrics:

```
[KAFKA_SERVICE_NAME]_feature_send_broker_topic_partition_entity_update_enabled=false
```

Replace [KAFKA\_SERVICE\_NAME] with the service name of Kafka obtained in step 1. The service name should always be in lower case.

- c. Click Save Changes.

## Schema Registry

Learn about the known issues and limitations in Schema Registry in this release:

### CDPD-40380: Authorization checking issue when Kerberos is disabled

Due to an issue in Ranger, when Kerberos is disabled then it is not possible to check authorization.

1. Open Schema Registry configuration in Cloudera Manager.
2. Find the ranger.plugin.schema-registry.service.name field.

3. Replace `GENERATED_RANGER_SERVICE_NAME` with the actual name of the service.
4. Restart the Schema Registry service.

**CDPD-49304: AvroConverter does not support composite default values**

AvroConverter cannot handle schemas containing a `STRUCT` type default value.

None.

**OPSAPS-70971: Schema Registry does not have permissions to use Atlas after an upgrade**

Following an upgrade, Schema Registry might not have the required permissions in Ranger to access Atlas. As a result, Schema Registry's integration with Atlas might not function in secure clusters where Ranger authorization is enabled.

1. Access the Ranger Console (Ranger Admin web UI).
2. Click the `cm_atlas` resource-based service.
3. Add the `schemaregistry` user to the `all - *` policies.
4. Click `Manage Service Edit Service`.
5. Add the `schemaregistry` user to the `default.policy.users` property.

**OPSAPS-69317: Kafka Connect Rolling Restart Check fails if SSL Client authentication is required**

The rolling restart action does not work in Kafka Connect when the `ssl.client.auth` option is set to required. The health check fails with a timeout which blocks restarting the subsequent Kafka Connect instances.

You can set `ssl.client.auth` to requested instead of required and initiate a rolling restart again. Alternatively, you can perform the rolling restart manually by restarting the Kafka Connect instances one-by-one and checking periodically whether the service endpoint is available before starting the next one.

## Streams Messaging Manager

Learn about the known issues in Streams Messaging Manager in this release.

**OPSAPS-59597: Streams Messaging Manager UI logs are not supported by Cloudera Manager**

Cloudera Manager does not display a Log Files menu for Streams Messaging Manager UI role (and Streams Messaging Manager UI logs cannot be displayed in the Cloudera Manager UI) because the logging type used by Streams Messaging Manager UI is not supported by Cloudera Manager.

View the Streams Messaging Manager UI logs on the host.

**CDPD-39313: Some numbers are not rendered properly in Streams Messaging Manager UI**

Very large numbers can be imprecisely represented on the UI. For example, bytes larger than 8 petabytes would lose precision.

None.

**OPSAPS-59553: Streams Messaging Manager's bootstrap server config should be updated based on Kafka's listeners**

Streams Messaging Manager does not show any metrics for Kafka or Kafka Connect when multiple listeners are set in Kafka.

Streams Messaging Manager cannot identify multiple listeners and still points to bootstrap server using the default broker port (9093 for `SASL_SSL`). You need to override bootstrap server URL (hostname:port as set in the listeners for broker). Add the bootstrap server details in Streams Messaging Manager safety valve in the following path:

1. In Cloudera Manager, go to `Streams Messaging Manager Configuration Streams Messaging Manager Rest Admin Server Advanced Configuration Snippet (Safety Valve) for streams-messaging-manager.yaml`

2. Add the following value for bootstrap servers.

```
streams.messaging.manager.kafka.bootstrap.servers=<comma-separated list of brokers>
```

3. Save your changes.
4. Restart Streams Messaging Manager.

#### Limitations

##### **CDPD-36422: 1MB flow.snapshot freezes Safari**

While importing large connector configurations, flow.snapshots reduces the usability of the Streams Messaging Manager when using Safari browser.

Use a different browser (Chrome/Firefox/Edge).

## Streams Replication Manager

Learn about the known issues and limitations in Streams Replication Manager in this release:

#### Known Issues

##### **CDPD-22089: Streams Replication Manager does not sync re-created source topics until the offsets have caught up with target topic**

Messages written to topics that were deleted and re-created are not replicated until the source topic reaches the same offset as the target topic. For example, if at the time of deletion and re-creation there are a 100 messages on the source and target clusters, new messages will only get replicated once the re-created source topic has 100 messages. This leads to messages being lost.

None

##### **CDPD-11079: Blacklisted topics appear in the list of replicated topics**

If a topic was originally replicated but was later disallowed (blacklisted), it will still appear as a replicated topic under the /remote-topics REST API endpoint. As a result, if a call is made to this endpoint, the disallowed topic will be included in the response. Additionally, the disallowed topic will also be visible in the Streams Messaging Manager UI. However, its Partitions and Consumer Groups will be 0, its Throughput, Replication Latency and Checkpoint Latency will show N/A.

None

##### **CDPD-30275: Streams Replication Manager may automatically re-create deleted topics on target clusters**

If auto.create.topics.enable is enabled, deleted topics might get automatically re-created on target clusters. This is a timing issue. It only occurs if remote topics are deleted while the replication of the topic is still ongoing.

1. Remove the topic from the topic allowlist with srm-control. For example:

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --remove [TOPIC1]
```

2. Wait until Streams Replication Manager is no longer replicating the topic.
3. Delete the remote topic in the target cluster.

#### Limitations

##### **Streams Replication Manager cannot replicate Ranger authorization policies to or from Kafka clusters**

Due to a limitation in the Kafka-Ranger plugin, Streams Replication Manager cannot replicate Ranger policies to or from clusters that are configured to use Ranger for authorization. If you are using Streams Replication Manager to replicate data to or from a cluster that uses Ranger, disable authorization policy synchronization in Streams Replication Manager. This can be achieved by clearing the Sync Topic Acls Enabled (sync.topic.acls.enabled) checkbox.

##### **Streams Replication Manager cannot ensure the exactly-once semantics of transactional source topics**

Streams Replication Manager data replication uses at-least-once guarantees, and as a result cannot ensure the exactly-once semantics (EOS) of transactional topics in the backup/target cluster.



**Note:** Even though EOS is not guaranteed, you can still replicate the data of a transactional source, but you must set `isolation.level` to `read_committed` for Streams Replication Manager's internal consumers. This can be done by adding `[***SOURCE CLUSTER ALIAS***]->[***TARGET CLUSTER ALIAS***].consumer.isolation.level=read_committed` to the Streams Replication Manager's Replication Configs Streams Replication Manager service property in Cloudera Manager. The `isolation.level` property can be set on a global connector or replication level. For example:

```
#Global connector level
connectors.consumer.isolation.level=read_committed
#Replication level
uswest->useast.consumer.isolation.level=read_committed
```

### Streams Replication Manager checkpointing is not supported for transactional source topics

Streams Replication Manager does not correctly translate checkpoints (committed consumer group offsets) for transactional topics. Checkpointing assumes that the offset mapping function is always increasing, but with transactional source topics this is violated. Transactional topics have control messages in them, which take up an offset in the log, but they are never returned on the consumer API. This causes the mappings to decrease, causing issues in the checkpointing feature. As a result of this limitation, failover operations for transactional topics is not possible.

## Cruise Control

Learn about the known issues and limitations in Cruise Control in this release:

### CDPD-44676: Rebalancing with Cruise Control does not work if the metric reporter fails to report the CPU usage metric

If the CPU usage metric is not reported, the `numValidWindows` in Cruise Control will be 0 and proposal generation as well as partition rebalancing will not work. If this issue is present, the following message will be included in the Kafka logs:

```
WARN com.linkedin.kafka.cruisecontrol.metricsreporter.CruiseControlMetricsReporter:
[CruiseControlMetricsReporterRunner]: Failed reporting CPU
util.
```

```
java.io.IOException: Java Virtual Machine recent CPU usage is not
available.
```

This issue is only known to affect Kafka broker hosts that have the following specifications:

- CPU: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
- OS: Linux 4.18.5-1.el7.elrepo.x86\_64 #1 SMP Fri Aug 24 11:35:05 EDT 2018 x86\_64
- Java version: 8-18

Move the broker to a different machine where the CPU is different. This can be done by moving the host to a different cluster. For more information, see [Moving a Host Between Clusters](#)



**Note:** Cluster nodes affected by this issue are not displayed as unhealthy.

## Known issues in Cloudera Streaming Analytics

Learn about the known issues in Cloudera Streaming Analytics clusters, the impact or changes to the functionality, and the workaround.

### Cloudera SQL Stream Builder

#### CSA-4858 - Kerberos encryption type detection does not always work correctly for Cloudera SQL Stream Builder

Cloudera SQL Stream Builder detects no supported encryption types even though there is a list of allowed encryption types in the `krb5.conf` file. This causes an error when generating keytabs from the principal and password pair.

1. Run `ktutil` on your cluster.
2. Change the configuration with the following commands:

```
addent -password -p <username> -k 1 -e aes256-cts  
wkt /tmp/new_keytab.keytab
```

3. Upload the new keytab on Streaming SQL Console.

### Flink

In Cloudera Streaming Analytics, the following SQL API features are in preview:

- Match recognize
- Top-N
- Stream-Table join (without rowtime input)

#### CSA-5525 - Illegal join reordering in Flink optimizer

Flink optimizer's reordering might violate certain clauses (for example `FOR SYSTEM_TIME AS OF`) that are supported only on a specific side of a join operation, resulting in an error.

Example error message:

```
Caused by: org.apache.flink.table.api.TableException: Temporal t  
able join only support apply FOR SYSTEM_TIME AS OF on the right  
table
```

Set `table.optimizer.join-reorder-enabled` to `false`, until the issue is fixed in upstream Flink.

#### Third-party dependencies upgraded in Cloudera on cloud might cause Flink jobs to fail

After upgrading Cloudera on cloud, Flink jobs might fail due to upgraded 3rd-party dependencies. For example, this could happen with `awssdk`, which has been updated to version 2.23.10 in Cloudera on cloud version 7.2.18.

Verify your application's dependency versions against the Cloudera-supported versions before upgrading to a newer version of Cloudera on cloud. For more information see [Updating Flink job dependencies](#).

#### DataStream conversion limitations

- Converting between Tables and POJO DataStreams is currently not supported in Cloudera Streaming Analytics.
- Object arrays are not supported for Tuple conversion.
- The `java.time` class conversions for Tuple DataStreams are only supported by using explicit `TypeInformation`: `LegacyInstantTypeInfo`, `LocalTimeTypeInfo`, `getInfoFor(LocalDate/LocalDateTime/LocalTime.class)`.

- Only `java.sql.Timestamp` is supported for rowtime conversion, `java.time.LocalDateTime` is not supported.

#### **Kudu catalog limitations**

- **CREATE TABLE**
  - Primary keys can only be set by the `kudu.primary-key-columns` property. Using the `PRIMARY KEY` constraint is not yet possible.
  - Range partitioning is not supported.
- When getting a table through the catalog, `NOT NULL` and `PRIMARY KEY` constraints are ignored. All columns are described as being nullable, and not being primary keys.
- Kudu tables cannot be altered through the catalog other than simply renaming them.

#### **Schema Registry catalog limitations**

- Currently, the Schema Registry catalog / format only supports reading messages with the latest enabled schema for any given Kafka topic at the time when the SQL query was compiled.
- No time-column and watermark support for Registry tables.
- No **CREATE TABLE** support. Schemas have to be registered directly in the SchemaRegistry to be accessible through the catalog.
- The catalog is read-only. It does not support table deletions or modifications.
- By default, it is assumed that Kafka message values contain the schema id as a prefix, because this is the default behaviour for the SchemaRegistry Kafka producer format. To consume messages with schema written in the header, the following property must be set for the Registry client: `store.schema.version.id.in.header: true`.

## Deprecation notices in Cloudera DataFlow for Data Hub 7.3.1

Certain features and functionalities have been removed or deprecated in Cloudera DataFlow for Data Hub 7.3.1. You must review these items to understand whether you must modify your existing configuration. You can also learn about the features that will be removed or deprecated in the future release to plan for the required changes.

### Terminology

Items in this section are designated as follows:

#### **Deprecated**

Technology that Cloudera is removing in a future release. Marking an item as deprecated gives you time to plan for removal in a future release.

#### **Moving**

Technology that Cloudera is moving from a future release and is making available through an alternative Cloudera offering or subscription. Marking an item as moving gives you time to plan for removal in a future release and plan for the alternative Cloudera offering or subscription for the technology.

#### **Removed**

Technology that Cloudera has removed from the product and is no longer available or supported as of this release. Take note of technology marked as removed since it can potentially affect your upgrade plans.



## Deprecation notices for Cloudera Streaming Analytics

Certain features and functionality are deprecated or removed in Cloudera Streaming Analytics. You must review these changes along with the information about the features in Cloudera Streaming Analytics that will be removed or deprecated in a future release.

### Deprecated

#### Support for JavaScript UDFs

Due to the deprecation of the Nashorn engine used in JDK 8 and 11, User-Defined Functions (UDFs) written in JavaScript are deprecated in Cloudera Streaming Analytics 1.13. Cloudera recommends that customers start using [Python UDFs](#) for all new developments, and start migrating their JavaScript UDFs to Python to prepare for future upgrades.

#### v1 API

The v1 REST API for Cloudera SQL Stream Builder has been deprecated and will be removed in a future version of Cloudera Streaming Analytics.

Customers are advised to migrate to the v2 API, available for Cloudera SQL Stream Builder.

For more information on the v2 API, see [Cloudera SQL Stream Builder REST API reference](#).

## Fixed issues in Cloudera DataFlow for Data Hub 7.3.1

This release addresses a number of issues previously reported through Cloudera Support or identified by the Cloudera Quality Engineering team. Some of these issues were listed in earlier versions under the *Known Issues* section.

Review the list of issues resolved in Cloudera DataFlow for Data Hub 7.3.1.

## Fixed issues in Flow Management

Review the list of Flow Management issues that are resolved in Cloudera DataFlow for Data Hub.

### 7.3.1.400

#### NiFi 1.28.1 with Cloudera Flow Management 2.2.9.400

Cloudera Flow Management 2.2.9.400 is based on Apache NiFi 1.28.1. It includes all fixed issues of this Apache NiFi release, as well as the following additional fixes:

- CFM-5252 Updating group in policies fails with KeyError: '/controller'
- CFM-4842 Fix mysql installation for Redhat 8.10
- CFM-4315 Fix pre upgrade check during NiFi start after upgrade to CFM-4.0.0.0 and upgrade path check
- CFM-5285 NiFi startup fails because some NARs were compiled using a more recent version of Java.
- CFM-3870 QueryAirtableTable processor is no longer working
- CFM-5262 Fix HBase extensions: IllegalAccessError
- CFM-5499 Remove ranger repo fails in tests because of timeout
- CFM-5249 Cloudera Manager fails to restart after NiFi CSD installation
- CFM-4827 Post upgrade to CFM 2.1.7 from CFM 2.1.6 json\* type processors and controllers becomes invalid
- CFM-4831 Atlas reporting is not working as expected
- CFM-4828 Upgrade commons-compress version to 1.27.1 to fix decompression issue
- CFM-4823 NIFI-14158 - DeleteHDFS processor ignores delete action return value
- CFM-4825 NIFI-13829 - MonitorActivity processor generating false inactivity files
- CFM-4830 NIFI-13742 - SelectHiveQL processors do not normalize column names when specified

- CFM-4819 NIFI-14174 - Protobuf Reader fails to generate schema for proto files containing circular reference
- CFM-4824 NIFI-13860 - IPLookupService throws exception when lookup fails
- CFM-4407 NIFI-13971 - (CVE Fix) Correct Parameter Context Logging on Flow Synchronization
- CFM-4382 NIFI-13927 - PublishGCPPubSub processor stop working and is stucked when using Record Oriented mode
- CFM-4552 NIFI-13971 - Correct Parameter Context Logging on Flow Synchronization
- CFM-4435 NIFI-14010 - (CVE fix) Upgrade Netty to 4.1.115 and Bouncy Castle to 1.79
- CFM-4596 NIFI-14158 - DeleteHDFS processor ignores delete action return value
- CFM-5459 NIFI-14277 - Add credential scope in GCP PubSub and BigQuery processors
- CFM-5463 NIFI-14287 - ExportSizeLimitExceeded error in FetchGoogleDrive
- CFM-5462 NIFI-14281 - SocketTimeoutException in FetchGoogleDrive
- CFM-5461 NIFI-14280 - NullPointerException in ListGoogleDrive
- CFM-5460 NIFI-14278 - Fix custom Storage API URL in PutGCSObject
- CFM-5482 NIFI-14376 - ListGoogleDrive fails to list Shared Drive subfolder

### **NiFi 2.3.0 with Cloudera Flow Management 4.2.1.400**

Cloudera Flow Management 4.2.1.400 is based on Apache NiFi 2.3.0. It includes all fixed issues of this Apache NiFi release.

### **7.3.1.0**

#### **NiFi 1.28.1 with Cloudera Flow Management 2.2.9.0**

Cloudera Flow Management 2.2.9.0 is based on Apache NiFi 1.28.1. It includes all fixed issues of this Apache NiFi release, as well as the following additional fixes:

- CFM-3673 Removed deprecation notice from downstream InvokeGRPC
- CFM-3673 Re-added InvokeGRPC processor to downstream
- CFM-3681 Be able to use downstream solr version
- CFM-3726 Added deprecation notice in GRPC processors
- CFM-3768 Exclude aws libs and put those into the common lib directory
- CFM-3775 Update scala-library version to 2.13.12 to mitigate CVE-2022-36944 and Reactor Netty client to 1.0.34 to mitigate CVE-2023-34062
- CFM-3775 Update org.json:json version to 20231013 to mitigate CVE-2023-5072
- CFM-3775 Updated snappy-java to 1.1.10.5 (CVE-2023-43642) excluded bcprov-ext references (CVE-2018-1000180, CVE-2018-1000613) updated snakeyaml to 2.2 (CVE-2017-18640)
- CFM-3775 Remove support for old Postgres versions, update Postgres to...
- CFM-3819 Change logging for debezium unit tests
- CFM-3846 Add missing Ozone dependency to PutIcebergCDC processor
- CFM-4125 CFM-4067 Bump postgresql driver version in debezium-connector-postgres lib to mitigate CVE-2024-1597
- CFM-4155 NIFI-13836 Dependency upgrades
- CFM-4201 Exclude log4j-slf4j-impl
- CFM-4289 NIFI-13720 Component is not reloaded when the isolation key depends on service property
- CFM-4289 NIFI-13722 Kerberos ticket renewal issue due static thread pool in Iceberg library
- CFM-4374 update azure-identity
- CFM-4411 Fix CVEs in bouncycastle
- CFM-4412 Fix CVE-2024-32888 in redshift
- CFM-4434 Use symlinks in nars for jar/war files (part 1+2)
- NIFI-1931 Add auto commit property to QueryDatabaseTable and QueryDatabaseTable processors to allow disabling auto commit so PostgreSQL Fetch Size will work
- NIFI-6379 Added SSL Context to PutSNS, DeleteSQS, GetSQS, and PutSQS

- NIFI-9677 Fixed issue that an empty JSON array causes flow file to be considered unmatched even though it should be considered as a match.
- NIFI-13993 Upgraded Netty to 4.1.115.Final
- NIFI-13993 Upgraded aws-java-sdk to 1.12.778 along with others
- NIFI-13993 Updated Docker version to 1.28.1
- NIFI-13993 Update Apache parent pom version to 33
- NIFI-13993 Removing pom versionsBackup files
- NIFI-13993 Set project version to 1.28.1-SNAPSHOT
- NIFI-13988 Adjusted Record number conversion to treat empty String as null
- NIFI-13963 Default to Drop Unknown Fields in JSON Record Reader
- NIFI-13991 Fix GetAwsTexttractJobStatus so that a ProvisionedThroughputExceededException properly sends the flowfile to the "throttled" relationship.
- NIFI-13994 Upgraded Jackson to 2.18.1 along with others
- NIFI-13922 Fixed SplitExcel to use the evaluated formula value for cell
- NIFI-13970 Added DISCLAIMER regarding NiFi version 1 to assemblies
- NIFI-13930 PutAzureDataLakeStorage sets close flag on file write so that Azure can emit FlushWithClose event
- NIFI-13927 Use synchronized lists in PublishGCPubSub
- NIFI-13897-RC1 prepare release nifi-1.28.0-RC1
- NIFI-13971 Removed Parameter Context debug logging in Flow Synchronizer
- NIFI-13633 Set JsonRecordSetWriter.AllowScientificNotation default value to 'true' on 1.x line in order to be backward compatible
- NIFI-13755 Improved Controller Service Enabling Process
- NIFI-13860 Avoid Throwing Exceptions for Failures in IPLookupService
- NIFI-13744 Corrected Excel Reader Cell Type Inferencing
- NIFI-13798 Renamed Airtable's API Key property to Personal Access Token and updated docs due to API Keys deprecation
- NIFI-13404 recreate s3Object before returning input stream
- NIFI-13726 Set cell style copy policy to false in order to avoid exceeding the maximum number of cell styles (64000) in a .xlsx Workbook
- NIFI-13335 Added ability for the XMLRecordReader to handle where an array of data has different types.
- NIFI-13842 Fixed truststore/keystore setup in AWS v2 components
- NIFI-13840 Fixed Proxy URL Configuration in AWS v2 components
- NIFI-12532 Ensure that when CommunicateAction completes (exceptionally or otherwise) that it gets removed from the list of all CommunicationActions
- NIFI-13819 Set Row Number and Sheet Name for ExcelReader Exceptions
- NIFI-13776 Updated CopyS3Object to Handle Files over 5 GB
- NIFI-13829 Mitigate false positive reports of MonitorActivity, in case of infrequent Flow Files
- NIFI-13831 Adding inheritance to versioned component synchronizer parameter context synchronization when considering referencing components to restart
- NIFI-13831 Adding inheritance to versioned component synchronizer parameter context synchronization when considering referencing components to restart
- NIFI-13836 Dependency upgrades
- NIFI-13837 QueryRecord changes timezones of Timestamp field
- NIFI-12016 Allow use of compatible NAR bundles when loading flow from cluster connection; when determining what bundles are compatible, consider not just any bundle if it's the only one but also any bundle whose version matches the framework version so that when NiFi is upgraded, it is handled more gracefully.
- NIFI-13549 Resolve ability to go to provenance event from lineage ui when nifi is clustered
- NIFI-13722 Kerberos ticket renewal issue due static thread pool in Iceberg library
- NIFI-13720 Component is not reloaded when the isolation key depends on service property
- NIFI-13763 Fixed HashSet Filtering for DeduplicateRecord
- NIFI-13742 Normalize column names in SelectHiveQL processors

- NIFI-13727 Add DeleteSFTP processor
- NIFI-13543 Backport HttpRecordSink service to NiFi 1.x
- NIFI-13723 Add standalone RecordPath function recordOf
- NIFI-13709 Added more meaningful message when validation fails with non-compliant XML and no schema is provided
- NIFI-13715 Fixed StandardProvenanceEventRecord.hashCode() to sort Parent/Child FlowFiles as equals() does
- NIFI-13620 Resolved MaxWaitTime Issue in QueryCassandra
- NIFI-13686 Make TestListFile.testFilterAge() more resilient to time delays
- NIFI-13692 Catch All Exceptions in ResizeImage
- NIFI-13691 Added Kerberos User Service to KuduLookupService
- NIFI-13690 Upgraded AWS SDK to 2.27.14 and other dependencies
- NIFI-13605 Make AbstractHadoopProcessor.KERBEROS\_USER\_SERVICE public
- NIFI-13675 Fixed Tooltip for Parameter Description
- NIFI-13669 Adding alternative processor suggestion in InvokeAWSGatewayApi deprecation notice
- NIFI-13666 Applied the change to 1.X
- NIFI-13576 Upgrade Iceberg from 1.5.2 to 1.6.0
- NIFI-13430 Added CopyS3Object and GetS3ObjectMetadata
- NIFI-13655 Upgrade 1.x Shared Dependencies including JacksonXML and others
- NIFI-13640 Extract Ranger Solr version to property
- NIFI-13627 Bump azure-sdk-bom to 1.2.25 and msal4j to 1.16.1 for nifi-property-protection-azure
- NIFI-13623 Bump gcp.sdk.version to 26.40.0 for nifi-property-protection-gcp
- NIFI-13621 Upgraded JGit to 5.13.3.202401111512 for CVE-2023-4759
- NIFI-13574 Upgraded Azure SDK BOM from 1.2.23 to 1.2.25
- NIFI-13439 Add performance tracking to ProcessGroupStatus
- NIFI-13439 Add performance tracking to ProcessGroupStatus
- NIFI-12741 Remove write permission requirement for the referenced controller service when changing component property referencing a controller service through parameter
- NIFI-13593 PutIceberg issue with decimal scale
- NIFI-13573 Bump google.libraries.version from 26.37.0 to 26.40.0
- NIFI-13557 Fixed TestExcelSchemaInference for Single Digit Month
- NIFI-13557 Fixed TestSchemaInferenceUtil for Single Digit Month
- NIFI-13557 Corrected Date Time Matcher to support single digit months
- NIFI-13566 Catch Throwable during JettyServer start to ensure any issue during start will exit
- NIFI-13550 Added documentation about the ExcelReader Starting Row Strategy
- NIFI-13542 Added missing Max String Length property for JSON Readers
- NIFI-12491 Added Starting Row Schema Strategy to ExcelReader
- NIFI-13538 Do not include exception details in FlowFile attributes in DeleteFile
- NIFI-13418 Updated ExcelReader to handle spreadsheets with shared formulas
- NIFI-13461 Added DeleteFile Processor
- NIFI-13420 Maintain consistent maxNonHeapBytes for clustered diagnostics
- NIFI-13031 Changed PG StatusSnapshotDTO to use cloned copy for status
- NIFI-13496 Included Hadoop configuration file paths in the classloader isolation key of HDFS processors
- NIFI-13478 Protobuf Reader fails to coerce type of repeated fields
- NIFI-13464 Replaced nifi-deprecation-log with logger in Registry
- NIFI-13304 Added SplitExcel Processor
- NIFI-13422 Use unique instances for ScanRule implementations for QueryNiFiReportingTask
- NIFI-12750 ExecuteStreamCommand incorrectly decodes std error stream
- NIFI-13429 Corrected EncryptContentPGP Packet Detection
- NIFI-13415 Deprecated Property Protection and Encrypt Config
- NIFI-10666 PrometheusReportingTask needs to use UTF-8 (not jvm default charset) for /metrics endpoint

- NIFI-13400 ensure container has time to startup before interacting with smb
- NIFI-12231 FetchSmb supports Move and Delete Completion Strategies
- NIFI-13356 Fixed ProtobufReader handling of repeated fields
- NIFI-13397 Updated PutDatabaseRecord to retry transient ProcessException causes
- NIFI-13411 Upgraded Spring to 5.3.37 along with other common dependencies
- NIFI-12704 Avoid NPE in escapeJson() for Root Path
- NIFI-13340 Fixed a bug in which an Output Port can leave a Process Group's DataValve open for output, but then the last FlowFile is terminated instead of going to an Output Port, ultimately resulting in the DataValve remaining open indefinitely. Now, this will be detected and the valve will be closed.
- NIFI-13379 Replaced use of deprecated com.nimbusds.oauth2.sdk.http.HTTPResponse method getContentAsJsonObject with API suggested replacement getBodyAsJsonObject.
- NIFI-13359 Tune ExecuteSQL/Record to create fewer transient flow files
- NIFI-13374 Fix tooltip for Parameter Context in new Process Group dialog
- NIFI-13030 Adding endpoint for comparing versions of registered flows (1.x support)
- NIFI-13323 Removed instantiation of Object arrays for log arguments
- NIFI-13315 Fixed ListAzureBlobStorage\_v12 fails when Record Writer is used
- NIFI-13298 Removed unused instantiated java.util.HashSet from RouteAttribute
- NIFI-13294 Deprecated Apache Knox SSO Authentication
- NIFI-13296 Deprecated Kerberos SPNEGO Authentication
- NIFI-11658 Streamline using single Parameter Context for nested PGs
- NIFI-12896 Added Endpoint Override URL property for PutSNS Processor
- NIFI-12669 Fixed ByteArrayOutputStream.toString() for Java 8 in EvaluateXQuery
- NIFI-12669 Fix EvaluateXQuery processor which incorrectly encodes result attributes in certain case
- NIFI-13072 Fix MonitorActivity problems with cluster scope flow monitoring
- NIFI-13156 Replaced JsonParser deprecated getCurrentName() with currentName()
- NIFI-13208 Increased PropertyDescriptor visibility in HadoopDBCPCConnectionPool
- NIFI-13172 Deprecated nifi-kafka-connect components
- NIFI-13201 Deprecated Accumulo Components for Removal
- NIFI-13191 Deprecated InvokeAWSGatewayApi
- NIFI-13181 Updated msal4j to 1.15.0
- NIFI-13151 Deprecated Couchbase Components
- NIFI-13152 Deprecated DataDogReportingTask
- NIFI-13133 RC1 prepare for next development iteration
- NIFI-13133 RC1 prepare release nifi-1.26.0-RC1
- NIFI-12231 FetchSmb supports Move and Delete Completion Strategies
- NIFI-13137 Switch to Zulu for MacOS Java 8 action
- NIFI-13008 CLI command to upgrade all instances of a versioned flow
- NIFI-13133 RC1 prepare for next development iteration
- NIFI-13133 RC1 prepare release nifi-1.26.0-RC1
- NIFI-13131 dependency updates
- NIFI-13121 Handle runtime exceptions in FetchHDFS
- NIFI-13103 Make AutoCommit default to no value set in PutDatabaseRecord
- NIFI-13006 Deprecated nifi-solr-bundle components
- NIFI-12960 Corrected default Protection Type in ExcelReader
- NIFI-12960 Support reading password-protected files in ExcelReader
- NIFI-13084 Backport Allow disabling scientific notation when writing JSON (NIFI-12697)
- NIFI-13090 Backport Improve handling of embedded JSON records (NIFI-12480)
- NIFI-12858 Corrected Order of Previous Property Values
- NIFI-13069 Deprecated ConvertAvroToJSON
- NIFI-12923 Added append avro mode to PutHDFS

- NIFI-13070 Upgraded Netty from 4.1.108 to 4.1.109
- NIFI-13066 Upgraded Bouncy Castle from 1.77 to 1.78.1
- NIFI-13064 Upgrade commons-configuration2 to 2.10.1 for Registry
- NIFI-12993 Add auto commit feature and add batch processing for the sql stmt type
- NIFI-13049 Upgraded nimbus-jose-jwt to 9.37.3 for Registry and Toolkit
- NIFI-13052 allow CRON driven components to be searchable
- NIFI-13046 Upgrade Solr dependencies to 8.11.3
- NIFI-13037 Upgraded Spring Framework from 5.3.31 to 5.3.34
- NIFI-13040 Upgraded Commons IO from 2.15.1 to 2.16.1
- NIFI-13025 Removed custom validation from NifiRegistryFlowRegistryClient
- NIFI-12890 Refactor HadoopDBCPCConnectionPool to extend AbstractDBCPCConnectionPool
- NIFI-12614 Create record reader service for Protobuf messages (1.x version)
- NIFI-12889 Retry Kerberos login on auth failure in HDFS processors
- NIFI-12837 Fix checkstyle issue following a manual cherrypick
- NIFI-13010 Fix UpdateDatabaseTable to work with DBCPCConnectionPoolLookup
- NIFI-12837 Added DFS support in SMB processors
- NIFI-13012 Upgraded Apache Tika from 2.9.1 to 2.9.2
- NIFI-12984 Bump Snowflake Ingest SDK to 2.1.0
- NIFI-12918 Corrected Nested Versioned Flows for Stateless
- NIFI-13002 Restored zstd Compression Level in CompressContent
- NIFI-12996 Moved zstd-jni to standard-shared-nar
- NIFI-12969 Fixed a typo in the #isTempDestinationNecessary method, where we were comparing existingConnection.getProcessGroup() to newDestination.getProcessGroup(), instead of comparing existingConnection.getDestination().getProcessGroup() to newDestination.getProcessGroup(). I.e., we were comparing the Destination's PG to the Connection's PG instead of comparing Destination's PG to Destination's PG. This resulted in using a temporary funnel when we shouldn't. And as a result it attempted to change a Connection's Destination while the destination was running.
- NIFI-12987 allow controller service type to be searchable
- NIFI-12980 Deprecated Hive 3 Components for Removal
- NIFI-12979 Upgraded Kotlin from 1.9.10 to 1.9.23
- NIFI-12966 Upgraded Netty from 4.1.106 to 4.1.108
- NIFI-12900 Avoid unnecessary file listing in PutSFTP
- NIFI-12957 Upgraded Azure SDK BOM from 1.2.19 to 1.2.21
- NIFI-12939 Retry Kerberos login on authentication failure in Iceberg processors
- NIFI-12888 In AbstractEmailProcessor check for expired oauth2 token when determining whether mail receiver needs to be recreated.
- NIFI-12954 Upgraded AWS BOM from 2.23.3 to 2.25.16
- NIFI-12949 Upgraded Box SDK from 4.6.1 to 4.8.0
- NIFI-12925 Updated ListenHTTP to return 405 for TRACE and OPTIONS
- NIFI-12930 Catch FlowFileAccessException in FetchFile
- NIFI-12887 Added Binary String Format property to PutDatabaseRecord
- NIFI-12943 Upgraded Hadoop from 3.3.6 to 3.4.0
- NIFI-12947 Upgraded MIME4J to 0.8.11
- NIFI-12936 ListGCSBucket resets its tracking state after configuration change
- NIFI-12944 Add PeerAddress as Attribute into the flowfile
- NIFI-12928 Added Simple Write strategy in PutAzureDataLakeStorage
- NIFI-12929 Fix logout infinite redirect loop in case of Knox
- NIFI-12895 Added Timeout property to GetSmbFile and PutSmbFile
- NIFI-12938 Upgrade Iceberg from 1.4.3 to 1.5.0
- NIFI-12931 Upgraded Commons Configuration from 2.9.0 to 2.10.1
- NIFI-12926 Upgraded Jackson from 2.16.2 to 2.17.0

- NIFI-12503 Render from-data in swagger.json and RestAPI docs correctly
- NIFI-12919 Deprecated Cassandra 3 Components for Removal
- NIFI-12877 Added Sensitive Dynamic Property Support to RestLookupService
- NIFI-12911 Upgraded Jagged from 0.3.0 to 0.3.2
- NIFI-12909 Upgraded Nimbus JOSE+JWT from 9.33.0 to 9.37.3
- NIFI-12700 refactored PutKudu to optimize memory handling for AUTO\_FLUSH\_SYNC flush mode (unbatched flush)
- NIFI-12901 Removed time units in description of time period properties
- NIFI-12906 Upgraded ZooKeeper from 3.9.1 to 3.9.2
- NIFI-12886 Upgraded Jackson JSON from 2.16.1 to 2.16.2
- NIFI-12840 Expose REMOTE\_POLL\_BATCH\_SIZE property for ListSFTP
- NIFI-12871 Upgraded Commons Compress from 1.25.0 to 1.26.1
- NIFI-12879 Upgraded Clojure from 1.11.1 to 1.11.2
- NIFI-12785 Corrected InvokeHTTP URL handling to avoid double encoding
- NIFI-12876 Upgraded Surefire Plugin from 3.1.2 to 3.2.5
- NIFI-12874 Upgraded Log4j from 2.20.0 to 2.23.0
- NIFI-12868 Upgraded Commons DBCP from 2.11.0 to 2.12.0
- NIFI-12860 Fixed NPE in ExtensionMetadata Constructor
- NIFI-12846 Fixed Region handling for AWS Assume Role Credentials with VPCE Endpoint URL
- NIFI-12645 Fix to correctly invoke onStopped method of scripted processor
- NIFI-12850 Prevent indexing of overly large filename attribute
- NIFI-12828 Added Mapping for BIT type to return INT and handled boolean case for postgres
- NIFI-12851 ConsumeKafka, remove limitation on count of subscribed topics
- NIFI-12847 Add Enum data type handling to Iceberg record converter
- NIFI-12843 Fix incorrect read of parquet data, when record.count is inherited
- NIFI-12843 Fix incorrect read of parquet data, when record.count is inherited
- NIFI-12839 Explicitly set nifiVersion for processor bundle archetype dependencies
- NIFI-11859 Ensure Hazelcast can not join a network when Cluster is NONE
- NIFI-12835 Upgraded node-ip from 1.1.8 to 1.1.9 for Registry
- NIFI-12826 Added timing lag in TestFTP method for improved stability
- NIFI-12784 Set Path Not Found as a Dependent Property in EvaluateJsonPath
- NIFI-12827 Upgraded PostgreSQL JDBC test driver from 42.6.0 to 42.7.2
- NIFI-12232 Corrected Group Component ID Handling for Clustered Flows
- NIFI-12818 Deprecated ReportLineageToAtlas for Removal
- NIFI-12808 Upgraded Commons Codec from 1.16.0 to 1.16.1
- NIFI-12810 Upgraded SLF4J from 2.0.11 to 2.0.12
- NIFI-12725 Upgraded json-schema-validator from 1.1.0 to 1.3.2
- NIFI-12789 fix broken link in couchbase additional details
- NIFI-12770 Deprecated Ranger Authorizers for Removal
- NIFI-12792 Deprecated nifi-spark-bundle components for removal
- NIFI-12777 Add support for UUID record field type in QueryRecord processor
- NIFI-12779 Upgraded Okio from 3.7.0 to 3.8.0
- NIFI-12769 Updated copyright year to 2024 in NOTICE file headers
- NIFI-12680 Fixed JAR for DefaultedDynamicClassPathModificationIT
- NIFI-12745 Fix AvroReader silently dropping malformed records
- NIFI-12749 Handled Forward Slash in Flow Name for nifi-toolkit-cli
- NIFI-12732 ListS3 resets its tracking state after configuration change
- NIFI-12728 Upgraded brotli4j from 1.13.0 to 1.16.0
- NIFI-12729 Upgraded unboundid-ldapsdk from 6.0.10 to 6.0.11
- NIFI-12730 Upgraded Spring Integration from 5.5.18 to 5.5.20

- NIFI-12441 Added No Tracking Strategy to ListS3
- NIFI-12726 Update commons-email to 1.6.0
- NIFI-12731 Ensure state is updated in GetHBase whenever the session is committed
- NIFI-12715 Updated Snowflake SDKs
- NIFI-12719 Upgraded metrics-core from 4.2.22 to 4.2.25
- NIFI-12718 Upgraded greenmail from 1.6.14 to 1.6.15
- NIFI-12717 Upgraded Gremlin from 3.7.0 to 3.7.1
- NIFI-12713 Upgraded mysql-binlog-connector from 0.28.3 to 0.29.0
- NIFI-12695 Enabled PKCE Support for OIDC Integration
- NIFI-12699 Set timeout to 10 seconds for TestStandardFlowFileQueue.testListFlowFilesResultsLimitedCollection
- NIFI-12691 Update okio to 3.7.0
- NIFI-12706 Update reactor-test to 3.5.14
- NIFI-12688 Upgrade mysql-connector-j to 8.3.0
- NIFI-12705 Update metrics-jvm to 4.2.25
- NIFI-12692 Update jline.version to 3.25.1
- NIFI-12690 Upgraded opentelemetry-proto from 1.0.0 to 1.1.0
- NIFI-12689 Upgraded Testcontainers from 1.19.3 to 1.19.4
- NIFI-12682 Fix MiNiFi agent manifest hash swaps
- NIFI-12677 Removed documentation of non-existent strategy for ExcelReader
- NIFI-12500 Add dynamic target for Get/Set/SendTrapSNMP

#### **NiFi 2.0.0 with Cloudera Flow Management 4.2.1.0**

Cloudera Flow Management 4.2.1.0 is based on Apache NiFi 2.0.0-M2. It includes all fixed issues of this Apache NiFi release, as well as the following additional fixes:

- CFM-2279 Added nifi-cdf-flow-analysis-rules-bundle and multiple rule implementations.
- CFM-2663 Add Kite module back - Revert "NIFI-9591 Removed nifi-kite-bundle" - Set NiFi version to 1.16.0
- CFM-2663 Add NARs shipped in CFM to nifi-assembly
- CFM-2900 Fix build error, use com.google.cloud.bigdataoss downstream versions There was a StackOverFlow error, because ozone-common downstream lib transitive dependency had a collision with com.google.cloud.bigdataoss dependencies. As turned out, com.google.cloud.bigdataoss has downstream versions as well, which are compatible with other downstream libs, so we have to use that one - CFM-2900: Fix atlas downstream errors Extend interface with new method + remove EmbeddedKafka as no longer support that way - CFM-2925: Be compatible with hive downstream library - CFM-2946: Ignoring failing Atlas tests, which work downstream - CFM-2946: Updating hive3 version to downstream version 3.1.3000.7.2.16.0-147 - CDPDFX-5826 - fix log4j exclusion - Removing log4j dependencies pulled in with hadoop version 3.1.1.7.2.16.0-171 - Adding more log4j exclusions for hadoop-cloud-storage - CDPDFX-6081 Resolve issues after rebase to 1.18.0 - CDPDFX-6081 Fix more versions after rebase
- CFM-2946 Updating version to 1.17.0, and excluding log4j from kite bundle
- CFM-3219 Downstream library derby lib scope is runtime which cause build failure, exclude extra org.slf4j:slf4j-reload4j dependencies come from downstream libs
- CFM-3219 exclude extra org.slf4j:slf4j-reload4j dependencies come from downstream libs
- CFM-3680 fix dependency collision issue
- CFM-3686 Fix snappy cve error + add some rules from apache nifi
- CFM-3719 Fix hive-jdbc version in nifi-hive-bundle
- CFM-3742 Exclude orc-core from hive3
- CFM-3761 In nifi-cdf-flow-analysis-rules adding RestrictThreadPoolSize in src/main/resources/META-INF/services/org.apache.nifi.flowanalysis.FlowAnalysisRule.
- CFM-3775 Remove support for old Postgres versions, update Postgres to...
- CFM-4140 NIFI-13675 Fixed Tooltip for Parameter Description
- CFM-4273 Fix concurrent issues - not use virtual thread for this release



- CFM-4360 Create configured python temp directory if it does not exists
- CFM-4360 Add python package install tempdir NiFi property
- CFM-4411 Fix CVEs in bouncycastle
- CFM-4412 Fix CVE-2024-32888 in redshift
- CFM-4374 update spring-core to latest 6.0.x
- CDPDFX-2216 Use the SAN from X509Certificate instead of CN
- CDPDFX-2216 Refactored methods incompatible with Java 8
- CDPDFX-2216 Corrected character set usage and dependency declaration
- CDPDFX-2216 Added HeaderX509CertificateExtractor
- CDPDFX-2216 Switch to using StringUtils.isNotBlank
- CDPDFX-2216 Add dev mode groups to anonymous user when dev mode is enabled
- CDPDFX-2470 Ensure anonymous user has idp groups populated - Remove RunOnceIT - CDPDFX-2597: Appending the context path to any entry in localStorage.
- CDPDFX-4049 Initial setup of FlowDesignerClient and Redis implementation
- CDPDFX-6081 More fixes after rebase to 1.18.0 - CDPDFX-6081 Fix ignite NAR version to 1.18.0 - CDPDFX-6081 Fix versions in nifi-hashicorp-vault poms - CDPDFX-6081 Use regular gcs version in poms, RE scripts will do version replacement - CDPDFX-6081 Add addition log4j 2 exclusions to ranger poms
- CDPDFX-6130 Adding null check before resolving sensitive parameter
- CDPDFX-6285 Use flow designer id in flow control event handler
- CDPDFX-6285 Passing through flow designer id during component creation
- CDPDFX-6813 Add more bouncycastle exclusions, fix poms
- CDPDFX-6813 Fix use of flow comparator in standard flow change event handler
- CDPDFX-6813 Fix pom versions to 1.20.0 after rebase - CFM Fix problems, caused by downstream library differences - CFM: Some reason releng build is unable to resolve guava from asana dep pom.xml [ERROR] Failed to execute goal on project nifi-asana-services-api: Could not resolve dependencies for project org.apache.nifi:nifi-asana-services-api:jar:1.18.0.2.1.6.0-33: Failed to collect dependencies at com.asana:asana:jar:1.0.0 -> com.google.guava:guava:jar:[31.1,): No versions available for com.google.guava:guava:jar:[31.1,) within specified range -> [Help 1] - CFM Fix asana guava dependency resolving - CFM - Fix guava resolving problem Some reason downstream library is not able to resolve guava version, when version is defined with [31.1,) format
- CDPDFX-6813 Remove duplicate NAR in assembly - CDPDFX-6831 Fix publishing of controller service bulletins
- CDPDFX-6848 Ensuring that all child components are included in each published BulletinGroup. This separation allows them to not be pruned when there are many other descendent components.
- CDPDFX-6854 Correcting process group ID mapping in flow bulletin task
- CDPDFX-7171 Correcting resolution of VCI for scheduled state changes
- CDPDFX-7192 Add lastModifier field to FlowChangeEvent, set lastModifier when publishing state change events
- CDPDFX-7379 Use identity state lookup when determining affected components for a PG sync, Ensure VCI is set back to original value if an exception is encountered during pg sync, Updated StandardScheduledStateListener to use VCI from the passed in FD PG
- CDPDFX-7395 Resolving bug in bulk actions as applied to nested groups
- CDPDFX-7445 Adding requestId to FlowChangeEvent
- CDPDFX-7517 Fixing NPE when publishing flow change events for bulk actions, publishing events on initially stopping components
- CDPDFX-7757 Fixes after rebasing to Apache 1.23.2
- CDPDFX-7815 Fix atlas dependencies for upgrading to CDP 7.2.17 dependencies
- CDPDFX-8320 Removing nifi-pulsar-nar, which was not compatible with 2.X
- CDPDFX-8348 Updating NiFiRecordSerDe for NIFI-9458 changes
- CDPDFX-8350 Adding python extensions autoloading
- CDPDFX-8623 Updating snakeyaml and postgresql versions
- CDPDFX-8623 Updating COS API version, Upgrading dependencies for CVEs, removing Kite, Upgrading dependencies for CVEs, Remediating Heimdall findings

- CDPDFX-8804 Adding Swagger 1 annotations to domain model
- CDPDFX-8878 Adding support for createConfigurationContext in StandardFlowChangeEventHandler - This allows Flow Designer to continue to work with recent changes from 2.0.0-M2
- CDPDFX-9164 Fixing FlowDesignerEventIT
- CDPDFX-9174 Correcting loadBalancerStrategy enum on VersionedConnection
- CDPDFX-9263 extend manifest generation to add dependencies for property descriptors in case of python processors
- NIFI-10976 Added Previous Cluster State Provider configuration
- NIFI-13722 Kerberos ticket renewal issue due static thread pool in Iceberg library
- NIFI-13593 PutIceberg issue with decimal scale
- NIFI-12939 Retry Kerberos login on authentication failure in Iceberg processors
- NIFI-12697 Allow disabling scientific notation when writing JSON
- NIFI-13872 extend manifest generation to add dependencies for property descriptors in case of python processors
- NIFI-13971 Removed Parameter Context debug logging in Flow Synchronizer
- NIFI-13831 Adding inheritance to versioned component synchronizer parameter context synchronization when considering referencing components to restart
- NIFI-13831 Adding inheritance to versioned component synchronizer parameter context synchronization when considering referencing components to restart
- NIFI-13184 mark invalid processors to start once they become valid
- NIFI-12755 Upgraded Jetty from 12.0.5 to 12.0.6
- NIFI-13365 Fix unit tests running in kubernetes pod
- NIFI-13604 Python Source processors can be triggered without creating new FlowFiles
- NIFI-13528 fixed Python processor's customValidate function
- NIFI-13427 Added FlowFileSource interface for Python Processors
- NIFI-13396 Added Python constant Allowable Values to Manifests
- NIFI-13394 Check candidate directory for python command
- NIFI-13324 Set FlowFile attributes for Python Processors on failure
- NIFI-13433 Fixed PutChroma handling of list values in properties
- NIFI-13118 Add LANGUAGE to property\_descriptor list
- NIFI-13042 Support Python 3.12 for Python Processors
- NIFI-12970 Generate documentation for Python Processors
- NIFI-12740 Fixed Python to Java Object Binding
- NIFI-12959 Support loading Python processors from NARs
- NIFI-12913 Corrected NPE for Python Log Listener ID
- NIFI-12514 Added Windows support for Python venv
- NIFI-12884 Corrected documentation for python debugging
- NIFI-11443 Route Python Framework Logging to SLF4J
- NIFI-12791 Added pillow-heif to ParseDocument Processor
- NIFI-12232 Corrected Group Component ID Handling for Clustered Flows
- NIFI-12693 Moved notification of python process that a Processor was removed to a background (virtual) thread. Also noted in testing that in one instance a Python Processor never became valid because it had cached property descriptors before the processor was fully initialized, so updated code to ensure that we do not cache values before initialization is completed.
- NIFI-12659 Respawn Python Processes on Unexpected Termination
- NIFI-12675 Fixed custom Relationships with Python Processors
- NIFI-12757 Issue GC commands to Python for FlowFileTransformResults and RecordTransformResults when no longer needed on Java side
- NIFI-12739 Import ProcessPoolExecutor to fix bug in python 3.9+
- NIFI-12740 Fixed Threading Bug with Java to Python Bound Objects
- NIFI-12675 Fixed custom Relationships with Python Processors
- NIFI-12616 Added Processor Documentation Support for Python

- NIFI-12591 Upgraded from Swagger Annotations 1.6.12 to 2.2.20
- NIFI-12766 Fixed Region handling for AWS Assume Role Credentials
- NIFI-12636 Upgrade dependencies for Pinecone, ChromaDB and OpenAI processors
- NIFI-12676 Fixed Servlet Registration in HandleHttpRequest
- NIFI-12394 Fixed Service references for Migrated Configurations
- NIFI-11739 Add ability to ignore missing fields in PutIceberg
- NIFI-11177 Add defensive code for null values for Iceberg
- NIFI-11716 Backported nifi-schema-inference-utils from NIFI-11241
- NIFI-11907 Backport from NIFI-11241 for JSON Schema Inference
- NIFI-11916 Iceberg processor extensibility improvement
- NIFI-11334 Fixed PutIceberg processor instance interference due to same class loader usage
- NIFI-11817 Improve ListHDFS extensibility
- NIFI-11178 Improve ListHDFS performance, incremental loading refactor.
- NIFI-10976 Replaced toUnmodifiableList() with toList() for Java 8
- NIFI-10975 Add Kubernetes Leader Election and State Provider
- NIFI-12646 Set Python Processor version to 2.0.0-M2
- NIFI-12668 Fix conflict in Registry Git provider with gpg.format=ssh
- NIFI-12664 Removed deprecated DMC in GetHBase
- NIFI-12676 Fixed Servlet Registration in HandleHttpRequest
- NIFI-12611 Introduce the ability to view and clear state for extension types that support state.
- NIFI-12500 Add dynamic target for Get/Set/SendTrapSNMP
- NIFI-12387 Initialize Controller Service Comments with Empty String
- NIFI-12666 Corrected Registry Data Source Configuration
- NIFI-12596 PutIceberg is missing case-insensitive Record type handling in List and Map types
- NIFI-12660 Added missing Filter property to QueryPinecone
- NIFI-12657 Removed MiNiFi C2 Server modules
- NIFI-12656 Upgraded vite for new frontend from 4.5.1 to 4.5.2
- NIFI-12653 Upgraded Spring from 6.0.15 to 6.0.16
- NIFI-12654 Upgraded Netty from 4.1.105 to 4.1.106
- NIFI-12652 Upgraded SLF4J from 2.0.9 to 2.0.11
- NIFI-12650 Upgraded json-path from 2.8.0 to 2.9.0
- NIFI-12506 Added Threading for Status Analytics Retrieval
- NIFI-12501 Encrypt MiNiFi bootstrap properties
- NIFI-12554 Moved JoltTransformJSON and JoltTransformRecord to nifi-jolt-nar
- NIFI-12647 Added MultiProcessorUseCase for ListFile/FetchFile together
- NIFI-8606 Added Disable & Configure button to the Controller Services Details dialog
- NIFI-12629 adding metadata filtering to QueryPinecone
- NIFI-12402 Added Wait for Activity to MonitorActivity
- NIFI-12506 Added Threading for Status Analytics Retrieval
- NIFI-12640 Moved servlet-api and jetty-schemas to nifi-jetty-bundle
- NIFI-12634 Ignored Blank Prefix Values in Kubernetes Components
- NIFI-12616 Added Processor Documentation Support for Python
- NIFI-12638 Add Use Case on how to use QueryRecord as a router
- NIFI-12637 Handle migrating Proxy properties for InvokeHTTP
- NIFI-12635 Upgraded Slack client from 1.36.1 to 1.37.0
- NIFI-12631 Upgraded Apache MINA SSHD from 2.11.0 to 2.12.0
- NIFI-12628 Upgraded Netty from 4.1.104 to 4.1.105
- NIFI-12627 Extract nifi-file-transfer from nifi-standard-processors
- NIFI-12613 Renamed asDescribedValue() to asAllowableValue()
- NIFI-11294 Support Component State Checkpoints in ConsumeAzureEventHub

- NIFI-12625 Listed Supported Python Versions in Docs
- NIFI-12619 Fixed Python dependencies in ParseDocument
- NIFI-12590 Added Prefix Properties for Kubernetes Leases and ConfigMaps
- NIFI-12394 Fixed Service references for Migrated Configurations
- NIFI-12604 Empty Queue
- NIFI-12621 Upgraded AWS SDK from 2.20.148 to 2.23.3
- NIFI-12620 Upgraded JLine from 3.24.1 to 3.25.0
- NIFI-8278 Fixed Proxy Service property in Azure Storage processors
- NIFI-12623 Expose ability to fetch User Details in ListenSlack and receive App Mention events
- NIFI-11958 Added PutAzureDataExplorer and StandardKustoQueryService
- NIFI-12596 PutIceberg is missing case-insensitive Record type handling in List and Map types
- NIFI-9458 Replaced SimpleDateFormat with DateTimeFormatter
- NIFI-12615 fix ExpressionChanged error on Counters page. NIFI-12618 Upgraded Azure SDK BOM from 1.2.18 to 1.2.19
- NIFI-11389 Fixed controller services's link to referencing controller
- NIFI-12441 Added No Tracking Strategy to ListS3 NIFI-12593 Added Include all violations property to ValidateCsv
- NIFI-12597 Introducing a common navigation bar across all pages
- NIFI-12612 In asn1 bundle handle OBJECT IDENTIFIER type as string.
- NIFI-12573 Improved support for Enums in PropertyDescriptor.Builder
- NIFI-11288 Add AWS STS dependency for AssumeRoleWithWebIdentity method
- NIFI-8278 Added Credentials Type to ADLSCredentialsControllerService
- NIFI-12608 Add nifi-standard-services-api-nar to Processor Archetype
- NIFI-12610 Corrected default\_value example in Python Developer guide
- NIFI-12594 ListS3 - observe min/max object age when entity state tracking is used
- NIFI-12588 Flow Analysis Rules
- NIFI-12606 Upgrade parent Apache POM to version 31
- NIFI-12607 remove kernel 2.6 TIMED\_WAIT documentation
- NIFI-12589 Queue Listing
- NIFI-12561 Fixed MergeContent DELIMITER\_STRATEGY\_NONE Handling
- NIFI-12599 Added READ\_FILESYSTEM Permissions to Lookup Services
- NIFI-12548 Policy Management
- NIFI-12602 Upgraded follow-redirects from 1.15.2 to 1.15.4
- NIFI-12600 Upgraded Apache Maven from 3.9.5 to 3.9.6
- NIFI-12592 Upgraded Apache Curator from 5.5.0 to 5.6.0
- NIFI-12530 Support CREATE TABLE in Oracle database adapters
- NIFI-12591 Upgraded from Swagger Annotations 1.6.12 to 2.2.20
- NIFI-12572 Updated nifi-azure-bundle using current API methods
- NIFI-12434 Upgraded Registry to Spring Framework 6.1.1
- NIFI-12089 Fix typo in additionalDetails of CSVReader
- NIFI-11583 Removed nifi-ignite-nar module from assembly

## Fixed issues in Edge Management [Technical Preview]

Learn about the fixed issues in Edge Management clusters, the impact or changes to the functionality, and any available workaround.

For Edge Management fixed issues, see the [Cloudera Edge Management documentation](#).

## Fixed Issues in Streams Messaging

Review the list of Streams Messaging issues that are resolved in Cloudera DataFlow for Data Hub 7.3.1.

### Kafka

#### **CDPD-65649: ReplicaAlterLogDirs stuck with Offset mismatch for the future replica**

This is a backported fix, see [KAFKA-9087](#) for more information.

#### **CDPD-66986: Mirrormaker 2 auto.offset.reset=latest not working**

This is a backported fix, see [KAFKA-13988](#) for more information.

#### **OPSAPS-71258: Kafka, Streams Replication Manager, and Streams Messaging Manager cannot process messages compressed with Zstd or Snappy if /tmp is mounted as noexec**

The issue is fixed by using JVM flags that point to a different temporary folder for extracting the native library.

#### **CDPD-71433: Connect logical type null values are not handled in AvroConnectTranslator**

When the time.precision.mode property is set to connect for the Debezium connector, the connect logical types are used and null values are now handled.

#### **OPSAPS-69481: Some Kafka Connect metrics missing from Cloudera Manager due to conflicting definitions**

Cloudera Manager now registers the metrics kafka\_connect\_connector\_task\_metrics\_batch\_size\_avg and kafka\_connect\_connector\_task\_metrics\_batch\_size\_max correctly.

### Schema Registry

#### **OPSAPS-68708: Schema Registry might fail to start if a load balancer address is specified in Ranger**

Schema Registry now always ensures that the address it uses to connect to Ranger ends with a trailing slash (/). As a result, Schema Registry no longer fails to start if Ranger has a load balancer address configured that does not end with a trailing slash.

### Streams Messaging Manager

#### **OPSAPS-71258: Kafka, Streams Replication Manager, and Streams Messaging Manager cannot process messages compressed with Zstd or Snappy if /tmp is mounted as noexec**

The issue is fixed by using JVM flags that point to a different temporary folder for extracting the native library.

#### **CDPD-72543: Security headers are not set for static files in Streams Messaging Manager**

Streams Messaging Manager now applies the following security-related headers to static files:

- Content-Security-Policy
- X-XSS-PROTECTION
- X-Content-Type-Options
- X-Frame-Options
- Strict-Transport-Security

#### **CDPD-73643: Unused CM\_USER parameter is visible in /cm-configs internal endpoint**

The unused CM\_USER field has been removed from the /cm-configs internal endpoint

#### **CDPD-70313: KNOX does not send Authentication header on FIPS configuration**

KNOX now sends the Authentication header on FIPS clusters.

### Streams Replication Manager

#### **OPSAPS-71258: Kafka, Streams Replication Manager, and Streams Messaging Manager cannot process messages compressed with Zstd or Snappy if /tmp is mounted as noexec**

The issue is fixed by using JVM flags that point to a different temporary folder for extracting the native library.

### **Cruise Control**

#### **OPSAPS-69978: Cruise Control capacity.py script fails on Python 3**

The script querying the capacity information is now fully compatible with Python 3.

## **Fixed Issues in Cloudera Streaming Analytics**

Review the list of Cloudera Streaming Analytics issues that are resolved in Cloudera DataFlow for Cloudera Data Hub 7.3.1.

**CSA-5423 - Extend Cloudera SQL Stream Builder diag bundle data points**

**CSA-5440 - Permit Spring Flyway plugin execution on PvC**

**CSA-5364 - Add number of topics/tables to successful data source validation message on UI**

**CSA-5306 - Cloudera SQL Stream Builder API does not validate catalog type**

**CSA-5362 - Update "ssb-sse" ASCII text banner to not contain special characters**

**CSA-5359 - Improve error message when creating a JS UDF with a Java version that doesn't support it**

**CSA-5296 - Samples table fields are limited to 32 characters in mysql and oracle dbs**

**CSA-5324 - Cloudera SQL Stream Builder default admin does not have admin privileges**

**CSA-5428 - Sampling renders null as Invalid Number in some cases**

**CSA-5474 - Cloudera SQL Stream Builder can't execute any jobs due to permission issue in the artifacts directory**

**CSA-5475 - Local-kafka connector template not showing in Cloudera SQL Stream Builder**

**CSA-5479 - Using Temp View based on kudu lookup table leaks eventpolls**

**CSA-5499 - Bump Avro to 1.11.4 in parcel to mitigate CVE-2024-47561**

**Cannot submit SQL jobs with UDF JARs when checkpoints enabled**

**CSA-5048 - Generate correct default log configuration for Cloudera SQL Stream Builder jobs**

**CSA-5055 - Backport FLINK-20539 to Cloudera Streaming Analytics**

**CSA-5065 - Artifact Storage request thread does not timeout when storage is offline, hanging the UI**

**CSA-5120 - Connector dependencies are missing from Cloudera SQL Stream Builder**

**CSA-5122 - Cloudera SQL Stream Builder keeps reconciling after a failed job**

**CSA-5126 - Undeterministic classloader behavior with flink-metrics-kafka and kafka-connector resulting in job failure**

**CSA-5161 - analyzeQuery returning false validation errors in some cases**

**CSA-5166 - Cloudera SQL Stream Builder Local Kafka data source doesn't work with a user-specified TLS truststore**

**CSA-5199 - Cloudera registry catalog type registry is not compatible with UI**

**CSA-5221 - SsbCatalog uses user session in prod mode**

**CSA-5236 - Implement MetricReporterFactory for KafkaMetricsReporter**

**CSA-5251 - UiConfigController throws NullPointerException**

**CSA-5270 - If custom log is set, the job cannot be saved**

**CSA-5282 - Check Flink job status before submitting Cloudera SQL Stream Builder job**

**CSA-5283 - [Cloudera SQL Stream Builder] Make all overloaded methods transactional in JobService**

**CSA-5303 - Fix Kerberos/SPNEGO authentication for Flink Deployments**

**CSA-5306 - Cloudera SQL Stream Builder API does not validate catalog type**

**CSA-5315 - Load balancer role cannot start**

**FLINK-20539 - Type mismatch when using ROW() in computed column**

## UI fixes and improvements

**CSA-4602 - Changing existing MV filter type can't be saved**

**CSA-5038 - Widget is empty when added to the dashboard before initialization completes**

**CSA-5140 - Fix No Rows To Show message when switching from sampler to MV in dashboard preview**

**CSA-5024 - Polling samples feedback is on even when polling is turned off**

**CSA-5025 - Cursor jumps to the end after the first keystroke when using templates**

**CSA-5026 - Oversize widget cannot be sized down**

**CSA-5294 - Add job save button to job settings component**

## Fixed CVEs in Cloudera DataFlow for Data Hub 7.3.1

Review the list of Common Vulnerabilities and Exposures (CVEs) that are resolved in Cloudera DataFlow for Data Hub 7.3.1.

## CVE-2021-45105 & CVE-2021-44832 remediation for Cloudera DataFlow for Data Hub

Learn more about the CVE-2021-45105 and CVE-2021-44832 remediation for the Flow Management, Streams Messaging and Streaming Analytics cluster templates in Cloudera DataFlow for Data Hub.

On February 1, 2022, Cloudera released a hotfix to Cloudera on cloud Runtime version 7.2.12. It addresses the CVE and other vulnerability concerns as listed below:

- [CVE-2021-45105](#) which affects Apache Log4j2 versions from 2.0-beta9 to 2.16.0, excluding 2.12.3
- [CVE-2021-44832](#) which affects Apache Log4j2 versions from 2.0-alpha7 to 2.17.0, excluding 2.3.2 and 2.12.4

The following table summarizes which template is impacted by the vulnerabilities:

Template	Impacted versions
Flow Management	All versions
Streams Messaging	Not impacted
Streaming Analytics	All versions from 7.2.10

As the Cloudera DataFlow for Data Hub cluster templates are running in the Cloudera on cloud environment powered by Runtime, Cloudera encourages users to upgrade the Cloudera services running Runtime versions from 7.2.7 so

that they include the latest hotfixes. You can update your existing data lakes and data hubs by doing a maintenance upgrade. For more information, see the [Data Lake upgrade](#) and [Data Hub upgrade](#) documentation.



**Note:** Maintenance upgrades are not supported for RAZ-enabled environments.

If you are running a version of Runtime lower than 7.2.7, contact Cloudera Support for details on how to upgrade Runtime.

For more information about the impacts of CVE-2021-45105, see the [TSB 2021-547: Critical vulnerability in log4j2 CVE-2021-45105 Knowledge Base article](#).

## Fixed CVEs in Flow Management

Review the list of Common Vulnerabilities and Exposures (CVEs) fixed in Flow Management in Cloudera DataFlow for Data Hub.

### 7.3.1.400

The following CVEs have been fixed in Flow Management in Cloudera Data Hub 2.2.9.400 and 4.2.1.400.

#### Cloudera Spring Framework

To resolve CVEs reported in the Spring Framework used in Cloudera products, Cloudera now maintains its own internal repository for the Spring Framework and applies necessary patches for identified CVEs. This fork is based on Spring Framework 5 with backports from Spring Framework 6 that inherit the official fixes and community best practices. This approach ensures no breaking changes are introduced into customer environments while adopting best practices from Spring Framework 6 to resolve any vulnerabilities.

In 2.1.7.0, the following critical Spring framework vulnerabilities have been addressed.

CVE#2024#38820: A locale-dependent issue in `String.toLowerCase()` and `toUpperCase()` introduced by the fix for CVE#2022#22968 could cause Spring DataBinder's disallowed fields check to fail. This is resolved by enforcing `Locale.ROOT` for consistent, locale-independent string casing.

CVE#2024#38819 and CVE#2024#38816: These patch path traversal vulnerabilities in Spring's functional web frameworks (`WebMvc.fn`, `WebFlux.fn`), where crafted HTTP requests could access files outside the intended static resource directory. The fix adopts Spring 6's centralized approach: path normalization and decoding are now handled early in the request cycle using a shared utility (`ResourceHandlerUtils`), replacing scattered validations across resource resolvers.

The Cloudera managed Spring library includes additional patches for the following CVEs:

- CVE#2024#38808 (fixed in 5.3.39)
- CVE#2024#38809 (fixed in 5.3.38)
- CVE#2024#22243 (fixed in 5.3.32)
- CVE#2024#38821 (fixed in Spring Security 5.8.16)

Due to the custom JAR naming convention (for example: 5.3.39-cloudera-5.3.44), some vulnerability scans may incorrectly flag the component for known CVEs. These are false positives, as Cloudera has addressed the vulnerabilities internally.

#### **CVE-2024-56128: Apache Kafka: SCRAM authentication vulnerable to replay attacks when used without encryption**

Incorrect Implementation of Authentication Algorithm in Apache Kafka's SCRAM implementation. Issue Summary: Apache Kafka's implementation of the Salted Challenge Response Authentication Mechanism (SCRAM) did not fully adhere to the requirements of RFC 5802 [1]. Specifically, as per RFC 5802, the server must verify that the nonce sent by the client in the second message matches



the nonce sent by the server in its first message. However, Kafka's SCRAM implementation did not perform this validation. Impact: This vulnerability is exploitable only when an attacker has plaintext access to the SCRAM authentication exchange. However, the usage of SCRAM over plaintext is strongly discouraged as it is considered an insecure practice [2]. Apache Kafka recommends deploying SCRAM exclusively with TLS encryption to protect SCRAM exchanges from interception [3]. Deployments using SCRAM with TLS are not affected by this issue. How to Detect If You Are Impacted: If your deployment uses SCRAM authentication over plaintext communication channels (without TLS encryption), you are likely impacted. To check if TLS is enabled, review your `server.properties` configuration file for `listeners` property. If you have `SASL_PLAINTEXT` in the listeners, then you are likely impacted. Fix Details: The issue has been addressed by introducing nonce verification in the final message of the SCRAM authentication exchange to ensure compliance with RFC 5802. Affected Versions: Apache Kafka versions 0.10.2.0 through 3.9.0, excluding the fixed versions below. Fixed Versions: 3.9.0 3.8.1 3.7.2 Users are advised to upgrade to 3.7.2 or later to mitigate this issue. Recommendations for Mitigation: Users unable to upgrade to the fixed versions can mitigate the issue by: - Using TLS with SCRAM Authentication: Always deploy SCRAM over TLS to encrypt authentication exchanges and protect against interception. - Considering Alternative Authentication Mechanisms: Evaluate alternative authentication mechanisms, such as PLAIN, Kerberos or OAuth with TLS, which provide additional layers of security.

#### **CVE-2024-43382:**

Snowflake JDBC driver versions  $\geq 3.2.6$  and  $\leq 3.19.1$  have an Incorrect Security Setting that can result in data being uploaded to an encrypted stage without the additional layer of protection provided by client side encryption.

#### **CVE-2024-41909: Apache MINA SSHD: integrity check bypass**

Like many other SSH implementations, Apache MINA SSHD suffered from the issue that is more widely known as CVE-2023-48795. An attacker that can intercept traffic between client and server could drop certain packets from the stream, potentially causing client and server to consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack The mitigations to prevent this type of attack were implemented in Apache MINA SSHD 2.12.0, both client and server side. Users are recommended to upgrade to at least this version. Note that both the client and the server implementation must have mitigations applied against this issue, otherwise the connection may still be affected.

#### **CVE-2024-38829: Spring LDAP sensitive data exposure for case-sensitive comparisons**

A vulnerability in Spring LDAP allows data exposure for case sensitive comparisons. This issue affects Spring LDAP: from 2.4.0 through 2.4.3, from 3.0.0 through 3.0.9, from 3.1.0 through 3.1.7, from 3.2.0 through 3.2.7, AND all versions prior to 2.4.0. The usage of `String.toLowerCase()` and `String.toUpperCase()` has some Locale dependent exceptions that could potentially result in unintended columns from being queried Related to CVE-2024-38820 <https://spring.io/security/cve-2024-38820>

#### **CVE-2024-38808: CVE-2024-38808: Spring Expression DoS Vulnerability**

In Spring Framework versions 5.3.0 - 5.3.38 and older unsupported versions, it is possible for a user to provide a specially crafted Spring Expression Language (SpEL) expression that may cause a denial of service (DoS) condition. Specifically, an application is vulnerable when the following is true: \* The application evaluates user-supplied SpEL expressions.

#### **CVE-2024-32888: Amazon JDBC Driver for Redshift SQL Injection via line comment generation**

The Amazon JDBC Driver for Redshift is a Type 4 JDBC driver that provides database connectivity through the standard JDBC application program interfaces (APIs) available in the Java Platform, Enterprise Editions. Prior to version 2.1.0.28, SQL injection is possible when using the non-default connection property ``preferQueryMode=simple`` in combination with application code which has a vulnerable SQL that negates a parameter value. There is no vulnerability in the driver when using the default, extended query mode. Note that ``preferQueryMode`` is not a supported

parameter in Redshift JDBC driver, and is inherited code from Postgres JDBC driver. Users who do not override default settings to utilize this unsupported query mode are not affected. This issue is patched in driver version 2.1.0.28. As a workaround, do not use the connection property ``preferQueryMode=simple``. (NOTE: Those who do not explicitly specify a query mode use the default of extended query mode and are not affected by this issue.)

#### **CVE-2024-30172:**

An issue was discovered in Bouncy Castle Java Cryptography APIs before 1.78. An Ed25519 verification code infinite loop can occur via a crafted signature and public key.

#### **CVE-2024-23450: Elasticsearch Uncontrolled Resource Consumption vulnerability**

A flaw was discovered in Elasticsearch, where processing a document in a deeply nested pipeline on an ingest node could cause the Elasticsearch node to crash.

#### **CVE-2024-23444: Elasticsearch elasticsearch-certutil csr fails to encrypt private key**

It was discovered by Elastic engineering that when elasticsearch-certutil CLI tool is used with the csr option in order to create a new Certificate Signing Requests, the associated private key that is generated is stored on disk unencrypted even if the `--pass` parameter is passed in the command invocation.

#### **CVE-2024-22257:**

In Spring Security, versions 5.7.x prior to 5.7.12, 5.8.x prior to 5.8.11, versions 6.0.x prior to 6.0.9, versions 6.1.x prior to 6.1.8, versions 6.2.x prior to 6.2.3, an application is possible vulnerable to broken access control when it directly uses the `AuthenticatedVoter#vote` passing a null Authentication parameter.

#### **CVE-2023-49921:**

An issue was discovered by Elastic whereby Watcher search input logged the search query results on DEBUG log level. This could lead to raw contents of documents stored in Elasticsearch to be printed in logs. Elastic has released 8.11.2 and 7.17.16 that resolves this issue by removing this excessive logging. This issue only affects users that use Watcher and have a Watch defined that uses the search input and additionally have set the search input's logger to DEBUG or finer, for example using: `org.elasticsearch.xpack.watcher.input.search`, `org.elasticsearch.xpack.watcher.input`, `org.elasticsearch.xpack.watcher`, or wider, since the loggers are hierarchical.

#### **CVE-2023-48795:**

The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in `chacha20-poly1305@openssh.com` and (if CBC is used) the `-etm@openssh.com` MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the

mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

#### **CVE-2023-46674: Elasticsearch-hadoop Unsafe Deserialization**

An issue was identified that allowed the unsafe deserialization of java objects from hadoop or spark configuration properties that could have been modified by authenticated users. Elastic would like to thank Yakov Shafranovich, with Amazon Web Services for reporting this issue.

#### **CVE-2023-46673:**

It was identified that malformed scripts used in the script processor of an Ingest Pipeline could cause an Elasticsearch node to crash when calling the Simulate Pipeline API.

#### **CVE-2023-46604: Apache ActiveMQ, Apache ActiveMQ Legacy OpenWire Module: Unbounded deserialization causes ActiveMQ to be vulnerable to a remote code execution (RCE) attack**

The Java OpenWire protocol marshaller is vulnerable to Remote Code Execution. This vulnerability may allow a remote attacker with network access to either a Java-based OpenWire broker or client to run arbitrary shell commands by manipulating serialized class types in the OpenWire protocol to cause either the client or the broker (respectively) to instantiate any class on the classpath. Users are recommended to upgrade both brokers and clients to version 5.15.16, 5.16.7, 5.17.6, or 5.18.3 which fixes this issue.

#### **CVE-2023-46120: RabbitMQ Java client's lack of message size limitation leads to remote DoS attack**

The RabbitMQ Java client library allows Java and JVM-based applications to connect to and interact with RabbitMQ nodes. `maxBodyLebgh` was not used when receiving Message objects. Attackers could send a very large Message causing a memory overflow and triggering an OOM Error. Users of RabbitMQ may suffer from DoS attacks from RabbitMQ Java client which will ultimately exhaust the memory of the consumer. This vulnerability was patched in version 5.18.0.

#### **CVE-2023-34042:**

The spring-security.xsd file inside the spring-security-config jar is world writable which means that if it were extracted it could be written by anyone with access to the file system. While there are no known exploits, this is an example of “CWE-732: Incorrect Permission Assignment for Critical Resource” and could result in an exploit. Users should update to the latest version of Spring Security to mitigate any future exploits found around this issue.

#### **CVE-2023-31419: Elasticsearch StackOverflow vulnerability**

A flaw was discovered in Elasticsearch, affecting the `\_search` API that allowed a specially crafted query string to cause a Stack Overflow and ultimately a Denial of Service.

#### **CVE-2023-31418: Elasticsearch uncontrolled resource consumption**

An issue has been identified with how Elasticsearch handled incoming requests on the HTTP layer. An unauthenticated user could force an Elasticsearch node to exit with an OutOfMemory error by sending a moderate number of malformed HTTP requests. The issue was identified by Elastic Engineering and we have no indication that the issue is known or that it is being exploited in the wild.

#### **CVE-2023-31417: Elasticsearch Insertion of sensitive information in audit logs**

Elasticsearch generally filters out sensitive information and credentials before logging to the audit log. It was found that this filtering was not applied when requests to Elasticsearch use certain deprecated URIs for APIs. The impact of this flaw is that sensitive information such as passwords and tokens might be printed in cleartext in Elasticsearch audit logs. Note that audit logging is disabled by default and needs to be explicitly enabled and even when audit logging is enabled, request bodies that could contain sensitive information are not printed to the audit log unless explicitly configured.

#### **CVE-2022-46751: Apache Ivy: XML External Entity vulnerability in Apache Ivy**

Improper Restriction of XML External Entity Reference, XML Injection (aka Blind XPath Injection) vulnerability in Apache Software Foundation Apache Ivy. This issue affects any version of Apache Ivy prior to 2.5.2. When Apache Ivy prior to 2.5.2 parses XML files - either its own configuration, Ivy files or Apache Maven POMs - it will allow downloading external document type definitions and expand any entity references contained therein when used. This can be used to exfiltrate data, access resources only the machine running Ivy has access to or disturb the execution of Ivy in different ways. Starting with Ivy 2.5.2 DTD processing is disabled by default except when parsing Maven POMs where the default is to allow DTD processing but only to include a DTD snippet shipping with Ivy that is needed to deal with existing Maven POMs that are not valid XML files but are nevertheless accepted by Maven. Access can be made more lenient via newly introduced system properties where needed. Users of Ivy prior to version 2.5.2 can use Java system properties to restrict processing of external DTDs, see the section about "JAXP Properties for External Access restrictions" inside Oracle's "Java API for XML Processing (JAXP) Security Guide".

**CVE-2022-46337: Apache Derby: LDAP injection vulnerability in authenticator**

A cleverly devised username might bypass LDAP authentication checks. In LDAP-authenticated Derby installations, this could let an attacker fill up the disk by creating junk Derby databases. In LDAP-authenticated Derby installations, this could also allow the attacker to execute malware which was visible to and executable by the account which booted the Derby server. In LDAP-protected databases which weren't also protected by SQL GRANT/REVOKE authorization, this vulnerability could also let an attacker view and corrupt sensitive data and run sensitive database functions and procedures. Mitigation: Users should upgrade to Java 21 and Derby 10.17.1.0. Alternatively, users who wish to remain on older Java versions should build their own Derby distribution from one of the release families to which the fix was backported: 10.16, 10.15, and 10.14. Those are the releases which correspond, respectively, with Java LTS versions 17, 11, and 8.

**CVE-2022-41678: Apache ActiveMQ: Insufficient API restrictions on Jolokia allow authenticated users to perform RCE**

Once an user is authenticated on Jolokia, he can potentially trigger arbitrary code execution. In details, in ActiveMQ configurations, jetty allows `org.jolokia.http.AgentServlet` to handle request to `/api/jolokia` `org.jolokia.http.HttpRequestHandler#handlePostRequest` is able to create `JmxRequest` through `JSONObject`. And calls to `org.jolokia.http.HttpRequestHandler#executeRequest`. Into deeper calling stacks, `org.jolokia.handler.ExecHandler#doHandleRequest` can be invoked through reflection. This could lead to RCE through via various mbeans. One example is unrestricted deserialization in `jdk.management.jfr.FlightRecorderMXBeanImpl` which exists on Java version above 11. 1 Call `newRecording`. 2 Call `setConfiguration`. And a webshell data hides in it. 3 Call `startRecording`. 4 Call `copyTo` method. The webshell will be written to a `.jsp` file. The mitigation is to restrict (by default) the actions authorized on Jolokia, or disable Jolokia. A more restrictive Jolokia configuration has been defined in default ActiveMQ distribution. We encourage users to upgrade to ActiveMQ distributions version including updated Jolokia configuration: 5.16.6, 5.17.4, 5.18.0, 6.0.0.

**CVE-2022-37866: Apache Ivy allows path traversal in the presence of a malicious repository**

When Apache Ivy downloads artifacts from a repository it stores them in the local file system based on a user-supplied "pattern" that may include placeholders for artifacts coordinates like the organisation, module or version. If said coordinates contain `"../"` sequences - which are valid characters for Ivy coordinates in general - it is possible the artifacts are stored outside of Ivy's local cache or repository or can overwrite different artifacts inside of the local cache. In order to exploit this vulnerability an attacker needs collaboration by the remote repository as Ivy will issue http requests containing `".."` sequences and a "normal" repository will not interpret them as part of the artifact coordinates. Users of Apache Ivy 2.0.0 to 2.5.1 should upgrade to Ivy 2.5.1.

**CVE-2022-37865: Apache Ivy allows creating/overwriting any file on the system**

With Apache Ivy 2.4.0 an optional packaging attribute has been introduced that allows artifacts to be unpacked on the fly if they used `pack200` or `zip` packaging. For artifacts using the `"zip"`, `"jar"`

or "war" packaging Ivy prior to 2.5.1 doesn't verify the target path when extracting the archive. An archive containing absolute paths or paths that try to traverse "upwards" using ".." sequences can then write files to any location on the local file system that the user executing Ivy has write access to. Ivy users of version 2.4.0 to 2.5.0 should upgrade to Ivy 2.5.1.

**CVE-2021-37937: Elasticsearch privilege escalation**

An issue was found with how API keys are created with the Fleet-Server service account. When an API key is created with a service account, it is possible that the API key could be created with higher privileges than intended. Using this vulnerability, a compromised Fleet-Server service account could escalate themselves to a super-user.

**CVE-2021-25738: Code exec via yaml parsing**

Loading specially-crafted yaml with the Kubernetes Java Client library can lead to code execution.

**CVE-2021-22147:**

Elasticsearch before 7.14.0 did not apply document and field level security to searchable snapshots. This could lead to an authenticated user gaining access to information that they are unauthorized to view.

**CVE-2020-8570: Kubernetes Java client libraries unvalidated path traversal in Copy implementation**

Kubernetes Java client libraries in version 10.0.0 and versions prior to 9.0.1 allow writes to paths outside of the current directory when copying multiple files from a remote pod which sends a maliciously crafted archive. This can potentially overwrite any files on the system of the process executing the client code.

**CVE-2019-10086:**

In Apache Commons Beanutils 1.9.2, a special BeanIntrospector class was added which allows suppressing the ability for an attacker to access the classloader via the class property available on all Java objects. We, however were not using this by default characteristic of the PropertyUtilsBean.

**CVE-2019-0210:**

In Apache Thrift 0.9.3 to 0.12.0, a server implemented in Go using TJSONProtocol or TSimpleJSONProtocol may panic when feed with invalid input data.

**CVE-2018-1337:**

In Apache Directory LDAP API before 1.0.2, a bug in the way the SSL Filter was setup made it possible for another thread to use the connection before the TLS layer has been established, if the connection has already been used and put back in a pool of connections, leading to leaking any information contained in this request (including the credentials when sending a BIND request).

**CVE-2017-15718:**

The YARN NodeManager in Apache Hadoop 2.7.3 and 2.7.4 can leak the password for credential store provider used by the NodeManager to YARN Applications.

**CVE-2017-3166:**

In Apache Hadoop versions 2.6.1 to 2.6.5, 2.7.0 to 2.7.3, and 3.0.0-alpha1, if a file in an encryption zone with access permissions that make it world readable is localized via YARN's localization mechanism, that file will be stored in a world-readable location and can be shared freely with any application that requests to localize that file.

**CVE-2016-5725:**

Directory traversal vulnerability in JCraft JSch before 0.1.54 on Windows, when the mode is ChannelSftp.OVERWRITE, allows remote SFTP servers to write to arbitrary files via a ..\ (dot dot backslash) in a response to a recursive GET command.

**CVE-2015-6420:**



Serialized-object interfaces in certain Cisco Collaboration and Social Media; Endpoint Clients and Client Software; Network Application, Service, and Acceleration; Network and Content Security Devices; Network Management and Provisioning; Routing and Switching - Enterprise and Service Provider; Unified Computing; Voice and Unified Communications Devices; Video, Streaming, TelePresence, and Transcoding Devices; Wireless; and Cisco Hosted Services products allow remote attackers to execute arbitrary commands via a crafted serialized Java object, related to the Apache Commons Collections (ACC) library.

#### **CVE-2014-0114:**

Apache Commons BeanUtils, as distributed in lib/commons-beanutils-1.8.0.jar in Apache Struts 1.x through 1.3.10 and in other products requiring commons-beanutils through 1.9.2, does not suppress the class property, which allows remote attackers to "manipulate" the ClassLoader and execute arbitrary code via the class parameter, as demonstrated by the passing of this parameter to the getClass method of the ActionForm object in Struts 1.

#### **CVE-2024-38816: Path traversal vulnerability in functional web frameworks**

Applications serving static resources through the functional web frameworks WebMvc.fn or WebFlux.fn are vulnerable to path traversal attacks. An attacker can craft malicious HTTP requests and obtain any file on the file system that is also accessible to the process in which the Spring application is running. Specifically, an application is vulnerable when both of the following are true: \* the web application uses RouterFunctions to serve static resources \* resource handling is explicitly configured with a FileSystemResource location However, malicious requests are blocked and rejected when any of the following is true: \* the Spring Security HTTP Firewall <https://docs.spring.io/spring-security/reference/servlet/exploits/firewall.html> is in use \* the application runs on Tomcat or Jetty

#### **CVE-2024-38819:**

Applications serving static resources through the functional web frameworks WebMvc.fn or WebFlux.fn are vulnerable to path traversal attacks. An attacker can craft malicious HTTP requests and obtain any file on the file system that is also accessible to the process in which the Spring application is running.

### **7.3.1.0**

All known NiFi CVEs are addressed in both clusters based on NiFi 1.28.1 and clusters based on NiFi 2.0-M2. See [Apache NiFi Security](#) for more information about NiFi's CVEs.

In Flow Management clusters using NiFi 1.28.1, vulnerability scanners may detect certain CVEs in some legacy components. For these components, it is not possible to update the client library NiFi depends on. You can find the list of affected components below. Although NiFi does not expose ways to exploit those vulnerabilities, you may want to remove the associated NARs. Note that these NARs are deprecated and no longer available in NiFi clusters using NiFi 2.0.0.

- nifi-kite-nar (CVE-2022-42889, CVE-2023-39410)
- nifi-kafka-1-0-nar, nifi-kafka-2-0-nar (CVE-2018-17196)
- nifi-couchbase-nar (CVE-2020-9040)

The following CVEs have been fixed in Flow Management in Data Hub 7.3.1.0.

#### **CVE-2022-40149, CVE-2022-40150**

Those using Jettison to parse untrusted XML or JSON data may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow. This effect may support a denial of service attack.

#### **CVE-2022-45685**

A stack overflow in Jettison before v1.5.2 allows attackers to cause a Denial of Service (DoS) via crafted JSON data.

**CVE-2022-45693**

Jettison before v1.5.2 was discovered to contain a stack overflow via the map parameter. This vulnerability allows attackers to cause a Denial of Service (DoS) via a crafted string.

**CVE-2023-1436**

An infinite recursion is triggered in Jettison when constructing a JSONArray from a Collection that contains a self-reference in one of its elements. This leads to a StackOverflowError exception being thrown.

**CVE-2021-23358**

The package underscore from 1.13.0-0 and before 1.13.0-2, from 1.3.2 and before 1.12.1 are vulnerable to Arbitrary Code Injection via the template function, particularly when a variable property is passed as an argument as it is not sanitized.

**CVE-2024-1597**

pgjdbc, the PostgreSQL JDBC Driver, allows attacker to inject SQL if using PreferQueryMode=SIMPLE. Note this is not the default. In the default mode there is no vulnerability. A placeholder for a numeric value must be immediately preceded by a minus. There must be a second placeholder for a string value after the first placeholder; both must be on the same line. By constructing a matching string payload, the attacker can inject SQL to alter the query, bypassing the protections that parameterized queries bring against SQL Injection attacks. Versions before 42.7.2, 42.6.1, 42.5.5, 42.4.4, 42.3.9, and 42.2.28 are affected.

**CVE-2022-31197**

PostgreSQL JDBC Driver (PgJDBC for short) allows Java programs to connect to a PostgreSQL database using standard, database independent Java code. The PGJDBC implementation of the `java.sql.ResultSet.refreshRow()` method is not performing escaping of column names so a malicious column name that contains a statement terminator, e.g. ``;``, could lead to SQL injection. This could lead to executing additional SQL commands as the application's JDBC user. User applications that do not invoke the `ResultSet.refreshRow()` method are not impacted. User application that do invoke that method are impacted if the underlying database that they are querying via their JDBC application may be under the control of an attacker. The attack requires the attacker to trick the user into executing SQL against a table name who's column names would contain the malicious SQL and subsequently invoke the `refreshRow()` method on the `ResultSet`. Note that the application's JDBC user and the schema owner need not be the same. A JDBC application that executes as a privileged user querying database schemas owned by potentially malicious less-privileged users would be vulnerable. In that situation it may be possible for the malicious user to craft a schema that causes the application to execute commands as the privileged user. Patched versions will be released as `42.2.26` and `42.4.1`. Users are advised to upgrade. There are no known workarounds for this issue.

**CVE-2022-41946**

pgjdbc is an open source postgresql JDBC Driver. In affected versions a prepared statement using either `PreparedStatement.setText(int, InputStream)` or `PreparedStatement.setBytea(int, InputStream)` will create a temporary file if the `InputStream` is larger than 2k. This will create a temporary file which is readable by other users on Unix like systems, but not MacOS. On Unix like systems, the system's temporary directory is shared between all users on that system. Because of this, when files and directories are written into this directory they are, by default, readable by other users on that same system. This vulnerability does not allow other users to overwrite the contents of these directories or files. This is purely an information disclosure vulnerability. Because certain JDK file system APIs were only added in JDK 1.7, this fix is dependent upon the version of the JDK you are using. Java 1.7 and higher users: this vulnerability is fixed in 4.5.0. Java 1.6 and lower users: no patch is available. If you are unable to patch, or are stuck running on Java 1.6, specifying the `java.io.tmpdir` system environment variable to a directory that is exclusively owned by the executing user will mitigate this vulnerability.

**CVE-2022-21724**

pgjdbc is the official PostgreSQL JDBC Driver. A security hole was found in the jdbc driver for postgresql database while doing security research. The system using the postgresql library will be attacked when attacker control the jdbc url or properties. pgjdbc instantiates plugin instances based on class names provided via `authenticationPluginClassName`, `sslhostnameverifier`, `socketFactory`, `sslfactory`, `sslpasswordcallback` connection properties. However, the driver did not verify if the class implements the expected interface before instantiating the class. This can lead to code execution loaded via arbitrary classes. Users using plugins are advised to upgrade. There are no known workarounds for this issue.

**CVE-2018-10936**

A weakness was found in postgresql-jdbc before version 42.2.5. It was possible to provide an SSL Factory and not check the host name if a host name verifier was not provided to the driver. This could lead to a condition where a man-in-the-middle attacker could masquerade as a trusted server by providing a certificate for the wrong host, as long as it was signed by a trusted CA.

**CVE-2020-13692**

PostgreSQL JDBC Driver (aka PgJDBC) before 42.2.13 allows XXE.

**CVE-2022-36944**

Scala 2.13.x before 2.13.9 has a Java deserialization chain in its JAR file. On its own, it cannot be exploited. There is only a risk in conjunction with Java object deserialization within an application. In such situations, it allows attackers to erase contents of arbitrary files, make network connections, or possibly run arbitrary code (specifically, Function0 functions) via a gadget chain.

**CVE-2022-36944**

Scala 2.13.x before 2.13.9 has a Java deserialization chain in its JAR file. On its own, it cannot be exploited. There is only a risk in conjunction with Java object deserialization within an application. In such situations, it allows attackers to erase contents of arbitrary files, make network connections, or possibly run arbitrary code (specifically, Function0 functions) via a gadget chain.

**CVE-2023-22899**

Zip4j through 2.11.2, as used in Threema and other products, does not always check the MAC when decrypting a ZIP archive.

**CVE-2024-22233**

In Spring Framework versions 6.0.15 and 6.1.2, it is possible for a user to provide specially crafted HTTP requests that may cause a denial-of-service (DoS) condition. Specifically, an application is vulnerable when all of the following are true: \* the application uses Spring MVC \* Spring Security 6.1.6+ or 6.2.1+ is on the classpath Typically, Spring Boot applications need the org.springframework.boot:spring-boot-starter-web and org.springframework.boot:spring-boot-starter-security dependencies to meet all conditions.

**CVE-2024-35255**

Azure Identity Libraries and Microsoft Authentication Library Elevation of Privilege Vulnerability

**GHSA-xpw8-rcwv-8f8p**

A client might overload the server by issue frequent RST frames. This can cause a massive amount of load on the remote system and so cause a DDOS attack.

**CVE-2023-34462**

Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. The `SniHandler` can allocate up to 16MB of heap for each channel during the TLS handshake. When the handler or the channel does not have an idle timeout, it can be used to make a TCP server using the `SniHandler` to allocate 16MB of heap. The `SniHandler` class is a handler that waits for the TLS handshake to configure



a `SslHandler` according to the indicated server name by the `ClientHello` record. For this matter it allocates a `ByteBuf` using the value defined in the `ClientHello` record. Normally the value of the packet should be smaller than the handshake packet but there are not checks done here and the way the code is written, it is possible to craft a packet that makes the `SslClientHelloHandler`. This vulnerability has been fixed in version 4.1.94.Final.

#### **GHSA-58qw-p7qm-5rvh**

There are no circumstances in a normally deployed Jetty server where potentially hostile XML is given to the `XmlParser` class without the attacker already having arbitrary access to the server. I.e. in order to exploit `XmlParser` the attacker would already have the ability to deploy and execute hostile code. Specifically, Jetty has no protection against malicious web application and potentially hostile web applications should only be run on an isolated virtualisation.

#### **CVE-2023-35887**

Exposure of Sensitive Information to an Unauthorized Actor vulnerability in Apache Software Foundation Apache MINA. In SFTP servers implemented using Apache MINA SSHD that use a `RootedFileSystem`, logged users may be able to discover "exists/does not exist" information about items outside the rooted tree via paths including parent navigation ("..") beyond the root, or involving symlinks. This issue affects Apache MINA: from 1.0 before 2.10. Users are recommended to upgrade to 2.10

#### **CVE-2023-34055**

In Spring Boot versions 2.7.0 - 2.7.17, 3.0.0-3.0.12 and 3.1.0-3.1.5, it is possible for a user to provide specially crafted HTTP requests that may cause a denial-of-service (DoS) condition. Specifically, an application is vulnerable when all of the following are true: \* the application uses Spring MVC or Spring WebFlux \* `org.springframework.boot:spring-boot-actuator` is on the classpath

#### **CVE-2022-39135**

Apache Calcite 1.22.0 introduced the SQL operators `EXISTS_NODE`, `EXTRACT_XML`, `XML_TRANSFORM` and `EXTRACT_VALUE` do not restrict XML External Entity references in their configuration, making them vulnerable to a potential XML External Entity (XXE) attack. Therefore any client exposing these operators, typically by using Oracle dialect (the first three) or MySQL dialect (the last one), is affected by this vulnerability (the extent of it will depend on the user under which the application is running). From Apache Calcite 1.32.0 onwards, Document Type Declarations and XML External Entity resolution are disabled on the impacted operators.

#### **GHSA-6g3j-p5g6-992f**

A flaw was discovered in OpenSearch, affecting the `_search` API that allowed a specially crafted query string to cause a Stack Overflow and ultimately a Denial of Service. The issue was identified by Elastic Engineering and corresponds to security advisory ESA-2023-14 (CVE-2023-31419).

#### **CVE-2023-50298**

Exposure of Sensitive Information to an Unauthorized Actor vulnerability in Apache Solr. This issue affects Apache Solr: from 6.0.0 through 8.11.2, from 9.0.0 before 9.4.1. Solr Streaming Expressions allows users to extract data from other Solr Clouds, using a `zkHost` parameter. When original SolrCloud is setup to use ZooKeeper credentials and ACLs, they will be sent to whatever `zkHost` the user provides. An attacker could setup a server to mock ZooKeeper, that accepts ZooKeeper requests with credentials and ACLs and extracts the sensitive information, then send a streaming expression using the mock server's address in `zkHost`. Streaming Expressions are exposed via the `/streaming` handler, with `read` permissions. Users are recommended to upgrade to version 8.11.3 or 9.4.1, which fix the issue. From these versions on, only `zkHost` values that have the same server address (regardless of chroot), will use the given ZooKeeper credentials and ACLs when connecting.

#### **CVE-2021-41561**

Improper Input Validation vulnerability in Parquet-MR of Apache Parquet allows an attacker to DoS by malicious Parquet files. This issue affects Apache Parquet-MR version 1.9.0 and later versions.

**CVE-2023-52428**

In Connect2id Nimbus JOSE+JWT before 9.37.2, an attacker can cause a denial of service (resource consumption) via a large JWE p2c header value (aka iteration count) for the PasswordBasedDecrypter (PBKDF2) component.

**CVE-2020-13955**

HttpUtils#getURLConnection method disables explicitly hostname verification for HTTPS connections making clients vulnerable to man-in-the-middle attacks. Calcite uses internally this method to connect with Druid and Splunk so information leakage may happen when using the respective Calcite adapters. The method itself is in a utility class so people may use it to create vulnerable HTTPS connections for other applications. From Apache Calcite 1.26 onwards, the hostname verification will be performed using the default JVM truststore.

**CVE-2024-36114**

Aircompressor is a library with ports of the Snappy, LZO, LZ4, and Zstandard compression algorithms to Java. All decompressor implementations of Aircompressor (LZ4, LZO, Snappy, Zstandard) can crash the JVM for certain input, and in some cases also leak the content of other memory of the Java process (which could contain sensitive information). When decompressing certain data, the decompressors try to access memory outside the bounds of the given byte arrays or byte buffers. Because Aircompressor uses the JDK class `sun.misc.Unsafe` to speed up memory access, no additional bounds checks are performed and this has similar security consequences as out-of-bounds access in C or C++, namely it can lead to non-deterministic behavior or crash the JVM. Users should update to Aircompressor 0.27 or newer where these issues have been fixed. When decompressing data from untrusted users, this can be exploited for a denial-of-service attack by crashing the JVM, or to leak other sensitive information from the Java process. There are no known workarounds for this issue.

**CVE-2024-29857**

An issue was discovered in ECCurve.java and ECCurve.cs in Bouncy Castle Java (BC Java) before 1.78, BC Java LTS before 2.73.6, BC-FJA before 1.0.2.5, and BC C# .Net before 2.3.1. Importing an EC certificate with crafted F2m parameters can lead to excessive CPU consumption during the evaluation of the curve parameters.

**CVE-2024-32888**

The Amazon JDBC Driver for Redshift is a Type 4 JDBC driver that provides database connectivity through the standard JDBC application program interfaces (APIs) available in the Java Platform, Enterprise Editions. Prior to version 2.1.0.28, SQL injection is possible when using the non-default connection property `preferQueryMode=simple` in combination with application code which has a vulnerable SQL that negates a parameter value. There is no vulnerability in the driver when using the default, extended query mode. Note that `preferQueryMode` is not a supported parameter in Redshift JDBC driver, and is inherited code from Postgres JDBC driver. Users who do not override default settings to utilize this unsupported query mode are not affected. This issue is patched in driver version 2.1.0.28. As a workaround, do not use the connection property `preferQueryMode=simple`. (NOTE: Those who do not explicitly specify a query mode use the default of extended query mode and are not affected by this issue.)

## Behavioral changes in Cloudera DataFlow for Data Hub 7.3.1

You can review the changes in certain features or functionalities of components that have resulted in a change in behavior from the previously released version to this version of Cloudera DataFlow for Data Hub 7.3.1.

### Behavioral changes in Flow Management

Review the list of Flow Management behavioral changes in Cloudera DataFlow for Data Hub.

#### Behavioral changes in Flow Management in Cloudera DataFlow for Data Hub 7.3.1.400

Review the list of Flow Management behavioral changes in Cloudera DataFlow for Data Hub 7.3.1.400.

##### Flow Management with NiFi 1

##### Secure communication between NiFi and ZooKeeper configured by default

If both ZooKeeper and NiFi services are secured, NiFi communication with ZooKeeper will be automatically configured as secured (TLS) using a new port, 2182. If you enforce TCP communication through a firewall and explicitly allow certain ports, you need to open them for port 2182.

If you do not want to use secure communication between ZooKeeper and NiFi, follow these steps to configure unsecured communication on port 2181:

1. Update the ZooKeeper connection string:
  - a. In Cloudera Manager, navigate to NiFi Configuration .
  - b. Set `nifi.zookeeper.connect.string` by replacing `${ZK_QUORUM}` with the unsecure ZK QUORUM string, which has port 2181.

To find your ZooKeeper quorum string from a NiFi node, run the following command as root:

```
NIFI_PROC=$(ls -td /var/run/cloudera-scm-agent/process/NIFI/ | head -1);
grep "Connect String" $NIFI_PROC/state-management.xml | cut -d\> -f2 |
cut -d\< -f1; unset NIFI_PROC
```

This command will provide your connect string. For example:

```
host1:2181,host2:2181,host3:2181
```

2. Add a safety valve for staging/state-management.xml in Cloudera Manager with the following property:
  - Name: `xml.state-management.cluster-provider.zk-provider.property.Connect String`
  - Value: `<YOUR ZOOKEEPER CONNECT STRING>`
3. After upgrading to version 2.1.7, uncheck the `nifi.zookeeper.client.secure` option in Cloudera Manager.

##### ScriptedTransformRecord processor requires proper schema name attribute for record writer

NIFI-11523 introduced a fix that ensures the ScriptedTransformRecord processor uses the correct schema defined for the record writer. Previously, if the schema name attribute was set in the writer but not in the flow, it was ignored, defaulting to the reader schema. This behavior has been corrected, which may cause the processor to fail after upgrading if the schema name attribute is not set in the flow.

The failure is typically logged as:

```
org.apache.nifi.schema.access.SchemaNotFoundException: ${schema.name} did not provide appropriate Schema Name
```

To prevent failures, ensure that the schema name attribute is properly configured in the flow or match it to the schema defined for the record reader for identical behavior.




## Flow Management with NiFi 2

### Required actions after upgrade to re-enable the default Atlas reporting task



After upgrading from Cloudera on cloud 7.2.18.700 to 7.3.1.400 with NiFi 2, the default Atlas reporting task may fail due to changes in how Kerberos is configured in NiFi 2.

In previous versions, Kerberos credentials could be configured directly in the reporting task or controller service properties. In NiFi 2, this is no longer supported, and Kerberos authentication for the default Atlas reporting task must be configured through a dedicated Kerberos user service.

#### Step 1: Create a Kerberos user service

1. In the Cloudera Data Hub Flow Management NiFi 2 UI, click **Global Menu** () **Controller Settings** from the top-right corner to open the **NiFi Settings** page.
2. Select the **Management Controller Services** tab.
3. Click the **+** icon to add a new controller service.
4. In the **Add Controller Service** window, filter for **KerberosKeytabUserService** and select it from the list.
5. Click **ADD**.
6. Configure the new service.
  - a. Click the **Configure** icon () in the far-right column.
  - b. Set the following properties (copied from the old configuration):
    - nifi-kerberos-keytab: /hadoopfs/fs4/working-dir/nifi.keytab
    - nifi-kerberos-principal: \${NIFI\_PRINCIPAL}
  - c. Click **APPLY**.
7. Click the **Enable** icon () to activate the new **KerberosKeytabUserService**.

#### Step 2: Update the default Atlas reporting task

1. Switch to the **Reporting Tasks** page from the **Management Controller Services** tab.
2. Locate the **Default Atlas Reporting Task** and click the **Edit** icon () to modify its properties.
3. Remove the following properties from the list:
  - kafka-kerberos-service-name
  - kafka-security-protocol
  - nifi-kerberos-principal
  - nifi-kerberos-keytab
  - kafka-bootstrap-servers
4. Change the **Atlas Authentication Method** from **Basic** to **Kerberos**.
5. For **Kerberos User Service**, select the **KerberosKeytabUserService** you created in Step 1.
6. Save the changes and click the **Enable** icon () to activate the reporting task.

### Key compatibility changes in NiFi 2

NiFi 2 introduces a lot of significant changes and enhancements, including some breaking changes. It is important to familiarize yourself with the following points before migrating your existing flows.

#### Java 21

Java 21 is the minimum Java version required with NiFi 2.0, so make sure you have Java 21 installed on your NiFi nodes before upgrading.

### **Templates and XML flow definitions**

The concept of templates in NiFi has been deprecated, and the XML templates are stored in memory in NiFi as well as in the persisted flow definition.

Additionally, flow.xml.gz no longer exists, only flow.json.gz can be used in NiFi clusters for defining flows in the canvas.

If you have templates, export those templates as JSON definitions or version the templates into a NiFi Registry instance. The best practice is to use a NiFi Registry in combination with NiFi when it comes to version control and share / reuse flow definitions.

### **Custom components / NARs**

Although not certain, it is very likely that a custom NAR designed for NiFi 1 will not be successfully loaded into NiFi 2. If your NiFi setup includes custom components or NARs, it is a requirement to update your dependencies to align with NiFi 2. This entails making the necessary adjustments and rebuilding your NARs using Java 21.

### **Variables replaced by parameters**

Variables and the variable registry have been removed from NiFi. Only Parameter contexts and parameters are available for use going forward. In future Cloudera Flow Management releases, tooling will be provided to help with the conversion of variables to parameters. In the meantime, this conversion should be done manually when migrating flows to NiFi 2. Any variables left will simply be ignored when loading the flow definition.

### **Event driven thread pool no longer exists**

The event driven scheduling strategy was an option available on some processors. This was an experimental feature in NiFi and did not prove to bring any significant performance improvements. The event driven thread pool has been removed, leaving only the time driven thread pool available. Any components previously configured using the event driven scheduling strategy should be switched to the time driven scheduling strategy.

### **Removed languages in scripted components**

In NiFi 2.0, support for certain languages in scripted components has been removed. The affected languages are: ECMAScript, Lua, Ruby, and Python. It is recommended to switch to Groovy or to leverage the new Python API feature for developing processors.

### **Removed components and replacement options**

The following list contains the list of the components that have been removed between clusters based on NiFi 1.26 and clusters based on NiFi 2.0, along with the recommended alternatives where available.

- Processors
  - Base64EncodeContent => EncodeContent
  - CompareFuzzyHash => no replacement
  - ConsumeEWS => no replacement
  - ConsumeKafka\_1\_0 => ConsumeKafka\_2\_6
  - ConsumeKafka\_2\_0 => ConsumeKafka\_2\_6
  - ConsumeKafkaRecord\_1\_0 => ConsumeKafkaRecord\_2\_6
  - ConsumeKafkaRecord\_2\_0 => ConsumeKafkaRecord\_2\_6
  - ConvertAvroSchema => ConvertRecord
  - ConvertAvroToORC => no replacement
  - ConvertCSVToAvro => ConvertRecord
  - ConvertExcelToCSVProcessor => ConvertRecord with ExcelReader
  - ConvertJSONToAvro => ConvertRecord
  - CryptographicHashAttribute => UpdateAttribute
  - DeleteAzureBlobStorage => DeleteAzureBlobStorage\_v12
  - DeleteRethinkDB => no replacement
  - EncryptContent => EncryptContentAge or EncryptContentPGP
  - ExecuteInfluxDBQuery => use [Influx Data NARs for NiFi](#)
  - ExtractCCDAAttributes => no replacement
  - FetchAzureBlobStorage => FetchAzureBlobStorage\_v12
  - FetchElasticsearchHttp => GetElasticsearch
  - FuzzyHashContent => no replacement
  - GetAzureQueueStorage => GetAzureQueueStorage\_v12
  - GetHTMLElement => no replacement
  - GetHTTP => InvokeHTTP
  - GetIgniteCache => no replacement
  - GetJMSQueue => ConsumeJMS
  - GetJMSTopic => ConsumeJMS
  - GetRethinkDB => no replacement
  - GetTCP => no replacement
  - GetTwitter => ConsumeTwitter
  - HashAttribute => CryptographicHashAttribute
  - HashContent => CryptographicHashContent
  - InferAvroSchema => ExtractRecordSchema
  - ListAzureBlobStorage => ListAzureBlobStorage\_v12
  - ModifyHTMLElement => no replacement
  - PostHTTP => InvokeHTTP
  - PostSlack => PublishSlack
  - PublishKafka\_1\_0 => PublishKafka\_2\_6
  - PublishKafka\_2\_0 => PublishKafka\_2\_6
  - PublishKafkaRecord\_1\_0 => PublishKafkaRecord\_2\_6
  - PublishKafkaRecord\_2\_0 => PublishKafkaRecord\_2\_6
  - PutAzureBlobStorage => PutAzureBlobStorage\_v12
  - PutAzureQueueStorage => PutAzureQueueStorage\_v12
  - PutBigQueryBatch => PutBigQuery
  - PutBigQueryStreaming => PutBigQuery
  - PutElasticsearchHttp => PutElasticsearchJson
  - PutElasticsearchHttpRecord => PutElasticsearchRecord
  - PutHiveQL => PutClouderaHiveQL
  - PutHiveStreaming => PutClouderaHiveStreaming

- PutHTMLElement => no replacement
- PutIgniteCache => no replacement
- PutInfluxDB => use [Influx Data NARs for NiFi](#)
- PutJMS => PublishJMS
- PutRethinkDB => no replacement
- PutRiemann => no replacement
- PutSlack => PublishSlack
- QueryElasticsearchHttp => PaginatedJsonQueryElasticsearch
- ScrollElasticsearchHttp => SearchElasticsearch
- SelectHiveQL => SelectClouderaHiveQL
- SpringContextProcessor => no replacement
- StoreInKiteDataset => no replacement
- UpdateHiveTable => UpdateClouderaHiveTable
- Controller services
  - ActionHandlerLookup => no replacement
  - AlertHandler => no replacement
  - AzureStorageCredentialsControllerService => AzureStorageCredentialsControllerService\_v12
  - AzureStorageCredentialsControllerServiceLookup => AzureStorageCredentialsControllerServiceLookup\_v12
  - AzureStorageEmulatorCredentialsControllerService => no replacement
  - EasyRulesEngineProvider => no replacement
  - EasyRulesEngineService => no replacement
  - ExpressionHandler => no replacement
  - GraphiteMetricReporterService => no replacement
  - GremlinClientService => no replacement
  - HBase\_1\_1\_2\_ClientMapCacheService => HBase\_2\_ClientMapCacheService
  - HBase\_1\_1\_2\_ClientService => HBase\_2\_ClientService
  - HBase\_1\_1\_2\_ListLookupService => no replacement
  - HBase\_1\_1\_2\_RecordLookupService => HBase\_2\_RecordLookupService
  - HiveConnectionPool => ClouderaHiveConnectionPool
  - HortonworksSchemaRegistry => ClouderaSchemaRegistry
  - KafkaRecordSink\_1\_0 => KafkaRecordSink\_2\_6
  - KafkaRecordSink\_2\_0 => KafkaRecordSink\_2\_6
  - KeytabCredentialsService => KerberosKeytabUserService
  - LogHandler => no replacement
  - OAuth2TokenProviderImpl => StandardOAuth2AccessTokenProvider
  - OpenCypherClientService => no replacement
  - RecordSinkHandler => no replacement
  - ScriptedActionHandler => no replacement
  - ScriptedRulesEngine => no replacement
- Reporting tasks
  - AmbariReportingTask => no replacement
  - MetricsEventReportingTask => no replacement
  - MetricsReportingTask => no replacement

- Components with new coordinates
  - InvokeGRPC => moved into nifi-cdf-grpc-nar
  - ListenGRPC => moved into nifi-cdf-grpc-nar
  - KerberosKeytabUserService => moved into nifi-kerberos-user-service-nar
  - KerberosPasswordUserService => moved into nifi-kerberos-user-service-nar
  - KerberosTicketCacheUserService => moved into nifi-kerberos-user-service-nar

Tooling will be provided in upcoming Cloudera Flow Management releases to automatically handle these changes. Currently, two options are available:

- Manually edit the flow.json.gz file to update the coordinates of the impacted components.
- Make the changes after the flow is imported in NiFi 2.0 by replacing the ghost components with the new implementations for each instance of the components listed above.
- Pulsar components

All Pulsar components have been removed. You can download the NARs from a public Maven repository and deploy them as custom NARs.

- [nifi-pulsar-nar](#)
- [nifi-pulsar-client-service-nar](#)

## Behavioral changes in Flow Management in Cloudera DataFlow for Data Hub 7.3.1.0

Review the list of Flow Management behavioral changes in Cloudera DataFlow for Data Hub 7.3.1.0.

### Flow Management with NiFi 1

#### Secure communication between NiFi and ZooKeeper configured by default

If both ZooKeeper and NiFi services are secured, NiFi communication with ZooKeeper will be automatically configured as secured (TLS) using a new port, 2182. If you enforce TCP communication through a firewall and explicitly allow certain ports, you need to open them for port 2182.

If you do not want to use secure communication between ZooKeeper and NiFi, follow these steps to configure unsecured communication on port 2181:

1. Update the ZooKeeper connection string:
  - a. In Cloudera Manager, navigate to NiFi Configuration .
  - b. Set `nifi.zookeeper.connect.string` by replacing `${ZK_QUORUM}` with the unsecure ZK QUORUM string, which has port 2181.

To find your ZooKeeper quorum string from a NiFi node, run the following command as root:

```
NIFI_PROC=$(ls -td /var/run/cloudera-scm-agent/process/NIFI/ | head -1);
grep "Connect String" $NIFI_PROC/state-management.xml | cut -d\> -f2 |
cut -d\< -f1; unset NIFI_PROC
```

This command will provide your connect string. For example:

```
host1:2181,host2:2181,host3:2181
```

2. Add a safety valve for staging/state-management.xml in Cloudera Manager with the following property:
  - Name: `xml.state-management.cluster-provider.zk-provider.property.Connect String`
  - Value: `<YOUR ZOOKEEPER CONNECT STRING>`
3. After upgrading to version 2.1.7, uncheck the `nifi.zookeeper.client.secure` option in Cloudera Manager.

#### ScriptedTransformRecord processor requires proper schema name attribute for record writer

[NiFi-11523](#) introduced a fix that ensures the ScriptedTransformRecord processor uses the correct schema defined for the record writer. Previously, if the schema name attribute was set in the writer but not in the flow, it was ignored,



defaulting to the reader schema. This behavior has been corrected, which may cause the processor to fail after upgrading if the schema name attribute is not set in the flow.

The failure is typically logged as:

```
org.apache.nifi.schema.access.SchemaNotFoundException: ${schema.name} did not provide appropriate Schema Name
```

To prevent failures, ensure that the schema name attribute is properly configured in the flow or match it to the schema defined for the record reader for identical behavior.

## Flow Management with NiFi 2

NiFi 2 introduces a lot of significant changes and enhancements, including some breaking changes for Flow Management clusters based on NiFi 2. It is important to familiarize yourself with the following points before migrating your existing flows.



**Important:** Currently, there is no upgrade path from NiFi 1-based Flow Management clusters to clusters with NiFi 2. You need to start new clusters using the NiFi 2 templates and migrate your existing flows to the new clusters. Note that these new NiFi 2-based clusters are available in Technical Preview. They are not production-ready and should not be used for critical workloads.

If you want to migrate a data flow, you need to export the process group as a JSON file from your NiFi 1 cluster and import this JSON file into your NiFi 2 cluster. Tooling to help with upgrades and automatically manage the breaking changes will be provided in an upcoming Flow Management release.

### Java 21

Java 21 is the minimum Java version required with NiFi 2. This version is automatically installed and configured on new Data Hub clusters using NiFi 2.

### Templates and XML flow definitions

The concept of templates in NiFi has been deprecated. Instead, versioning flows should be managed using the DataFlow Catalog and/or the NiFi Registry. It is highly recommended to handle any existing templates in your NiFi 1.x clusters by:

- Versioning the templates into the desired registry (DataFlow Catalog, NiFi Registry)
- Deleting the templates from NiFi process groups

Additionally, flow.xml.gz no longer exists, only flow.json.gz can be used in NiFi clusters for defining flows in the canvas.

### Custom components / NARs

Although not certain, it is very likely that a custom NAR designed for NiFi 1 will not be successfully loaded into NiFi 2. If your NiFi setup includes custom components or NARs, it is a requirement to update your dependencies to align with NiFi 2. This entails making the necessary adjustments and rebuilding your NARs using Java 21.

### Variables are removed in favor of parameters

Variables and the variable registry have been removed from NiFi. Only Parameter Contexts and parameters should be used going forward. In future releases, tools will be provided to help with the conversion of variables to parameters. In the meantime, this conversion should be done manually when migrating flows to NiFi 2. Any variables left will simply be ignored when loading the flow definition.

### Event driven thread pool no longer exists

The event driven thread pool has been removed, leaving only the time driven thread pool available. Any components previously configured using the event driven scheduling strategy should be switched to the time driven scheduling strategy.

**Removed languages in scripted components**

In NiFi 2, support for certain languages in scripted components has been removed. The affected languages are: ECMAScript, Lua, Ruby, and Python. It is recommended to switch to Groovy or to leverage the new Python API feature for developing processors.

**Removed components and replacement options**

The following list contains the list of the components that have been removed between clusters based on NiFi 1.28 and clusters based on NiFi 2, along with the recommended alternatives where available.

- Processors
  - Base64EncodeContent => EncodeContent
  - CompareFuzzyHash => no replacement
  - ConsumeEWS => no replacement
  - ConsumeKafka\_1\_0 => ConsumeKafka\_2\_6
  - ConsumeKafka\_2\_0 => ConsumeKafka\_2\_6
  - ConsumeKafkaRecord\_1\_0 => ConsumeKafkaRecord\_2\_6
  - ConsumeKafkaRecord\_2\_0 => ConsumeKafkaRecord\_2\_6
  - ConvertAvroSchema => ConvertRecord
  - ConvertAvroToORC => no replacement
  - ConvertCSVToAvro => ConvertRecord
  - ConvertExcelToCSVProcessor => ConvertRecord with ExcelReader
  - ConvertJSONToAvro => ConvertRecord
  - CryptographicHashAttribute => UpdateAttribute
  - DeleteAzureBlobStorage => DeleteAzureBlobStorage\_v12
  - DeleteRethinkDB => no replacement
  - EncryptContent => EncryptContentAge or EncryptContentPGP
  - ExecuteInfluxDBQuery => use [Influx Data NARs for NiFi](#)
  - ExtractCCDAAttributes => no replacement
  - FetchAzureBlobStorage => FetchAzureBlobStorage\_v12
  - FetchElasticsearchHttp => GetElasticsearch
  - FuzzyHashContent => no replacement
  - GetAzureQueueStorage => GetAzureQueueStorage\_v12
  - GetHTMLElement => no replacement
  - GetHTTP => InvokeHTTP
  - GetIgniteCache => no replacement
  - GetJMSQueue => ConsumeJMS
  - GetJMSTopic => ConsumeJMS
  - GetRethinkDB => no replacement
  - GetTCP => no replacement
  - GetTwitter => ConsumeTwitter
  - HashAttribute => CryptographicHashAttribute
  - HashContent => CryptographicHashContent
  - InferAvroSchema => ExtractRecordSchema
  - ListAzureBlobStorage => ListAzureBlobStorage\_v12
  - ModifyHTMLElement => no replacement
  - PostHTTP => InvokeHTTP
  - PostSlack => PublishSlack
  - PublishKafka\_1\_0 => PublishKafka\_2\_6
  - PublishKafka\_2\_0 => PublishKafka\_2\_6
  - PublishKafkaRecord\_1\_0 => PublishKafkaRecord\_2\_6
  - PublishKafkaRecord\_2\_0 => PublishKafkaRecord\_2\_6
  - PutAzureBlobStorage => PutAzureBlobStorage\_v12
  - PutAzureQueueStorage => PutAzureQueueStorage\_v12
  - PutBigQueryBatch => PutBigQuery
  - PutBigQueryStreaming => PutBigQuery
  - PutElasticsearchHttp => PutElasticsearchJson
  - PutElasticsearchHttpRecord => PutElasticsearchRecord
  - PutHiveQL => PutClouderaHiveQL
  - PutHiveStreaming => PutClouderaHiveStreaming

- PutHTMLElement => no replacement
- PutIgniteCache => no replacement
- PutInfluxDB => use [Influx Data NARs for NiFi](#)
- PutJMS => PublishJMS
- PutRethinkDB => no replacement
- PutRiemann => no replacement
- PutSlack => PublishSlack
- QueryElasticsearchHttp => PaginatedJsonQueryElasticsearch
- ScrollElasticsearchHttp => SearchElasticsearch
- SelectHiveQL => SelectClouderaHiveQL
- SpringContextProcessor => no replacement
- StoreInKiteDataset => no replacement
- UpdateHiveTable => UpdateClouderaHiveTable
- Controller services
  - ActionHandlerLookup => no replacement
  - AlertHandler => no replacement
  - AzureStorageCredentialsControllerService => AzureStorageCredentialsControllerService\_v12
  - AzureStorageCredentialsControllerServiceLookup => AzureStorageCredentialsControllerServiceLookup\_v12
  - AzureStorageEmulatorCredentialsControllerService => no replacement
  - EasyRulesEngineProvider => no replacement
  - EasyRulesEngineService => no replacement
  - ExpressionHandler => no replacement
  - GraphiteMetricReporterService => no replacement
  - GremlinClientService => no replacement
  - HBase\_1\_1\_2\_ClientMapCacheService => HBase\_2\_ClientMapCacheService
  - HBase\_1\_1\_2\_ClientService => HBase\_2\_ClientService
  - HBase\_1\_1\_2\_ListLookupService => no replacement
  - HBase\_1\_1\_2\_RecordLookupService => HBase\_2\_RecordLookupService
  - HiveConnectionPool => ClouderaHiveConnectionPool
  - HortonworksSchemaRegistry => ClouderaSchemaRegistry
  - KafkaRecordSink\_1\_0 => KafkaRecordSink\_2\_6
  - KafkaRecordSink\_2\_0 => KafkaRecordSink\_2\_6
  - KeytabCredentialsService => KerberosKeytabUserService
  - LogHandler => no replacement
  - OAuth2TokenProviderImpl => StandardOAuth2AccessTokenProvider
  - OpenCypherClientService => no replacement
  - RecordSinkHandler => no replacement
  - ScriptedActionHandler => no replacement
  - ScriptedRulesEngine => no replacement
- Reporting tasks
  - AmbariReportingTask => no replacement
  - MetricsEventReportingTask => no replacement
  - MetricsReportingTask => no replacement

- Components with new coordinates
  - InvokeGRPC => moved into nifi-cdf-grpc-nar
  - ListenGRPC => moved into nifi-cdf-grpc-nar
  - KerberosKeytabUserService => moved into nifi-kerberos-user-service-nar
  - KerberosPasswordUserService => moved into nifi-kerberos-user-service-nar
  - KerberosTicketCacheUserService => moved into nifi-kerberos-user-service-nar

Tooling will be provided in upcoming releases to automatically handle these changes. Currently, two options are available:

- Manually edit the flow.json.gz file to update the coordinates of the impacted components.
  - Make the changes after the flow is imported in NiFi 2 by replacing the ghost components with the new implementations for each instance of the components listed above.
- Pulsar components

All Pulsar components have been temporarily removed. They will be reintroduced in an upcoming release. In the meantime, you can download the NARs from a public Maven repository and deploy them as custom NARs.

- [nifi-pulsar-nar](#)
- [nifi-pulsar-client-service-nar](#)

## Behavioral changes in Streams Messaging

Review the list of Streams Messaging behavioral changes in Cloudera DataFlow for Data Hub 7.3.1.

### Kafka

There are no behavioral changes for Kafka in Cloudera DataFlow for Data Hub 7.3.1.

### Schema Registry

There are no behavioral changes for Schema Registry in Cloudera DataFlow for Data Hub 7.3.1.

### Streams Messaging Manager

There are no behavioral changes for Streams Messaging Manager in Cloudera DataFlow for Data Hub 7.3.1.

### Streams Replication Manager

There are no behavioral changes for Streams Replication Manager in Cloudera DataFlow for Data Hub 7.3.1.

### Cruise Control

There are no behavioral changes for Cruise Control in Cloudera DataFlow for Data Hub 7.3.1.

## Behavioral changes in Cloudera Streaming Analytics

Review the list of Cloudera Streaming Analytics behavioral changes in Cloudera DataFlow for Cloudera Data Hub 7.3.1.

### Cloudera SQL Stream Builder

#### Summary:

CSA-5068 - Page refresh is not required to update result samples.

Previous behavior:

Sample ids could become invalid if another user restarted the job while polling in the UI.

New behavior:

Sample ids are updated before and during polling.

**Summary:**

CSA-5238 - Yarn is set as default token renewer automatically based on configuration.

Previous behavior:

Configuration check wasn't performed and yarn wasn't set as the default renewer.

New behavior:

Yarn is set as the default token renewer if the deployment target is yarn-session or yarn-per-job.

**Summary:**

CSA-5258 - Extended validation for dynamic MV parameter names.

Previous behavior:

Dynamic pattern validation was restrictive which could result in errors when querying MV.

New behavior:

Validation pattern is extended to `[A-Za-z\_\\-0-9\\.\\~]+`

**Summary:**

CSA-5214 - Add option to Cloudera SQL Stream Builder service to customize default truststore

Previous behavior:

Default Kafka TrustStore configurations could only be customized in Flink configurations.

New behavior:

Customizing default Kafka TrustStore configurations was added to Streaming SQL Console. Kafka TrustStore can be configured during adding Kafka as a Data Source on the UI.

**Summary:**

CSA-5199 - API change: the Cloudera registry catalog type registry is now named cloudera-registry

Previous behavior:

When creating a Schema Registry data source catalog type property had to be registry

New behavior:

Catalog type property now has to be cloudera-registry when creating a Schema Registry data source, but registry is backward compatible.

**Summary:**

CSA-5306 - Cloudera SQL Stream Builder API does not validate data sources before saving

Previous behavior:

There was no validation before saving a data source via API. If the user wanted to make sure to save a valid data source they had to use the validate endpoint before saving. When creating a data source on SSB UI, the user had the option to save an invalid data source.

New behavior:

Saving the data source is only allowed if the data source is valid. SSB API validates the data source before saving and on the SSB UI Create/Save button is only active if the validation is successful.