

Cloudera Flow Management 4.13.0

Cloudera Flow Management Release Notes

Date published: 2019-06-26

Date modified: 2026-10-21

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2026. 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

What's new in Cloudera Flow Management 4.13.0.....	4
Support matrix.....	6
Component versions.....	6
System requirements.....	8
Supported operating systems.....	9
Supported NiFi Registry databases.....	9
Supported NiFi processors.....	9
Supported NiFi controller services.....	13
Supported NiFi reporting tasks.....	15
Supported NiFi parameter providers.....	15
Supported NiFi flow analysis rules.....	16
Supported NiFi flow registry clients.....	16
Supported Cloudera exclusive components.....	17
Download locations.....	19
Unsupported features.....	21
Behavioral changes.....	21
Known issues in Cloudera Flow Management.....	21
Fixed issues in Cloudera Flow Management.....	22
Fixed CVEs in Cloudera Flow Management.....	22

What's new in Cloudera Flow Management 4.13.0

Explore the latest features and improvements in Cloudera Flow Management and how they support modern data pipeline development and operations.

Cloudera Flow Management 4.13.0 is based on Apache NiFi 2.11.0 and runs with Cloudera Manager on Cloudera on premises 7.3.2 SP1.

This release expands support for Apache Iceberg version 3 tables, adds integrations for cloud services and external systems, improves metrics reporting, and extends platform support. It also includes fixes for multiple Common Vulnerabilities and Exposures (CVEs).

The following sections highlight the main changes.

Rebase to Apache NiFi 2.11

This release updates the underlying Apache NiFi version to 2.11, bringing upstream fixes and improvements to the NiFi core and its components. For the complete list of components supported in Cloudera Flow Management 4.13.0, see the [Support Matrix](#).

Platform support

Red Hat Enterprise Linux 9.6

Cloudera Flow Management 4.13.0 adds support for RHEL 9.6.

Spring Framework 7

The cdf-extensions dependencies have been updated to Spring Framework 7.

Apache Iceberg version 3 support

Cloudera Flow Management 4.13.0 upgrades Apache Iceberg to version 1.10.1 and adds support for the following Iceberg version 3 capabilities:

- **Deletion vectors:** Process position deletes stored as deletion vectors in Iceberg tables.
- **Variant data type:** Ingest and process semi-structured data represented using the Iceberg variant type.
- **Nanosecond timestamp precision:** Preserve nanosecond precision when processing supported timestamp values.
- **Column default values:** Apply Iceberg schema defaults when records omit fields that have defined default values.

The PutIceberg and PutIcebergCDC processors can also write to a specified Iceberg branch. This supports workflows that ingest data into a staging or validation branch before it is merged into the main branch.

New NiFi components

Processors

PutSnowflakeStreaming

It uses the Snowflake Ingest SDK and Snowpipe Streaming API to stream data directly from NiFi dataflows into Snowflake tables without intermediate staging files. It supports high-throughput, low-latency ingestion for pipelines that require lower latency than batch ingestion.

For limitations, see [Authentication, SSL, and proxy limitations in PutSnowflakeStreaming](#).

EnrichGraphRecord

It enables you to build graph-based dataflows by creating or updating graph nodes and relationships and enriching their properties with incoming record data.

The updated GraphClientService API generates database-specific statements, such as Cypher, without requiring users to write queries manually.

Graph component types and labels allow the processor to take advantage of graph database indexes.

Parameter provider

ClouderaOnCloudParameterProvider

You can discover supported service connection details in Cloudera on cloud and populate NiFi Parameter Contexts with the help of this parameter provider reducing manual configuration needs. With the required environment credentials and endpoint information, the provider queries the cloud APIs to discover supported resources, including Data Hub clusters, Cloudera Data Warehouse virtual warehouse JDBC endpoints, Cloudera Operational Database connection details, and Kafka broker endpoints.

The provider automatically organizes the discovered details into parameter groups to simplify the configuration of NiFi processors and Controller Services.

Reporting task

OpenTelemetryReportingTask

It exports NiFi operational, flow, and JVM metrics to OpenTelemetry collectors and compatible observability platforms using the OpenTelemetry Protocol (OTLP).

It supports unauthenticated endpoints, Basic and Bearer token authentication, and secure connections through TLS or mutual TLS (mTLS).

Flow analysis rule

RequireLoadBalancedConnectionAfterPrimaryNodeSourceProcessor

This new flow analysis rule flags connections without load balancing downstream of source processors configured to run only on the Primary Node. It helps identify flow designs that could concentrate processing on a single node instead of distributing FlowFiles across the cluster.

For a full list of supported NiFi components, see the [Support Matrix](#).

Enhancements to existing components and services

JSON Schema support in Cloudera Schema Registry

Cloudera Flow Management 4.13.0 adds JSON Schema support to the ClouderaSchemaRegistry Controller Service. You can register and retrieve JSON Schemas to validate records. Apache Avro schema support remains available.

The service also supports OAuth authentication through an OAuth2AccessTokenProvider Controller Service, providing an alternative to Kerberos or Basic authentication. OAuth support is also available for Confluent Schema Registry.

Apache Atlas lineage for Iceberg

The Atlas Reporting Task now analyzes provenance events from Iceberg processors, allowing their data lineage to be tracked in Apache Atlas.

Kerberos authentication for SMB

SmbjClientProviderService now supports KerberosUserService, enabling Kerberos authentication when accessing SMB shares.

External password sources for Impala

The ImpalaConnectionPool controller service has a new Password Source property for resolving passwords dynamically, including from supported external secret providers.

TLS for NiFi health metrics

Communication between NiFi and Cloudera Manager for service health and status metrics now supports TLS encryption.

JSO

Flow Migration Tool

The available Cloudera Flow Migration Tool versions do not support migration to Cloudera Flow Management 4.13.0. A new release of the Migration Tool is in development to support this migration target.

For currently supported migration paths, see the Migration Tool documentation.

Upgrade and migration options

Installation

To perform a fresh installation of Cloudera Flow Management 4.13.0, follow the [Cloudera Flow Management installation workflow](#).

Support matrix

Review the support matrix before installing or upgrading Cloudera Flow Management.

Component versions

Review the Cloudera Flow Management component versions for compatibility with other applications.



Note: NiFi is compatible with the version of NiFi Registry bundled with your Cloudera Flow Management release as well as any later version.

Cloudera Flow Management 4.13.0

- Apache NiFi 2.11.0.4.13.0.0
- Apache NiFi Registry 2.11.0.4.13.0.0

Cloudera Flow Management 4.12.0.20000

- Apache NiFi 2.6.0.4.12.0.20000
- Apache NiFi Registry 2.6.0.4.12.0.20000

Cloudera Flow Management 4.12.0.10001

- Apache NiFi 2.6.0.4.12.0.10001
- Apache NiFi Registry 2.6.0.4.12.0.10001

Cloudera Flow Management 4.12.0.10000

Not available

Cloudera Flow Management 4.12.0.1

- Apache NiFi 2.6.0.4.12.0.1
- Apache NiFi Registry 2.6.0.4.12.0.1

Cloudera Flow Management 4.12.0.0

- Apache NiFi 2.6.0.4.12.0.0
- Apache NiFi Registry 2.6.0.4.12.0.0

Cloudera Flow Management 4.11.0

- Apache NiFi 2.4.0.4.11.0.0
- Apache NiFi Registry 2.4.0.4.11.0.0

Cloudera Flow Management 4.10.0

- Apache NiFi 2.3.0.4.10.0.0
- Apache NiFi Registry 2.3.0.4.10.0.0

Cloudera Flow Management 4.0.0 [Technical Preview]

- Apache NiFi 2.0.0.4.0.0.0
- Apache NiFi Registry 2.0.0.4.0.0.0

Cloudera Flow Management 2.1.7.2000 (SP2)

- Apache NiFi 1.28.1.2.1.7.2000
- Apache NiFi Registry 1.28.1.2.1.7.2000

Cloudera Flow Management 2.1.7.1000 (SP1)

- Apache NiFi 1.26.0.2.1.7.1000
- Apache NiFi Registry 1.26.0.2.1.7.1000

Cloudera Flow Management 2.1.7

- Apache NiFi 1.26.0.2.1.7.0
- Apache NiFi Registry 1.26.0.2.1.7.0

Cloudera Flow Management 2.1.6.1000 (SP1)

- Apache NiFi 1.23.1.2.1.6.1000
- Apache NiFi Registry 1.23.1.2.1.6.1000

Cloudera Flow Management 2.1.6

- Apache NiFi 1.23.1.2.1.6.0
- Apache NiFi Registry 1.23.1.2.1.6.0

Cloudera Flow Management 2.1.5.1000 (SP1)

- Apache NiFi 1.18.0.2.1.5.1000
- Apache NiFi Registry 1.18.0.2.1.5.1000

Cloudera Flow Management 2.1.5

- Apache NiFi 1.18.0.2.1.5.0
- Apache NiFi Registry 1.18.0.2.1.5.0

Cloudera Flow Management 2.1.4.1000 (SP1)

- Apache NiFi 1.16.0.2.1.4.1000
- Apache NiFi Registry 1.16.0.2.1.4.1000

Cloudera Flow Management 2.1.4

- Apache NiFi 1.16.0.2.1.4.0

- Apache NiFi Registry 1.16.0.2.1.4.0

Cloudera Flow Management 2.1.3

- Apache NiFi 1.15.2.2.1.3.0
- Apache NiFi Registry 1.15.2.2.1.3.0



Note: Apache NiFi and Apache NiFi Registry versions are unified in the 1.15.x release.

Cloudera Flow Management 2.1.2

- Apache NiFi 1.13.2.2.1.2.0
- Apache NiFi Registry 0.8.0.2.1.2.0

Cloudera Flow Management 2.1.1

- Apache NiFi 1.13.2.2.1.1.0
- Apache NiFi Registry 0.8.0.2.1.1.0

Cloudera Flow Management 2.0.4

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

Cloudera Flow Management 2.0.1

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

System requirements

Review the system requirements before installing or upgrading Cloudera Flow Management.

Supported Cloudera platform versions

Cloudera Flow Management 4.13.0 supports the following versions of Cloudera Base on premises:

- 7.3.2 SP1

Supported JAVA Development Kits (JDK)

Cloudera Flow Management requires a minimum of JDK 21 for proper functionality. Ensure your environment meets this requirement before installation.

Other system requirements

ZooKeeper

You need to install the ZooKeeper service included with your Cloudera Base on premises cluster.

Python

- Minimum requirement: Python 3.11
- Recommended version: Python 3.12

Number of cores

- Minimum: Four cores per NiFi node are required for Cloudera support.

- Recommended: Eight cores per NiFi node, which typically provides an optimal starting point for most common use cases.

Supported operating systems

Review the list of operating systems supported in Cloudera Flow Management 4.13.0.

Operating system	Versions
RHEL	• 8.8 • 8.10 • 9.1 • 9.2 • 9.4 • 9.6
SLES	• 15 SP5
Oracle	• 8.8
Ubuntu	• 22.04 • 24.04
Windows	• 10 • Server 2016 • Server 2019 • Server 2022 • Server 2025



Note:

NiFi on Windows is only supported in standalone mode, not managed by Cloudera Manager or as part of a Cloudera cluster, and as a single instance installation. Clustering NiFi on Windows is not supported.

NiFi Registry is not supported on Windows.

Supported NiFi Registry databases

Review the list of databases supported by NiFi Registry in Cloudera Flow Management 4.13.0.

- H2
- PostgreSQL 14.x
- PostgreSQL 15.x
- PostgreSQL 16.x
- PostgreSQL 17.x
- MySQL 8.0
- MariaDB 10.6

Supported NiFi processors

Review the list of Apache NiFi processors supported in Cloudera Flow Management 4.13.0.

Apache NiFi provides a wide range of processors for building dataflows. Cloudera Flow Management 4.13.0 is based on Apache NiFi 2.6.0 and includes most of these processors, but not all are supported by Cloudera.

Some processors available in Apache NiFi are developed and maintained by the community and are not supported in Cloudera. These processors may be excluded due to factors such as limited testing, insufficient reliability, lack of production readiness, or misalignment with Cloudera best practices.

To ensure stability, performance, and full support coverage, use only processors that are officially supported by Cloudera in production environments.

Table 1: List of NiFi processors supported in Cloudera Flow Management 4.13.0

AttributesToCSV	GetAwsTranscribeJobStatus
AttributesToJSON	GetAwsTranslateJobStatus
CalculateParquetOffsets	GetAzureEventHub
CalculateParquetRowGroupOffsets	GetAzureQueueStorage_v12
CalculateRecordStats	GetBoxFileCollaborators
CaptureChangeDebeziumDB2	GetBoxGroupMembers
CaptureChangeDebeziumMongoDB	GetCouchbaseKey
CaptureChangeDebeziumMySQL	GetElasticsearch
CaptureChangeDebeziumOracle	GetFile
CaptureChangeDebeziumPostgreSQL	GetFTP
CaptureChangeDebeziumSQLServer	GetGcpVisionAnnotateFilesOperationStatus
CaptureChangeMySQL	GetGcpVisionAnnotateImagesOperationStatus
ChunkDocument	GetHBase
CompressContent	GetHDFS
ConnectWebSocket	GetHDFSFileInfo
ConsumeAMQP	GetHDFSSequenceFile
ConsumeAzureEventHub	GetHubSpot
ConsumeBoxEnterpriseEvents	GetJiraIssue
ConsumeBoxEvents	GetMongoRecord
ConsumeElasticsearch	GetS3ObjectTags
ConsumeGCPubSub	GetSFTP
ConsumeGCPubSubLite	GetShopify
ConsumeJMS	GetSNMP
ConsumeKafka_2_6	GetSnowflakeIngestStatus
ConsumeKafka2CDP	GetSolr
ConsumeKafka2RecordCDP	GetSplunk
ConsumeKafkaRecord_2_6	GetSQS
ConsumeKinesis	GetWorkdayReport
ConsumeKinesisStream	GetZendesk
ConsumeMQTT	HandleHttpRequest
ConsumeMQTTIIoT	HandleHttpResponse
ConsumePLC	IdentifyMimeType
ConsumeSlack	InvokeAWSGatewayApi
ConsumeTwitter	InvokeGRPC
ConsumeWindowsEventLog	InvokeHTTP
ControlRate	InvokeScriptedProcessor
ConvertAvroToJSON	JoinEnrichment

ConvertAvroToParquet	JoltTransformJSON
ConvertCharacterSet	JoltTransformRecord
ConvertJSONToSQL	JSLTTransformJSON
ConvertProtobuf	JsonQueryElasticsearch
ConvertRecord	ListAzureBlobStorage_v12
CopyAzureBlobStorage_v12	ListAzureDataLakeStorage
CountText	ListBoxFile
CreateBoxFileMetadataInstance	ListBoxFileInfo
CreateBoxMetadataTemplate	ListBoxFileMetadataInstances
CreateHadoopSequenceFile	ListBoxFileMetadataTemplates
CryptographicHashContent	ListCDPObjectStore
DecryptContentAge	ListDatabaseTables
DecryptContentPGP	ListDropbox
DeduplicateRecord	ListenBeats
DeleteAzureBlobStorage_v12	ListenFTP
DeleteAzureDataLakeStorage	ListenGRPC
DeleteBoxFileMetadataInstance	ListenHTTP
DeleteByQueryElasticsearch	ListenNetFlow
DeleteCDPObjectStore	ListenOTLP
DeleteDynamoDB	ListenRELP
DeleteGCSObject	ListenSlack
DeleteGridFS	ListenSyslog
DeleteHBaseCells	ListenTCP
DeleteHBaseRow	ListenTCPRecord
DeleteHDFS	ListenTrapSNMP
DeleteS3Object	ListenUDP
DeleteSQS	ListenUDPRecord
DetectDuplicate	ListenWebSocket
DistributeLoad	ListFile
DuplicateFlowFile	ListFTP
EncodeContent	ListGCSBucket
EncryptContentAge	ListGoogleDrive
EncryptContentPGP	ListHDFS
EnforceOrder	ListS3
EnrichGraphRecord	ListSFTP
EvaluateJsonPath	ListSmb
EvaluateXPath	LogAttribute
EvaluateXQuery	LogMessage
ExecuteGraphQuery	LookupAttribute
ExecuteGraphQueryRecord	LookupRecord

ExecuteGroovyScript	MergeContent
ExecuteProcess	MergeRecord
ExecuteScript	ModifyCompression
ExecuteSQL	MonitorActivity
ExecuteSQLRecord	MoveAzureDataLakeStorage
ExecuteStateless	MoveHDFS
ExecuteStreamCommand	Notify
ExtractAvroMetadata	PackageFlowFile
ExtractGrok	PaginatedJsonQueryElasticsearch
ExtractHL7Attributes	ParseCEF
ExtractImageMetadata	ParseDocument
ExtractRecordSchema	ParseEvtx
ExtractStructuredBoxFileMetadata	ParseSyslog
ExtractText	PartitionRecord
FetchAzureBlobStorage_v12	PromptAzureOpenAI
FetchAzureDataLakeStorage	PromptChatGPT
FetchBoxFile	PromptClaude
FetchBoxFileInfo	PromptOpenAI
FetchBoxFileMetadataInstance	PublishAMQP
FetchBoxFileRepresentation	PublishGCPubSub
FetchCDPObjectStore	PublishGCPubSubLite
FetchDistributedMapCache	PublishJMS
FetchDropbox	PublishKafka_2_6
FetchFile	PublishKafka2CDP
FetchFTP	PublishKafka2RecordCDP
FetchGCSObject	PublishKafkaRecord_2_6
FetchGoogleDrive	PublishMQTT
FetchGridFS	PublishSlack
FetchHBaseRow	PutAccumuloRecord
FetchHDFS	PutAzureBlobStorage_v12
FetchParquet	PutAzureCosmosDBRecord
FetchPLC	PutAzureDataLakeStorage
FetchS3Object	PutAzureEventHub
FetchSFTP	PutAzureQueueStorage_v12
FetchSmb	PutBigQuery
FilterAttribute	PutBoxFile
FlattenJson	PutCassandraQL
ForkEnrichment	PutCassandraRecord
ForkRecord	PutCDPObjectStore
GenerateFlowFile	PutChroma

GenerateRecord	PutClouderaHiveQL
GenerateTableFetch	PutClouderaHiveStreaming
GeoEnrichIP	PutClouderaORC
GeoEnrichIPRecord	PutCloudWatchMetric
GeohashRecord	PutCouchbaseKey
GetAwsPollyJobStatus	PutDatabaseRecord
GetAwsTextractJobStatus	PutDistributedMapCache
	PutDropbox

Related Information

[Supported NiFi controller services](#)

[Supported NiFi reporting tasks](#)

[Supported NiFi parameter providers](#)

[Supported NiFi flow analysis rules](#)

[Supported NiFi flow registry clients](#)

Supported NiFi controller services

Review the list of Apache NiFi controller services supported in Cloudera Flow Management 4.13.0.

Apache NiFi provides a wide range of controller services for building dataflows. Cloudera Flow Management 4.13.0 is based on Apache NiFi 2.6.0 and includes most of these controller services, but not all are supported by Cloudera.

Some controller services available in Apache NiFi are developed and maintained by the community and are not supported in Cloudera. These controller services may be excluded due to factors such as limited testing, insufficient reliability, lack of production readiness, or misalignment with Cloudera best practices.

To ensure stability, performance, and full support coverage, use only controller services that are officially supported by Cloudera in production environments.

Table 2: List of NiFi controller services supported in Cloudera Flow Management 4.13.0

AccumuloService	DistributedSetCacheServer
ActiveMQJMSConnectionFactoryProvider	EBCDICRecordReader
ADLSCredentialsControllerService	ElasticSearchClientServiceImpl
ADLSCredentialsControllerServiceLookup	ElasticSearchLookupService
ADLSIDBrokerCloudCredentialsProviderControllerService	ElasticSearchStringLookupService
AmazonGlueSchemaRegistry	EmailRecordSink
ApicurioSchemaRegistry	EmbeddedHazelcastCacheManager
AvroReader	ExcelReader
AvroRecordSetWriter	ExternalHazelcastCacheManager
AvroSchemaRegistry	FreeFormTextRecordSetWriter
AWSCredentialsProviderControllerService	GCPCredentialsControllerService
AWSIDBrokerCloudCredentialsProviderControllerService	GCSFileResourceService
AzureBlobIDBrokerCloudCredentialsProviderControllerService	GenericPLC4XConnectionPool
AzureCosmosDBClientService	GrokReader
AzureEventHubRecordSink	HadoopCatalogService

AzureServiceBusJMSConnectionFactoryProvider	HadoopDBCPCConnectionPool
AzureStorageCredentialsControllerService_v12	HazelcastMapCacheClient
AzureStorageCredentialsControllerServiceLookup_v12	HBase_2_ClientMapCacheService
CassandraDistributedMapCache	HBase_2_ClientService
CassandraSessionProvider	HBase_2_RecordLookupService
CdpCredentialsProviderControllerService	Hive3ConnectionPool
CdpOAuth2AccessTokenProviderControllerService	HiveCatalogService
CEFReader	ImpalaConnectionPool
CiscoEmblemSyslogMessageReader	IPFIXReader
ClouderaAttributeSchemaReferenceReader	IPLookupService
ClouderaAttributeSchemaReferenceWriter	JASN1Reader
ClouderaEncodedSchemaReferenceReader	JiraRecordSink
ClouderaEncodedSchemaReferenceWriter	JMSConnectionFactoryProvider
ClouderaHiveConnectionPool	JndiJmsConnectionFactoryProvider
ClouderaHiveConnectionPoolLookup	JsonConfigBasedBoxClientService
ClouderaSchemaRegistry	JsonPathReader
CMLLookupService	JsonRecordSetWriter
ConfluentEncodedSchemaReferenceReader	JsonTreeReader
ConfluentEncodedSchemaReferenceWriter	JWTBearerOAuth2AccessTokenProvider
ConfluentSchemaRegistry	KafkaRecordSink_2_6
CouchbaseClusterService	KerberosKeytabUserService
CouchbaseKeyValueLookupService	KerberosPasswordUserService
CouchbaseMapCacheClient	KerberosTicketCacheUserService
CouchbaseRecordLookupService	KuduLookupService
CSVReader	LoggingRecordSink
CSVRecordLookupService	MongoDBControllerService
CSVRecordSetWriter	MongoDBLookupService
DatabaseRecordLookupService	MQTTIIoTReader
DatabaseRecordSink	Neo4JCypherClientService
DatabaseTableSchemaRegistry	ParquetReader
DeveloperBoxClientService	ParquetRecordSetWriter
DBCPCConnectionPool	PEMEncodedSSLContextProvider
DBCPCConnectionPoolLookup	PhoenixThickConnectionPool
DistributedMapCacheClientService	PhoenixThinConnectionPool
DistributedMapCacheLookupService	PostgreSQLConnectionPool
DistributedMapCacheServer	PrometheusRecordSink
DistributedSetCacheClientService	PropertiesFileLookupService

Related Information

[Supported NiFi processors](#)

[Supported NiFi reporting tasks](#)

[Supported NiFi parameter providers](#)

[Supported NiFi flow analysis rules](#)[Supported NiFi flow registry clients](#)

Supported NiFi reporting tasks

Review the list of Apache NiFi reporting tasks supported in Cloudera Flow Management 4.13.0.

Cloudera Flow Management is based on Apache NiFi and includes a set of reporting tasks, most of which are supported by Cloudera. To ensure optimal performance and reliable support, it is crucial to use only supported reporting tasks and avoid deploying unsupported ones in production environments.

- `ControllerStatusReportingTask`
- `MonitorDiskUsage`
- `MonitorMemory`
- `OpenTelemetryReportingTask`
- `QueryNiFiReportingTask`
- `ReportLineageToAtlas`
- `ScriptedReportingTask`
- `SiteToSiteBulletinReportingTask`
- `SiteToSiteMetricsReportingTask`
- `SiteToSiteProvenanceReportingTask`
- `SiteToSiteStatusReportingTask`

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.

Related Information

[Supported NiFi processors](#)[Supported NiFi controller services](#)[Supported NiFi parameter providers](#)[Supported NiFi flow analysis rules](#)[Supported NiFi flow registry clients](#)

Supported NiFi parameter providers

Review the list of Apache NiFi parameter providers supported in Cloudera Flow Management 4.13.0.

Cloudera Flow Management is shipped with Apache NiFi and includes a set of parameter providers, most of which are supported by Cloudera. To ensure optimal performance and reliable support, it is crucial to use only supported parameter providers and avoid deploying unsupported ones in production environments.

- `AwsSecretsManagerParameterProvider`
- `AzureKeyVaultSecretsParameterProvider`
- `ClouderaOnCloudParameterProvider`
- `CyberArkConjurParameterProvider`
- `DatabaseParameterProvider`
- `EnvironmentVariableParameterProvider`
- `GcpSecretManagerParameterProvider`
- `HashiCorpVaultParameterProvider`
- `KubernetesSecretParameterProvider`
- `OnePasswordParameterProvider`
- `PropertiesFileParameterProvider`

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.

Related Information

[Supported NiFi processors](#)

[Supported NiFi controller services](#)

[Supported NiFi reporting tasks](#)

[Supported NiFi flow analysis rules](#)

[Supported NiFi flow registry clients](#)

Supported NiFi flow analysis rules

Review the list of Apache NiFi processors supported in Cloudera Flow Management 4.13.0.

Flow Analysis Rules allow analyzing components or parts of a flow to help maintain optimal design. Rules can be set as Recommendations (informational only) or Policies (enforceable). Recommendation violations are logged but do not affect functionality, while Policy violations invalidate components until resolved.

- `DisallowComponentType`
- `DisallowConsecutiveConnectionsWithRoundRobinLB`
- `DisallowDeadEnd`
- `DisallowDeprecatedProcessor`
- `DisallowExtractTextForFullContent`
- `RecommendRecordProcessor`
- `RequireHandleHttpResponseAfterHandleHttpRequest`
- `RequireLoadBalancedConnectionAfterPrimaryNodeSourceProcessor`
- `RequireMergeBeforePutIceberg`
- `RequireServerSSLContextService`
- `RestrictBackpressureSettings`
- `RestrictComponentNaming`
- `RestrictConcurrentTasksVsThreadPoolSizeInProcessors`
- `RestrictFlowFileExpiration`
- `RestrictProcessorConcurrency`
- `RestrictSchedulingForListProcessors`
- `RestrictThreadPoolSize`
- `RestrictYieldDurationForConsumeKafkaProcessors`

Related Information

[Supported NiFi processors](#)

[Supported NiFi controller services](#)

[Supported NiFi reporting tasks](#)

[Supported NiFi parameter providers](#)

[Supported NiFi flow registry clients](#)

Supported NiFi flow registry clients

Review the list of Apache NiFi flow registry clients supported in Cloudera Flow Management 4.13.0.

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.

- `AzureDevOpsFlowRegistryClient` [Technical Preview]
- `BitbucketFlowRegistryClient`

- ClouderaDataFlowRegistryClient
- ClouderaFlowLibraryFlowRegistryClient
- GitHubFlowRegistryClient
- GitLabFlowRegistryClient
- NifiRegistryFlowRegistryClient

Related Information

[Supported NiFi processors](#)

[Supported NiFi controller services](#)

[Supported NiFi reporting tasks](#)

[Supported NiFi parameter providers](#)

[Supported NiFi flow analysis rules](#)

Supported Cloudera exclusive components

Review the list of Cloudera exclusive components supported in Cloudera Flow Management 4.13.0.

Cloudera Flow Management provides a set of NiFi components available only to Cloudera customers. These components provide additional functionality and are tailored to enhance the Cloudera NiFi experience. The list of these components is provided below.

Processors

- CaptureChangeDebeziumDB2
- CaptureChangeDebeziumMongoDB
- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- ConsumeKafka2CDP
- ConsumeKafka2RecordCDP
- ConsumeMQTTIIoT
- ConsumePLC
- ConvertProtobuf
- DeleteCDPObjectStore
- FetchPLC
- GetJiraIssue
- InvokeGRPC
- ListCDPObjectStore
- ListenGRPC
- ListenNetFlow
- PromptAzureOpenAI
- 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
- ClouderaAttributeSchemaReferenceReader
- ClouderaAttributeSchemaReferenceWriter
- ClouderaEncodedSchemaReferenceReader
- ClouderaEncodedSchemaReferenceWriter
- ClouderaSchemaRegistry
- CMLLookupService
- EBCDICRecordReader
- GenericPLC4XConnectionPool
- ImpalaConnectionPool
- IPFIXReader
- JiraRecordSink
- MQTTIIoTReader
- PhoenixThickConnectionPool
- PhoenixThinConnectionPool
- PostgreSQLConnectionPool
- ProxyPLC4XConnectionPool
- RabbitMQJMSConnectionFactoryProvider
- RedshiftConnectionPool
- RESTCatalogService
- StandardPLC4XConnectionPool

Parameter providers

- ClouderaOnCloudParameterProvider
- CyberArkConjurParameterProvider
- PropertiesFileParameterProvider

Flow registry clients

- ClouderaDataFlowRegistryClient
- ClouderaFlowLibraryFlowRegistryClient

Download locations

You can download the Cloudera Flow Management software artifacts from the Cloudera Archive. Artifacts are available for different operating systems, deployment types, and components.

Use the following tables to select the latest available Cloudera Flow Management 4.13.0 artifacts that match your operating system and deployment requirements.



Note:

You must have valid credentials to access and download the Cloudera Flow Management files. These credentials are different from those you use to access the Cloudera Support Portal.

You can obtain download credentials in one of the following ways:

- Contact your Cloudera sales representative.
- Check the welcome email you received for your Cloudera Flow Management account.
- File a non-technical case through the [Cloudera Support Portal](#) and ask Cloudera Support for assistance.

Cloudera Flow Management download links

Table 3: RHEL 8

File	Location
Manifest	manifest.json
Parcel	CFM-4.13.0.0-959-el8.parcel
Parcel sha file	CFM-4.13.0.0-959-el8.parcel.sha
CSD	NiFi: NIFI-2.11.0.4.13.0.0-959.jar NiFi Registry: NIFIREGISTRY-2.11.0.4.13.0.0-959.jar

Table 4: RHEL 9

File	Location
Manifest	manifest.json
Parcel	CFM-4.13.0.0-959-el9.parcel
Parcel sha file	CFM-4.13.0.0-959-el9.parcel.sha
CSD	NiFi: NIFI-2.11.0.4.13.0.0-959.jar NiFi Registry: NIFIREGISTRY-2.11.0.4.13.0.0-959.jar

Table 5: SLES 15

File	Location
Manifest	manifest.json
Parcel	CFM-4.13.0.0-959-sles15.parcel
Parcel sha file	CFM-4.13.0.0-959-sles15.parcel.sha

File	Location
CSD	NiFi: NIFI-2.11.0.4.13.0.0-959.jar NiFi Registry: NIFIREGISTRY-2.11.0.4.13.0.0-959.jar

Table 6: Ubuntu 22

File	Location
Manifest	manifest.json
Parcel	CFM-4.13.0.0-959-jammy.parcel
Parcel SHA file	CFM-4.13.0.0-959-jammy.parcel.sha
CSD	NiFi: NIFI-2.11.0.4.13.0.0-959.jar NiFi Registry: NIFIREGISTRY-2.11.0.4.13.0.0-959.jar

Table 7: Ubuntu 24

File	Location
Manifest	manifest.json
Parcel	CFM-4.13.0.0-959-noble.parcel
Parcel SHA file	CFM-4.13.0.0-959-noble.parcel.sha
CSD	NiFi: NIFI-2.11.0.4.13.0.0-959.jar NiFi Registry: NIFIREGISTRY-2.11.0.4.13.0.0-959.jar

Table 8: Windows

File	Location
NiFi MSI	nifi-4.13.0.0-959.msi
NiFi MSI SHA file	nifi-4.13.0.0-959.msi.sha256

Cloudera Flow Migration Tool download links

The available Cloudera Flow Migration Tool versions do not support migration to Cloudera Flow Management 4.13.0. A new release of the Migration Tool is in development to support this migration target.

Select the Cloudera Flow Migration Tool version that supports both your source and target Cloudera Flow Management versions.

Cloudera Cloudera Flow Migration Tool version	Source Cloudera Flow Management version	Target Cloudera Flow Management version	Cloudera Flow Migration Tool repository
7.0.2	2.1.7.300x	4.12.0.x	Cloudera Flow Migration Tool 7.0.2 repository
7.0.1		4.12.0.x	Cloudera Flow Migration Tool 7.0.1 repository
7.0.0			Cloudera Flow Migration Tool 7.0.0 repository

Unsupported features

There are no unsupported features in Cloudera Flow Management 4.13.0.

Behavioral changes

Learn about behavioral changes in Cloudera Flow Management 4.13.0.

Iceberg column defaults

PutIceberg and PutIcebergCDC no longer have the UnmatchedColumnBehavior property. When a record omits a column, the processors use the default value defined in the Iceberg table schema, where applicable. Review flows that previously relied on this property.

Cloud service authentication

When configuring GetAzureEventHub, PutAzureEventHub, or GCPCredentialsControllerService, you must now explicitly select an authentication strategy or type. Review existing configurations when upgrading.

Known issues in Cloudera Flow Management

Review the list of known issues and limitations in Cloudera Flow Management 4.13.0.

Known issues

CFM-7259 - PartitionPdf Python processor fails to initialize on Ubuntu 22

On Ubuntu 22 hosts, the PartitionPdf Python processor fails to initialize with the following error:

```
ImportError: ERROR: recursion is detected during loading of "cv2"
binary extensions. Check OpenCV installation.
```

The failure occurs when the native OpenGL library (libGL.so.1), required by the processor's OpenCV (cv2) dependency, is missing. Other Python processors are unaffected.

Install the libgl1 package on all affected Ubuntu 22 NiFi hosts before starting NiFi or creating a PartitionPdf processor: `sudo apt-get update && sudo apt-get install -y libgl1`

If NiFi is already running, restart the NiFi service or allow the processor to reinitialize its Python environment after installing the package.

Limitations

Authentication, SSL, and proxy limitations in PutSnowflakeStreaming

The PutSnowflakeStreaming processor in Cloudera Flow Management 4.13.0 has the following limitations:

- Authentication: Only key-pair (private key) authentication is supported. OAuth 2.0 authentication is not supported.
- SSL configuration: Custom SSL configuration through processor properties or NiFi's SSLContextService is not supported.
- Proxy configuration: Proxy configuration through processor properties or NiFi's ProxyConfigurationService is not supported.

Use key-pair authentication and ensure that the NiFi hosts can connect directly to the required Snowflake endpoints.

Fixed issues in Cloudera Flow Management

Review the list of issues resolved in Cloudera Flow Management 4.13.0.

CFM-7168 - NiFi extension loading

Fixed an archive packaging issue that truncated file paths longer than 100 characters. Affected NiFi Archive (NAR) extensions could be missing or fail to load after installation, causing `ClassNotFoundException` errors at startup.

CFM-6390 - Apache Phoenix JDBC drivers

Fixed dependency and driver loading errors that prevented the Phoenix Thick and Thin drivers from initializing or caused SQL queries to fail.

CFM-7078 - Kafka connections over TLS

Fixed certificate path validation errors that prevented `PublishKafka2CDP` and `ConsumeKafka2CDP` from connecting to Kafka over TLS.

CFM-7259 - Resolved PartitionPdf initialization failure caused by missing OpenCV dependencies on Ubuntu 22

Fixed an issue that caused the `PartitionPdf` Python processor to fail to initialize on Ubuntu 22 hosts when the native OpenGL library (`libGL.so.1`) was missing. The failure produced an `OpenCV (cv2)` binary extension recursion error. The processor now initializes and partitions PDF documents successfully on Ubuntu 22.

Fixed CVEs in Cloudera Flow Management

Review the list of Common Vulnerabilities and Exposures (CVE) resolved in Cloudera Flow Management 4.13.0.

[CVE-2026-68981](#): Apache NiFi: Uncontrolled Resource Consumption through Decompression of HTTP Requests

Apache NiFi 1.5.0 through 2.10.0 support gzip-encoded HTTP requests for the application REST API using a Jersey encoding filter. The framework enforced a configurable maximum request size on the compressed payload rather than the decompressed output, allowing a malicious client to send crafted requests that could consume excessive amounts of memory. Upgrading to Apache NiFi 2.11.0 is the recommended mitigation, which relocates response compression to Jetty Server and disables decompression of gzip-encoded HTTP requests.

[CVE-2026-68980](#): Apache NiFi: Authorization Bypass for Parameter Context Asset Deletion

Apache NiFi 2.0.0 through 2.10.0 support creating, reading, and deleting Assets associated with Parameter Contexts through the REST API. The framework authorizes asset deletion against the owning Parameter Context using the supplied Parameter Context Identifier and Asset Identifier. The framework performed authorized based on the supplied Parameter Context Identifier without verifying the requested Identifier against the stored Identifier. Apache NiFi installations that do not implement different levels of authorization across Parameter Contexts are not subject to this vulnerability, because the framework enforces write permissions as the security boundary. Upgrading to Apache NiFi 2.11.0 is the recommended mitigation, which verifies Parameter Context ownership of the requested Asset before deletion using the same strategy applied to Asset read operations.

[CVE-2026-62354](#): Apache NiFi: Incorrect Authorization for Parameter Context Validation Requests

Authorization handling for Parameter Context validation requests in Apache NiFi 1.10.0 through 2.10.0 allows clients with read access to submit proposed Parameter values. The proposed values override current configuration, enabling users with read access to invoke predefined component validation methods with alternative settings. Apache NiFi installations that do not implement different levels of authorization for viewing and modifying Parameter Context configuration are not subject to this vulnerability. Upgrading to Apache NiFi 2.11.0 is the recommended mitigation, requiring write access to submit Parameter Context validation requests.

CVE-2026-68979: Apache NiFi: Missing Authorization for Components Referenced by Parameter Context Updates

Apache NiFi 1.10.0 through 2.10.0 provide a Parameter Context update REST API method that does not enforce authorization checking on components referencing Parameter values. Updating a Parameter Context can change parameter values that affect referencing components, but framework authorization was limited to read and write privileges on the Parameter Context itself. As a result of the missing authorization, an authenticated user authorized to modify a Parameter Context, but not authorized on referencing components, could alter Parameter values affecting those components. In deployments where a Parameter value contains executable scripting content, updating a Parameter can result in code execution during automatic component validation, without starting the referencing component. The impact was limited to stopped components by existing verification checks, and the issue applies only to deployments that use component-level authorization policies. Upgrading to Apache NiFi 2.11.0 is the recommended mitigation, which aligns the Parameter Context update method authorization with other methods, adding authorization checking on affected components.

CVE-2026-87976: Apache NiFi Registry: Improper Limitation of Pathname in Persisted Extension Bundles

Apache NiFi Registry 0.4.0 through 2.11.0 are subject to path manipulation when storing extension bundle content using group, artifact, and version coordinates from uploaded NAR manifests. The default file persistence provider used coordinates as filesystem path components without rejected parent-directory names, and the path-containment check compared an unnormalized resolved path. An authenticated user authorized to write and delete bundles in a bucket can upload a NAR with a crafted manifest resulting in file system operations outside of the file persistence directory. Upgrading to Apache NiFi Registry 2.12.0 is the recommended mitigation, which rejects parent-directory coordinates and requires a normalized path to remain a strict child of the storage root location.

CVE-2026-86089: Apache NiFi: Missing Process Group Authorization for Connector Migration

Apache NiFi 2.11.0 supports migrating the contents of a version-controlled Process Group into a Connector using REST API methods that list eligible migration sources and submit migration requests. The framework authorized both methods against the target Connector alone, without evaluating access to the Process Groups involved. The absence of Process Group authorization allowed an authenticated user with read access to a Connector to enumerate the identifiers, names, and flow registry details of version-controlled Process Groups outside the scope of granted read policies. It also allowed a user with write access to a Connector to migrate a Process Group without write access to that Process Group, copying the flow definition, referenced assets, and component state into the Connector, and leaving the source Process Group disabled and renamed. Migration excludes sensitive property values and requires the source Process Group to be stopped with empty queues, which limits the scope of exposure. Apache NiFi installations that do not implement component-level authorization policies for Process Groups are not subject to this vulnerability, because the framework enforces Connector write permissions as the security boundary. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which filters migration sources to Process Groups the requesting user is authorized to read, and requires write access to the source Process Group when submitting a migration request.

CVE-2026-82561: Apache NiFi: Missing Authorization for Components Referenced in Flow Update Methods

Apache NiFi 1.5.0 through 2.11.0 provide REST API methods that replace the entire contents of a Process Group using a client-supplied flow definition, covering Process Group flow replacement together with versioned flow update and rebase operations. Framework authorization for these methods was limited to read and write privileges on the Process Group itself, unlike the corresponding asynchronous update request methods, which also authorize the components encapsulated in the Process Group along with referenced Controller Services, Parameter Contexts, and Parameter Providers. As a result of the missing authorization, an authenticated user with write access to a Process Group could supply a flow definition that modifies or removes components in descendant Process Groups protected by more restrictive access policies, and could bind components to Controller Services and Parameter Contexts without authorization for those referenced components. Existing verification checks limited the impact to stopped components, and the issue applies only to deployments that use component-level authorization policies, because the framework enforces write permissions as the security boundary. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which applies consistent reference resolution and component authorization across Process Group replacement and versioned flow update methods.

CVE-2026-81866: Apache NiFi: Missing Authorization for Assets and Secrets Referenced by Connector Configuration

Apache NiFi 2.9.0 through 2.11.0 provide Connector configuration update and verification REST API methods that do not enforce authorization checking on Assets and Secrets referenced in proposed configuration. Updating or

verifying a Connector configuration step can apply Asset and Secret references, but framework authorization was limited to write privileges on the Connector itself. As a result of the missing authorization, an authenticated user authorized to modify a Connector, but not authorized to read a referenced Parameter Provider, could apply Secret values backed by that Parameter Provider. The same methods also accepted Asset identifiers without verifying that the Asset belonged to the Connector being configured. Apache NiFi installations that do not implement different levels of authorization across Connectors and Parameter Providers are not subject to this vulnerability, because the framework enforces write permissions on the Connector as the security boundary. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which authorizes read access to referenced Parameter Providers and verifies Connector ownership of referenced Assets during configuration update and verification.

CVE-2026-70469: Apache NiFi: Improper Handling of Case Sensitivity for Content-Encoding in HTTP Requests

Apache NiFi 2.11.0 disabled support for gzip-encoded HTTP requests for the application REST API and rejected requests that included the standard Content-Encoding header indicating gzip encoding. The framework enforcement filter did not check multiple instances of the Content-Encoding header and did not reject non-standard identifiers for gzip encoding, allowing a malicious client to send crafted requests that could consume excessive amounts of memory. Upgrading to Apache NiFi 2.12.0 is the recommended mitigation, which disables decompression of gzip-encoded HTTP requests regardless of header number or encoding identifiers.