Cloudera DataFlow for Data Hub 7.2.18

Cloudera DataFlow for Data Hub Release Notes

Date published: 2019-12-16
Date modified: 2024-04-03

CLOUDERA

Legal Notice

Contents

What's New in Cloudera DataFlow for Data Hub 7.2.18

Cloudera DataFlow for Data Hub 7.2.18 includes components for Flow Management, Streaming Analytics, and Streams Messaging. Learn about the new features and improvements in each of these components.

What's new in Flow Management with NiFi 1

Learn about the new features of Flow Management using NiFi 1.25 in Cloudera DataFlow for Data Hub 7.2.18.

Flow Management Data Hub in CDP Public Cloud 7.2.18 is compatible with both NiFi 1 and NiFi 2. This section provides details about Flow Management Data Hub based on Apache NiFi 1.25.0. Below are the most important new features, improvements, and fixes included in this release:

Rebase on NiFi 1.25

> The upgrade offers access to the newest Apache NiFi features and enhancements on the 1.x branch. Transitioning to NiFi 2.x later oon has the prerequisite of already using NiFi 1.25.
>
> ⚠️ **Important:** Direct upgrades from NiFi 1.25 to NiFi 2.0 are currently not supported. You can create new Flow Management clusters based on NiFi 2 (Technical Preview) using the new templates and migrate your existing flows to the new clusters.

New components

- Processors

  - CalculateParquetOffsets and CalculateParquetRowGroupOffsets

    These processors can be used in combination with ConvertRecord and Parquet Reader to significantly reduce the amount of time required to convert very large Parquet files into another format.

  - CaptureChangeDebeziumDB2, CaptureChangeDebeziumMySQL, CaptureChangeDebeziumOracle, CaptureChangeDebeziumPostgreSQL, CaptureChangeDebeziumSQLServer

    These processors, currently in Technical Preview, leverageare the Debezium project to ingest Change Data Capture events from external databases.

  - DecryptContentAge and EncryptContentAge

    These are a new generation of processors for data encryption / decryption. For more information, see Modernizing Streaming Encryption with age in Apache NiFi.

  - ListenNetFlow

    This processor enables NiFi to receive Netflow data from network equipment. For more information, see Collecting NetFlow Records with Cloudera DataFlow.

  - ListenOTLP

    This processor allows NiFi to function as a destination for OLTP agents to receive OpenTelemetry data from external applications. You can learn more about this new processor Building OpenTelemetry Collection in Apache NiFi with Netty.

  - PutAzureQueueStorage_v12 and GetAzureQueueStorage_v12

    These processors use the latest library for interacting with Azure Queue Storage. It is highly recommended to switch to these new processors.

  - PutClouderaHiveQL, PutClouderaHiveStreaming, PutClouderaORC, SelectClouderaHiveQL, UpdateClouderaHiveTable, TriggerClouderaHiveMetaStoreEvent

    This is a set of Cloudera exclusive components to interact with Hive-based components in the Cloudera Data Platform. It is highly recommended to switch to these components as they leverage features that are part of Hive 4, not yet released in the Apache project.

  - PutIcebergCDC

    This processor in Technical Preview can be used in combination with the new Debezium processors to implement CDC pipelines with Iceberg tables as the destination.

  - ExtractRecordSchema

    This processor should be used as a replacement of the InferAvroSchema processor, which will be deprecated in NiFi 2.

  - PutJiraIssue
  - PutZendeskTicket
  - QueryIoTDBRecord
  - RemoveRecordField
  - ConsumeElasticsearch
  - FilterAttribute
  - PackageFlowFile
  - PublishSlack

- Controller services
  - ActiveMQJMSConnectionFactoryProvider

    This controller service allows you to interact with ActiveMQ without the need to deploy the JMS client on all of the NiFi nodes.
  - ADLSCredentialsControllerServiceLookup
  - AmazonGlueSchemaRegistry
  - AzureServiceBusJMSConnectionFactoryProvider

    This controller service allows you to interact with Azure Service Bus without the need to deploy the required dependencies on all of the NiFi nodes.
  - AzureStorageCredentialsControllerServiceLookup_v12
  - ClouderaHiveConnectionPool

    This controller service allows you to interact with Hive without the need to deploy the required dependencies on all of the NiFi nodes.
  - CMLLookupService

    This controller service can be used to enrich the data going through NiFi by calling exposed Machine Learning models running in Cloudera Machine Learning.
  - DatabaseTableSchemaRegistry

    This controller service enables you to retrieve the schema associated with a table from an external database. This allows you to validate the data going through NiFi against that schema before pushing the data into this table.
  - EBCDICRecordReader

    This controller service allows you to read and convert Mainframe data into another structured format like JSON, Avro, and so on. For more information, see *here*.
  - ExcelReader
  - ImpalaConnectionPool

    This controller service allows you to interact with Impala without the need to deploy the required dependencies on all NiFi nodes.
  - JiraRecordSink
  - RabbitMQJMSConnectionFactoryProvider

    This controller service allows you to interact with RabbitMQ without the need to deploy the JMS client on all NiFi nodes.
  - RedshiftConnectionPool

    This controller service allows you to interact with Redshift over JDBC without the need to care about deploying the JDBC driver on all of the NiFi nodes.
  - SimpleRedisDistributedMapCacheClientService
  - StandardFileResourceService
  - StandardJsonSchemaRegistry
  - YamlTreeReader
  - ZendeskRecordSink
- Parameter provider
  - CyberArkConjurParameterProvider

    This parameter provider allows you to retrieve the value associated with your parameters from an external CyberArk Conjur instance.

What's new in Flow Management with NiFi 2 [Technical Preview]

Learn about the new features of Flow Management using NiFi 2.0 in Cloudera DataFlow for Data Hub 7.2.18.

Flow Management Data Hub in CDP Public Cloud 7.2.18 is compatible with both NiFi 1 and NiFi 2. This section provides details about Flow Management Data Hub based on Apache NiFi 2.0. Two new Data Hub templates are introduced in Technical Preview for deploying Flow Management clusters using NiFi 2.0.

Currently there is no upgrade path from Flow Management clusters based on NiFi 1 to clusters with NiFi 2. The only approach is to start new clusters using the new templates and migrate your existing flows to these new clusters. However, these new clusters are not production ready and should not be used for critical workloads.

There are a significant number of changes, including breaking ones, between NiFi 1 and NiFi 2. See Behavioral changes for more information about these changes. Additionally, you can anticipate further breaking changes in the upcoming releases, particularly concerning components completely removed in favor of better and more efficient alternatives.

Here are the most important new features and other significant improvements of this release:

Python API - Technical Preview

There is now a first-class citizen Python API allowing users to develop NiFi processors using Python. For more information, see the NiFi Python Developer's Guide.

> ⚠️ **Warning:** An invalid Python version is defined for the NiFi service, which affects the usability of Python based NiFi processors. For more information and workaround, see Known issues in Flow Management.

Stateless Engine at Process Group level - Tech Preview

It is now possible to configure a Process Group to use the Stateless Engine for executing flows. This is particularly useful for transactional use cases such as CDC, or scenarios where a message broker is the source, aiming to achieve exactly-once semantics. For more information, see the Apache NiFi User Guide.

Rules Engine - Technical Preview

This feature allows NiFi administrators to define a set of rules to enforce best practices in flow design in NiFi. The feature version available in this release is not complete, it is still under development. More details will be available once the feature is finalized and released.

New NiFi components

The below list contains the components that are added specifically in Flow Management 7.2.18 clusters using NiFi 2 compared to Flow Management 7.2.18 clusters using NiFi 1. A large number of components have been added between 7.2.17 and 7.2.18. For a comprehensive list, see What's new in Flow Management with NiFi 1.

- Processors

  - ParseDocument & ChunkDocument

    These processors handle documents and extract text content to help with Generate AI use cases.
  - ConsumePLC & PutPLC & FetchPLC

    These processors based on the Apache PLC4X project interact with systems over common IoT protocol systems such as Siemens S7, Modbus, CAN Open, Allen Bradley ETH, OPC-UA and more.
  - ConsumeSlack
  - CopyAzureBlobStorage_v12
  - ListenSlack
  - PromptChatGPT
  - PutChroma & QueryChroma

    These processors interact with the ChromaDB vector database and compute embeddings while pushing documents.
  - PutMongoBulkOperations
  - PutPinecone & QueryPinecone

    These processors interact with the Pinecone vector database and compute embeddings while pushing documents.
  - QueryAzureDataExplorer
  - RenameRecordField
- Controller services

  - ApicurioSchemaRegistry
  - ConfluentEncodedSchemaReferenceReader
  - ConfluentEncodedSchemaReferenceWriter
  - GCSFileResourceService
  - GenericPLC4XConnectionPool
  - ProxyPLC4XConnectionPool
  - SlackRecordSink
  - StandardKustoQueryService
  - StandardPLC4XConnectionPool
  - TinkerpopClientService
- Parameter providers

  - OnePasswordParameterProvider
- Rules

  - DisallowComponentType

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 (CDF) for Data Hub 7.2.18 in CDP Public Cloud.

Edge Flow Management cluster definitions have been updated with Cloudera Edge Management (CEM) 2.0.0 functionalities, providing an enhanced user experience. Edge Flow Manager now integrates with CDP User Management, making it easier to manage users and groups. For more information about the integration between Edge Flow Manager and CDP 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 7.2.18.

Kafka

Rebase on Kafka 3.4.1

> Kafka shipped with this version of Cloudera Runtime is based on Apache Kafka 3.4.1. For more information, see the following upstream resources:
>
> Apache Kafka Notable Changes:
>
> - 3.2.0
> - 3.3.0 and 3.3.1
> - 3.4.0
>
> Apache Kafka Release Notes:
>
> - 3.2.0
> - 3.3.0
> - 3.3.1
> - 3.4.0
> - 3.4.1

Kafka log directory monitoring improvements

> A new Cloudera Manager chart, trigger, and action is added for the Kafka service. These assist you in monitoring the log directory space of the Kafka Brokers, and enable you to prevent Kafka disks from filling up.
>
> The chart is called Log Directory Free Capacity. It shows the capacity of each Kafka Broker log directory.
>
> The trigger is called Broker Log Directory Free Capacity Check. It is triggered if the capacity of any log directory falls below 10%. The trigger is automatically created for all newly deployed Kafka services, but must be created with the Create Kafka Log Directory Free Capacity Check action for existing services following an upgrade.
>
> The chart and trigger are available on the Kafka service Status page. The action is available in Kafka service Actions .

Kafka is safely stopped during operating system upgrades

> During OS upgrades, Cloudera Manager now ensures that Kafka brokers are safely stopped. Specifically, Cloudera Manager now performs a rolling restart check before stopping a broker. This ensures that the Kafka service stays healthy during the upgrade. The level of health guarantee that Cloudera Manager ensures is determined by the restart check type set in the Cluster Health Guarantee During Rolling Restart Kafka property. Cloudera recommends that you set this property to all partitions stay healthy to avoid service outages. For more information, see Rolling restart checks.

useSubjectCredsOnly set to true by default in Kafka Connect

> In previous versions, the javax.security.auth.useSubjectCredsOnly JVM property was set to false in Kafka Connect. Because of this, connectors running with an invalid or no JAAS configuration could use the credentials of other connectors to establish connections. Starting with this release, useSubjectCredsOnly is set to true by default. As a result, connectors are required to use their own credentials.
>
> This default change is true for newly provisioned clusters. On upgraded clusters, useSubjectCredsOnly remains set to false to ensure backwards compatibility. If you are migrating connectors from a cluster running a previous version of Runtime to a new cluster running 7.2.18 or later, you

must ensure that credentials are added to the connector configuration when migrated. Otherwise, migrated connectors may not work on the new cluster.

In addition to the default value change, a new Kafka Connect property is introduced in Cloudera Manager that you can use to set useSubjectCredsOnly. The property is called Add Use Subject Credentials Only JVM Option With True Value. Setting this property to false does not expressly set useSubjectCredsOnlyto false. Instead, it sets useSubjectCredsOnly to the cluster default value.

Kafka Connect metrics reporter security configurable in Cloudera Manager

New, dedicated Cloudera Manager properties are introduced for the security configuration of the Kafka Connect metrics reporter. As a result, you are no longer required to use advanced security snippets if you want to secure the metrics reporter and its endpoint. The new properties introduced are as follows:

- Secure Jetty Metrics Port
- Enable Basic Authentication for Metrics Reporter
- Jetty Metrics User Name
- Jetty Metrics Password

A dedicated property to enable TLS/SSL for the metrics reporter is not available. Instead, you must select Enable TLS/SSL for Kafka Connect which enables TLS/SSL for the Kafka Connect role including the metrics reporter. For more information regarding these properties, see Cloudera Manager Configuration Properties Reference.

As a result of these changes, the setup steps required to configure Prometheus as the metrics store for SMM are changed. For updated deployment instructions, see Setting up Prometheus for Streams Messaging Manager.

Kafka load balancer is automatically configured with the LDAP handler if LDAP authentication is configured

When a load balancer and LDAP authentication is configured for Kafka, the PLAIN mechanism is automatically added to the enabled authentication mechanisms of the load balancer listener. Additionally, the load balancer is automatically configured to use `LdapPlainServerCallbackHandler` as the callback handler.

Kafka Connect now supports Kerberos auth-to-local (ATL) rules with SPNEGO authentication

Kafka Connect now uses the cluster-wide Kerberos auth-to-local (ATL) rules by default. A new configuration property called Kafka Connect SPNEGO Auth To Local Rules is introduced. This property is used to manually specify the ATL rules. During an upgrade, the property is set to DEFA ULT to ensure backward compatibility. Following an upgrade, if you want to use the cluster-wide rules, clear the existing value from the Kafka Connect SPNEGO Auth To Local Rules property.

Schema Registry

AvroConverter support for KConnect logical types

AvroConverter now converts between Connect and Avro temporal and decimal types.

Support for alternative jersey connectors in SchemaRegistryClient

connector.provider.class can be configured in Schema Registry Client. If it is configured, schema.r egistry.client.retry.policy should also be configured to be different than default.

This also fixes the issue with some third party load balancers where the client is expected to follow redirects and authenticate while doing that.

Remove modules section from registry.yaml

In previous versions, the registry.yaml configuration file contained a modules section. This section was used to list pluggable modules that extended Schema Registry's functionality. However, modules were never fully supported and have been removed in a previous release. The modules section in registry.yaml was kept for backwards compatibility. Starting with this version, the modules section is removed by default from registry.yaml.

Upgraded Avro version to 1.11.1

Avro got upgraded from version 1.9.1 to 1.11.1.

New fingerprint version is added to Schema Registry with configuring option

A new fingerprint version, V2 is available in Schema Registry that contains the missing schema parts from the previous version. Newly created 7.2.18 clusters use the V2 fingerprint version. Upgraded clusters still use the V1 fingerprint version, but the schema.registry.fingerprint.version property can be used to change the fingerprint version in Schema Registry. Cloudera recommends to change the fingerprint version to V2 after upgrading to 7.2.18.

Support for additional JVM options

Additional JVM options can be passed to Schema Registry using the schema.registry.additional.java.options property in Cloudera Manager.

Streams Messaging Manager

UI updates

The style of SMM UI is updated. This update includes various changes to the colors, fonts, and overall style of the UI. Additionally, the following functional changes and improvements are made:

Cruise Control 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 SMM using the 🖥 on the navigation sidebar.

For more information about Cruise Control in SMM, see Monitoring and managing Kafka cluster rebalancing.

Data Explorer

- When you view Avro data in the Data Explorer, logicalTypes are converted by default. That is, instead of showing the underlying type, (for example, byte) the Data Explorer displays proper deserialized values.
- Avro messages are now pretty printed when you open them using the Show More option.

Kafka Connect

- Hovering over the status icons of connector tasks now displays the status text instead of the name of the icon.
- The Add missing configurations option now populates missing properties with default values.
- Adding flow.snapshot into a key field of a password type property clears password placeholders.
- The value field of the flow.snapshot property is now always a text area. Previously, if the property was added manually, the value field was a text field instead of an area.
- Text found in the ⑦ (Help) tooltip of connector property values now displays properly. Long strings no longer overflow the tooltip. Additionally, property descriptions are truncated. Clicking ⑦ or more... displays a pop-up containing the full description.
- Validating a connector configuration when the Kafka service is stopped returns a Connection refused error instead of validation passing.
- Deploying a new connector that has the same name as an existing connector no longer updates the existing connector. Instead, the connector deployment fails with Connector [***NAME***] already exists.
- A loading animation is displayed when loading connector templates.

- The horizontal divider found in the ⋮ context menu of properties is no longer displayed if the **ABC** String , ⬭ Boolean, **123** Number, and ⌀ Password options are not available for the property

  For more information regarding the various new features and options related to Kafka Connect, see Managing and monitoring Kafka Connect using Streams Messaging Manager.

  Other

  All charts present on the UI received a visual update. Additionally, more details are presented about the data they display.

Changes in Prometheus setup and configuration

Kafka Connect is now capable of securing its metric reporter with TLS/SSL and Basic Authentication. As a result, the setup steps required to configure Prometheus as the metrics store for SMM are changed. For updated deployment instructions, see Setting up Prometheus for Streams Messaging Manager.

SMM internal Kafka topics are created with a replication factor of 3

From now on the __smm* internal SMM topics are created with a replication factor of 3. This change is only true for newly deployed clusters. The replication factor is not updated during the upgrade. Cloudera recommends that you increase the replication factor of these topics to 3 with kafka-reassign-partitions following an upgrade.

Remove keystore from SMM Schema Registry client configuration if Kerberos is enabled for Schema Registry

SMM uses a Schema Registry client to fetch schemas from Schema Registry. This Schema Registry client has Kerberos authentication properties and keystore properties for mTLS. Typically, the Schema Registry server, by default, does not allow mTLS authentication. But if mTLS is enabled in the Schema Registry server, then mTLS authentication has a higher precedence than Kerberos. Therefore, the mTLS principal (from the keystore) is used for authorization with Ranger rather than the Kerberos principal. This might result in authorization failures if the mTLS principal is not added to Ranger to access the Schema Registry resources.

From now on, the Schema Registry client used by SMM does not have keystore properties for mTLS when Kerberos is enabled. As a result, even if mTLS is enabled for the Schema Registry server, the Kerberos principal is used for authentication and authorization with Ranger.

Dedicated endpoint for connector creation

A dedicated endpoint for connector creation is introduced. The endpoint is `POST /api/v1/admin/kafka-connect/connectors`. Requests made to this endpoint fail with the following error message if the connector name specified in the request exists.

```
{
  "error_code": 409,
  "message": "Connector [***NAME***] already exists"
}
```

In previous versions, `PUT api/v1/admin/kafka-connect/connectors/{connector}` was the only endpoint you could use to create connectors. However, this endpoint is also used to update connectors. If you specify the name of an existing connector in the request, the endpoint updates existing connector. As a result, Cloudera recommends that you use the new endpoint for connector creation going forward. The SMM UI is also updated and uses the new endpoint for connector creation.

Jersey client timeout now configurable

SMM uses internal Jersey clients to make requests to Kafka Connect and Cruise Control. The connection and read timeouts for these clients was previously hard-coded to 30 seconds. Configuring them was not possible. This release introduces new properties, which enable you

to configure the connection and read timeouts of these clients. The default timeout remains 30 seconds. The properties introduced are as follows.

- Kafka Connect Client Connect timeout
- Kafka Connect Client Read timeout
- Cruise Control Client Connect timeout
- Cruise Control Client Read timeout

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.

Verification of internal metrics topics can now be disabled

A new property, Verify Partition Count Of The Metrics Topic is introduced for the SRM service. This property controls whether SRM verifies the partition count of the srm-metrics-*[***SOURCE CLUSTER ALIAS***]*.internal topics (raw metric topics) when SRM is started. This property is selected by default. Cloudera recommends that you keep this property selected. During certain upgrades, the property is set to false automatically for the duration of the upgrade to avoid upgrade issues.

Prefixless replication with the IdentityReplicationPolicy

Full support, including replication monitoring, is introduced for the `IdentityReplicationPolicy`. Unlike the `DefaultReplicationPolicy`, this policy does not rename remote (replicated) topics on target clusters. That is, the topics that you replicate will have the same name in both source and target clusters. This replication policy is recommended for deployments where SRM is used to aggregate data from multiple streaming pipelines. Alternatively, this replication policy can also be used if the deployment requires MirrorMaker1 (MM1) compatible replication.

Prefixless replication is enabled in Cloudera Manager with the Enable Prefixless Replication property. This property configures SRM to use the `IdentityReplicationPolicy` and enables internal topic based remote topic discovery, which is required for replication monitoring.

Limitations:

- Replication loop detection is not supported. As a result, you must ensure that topics are **not** replicated in a loop between your source and target clusters.
- The /v2/topic-metrics/{target}/{downstreamTopic}/{metric} endpoint of SRM Service v2 API does not work properly with prefixless replication. Use the /v2/topic-metrics/{source}/{target}/ {upstreamTopic}/{metric} endpoint instead.
- The replication metric graphs shown on the **Topic Details** page of the SMM UI do not work with prefixless replication.

> ⚠️ **Important:** If you have been using the `IdentityReplicationPolicy` in a previous version of Cloudera Runtime, ensure that you transition your configuration and set the `IdentityReplicationPolicy` with Enable Prefixless Replication. If you do not transition your configuration, replication monitoring will not function.

For more information, see

- [Streams Replication Manager replication flows and replication policies](#)
- [Enabling prefixless replication](#)

Internal topic based remote topic discovery

From now on, SRM uses an internal Kafka topic to keep track of remote (replicated) topics. Previously, SRM relied on the naming conventions (prefixes) used by the `DefaultReplicationPolicy` to discover and track remote topics.

This feature enables SRM to provide better monitoring insights on replications. Additionally, if the feature is enabled, SRM is capable of providing replication monitoring even if a

replication policy other than the `DefaultReplicationPolicy` is in use. Most notably, this enables replication monitoring when SRM is configured for prefixless replication with the `IdentityReplicationPolicy`.

This feature is enabled in Cloudera Manager by selecting the Remote Topics Discovery With Internal Topic property. The property is selected by default on newly deployed clusters, but must be enabled manually for existing clusters after an upgrade. Cloudera recommends that you enable this feature no matter what replication policy you are using.

For more information, see Streams Replication Manager remote topic discovery .

Configurations to customize replication-records-lag metric calculation

Three new properties are introduced that enable you to control how SRM calculates the replicat ion-records-lag metric. This metric provides information regarding the replication lag based on offsets. The metric is available both on the cluster and the topic level. The following new properties are introduced because the calculation of the metric with default configurations might add latency to replications and impact SRM performance. While these properties are configured in Cloudera Manager, they do not have dedicated configuration entries. Instead, you add them to Streams Replication Manager's Replication Configs to configure them.

Table 1:

Property	Default Value	Description
replication.records.lag.calc.enabled	true	Controls whether the replication-records-lag metric is calculated. This metric provides information regarding the replication lag based on offsets. The metric is available both on the cluster and the topic level. The calculation of this metric might add latency to replications and impact SRM performance. If you are experiencing performance issues, you can try setting this property to false to disable the calculation of replication-records-lag. Alternatively, you can try fine-tuning how SRM calculates replicat ion-records-lag with the replication. records.lag.calc.period.ms and repl ication.records.lag.end.offset.timeo ut.ms properties.
replication.records.lag.calc.period. ms	0	Controls how frequently SRM calculates the replication-records-lag metric. The default value of 0 means that the metric is calculated continuously. Cloudera recommends configuring this property to 15000 ms (15 seconds) or higher if you are experiencing issues related to the calculation of replication-records-lag. A calculation frequency of 15 seconds or more results in the metric being available for consumption without any significant impact on SRM performance.

Property	Default Value	Description
replication.records.lag.end.offset.timeout.ms	6000	Specifies the Kafka end offset timeout value used for calculating the replication-records-lag metric. Setting this property to a lower value than the default 6000 ms (1 minute) might reduce latency in calculating replication-records-lag, however, replication-records-lag calculation might fail. A value higher than the default can help avoid metric calculation failures, but might increase replication latency and lower SRM performance.

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 SMM using the on the navigation sidebar.

For more information about Cruise Control in SMM, see Monitoring and managing Kafka cluster rebalancing.

What's new in Streaming Analytics

Learn about the new Streaming Analytics features in Cloudera DataFlow for Data Hub 7.2.18.

The following new features are introduced in Streaming Analytics CDF for Data Hub 7.2.18:

Rebase to Apache Flink 1.18.0

Apache Flink 1.18.0 is supported in the Streaming Analytics 7.2.18 cluster definition.

For more information on what is included in the Apache Flink 1.18.0 version, see the Apache Flink 1.18.0 Release Announcementand Release Notes.

Application Mode

Support for Flink's Application Mode is added. Application Mode enables you to submit Flink applications that run entirely on the cluster side, differently from Per-Job and Session modes where the application driver code runs on the client side. In Application Mode, each application can execute multiple jobs that share cluster resources but are isolated from other applications. For more details, please check the Running a Flink job.

Support for custom data port range in Task Manager

The taskmanager.data.port Flink service configuration now supports multiple ports and port ranges for Flink jobs' Task Manager processes. The taskmanager.data.port property can be configured for Flink in Cloudera Manager using a safety valve, and for SSB the property can be configured with SET statement.

PyFlink Support

Python API for Flink is supported in Cloudera Streaming Analytics that enables you to create Flink DataStream applications using Python.

Component Support in Cloudera DataFlow for Data Hub 7.2.18

Cloudera DataFlow for Data Hub 7.2.18 includes the following components.

Flow Management clusters with NiFi 1

- Apache NiFi 1.25.0.2.2.8.0
- Apache NiFi Registry 1.25.0.2.2.8.0
- Schema Registry 0.10.0

Flow Management clusters with NiFi 2 [Technical Preview]

- Apache NiFi 2.0.0.4.2.0.0
- Apache NiFi Registry 2.0.0.4.2.0.0
- Schema Registry 0.10.0

Edge Management clusters [Technical Preview]

- Edge Flow Manager 2.0.0.0

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 1.18.0

Supported NiFi extensions

Apache NiFi versions 1.25 and 2.0 include an extensive set of extensions, most of which are supported by Cloudera.

Note: Avoid using any unsupported extensions in your production environments to ensure seamless operation and support.

Review the list of supported extensions for each version.

Supported NiFi processors

This release is based on Apache NiFi 1.25 and 2.0.0 and includes a set of processors, most of which are supported by Cloudera Support.

You should be familiar with the available supported processors and avoid using any unsupported processors in production environments.

While additional processors are developed and tested by the Cloudera community but are not officially supported by Cloudera. Processors are excluded for a variety of reasons, including insufficient reliability, incomplete test case coverage, community declaration of non-production readiness, or feature deviation from Cloudera best practices.

NiFi 1.25

AttributesToCSV	GetElasticsearch	PutDropbox
AttributesToJSON	GetFile	PutDynamoDB1
Base64EncodeContent	GetFTP	PutDynamoDBRecord
CalculateParquetOffsets	GetGcpVisionAnnotateFilesOperationStatus	PutElasticsearchHttp1
CalculateParquetRowGroupOffsets	GetGcpVisionAnnotateImagesOperationStatus	PutElasticsearchHttpRecord
CalculateRecordStats	GetHBase	PutElasticsearchJson
CaptureChangeDebeziumDB2 [Technical Preview]	GetHDFS	PutElasticsearchRecord
CaptureChangeDebeziumMySQL [Technical Preview]	GetHDFSFileInfo	PutEmail1
CaptureChangeDebeziumOracle [Technical Preview]	GetHDFSSequenceFile	PutFile
CaptureChangeDebeziumPostgreSQL [Technical Preview]	GetHTMLElement	PutFTP
CaptureChangeDebeziumSQLServer [Technical Preview]	GetHTTP	PutGCSObject
CaptureChangeMySQL	GetHubSpot	PutGoogleDrive
CompressContent12	GetIgniteCache	PutGridFS
ConnectWebSocket	GetJiraIssue	PutHBaseCell1
ConsumeAMQP	GetJMSQueue	PutHBaseJSON
ConsumeAzureEventHub	GetJMSTopic	PutHBaseRecord
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	PutJMS
ConsumeKafkaRecord_2_0	GetZendesk	PutKinesisFirehose
ConsumeKafkaRecord_2_6	HandleHttpRequest	PutKinesisStream
ConsumeKinesisStream	HandleHttpResponse	PutKudu
ConsumeMQTT	HashAttribute	PutLambda
ConsumeTwitter	HashContent	PutMongoRecord
ConsumeWindowsEventLog	IdentifyMimeType	PutORC
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	PutSplunkHTTP1
DecryptContent	ListCDPObjectStore	PutSQL
DecryptContentAge	ListDatabaseTables	PutSQS
DecryptContentCompatibility	ListDropbox	PutSyslog
DecryptContentPGP	ListenBeats	PutTCP
DeduplicateRecord	ListenFTP	PutUDP
DeleteAzureBlobStorage	ListenGRPC	PutWebSocket1
DeleteAzureBlobStorage_v12	ListenGRPC	PutZendeskTicket
DeleteAzureDataLakeStorage	ListenHTTP	QueryAirtableTable
DeleteByQueryElasticsearch	ListenNetFlow	QueryCassandra
DeleteCDPObjectStore	ListenOTLP	QueryDatabaseTable
DeleteDynamoDB	ListenRELP	QueryDatabaseTableRecord
DeleteGCSObject	ListenSyslog	QueryElasticsearchHttp
DeleteGridFS	ListenTCP	QueryRecord
DeleteHBaseCells	ListenTCPRecord	QuerySalesforceObject
DeleteHBaseRow	ListenTrapSNMP	QuerySolr
DeleteHDFS	ListenUDP	QuerySplunkIndexingStatus
DeleteS3Object	ListenUDPRecord	QueryWhois
DeleteSQS	ListenWebSocket	RemoveRecordField
DetectDuplicate	ListFile	ReplaceText1
DistributeLoad	ListFTP	ReplaceTextWithMapping
DuplicateFlowFile	ListGCSBucket	ResizeImage
EncodeContent	ListGoogleDrive	RetryFlowFile
EncryptContent	ListHDFS	RouteHL7
EncryptContentAge	ListS3	RouteOnAttribute
EncryptContentPGP	ListSFTP	RouteOnContent
EnforceOrder	ListSmb	RouteText
EvaluateJsonPath	LogAttribute	SampleRecord1
EvaluateXPath	LogMessage	ScanAccumulo
EvaluateXQuery	LookupAttribute	ScanAttribute

ExecuteGroovyScript	LookupRecord	ScanContent
ExecuteInfluxDBQuery	MergeContent1	ScanHBase
ExecuteProcess	MergeRecord	ScriptedFilterRecord
ExecuteScript	ModifyCompression	ScriptedPartitionRecord
ExecuteSQL	ModifyHTMLElement	ScriptedTransformRecord
ExecuteSQLRecord	MonitorActivity	ScriptedValidateRecord
ExecuteStateless	MoveAzureDataLakeStorage	ScrollElasticsearchHttp
ExecuteStreamCommand	MoveHDFS	SearchElasticsearch
ExtractAvroMetadata	Notify	SegmentContent
ExtractGrok	PackageFlowFile	SelectClouderaHiveQL
ExtractHL7Attributes	PaginatedJsonQueryElasticsearch	SelectHive3QL
ExtractImageMetadata	ParseCEF	SelectHiveQL
ExtractRecordSchema	ParseEvtx	SendTrapSNMP
ExtractText	ParseSyslog	SetSNMP
FetchAzureBlobStorage	PartitionRecord	SignContentPGP
FetchAzureBlobStorage_v12	PostHTTP	SplitAvro1
FetchAzureDataLakeStorage	PublishAMQP1	SplitContent1
FetchBoxFile	PublishGCPubSub1	SplitJson1
FetchCDPObjectStore	PublishGCPubSubLite	SplitRecord
FetchDistributedMapCache	PublishJMS1	SplitText1
FetchDropbox	PublishKafka_1_0	SplitXml1
FetchElasticsearchHttp	PublishKafka_2_0	StartAwsPollyJob
FetchFile	PublishKafka_2_6	StartAwsTextractJob
FetchFTP	PublishKafka2CDP	StartAwsTranscribeJob
FetchGCSObject	PublishKafka2RecordCDP	StartAwsTranslateJob
FetchGoogleDrive	PublishKafkaRecord_1_0	StartGcpVisionAnnotateFilesOperation
FetchGridFS	PublishKafkaRecord_2_0	StartGcpVisionAnnotateImagesOperation
FetchHBaseRow	PublishKafkaRecord_2_6	StartSnowflakeIngest
FetchHDFS	PublishMQTT1	TagS3Object
FetchParquet	PublishSlack	TailFile
FetchS3Object	PutAccumuloRecord	TransformXml
FetchSFTP	PutAzureBlobStorage	TriggerClouderaHiveMetaStoreEvent
FetchSmb	PutAzureBlobStorage_v12	TriggerHiveMetaStoreEvent
FilterAttribute	PutAzureCosmosDBRecord1	UnpackContent
FlattenJson	PutAzureDataLakeStorage	UpdateAttribute
ForkEnrichment	PutAzureEventHub1	UpdateByQueryElasticsearch
ForkRecord	PutAzureQueueStorage	UpdateClouderaHiveTable
GenerateFlowFile	PutAzureQueueStorage_v12	UpdateCounter
GenerateRecord	PutBigQuery	UpdateDatabaseTable
GenerateTableFetch	PutBigQueryBatch	UpdateDeltaLakeTable [Technical Preview]

GeoEnrichIP	PutBigQueryStreaming1	UpdateHive3Table
GeoEnrichIPRecord	PutBoxFile	UpdateHiveTable
GeohashRecord	PutCassandraQL1	UpdateRecord
GetAsanaObject	PutCassandraRecord	ValidateCsv
GetAwsPollyJobStatus	PutCDPObjectStore	ValidateJson
GetAwsTextractJobStatus	PutClouderaHiveQL	ValidateRecord
GetAwsTranscribeJobStatus	PutClouderaHiveStreaming	ValidateXml
GetAwsTranslateJobStatus	PutClouderaORC	VerifyContentMAC
GetAzureEventHub	PutCloudWatchMetric	VerifyContentPGP
GetAzureQueueStorage	PutCouchbaseKey1	Wait
GetAzureQueueStorage_v12	PutDatabaseRecord	YandexTranslate
GetCouchbaseKey	PutDistributedMapCache	

Footnotes

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

NiFi 2.0

AttributesToCSV	GetCouchbaseKey	PutElasticsearchRecord
AttributesToJSON	GetElasticsearch	PutEmail
CalculateParquetOffsets	GetFile	PutFile
CalculateParquetRowGroupOffsets	GetFTP	PutFTP
CalculateRecordStats	GetGcpVisionAnnotateFilesOperationStatus	PutGCSObject
CaptureChangeDebeziumDB2 [Technical Preview]	GetGcpVisionAnnotateImagesOperationStatus	PutGoogleDrive
CaptureChangeDebeziumMySQL [Technical Preview]	GetHBase	PutGridFS
CaptureChangeDebeziumOracle [Technical Preview]	GetHDFS	PutHBaseCell
CaptureChangeDebeziumPostgreSQL [Technical Preview]	GetHDFSFileInfo	PutHBaseJSON
CaptureChangeDebeziumSQLServer [Technical Preview]	GetHDFSSequenceFile	PutHBaseRecord
CaptureChangeMySQL	GetHubSpot	PutHDFS
ChunkDocument [Technical Preview]	GetJiraIssue	PutHive3QL
CompressContent	GetMongoRecord	PutHive3Streaming
ConnectWebSocket	GetSFTP	PutIceberg
ConsumeAMQP	GetShopify	PutIcebergCDC [Technical Preview]
ConsumeAzureEventHub	GetSNMP	PutJiraIssue
ConsumeElasticsearch	GetSnowflakeIngestStatus	PutKinesisFirehose
ConsumeGCPubSub	GetSolr	PutKinesisStream
ConsumeGCPubSubLite	GetSplunk	PutKudu
ConsumeJMS	GetSQS	PutLambda

ConsumeKafka_2_6	GetWorkdayReport	PutMongoBulkOperations
ConsumeKafka2CDP	GetZendesk	PutMongoRecord
ConsumeKafka2RecordCDP	HandleHttpRequest	PutORC
ConsumeKafkaRecord_2_6	HandleHttpResponse	PutParquet
ConsumeKinesisStream	IdentifyMimeType	PutPinecone [Technical Preview]
ConsumeMQTT	InvokeAWSGatewayApi	PutPLC [Technical Preview]
ConsumePLC [Technical Preview]	InvokeGRPC	PutRecord
ConsumeSlack	InvokeHTTP	PutRedisHashRecord [Technical Preview]
ConsumeTwitter	InvokeScriptedProcessor	PutS3Object
ConsumeWindowsEventLog	JoinEnrichment	PutSalesforceObject
ControlRate	JoltTransformJSON	PutSFTP
ConvertAvroToJSON	JoltTransformRecord	PutSmbFile
ConvertAvroToParquet	JSLTTransformJSON	PutSnowflakeInternalStage
ConvertCharacterSet	JsonQueryElasticsearch	PutSNS
ConvertJSONToSQL	ListAzureBlobStorage_v12	PutSolrContentStream
ConvertProtobuf	ListAzureDataLakeStorage	PutSolrRecord
ConvertRecord	ListBoxFile	PutSplunk
CopyAzureBlobStorage_v12	ListCDPObjectStore	PutSplunkHTTP
CreateHadoopSequenceFile	ListDatabaseTables	PutSQL
CryptographicHashContent	ListDropbox	PutSQS
DecryptContent	ListenBeats	PutSyslog
DecryptContentAge	ListenFTP	PutTCP
DecryptContentCompatibility	ListenGRPC	PutUDP
DecryptContentPGP	ListenHTTP	PutWebSocket
DeduplicateRecord	ListenNetFlow	PutZendeskTicket
DeleteAzureBlobStorage_v12	ListenOTLP	QueryAirtableTable
DeleteAzureDataLakeStorage	ListenRELP	QueryCassandra
DeleteByQueryElasticsearch	ListenSlack	QueryChroma [Technical Preview]
DeleteCDPObjectStore	ListenSyslog	QueryDatabaseTable
DeleteDynamoDB	ListenTCP	QueryDatabaseTableRecord
DeleteGCSObject	ListenTCPRecord	QueryPinecone [Technical Preview]
DeleteGridFS	ListenTrapSNMP	QueryRecord
DeleteHBaseCells	ListenUDP	QuerySalesforceObject
DeleteHBaseRow	ListenUDPRecord	QuerySolr
DeleteHDFS	ListenWebSocket	QuerySplunkIndexingStatus
DeleteS3Object	ListFile	QueryWhois
DeleteSQS	ListFTP	RemoveRecordField
DetectDuplicate	ListGCSBucket	RenameRecordField
DistributeLoad	ListGoogleDrive	ReplaceText
DuplicateFlowFile	ListHDFS	ReplaceTextWithMapping

EncodeContent	ListS3	ResizeImage
EncryptContentAge	ListSFTP	RetryFlowFile
EncryptContentPGP	ListSmb	RouteHL7
EnforceOrder	LogAttribute	RouteOnAttribute
EvaluateJsonPath	LogMessage	RouteOnContent
EvaluateXPath	LookupAttribute	RouteText
EvaluateXQuery	LookupRecord	SampleRecord
ExecuteGroovyScript	MergeContent	ScanAccumulo
ExecuteProcess	MergeRecord	ScanAttribute
ExecuteScript	ModifyCompression	ScanContent
ExecuteSQL	MonitorActivity	ScanHBase
ExecuteSQLRecord	MoveAzureDataLakeStorage	ScriptedFilterRecord
ExecuteStateless	MoveHDFS	ScriptedPartitionRecord
ExecuteStreamCommand	Notify	ScriptedTransformRecord
ExtractAvroMetadata	PackageFlowFile	ScriptedValidateRecord
ExtractGrok	PaginatedJsonQueryElasticsearch	SearchElasticsearch
ExtractHL7Attributes	ParseCEF	SegmentContent
ExtractImageMetadata	ParseDocument [Technical Preview]	SelectClouderaHiveQL
ExtractRecordSchema	ParseEvtx	SelectHive3QL
ExtractText	ParseSyslog	SendTrapSNMP
FetchAzureBlobStorage_v12	PartitionRecord	SetSNMP
FetchAzureDataLakeStorage	PromptChatGPT [Technical Preview]	SignContentPGP
FetchBoxFile	PublishAMQP	SplitAvro
FetchCDPObjectStore	PublishGCPubSub	SplitContent
FetchDistributedMapCache	PublishGCPubSubLite	SplitJson
FetchDropbox	PublishJMS	SplitRecord
FetchFile	PublishKafka_2_6	SplitText
FetchFTP	PublishKafka2CDP	SplitXml
FetchGCSObject	PublishKafka2RecordCDP	StartAwsPollyJob
FetchGoogleDrive	PublishKafkaRecord_2_6	StartAwsTextractJob
FetchGridFS	PublishMQTT	StartAwsTranscribeJob
FetchHBaseRow	PublishSlack	StartAwsTranslateJob
FetchHDFS	PutAccumuloRecord	StartGcpVisionAnnotateFilesOperation
FetchParquet	PutAzureBlobStorage_v12	StartGcpVisionAnnotateImagesOperation
FetchPLC [Technical Preview]	PutAzureCosmosDBRecord	StartSnowflakeIngest
FetchS3Object	PutAzureDataLakeStorage	TagS3Object
FetchSFTP	PutAzureEventHub	TailFile
FetchSmb	PutAzureQueueStorage_v12	TransformXml
FilterAttribute	PutBigQuery	TriggerClouderaHiveMetaStoreEvent
FlattenJson	PutBoxFile	TriggerHiveMetaStoreEvent

ForkEnrichment	PutCassandraQL	UnpackContent
ForkRecord	PutCassandraRecord	UpdateAttribute
GenerateFlowFile	PutCDPObjectStore	UpdateByQueryElasticsearch
GenerateRecord	PutChroma [Technical Preview]	UpdateClouderaHiveTable
GenerateTableFetch	PutClouderaHiveQL	UpdateCounter
GeoEnrichIP	PutClouderaHiveStreaming	UpdateDatabaseTable
GeoEnrichIPRecord	PutClouderaORC	UpdateDeltaLakeTable [Technical Preview]
GeohashRecord	PutCloudWatchMetric	UpdateHive3Table
GetAsanaObject	PutCouchbaseKey	UpdateRecord
GetAwsPollyJobStatus	PutDatabaseRecord	ValidateCsv
GetAwsTextractJobStatus	PutDistributedMapCache	ValidateJson
GetAwsTranscribeJobStatus	PutDropbox	ValidateRecord
GetAwsTranslateJobStatus	PutDynamoDB	ValidateXml
GetAzureEventHub	PutDynamoDBRecord	VerifyContentMAC
GetAzureQueueStorage_v12	PutElasticsearchJson	VerifyContentPGP
		Wait
		YandexTranslate

Supported NiFi controller services

This release is based on Apache NiFi 1.25 and 2.0 and includes a set of controller services, most of which are supported by Cloudera Support.

You should be familiar with the available supported controller services and avoid using any unsupported controller services in production environments.

While additional controller services are developed and tested by the Cloudera community but are not officially supported by Cloudera. Controller services are excluded for a variety of reasons, including insufficient reliability, incomplete test case coverage, community declaration of non-production readiness, or feature deviation from Cloudera best practices.

NiFi 1.25

AccumuloService	ImpalaConnectionPool
ActionHandlerLookup	IPFIXReader
ActiveMQJMSConnectionFactoryProvider	IPLookupService
ADLSCredentialsControllerService	JASN1Reader
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	RabbitMQJMSConnectionFactoryProvider
ClouderaSchemaRegistry	ReaderLookup
CMLLookupService	RecordSetWriterLookup
ConfluentSchemaRegistry	RecordSinkHandler
CouchbaseClusterService	RecordSinkServiceLookup
CouchbaseKeyValueLookupService	RedisConnectionPoolService
CouchbaseMapCacheClient	RedisDistributedMapCacheClientService
CouchbaseRecordLookupService	RedshiftConnectionPool
CSVReader	RestLookupService
CSVRecordLookupService	ScriptedActionHandler
CSVRecordSetWriter	ScriptedLookupService
DatabaseRecordLookupService	ScriptedReader
DatabaseRecordSink	ScriptedRecordSetWriter
DatabaseTableSchemaRegistry	ScriptedRecordSink
DBCPConnectionPool	ScriptedRulesEngine
DBCPConnectionPoolLookup	SimpleDatabaseLookupService
DistributedMapCacheClientService	SimpleKeyValueLookupService
DistributedMapCacheLookupService	SimpleRedisDistributedMapCacheClientService
DistributedMapCacheServer	SimpleScriptedLookupService
DistributedSetCacheClientService	SiteToSiteReportingRecordSink
DistributedSetCacheServer	SmbjClientProviderService
EasyRulesEngineProvider	SnowflakeComputingConnectionPool
EasyRulesEngineService	StandardAsanaClientProviderService
EBCDICRecordReader [Technical Preview]	StandardAzureCredentialsControllerService
ElasticSearchClientServiceImpl	StandardDropboxCredentialService

ElasticSearchLookupService	StandardFileResourceService
ElasticSearchStringLookupService	StandardHashiCorpVaultClientService
EmailRecordSink	StandardHttpContextMap
EmbeddedHazelcastCacheManager	StandardJsonSchemaRegistry [Technical Preview]
ExcelReader	StandardOauth2AccessTokenProvider
ExpressionHandler	StandardPGPPrivateKeyService
ExternalHazelcastCacheManager	StandardPGPPublicKeyService
FreeFormTextRecordSetWriter	StandardPrivateKeyService
GCPCredentialsControllerService	StandardProxyConfigurationService
GrokReader	StandardRestrictedSSLContextService
HadoopCatalogService	StandardS3EncryptionService
HadoopDBCPConnectionPool	StandardSnowflakeIngestManagerProviderService
HazelcastMapCacheClient	StandardSSLContextService
HBase_1_1_2_ClientMapCacheService	StandardWebClientServiceProvider
HBase_1_1_2_ClientService	Syslog5424Reader
HBase_1_1_2_ListLookupService	SyslogReader
HBase_1_1_2_RecordLookupService	UDPEventRecordSink
HBase_2_ClientMapCacheService	VolatileSchemaCache
HBase_2_ClientService	WindowsEventLogReader
HBase_2_RecordLookupService	XMLReader
Hive3ConnectionPool	XMLRecordSetWriter
HiveCatalogService	YamlTreeReader
HiveConnectionPool	ZendeskRecordSink
HortonworksSchemaRegistry	

NiFi 2.0

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	KeytabCredentialsService

AzureEventHubRecordSink	KuduLookupService
AzureServiceBusJMSConnectionFactoryProvider	LoggingRecordSink
AzureStorageCredentialsControllerService_v12	MongoDBControllerService
AzureStorageCredentialsControllerServiceLookup_v12	MongoDBLookupService
CassandraDistributedMapCache	ParquetReader
CassandraSessionProvider	ParquetRecordSetWriter
CdpCredentialsProviderControllerService	PostgreSQLConnectionPool
CdpOauth2AccessTokenProviderControllerService	PrometheusRecordSink
CEFReader	ProxyPLC4XConnectionPool [Technical Preview]
CiscoEmblemSyslogMessageReader	RabbitMQJMSConnectionFactoryProvider
ClouderaHiveConnectionPool	ReaderLookup
ClouderaSchemaRegistry	RecordSetWriterLookup
CMLLookupService	RecordSinkServiceLookup
ConfluentEncodedSchemaReferenceReader	RedisConnectionPoolService
ConfluentEncodedSchemaReferenceWriter	RedisDistributedMapCacheClientService
ConfluentSchemaRegistry	RedshiftConnectionPool
CouchbaseClusterService	RestLookupService
CouchbaseKeyValueLookupService	ScriptedLookupService
CouchbaseMapCacheClient	ScriptedReader
CouchbaseRecordLookupService	ScriptedRecordSetWriter
CSVReader	ScriptedRecordSink
CSVRecordLookupService	SimpleDatabaseLookupService
CSVRecordSetWriter	SimpleKeyValueLookupService
DatabaseRecordLookupService	SimpleRedisDistributedMapCacheClientService
DatabaseRecordSink	SimpleScriptedLookupService
DatabaseTableSchemaRegistry	SiteToSiteReportingRecordSink
DBCPConnectionPool	SlackRecordSink
DBCPConnectionPoolLookup	SmbjClientProviderService
DistributedMapCacheClientService	SnowflakeComputingConnectionPool
DistributedMapCacheLookupService	StandardAsanaClientProviderService
DistributedMapCacheServer	StandardAzureCredentialsControllerService
DistributedSetCacheClientService	StandardDropboxCredentialService
DistributedSetCacheServer	StandardFileResourceService
EBCDICRecordReader [Technical Preview]	StandardHashiCorpVaultClientService
ElasticSearchClientServiceImpl	StandardHttpContextMap
ElasticSearchLookupService	StandardJsonSchemaRegistry [Technical Preview]
ElasticSearchStringLookupService	StandardOauth2AccessTokenProvider
EmailRecordSink	StandardPGPPrivateKeyService
EmbeddedHazelcastCacheManager	StandardPGPPublicKeyService
ExcelReader	StandardPLC4XConnectionPool [Technical Preview]

ExternalHazelcastCacheManager	StandardPrivateKeyService
FreeFormTextRecordSetWriter	StandardProxyConfigurationService
GCPCredentialsControllerService	StandardRestrictedSSLContextService
GCSFileResourceService	StandardS3EncryptionService
GenericPLC4XConnectionPool [Technical Preview]	StandardSnowflakeIngestManagerProviderService
GrokReader	StandardSSLContextService
HadoopCatalogService	StandardWebClientServiceProvider
HadoopDBCPConnectionPool	Syslog5424Reader
HazelcastMapCacheClient	SyslogReader
HBase_2_ClientMapCacheService	UDPEventRecordSink
HBase_2_ClientService	VolatileSchemaCache
HBase_2_RecordLookupService	WindowsEventLogReader
Hive3ConnectionPool	XMLReader
HiveCatalogService	XMLRecordSetWriter
ImpalaConnectionPool	YamlTreeReader
IPFIXReader	ZendeskRecordSink

Supported NiFi reporting tasks

This release is based on Apache NiFi 1.25 and 2.0 and includes a set of reporting tasks, most of which are supported by Cloudera Support.

You should be familiar with the available supported reporting tasks and avoid using any unsupported reporting tasks in production environments.

While additional reporting tasks are developed and tested by the Cloudera community but are not officially supported by Cloudera. Reporting tasks are excluded for a variety of reasons, including insufficient reliability, incomplete test case coverage, community declaration of non-production readiness, or feature deviation from Cloudera best practices.

NiFi 1.25

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

NiFi 2.0

- ControllerStatusReportingTask
- MonitorDiskUsage

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

Supported NiFi flow analysis rules [Technical Preview]

This release is based on Apache NiFi 1.25 and 2.0.0. Apache NiFi 2.0.0 introduces a set of flow analysis rules, most of which are supported by Cloudera.

You should be familiar with the available supported flow analysis rules and avoid using any unsupported flow analysis rules in production environments.

⚠ **Important:**

The Flow Analysis Rules feature is provided in technical preview. Work is still in progress and breaking changes may occur in the next release. Do not use this feature in production.

- DisallowComponentType

While additional flow analysis rules are developed and tested by the Cloudera community, they are not officially supported by Cloudera. Flow analysis rules may be excluded due to various reasons, such as insufficient reliability, incomplete test case coverage, community declaration of non-production readiness, or deviation from Cloudera best practices.

Supported NiFi Python components [Technical Preview]

This release is based on Apache NiFi 1.25 and 2.0.0. Apache NiFi 2.0 introduces a set of NiFi components written in Python, most of which are supported by Cloudera.

You should be familiar with the available supported Python components and avoid using any unsupported Python component in production environments.

⚠ **Important:**

The Python API feature is provided in technical preview. Work is still in progress and breaking changes may occur in the next release. Do not use this feature in production.

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

Supported Python components:

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.

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.

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.

QueryChroma

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

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.

QueryPinecone

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

Cloudera exclusive components [Technical Preview]

Cloudera provides a set of NiFi components that are only available to its customers. Below you can find the list of these components.

In NiFi 1.25 clusters

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 NiFi 2.0 clusters

Processors

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

- ListenGRPC
- ListenNetFlow
- PublishKafka2CDP
- PublishKafka2RecordCDP
- PutCDPObjectStore
- PutClouderaHiveQL
- PutClouderaHiveStreaming
- PutClouderaORC
- PutIcebergCDC
- PutJiraIssue
- PutPLC
- SelectClouderaHiveQL
- TriggerClouderaHiveMetaStoreEvent
- UpdateClouderaHiveTable
- UpdateDeltaLakeTable

Controller Services

- ActiveMQJMSConnectionFactoryProvider
- ADLSIDBrokerCloudCredentialsProviderControllerService
- AWSIDBrokerCloudCredentialsProviderControllerService
- AzureBlobIDBrokerCloudCredentialsProviderControllerService
- AzureServiceBusJMSConnectionFactoryProvider
- CdpCredentialsProviderControllerService
- CdpOauth2AccessTokenProviderControllerService
- CiscoEmblemSyslogMessageReader
- ClouderaHiveConnectionPool
- ClouderaSchemaRegistry
- CMLLookupService
- EBCDICRecordReader
- GenericPLC4XConnectionPool
- ImpalaConnectionPool
- IPFIXReader
- JiraRecordSink
- PostgreSQLConnectionPool
- ProxyPLC4XConnectionPool
- RabbitMQJMSConnectionFactoryProvider
- RedshiftConnectionPool
- StandardPLC4XConnectionPool

Parameter Providers

- CyberArkConjurParameterProvider

Components supported by partners

This release is based on Apache NiFi 1.25 and 2.0 and includes a set of components built, maintained and supported by Cloudera partners. You should reach out directly to these partners in case you need assistance.

These components are not officially supported by Cloudera Support even though Cloudera Quality Engineering teams added test coverage for these components.

> **Note:** In Flow Management with NiFi 2 clusters, the Pulsar components are not included. This is temporary and the components will be added in upcoming releases. Meanwhile, you can manually download the components from a Maven repository and add them into your cluster.

NiFi 1.25 processors supported by partners

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

NiFi 1.25 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.

Unsupported Features in Cloudera DataFlow for Data Hub 7.2.18

Some features exist within Cloudera DataFlow for Data Hub 7.2.18 components, but are not supported by Cloudera.

Unsupported Flow Management features

Some Flow Management features exist in Cloudera DataFlow for Data Hub 7.2.18, but are not supported by Cloudera.

NiFi 1.25

The following technical preview features are available in Flow Management in Cloudera DataFlow for Data Hub 7.2.18 but are not ready for production deployment. Cloudera encourages you to explore these technical preview features in non-production environments and provide feedback on your experiences through the Cloudera Community Forums.

Rules engine, handler in Technical Preview:

- ActionHandlerLookup
- AlertHandler
- EasyRulesEngineProvider
- EasyRulesEngineService
- ExpressionHandler
- LogHandler
- RecordSinkHandler
- ScriptedActionHandler
- ScriptedRulesEngine
- ScriptedRulesEngine

Processors in Technical Preview:

- CaptureChangeDebeziumDB2

- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- PutIcebergCDC
- PutRedisHashRecord
- UpdateDeltaLakeTable

Controller services in Technical Preview:

- EBCDICRecordReader
- StandardJsonSchemaRegistry

NiFi 2.0

The following technical preview features are available in Flow Management in Cloudera DataFlow for Data Hub 7.2.18 but are not ready for production deployment. Cloudera encourages you to explore these technical preview features in non-production environments and provide feedback on your experiences through the Cloudera Community Forums.

- The Flow Analysis Rules engine is in Technical Preview and all the provided rules are also in Technical Preview.
- The Python API feature is in Technical Preview and all the Python extensions using this feature are also in Technical Preview.
- The ability to run a Process Group using the Stateless Engine is in Technical Preview.

Processors in Technical Preview:

- CaptureChangeDebeziumDB2
- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- ChunkDocument
- ConsumePLC
- FetchPLC
- ParseDocument
- PromptChatGPT
- PutChroma
- PutIcebergCDC
- PutPinecone
- PutPLC
- PutRedisHashRecord
- QueryChroma
- QueryPinecone

Controller services in Technical Preview:

- UpdateDeltaLakeTable
- EBCDICRecordReader
- GenericPLC4XConnectionPool
- ProxyPLC4XConnectionPool
- StandardJsonSchemaRegistry
- StandardPLC4XConnectionPool

Flow analysis rules in Technical Preview:

- DisallowComponentType

NiFi Registry

There are no unsupported features in this release.

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 7.2.18, 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 Streaming Analytics features

Some Streaming Analytic features exist in Cloudera DataFlow for Data Hub 7.2.18, 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*.

Flink

- Apache Flink batch (DataSet) API
- GPU Resource Plugin
- SQL Client
- 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 7.2.18

You must be aware of the known issues and limitations, the areas of impact, and workaround in Cloudera DataFlow for Data Hub 7.2.18.

Known issues in Flow Management

Learn about the known issues and limitations in Flow Management clusters, the impact or changes to the functionality, and the workaround.

NiFi 1.25

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 CDP 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.

> Workaround: 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

> To ensure that Cloudera Manager accurately reflects the NiFi metrics for the NiFi service, you need to update the Flow NiFi access policies in Ranger and include the 'nifi' group.

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

⚠️ **Important:** The current version of Apache NiFi 2.0 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.

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.

Workaround:

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 CDP 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.

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

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: SMM's bootstrap server config should be updated based on Kafka's listeners

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

Workaround: SMM cannot identify multiple listeners and still points to bootstrap server using the default broker port (9093 for SASL_SSL). You would have to override bootstrap server URL (hostname:port as set in the listeners for broker) in the following path:

Cloudera Manager > SMM > Configuration > Streams Messaging Manager Rest Admin Server Advanced Configuration Snippet (Safety Valve) for streams-messaging-manager.yaml > Save Changes > Restart SMM.

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.

OPSAPS-43236: Kafka garbage collection logs are written to the process directory

By default Kafka garbage collection logs are written to the agent process directory. Changing the default path for these log files is currently unsupported.

None

CDPD-45183: Kafka Connect active topics might be visible to unauthorised users

The Kafka Connect active topics endpoint (/connectors/*[***CONNECTOR NAME***]*/topics) and the Connect Cluster page on the SMM UI disregard the user permissions configured for the Kafka service in Ranger. As a result, all active topics of connectors might become visible to users who do not have permissions to view them. Note that user permission configured for Kafka Connect in Ranger are not affected by this issue and are correctly applied.

None.

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 transacti
onal method because we are in an error state
    at org.apache.kafka.clients.producer.internals.TransactionM
anager.maybeFailWithError(TransactionManager.java:1125)
    at org.apache.kafka.clients.producer.internals.TransactionMa
nager.maybeAddPartition(TransactionManager.java:442)
    at org.apache.kafka.clients.producer.KafkaProducer.doSend(K
afkaProducer.java:1000)
    at org.apache.kafka.clients.producer.KafkaProducer.send(Kafk
aProducer.java:914)
    at org.apache.kafka.clients.producer.KafkaProducer.send(Kafk
aProducer.java:800)
    .
    .
    .
Caused by: org.apache.kafka.common.errors.ClusterAuthorization
Exception: 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.idempoten ce to false.
- Update your policies in Ranger and ensure that producers have Idempotent Write permission on the cluster resource.

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.

Unsupported features

The following Kafka features are not supported in Cloudera Data Platform:

- 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's Performance to Degrade

If the Kafka service operates with a large number of partitions, collection of partition level metrics may cause Cloudera Manager's 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 SMM to monitor Kafka or Cruise Control for rebalancing Kafka partitions, be aware that both SMM and Cruise Control rely on partition level metrics. If partition level metric collection is disabled, SMM 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_ent
      ity_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:

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

Schema Registry does not start if the address specified in the Load Balancer Address Ranger property does not end with a trailing slash (/).

Set the value of the RANGER_REST_URL Schema Registry environment variable to an address that includes a trailing slash.

1. In Cloudera Manager, select the Schema Registry service.
2. Go to Configuration.
3. Find the Schema Registry Server Environment Advanced Configuration Snippet (Safety Valve) property and add the following:

   ```
   Key: RANGER_REST_URL
   Value: [***RANGER REST API URL***]
   ```

   Replace *[***RANGER REST API URL***]* with an address that can be used by Schema Registry to access Ranger. Ensure that the address ends with a trailing slash. For example: http s://ranger-1.cloudera.com:6182/
4. Restart the Schema Registry service.

Streams Messaging Manager

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

CDPD-39313: Some numbers are not rendered properly in SMM UI

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

> None.

CDPD-45183: Kafka Connect active topics might be visible to unauthorised users

> The Kafka Connect active topics endpoint (/connectors/*[***CONNECTOR NAME***]*/topics) and the Connect Cluster page on the SMM UI disregard the user permissions configured for the Kafka service in Ranger. As a result, all active topics of connectors might become visible to users who do not have permissions to view them. Note that user permission configured for Kafka Connect in Ranger are not affected by this issue and are correctly applied.

> None.

OPSAPS-59553: SMM's bootstrap server config should be updated based on Kafka's listeners

> SMM does not show any metrics for Kafka or Kafka Connect when multiple listeners are set in Kafka.

> SMM cannot identify multiple listeners and still points to bootstrap server using the default broker port (9093 for SASL_SSL). You would have to override bootstrap server URL (hostname:port as set in the listeners for broker). Add the bootstrap server details in SMM safety valve in the following path:

> Cloudera Manager SMM Configuration Streams Messaging Manager Rest Admin Server Advanced Configuration Snippet (Safety Valve) for streams-messaging-manager.yaml Add the following value for bootstrap servers Save Changes Restart SMM .

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

OPSAPS-59597: SMM UI logs are not supported by Cloudera Manager

> Cloudera Manager does not support the log type used by SMM UI.

> View the SMM UI logs on the host.

Limitations

CDPD-36422: 1MB flow.snapshot freezes safari

> Importing large connector configurations/ flow.snapshots reduces the usability of the Streams Messaging Manager's Connectors page 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: SRM 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 SMM UI. However, it's Partitions and Consumer Groups will be 0, its Throughput, Replication Latency and Checkpoint Latency will show N/A.

None

CDPD-30275: SRM 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 [TARGE
   T_CLUSTER] --remove [TOPIC1]
   ```

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

Limitations
SRM cannot replicate Ranger authorization policies to or from Kafka clusters

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

Cruise Control

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

Rebalancing with Cruise Control does not work due to the metric reporter failing to report the CPU usage metric

On the Kafka broker, the Cruise control metric reporter plugin may fail 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.CruiseCont
rolMetricsReporter:
        [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 performing a manual repair on the affected nodes. For more information, see the Data Hub documentation.

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

Known issues in Streaming Analytics

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

SQL Stream Builder

CSA-5138 - SQL job submissions with UDF JARs fail when checkpointing is enabled

Due to the handling of ClassLoaders for custom JARs, uploading any Java UDFs with checkpoints enabled will cause the SQL job to fail with the following error:

```
ERROR com.cloudera.ssb.sqlio.service.SqlExecutorService: Error w
hile submitting streaming job
org.apache.flink.util.FlinkRuntimeException: org.apache.flink.
api.common.InvalidProgramException: Table program cannot be comp
iled.
```

Once the SQL job fails, the session on Streaming SQL Console must be reset before resubmitting the job without checkpointing.

None

CSA-4858 - Kerberos encryption type detection does not always work correctly for SSB

SSB 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)

INSIGHT-6486: Third-party dependencies upgraded in CDP Public Cloud might cause Flink jobs to fail

After upgrading CDP Public 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 CDP Public Cloud version 7.2.18.

Verify your application's dependency versions against the Cloudera-supported versions before upgrading to a newer version of CDP Public Cloud. For more information see Updating Flink job dependencies.

DataStream conversion limitations

- Converting between Tables and POJO DataStreams is currently not supported in CSA.
- 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 PRIM ARY 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.

Fixed Issues in Cloudera DataFlow for Data Hub 7.2.18

Fixed issues represent selected issues that were previously logged through Cloudera Support, but are addressed in the current release. These issues may have been reported in previous versions within the Known Issues section; meaning they were reported by customers or identified by Cloudera Quality Engineering team.

Review the list of issues that are resolved in Cloudera DataFlow for Data Hub 7.2.18.

Fixed issues in Flow Management

Review the list of Flow Management issues that are resolved in Cloudera DataFlow for Data Hub 7.2.18.

NiFi 1.25 with CFM 2.2.8

CFM 2.2.8 is based on Apache NiFi 1.25.0. It includes all fixed issues of this Apache NiFi release, as well as the following additional fixes:

- NIFI-12843: Fix incorrect read of parquet data, when record.count is inherited
- NIFI-12596: PutIceberg is missing case-insensitive Record type handling in List and Map types
- NIFI-12506 Added Threading for Status Analytics Retrieval
- NIFI-9464 Fixed race condition between "Timer-Driven" threads when running SiteToSiteProvenanceReportingTask.onTrigger and "Compress Provenance Log" threads
- NIFI-12105: ExecuteStateless processor accepts additional NAR directories
- NIFI-11149 Added PutRedisHashRecord Processor
- CFM-2713 Resource Fetcher over HTTPS with Basic Authentication
- CFM-3673 Re-added InvokeGRPC processor to downstream
- CFM-3503 Create PutJiraIssue and JiraRecordSink
- CFM-3712 Align PutIcebergCDC eq-delete strategy with Flink

- CFM-3616 Create Impala JDBC Controller Service
- CFM-3687 Update Commons-text-1.9.jar in nifi-cdf-iceberg-nar
- CFM-3676 Improve documentation about cache.
- CFM-3418 Add nifi-cdf-hive-bundle with CLDR-specific Hive components
- CFM-3399 Added Debezium-based processor for SQL Server
- CFM-3397 Added Debezium-based processor for PostgreSQL
- CFM-3395 Added Debezium-based processor for DB2
- CFM-3393 Added Debezium-based processor for Oracle
- CFM-3607 Add provenance event to PutIcebergCDC processor
- CFM-3090 Create RabbitMQ JMS Controller Service
- CFM-3400 Added abstract Debezium-based processor and MySQL implementation.
- CFM-3091 Create ActiveMQ JMS Controller Service
- CFM-3605 Add missing Ozone support to UpdateDeltaLakeTable
- CFM-3574 Allow override of fs.defaultFS in UpdateDeltaLakeTable
- CFM-3458 In 'UpdateDeltaLakeTable' handling when a 'DeltaConcurrentModificationException' occurs
- CFM-3329 CML Lookup Service
- CFM-2853 Added CyberarkConjurParameterProvider.
- CFM-3488 Correct padding handling for ListenNetFlow Options Templates
- CFM-3335 PutIcebergCDC processor
- CFM-2853 Added CyberarkConjurParameterProvider.
- CFM-3335 PutIcebergCDC processor
- CFM-3425 Create AzureServiceBusJMSConnectionFactoryProvider to consumer Azure
- CFM-3168 Added EBCDICRecordReader
- CFM-3256 Allowing ClouderFlowLibraryClient to work with unknown fields (will be ignored)
- CFM-3325 Redshift controller service
- CFM-3378 Automatic nar delivery should not die, if 404 happens
- CFM-3328 Add ListenNetFlow Processor
- CFM-3088 Add PostgreSQLConnectionPool
- CFM-3159 Add registry client for Cloudera Flow Library
- CFM-3239 ClouderaSchemaRegistry controller service
- CFM-1822 Handle custom schema in ListenGRPC

NiFi 2.0 with CFM 4.2.0

CFM 4.2.0 is based on Apache NiFi 2.0.0-M1. It includes all fixed issues of this Apache NiFi release, as well as the following additional fixes:

- 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-12676 Fixed Servlet Registration in HandleHttpRequest
- NIFI-12394 Fixed Service references for Migrated Configurations
- NIFI-12596 PutIceberg is missing case-insensitive Record type handling in List and Map types
- NIFI-12506 Added Threading for Status Analytics Retrieval
- NIFI-12434 Upgraded Registry to Spring Framework 6.1.1
- NIFI-12587 improve error message in ValidateCSV
- NIFI-12581 Upgraded geoip2 from 4.1.0 to 4.2.0
- NIFI-12580 Upgraded maven-artifact plugin from 3.8.8 to 3.9.6
- NIFI-12578 Updated nifi-compress-bundle using current API methods
- NIFI-12577 Updated nifi-cdc-bundle using current API methods

- NIFI-12575 Updated nifi-asana-bundle using current API methods
- NIFI-9464 Fixed race condition between "Timer-Driven" threads
- NIFI-12582 Update mockito.version to 5.8.0
- NIFI-12579 Update icu4j to 74.2
- NIFI-12583 Update SMBJ to 0.13.0
- NIFI-12576 MQTT processors allow path component in Broker URI
- NIFI-12571 Upgraded Logback from 1.3.14 to 1.4.14
- NIFI-12570 Upgraded Apache IoTDB from 1.2.2 to 1.3.0
- NIFI-12569 Upgraded Iceberg from 1.4.2 to 1.4.3
- NIFI-12562 Upgraded json-schema-validator from 1.0.87 to 1.1.0
- NIFI-12521 Upgrade ca.uhn.hapi dependencies to 2.5.1
- NIFI-11703 Upgraded to Spring Framework 6 and Jetty 12
- NIFI-12395 Upgraded Jackson JSON from 2.15.3 to 2.16.1
- NIFI-12382 Added DatabaseTableSchemaRegistry Service
- NIFI-12538 Added mapOf Record Standalone Function
- NIFI-12558 Upgraded Jagged from 0.2.0 to 0.3.0
- NIFI-12524 Fixed null handling for auditing Connect Details
- NIFI-12559 Upgraded SSHJ from 0.37.0 to 0.38.0
- NIFI-12531 Parameter references are removed after property migration
- NIFI-12526 Fixed handling of Fetch Size in QueryCassandra, added fragment attributes
- NIFI-12556 Updated nifi-cipher-bundle using current API methods
- NIFI-12557 Updated nifi-kafka-bundle using current API methods
- NIFI-12517 Updated isJson function to improve space handling
- NIFI-12513 Added UriUtils to nifi-utils and updated InvokeHTTP
- NIFI-12546 Updated nifi-snowflake-bundle using current API methods
- NIFI-12545 Updated nifi-iceberg-bundle using current API methods
- NIFI-11627 Added JsonSchemaRegistry for ValidateJson
- NIFI-12334 Added GCSFileResourceService
- NIFI-12542 Updated Zendesk Components using Current APIs
- NIFI-12532 Ensure that when CommunicateAction completes (exceptionally or otherwise) that it gets removed from the list of all CommunicationActions
- NIFI-12528 Fixed bug that resulted in StackOverflowError when deleting loop containing only funnels
- NIFI-12536 Used proper method for converting byte array to String
- NIFI-12481 Filtering out unauthorized registry clients to avoid unhandled error
- NIFI-12525 Added property migration for the removed Consumer Hostname property in ConsumeAzureEventHub
- NIFI-12515 Removed deprecated Client Auth property from AMQP processors
- NIFI-12527 Removed deprecated Session Cache Size property from JMS processors
- NIFI-12487 Added CSRF Protection to Registry
- NIFI-12418 Corrected Provider Groups Missing in Refreshed Tokens
- NIFI-11308 Added nifi.framework.version System Property for EL
- NIFI-12480 Updated MapRecord's toString() method to use the SerializedForm of the record when available and fixed bugs around ensuring that the serialized form is properly set
- NIFI-12522 Upgraded Netty from 4.1.101 to 4.1.104
- NIFI-12408 Added Pretty Print Property to AttributesToJSON
- NIFI-12371 Support tombstone messages in non-record Kafka processors
- NIFI-12446 Refactor FilterAttribute to align with code conventions
- NIFI-12520 ExtractHL7Attributes processor ignores repeatable field values
- NIFI-12518 Upgraded Calcite Avatica from 1.23.0 to 1.24.0
- NIFI-12516 Corrected Cluster Replicated Response Headers for HTTP/2

- NIFI-6730 AMQP QoS support
- NIFI-12509 Changing default TTL of HazelcastMapCacheClient
- NIFI-12474 remove TriggerWhenEmpty for DeleteHDFS
- NIFI-12496 Moved Python extensions unpacking to common.xml
- NIFI-12497 Upgraded snowflake-jdbc to 3.14.4
- NIFI-12241 Efficient Parquet Splitting
- NIFI-12479 Add pg-export to toolkit CLI
- NIFI-12492 Moved QuestDB Status Repository to Separate NAR
- NIFI-12475 Disabled Bypass Validation by Default in PutMongoRecord
- NIFI-12478 Return Message Type as body for JMS Object Messages
- NIFI-12489 Upgraded Apache POI from 5.2.4 to 5.2.5
- NIFI-12105 ExecuteStateless processor accepts additional NAR directories
- NIFI-12483 Streamlined JMX Metrics Filtering Parameters
- NIFI-12484 Bumping dependency to latest minor/incremental release
- NIFI-12470 Fixing forEach callback for usage with Object.entries() to address layout issue in Status History
- NIFI-12473 Deleted removeByPattern Methods from Cache Services
- NIFI-12477 Update org.apache.commons.io.version to 2.15.1
- NIFI-12476 Exclude nifi-framework-core from minifi-bootstrap
- NIFI-12453 Added cluster-status command to nifi.sh
- NIFI-12331 Added PublishSlack Processor
- NIFI-12410 Support EL for Starting Row in ExcelReader
- NIFI-12412 Support Proxies for Blob Checkpoints in ConsumeAzureEventHub
- NIFI-12427 Add channel name attribute to ConsumeSlack
- NIFI-12318 Fixed byte array generation in GenerateRecord
- NIFI-12462 Upgraded Logback from 1.3.13 to 1.3.14
- NIFI-12466 Upgraded MongoDB Driver from 4.10.2 to 4.11.1
- NIFI-12465 Upgraded mysql-binlog-connector-java from 0.28.1 to 0.28.3
- NIFI-12463 Upgraded maven-artifact from 3.8.5 to 3.8.8
- NIFI-12461 Upgraded jsoup from 1.16.2 to 1.17.1
- NIFI-12460 Upgraded OpenCSV from 5.8 to 5.9
- NIFI-12455 Upgraded JAXB Runtime from 2.3.8 to 2.3.9
- NIFI-12451 Removed Bootstrap Notification Services
- NIFI-12457 Added additional use case based documentation and fixed some typos/made some clarifications in related docs
- NIFI-12458 Upgrade Groovy to 4.0.16 from 4.0.15
- NIFI-11992 Processor and sink service for filing tickets in Zendesk
- NIFI-12439 Upgraded Kubernetes Client from 6.8.1 to 6.9.2
- NIFI-12448 Removed InfluxDB Processors
- NIFI-12432 Upgraded Spring Framework from 5.3.30 to 5.3.31
- NIFI-12440 Upgraded HikariCP from 5.0.1 to 5.1.0
- NIFI-11129 Added PutMongoBulkOperations Processor
- NIFI-12386 Adds processor FilterAttribute attribute uuid is not removed by removeAttribute(s) in MockProcessSession
- NIFI-12436 Upgraded datafaker from 2.0.1 to 2.0.2
- NIFI-12433 Upgraded mariadb-java-client from 3.2.0 to 3.3.0
- NIFI-12431 Upgraded testcontainers from 1.19.1 to 1.19.3
- NIFI-12430 Upgraded unboundid-ldapsdk from 6.0.9 to 6.0.10
- NIFI-12438 Upgraded Logback from 1.3.11 to 1.3.13
- NIFI-12366 Add HuggingFace support to Pinecone processors
- NIFI-12404 Added documentation indicating this controller service does not resolve YAML aliases

- NIFI-12416 Relocated the additionalDetails.html to the appropriate bundle in order for it to be seen when generating the documentation
- NIFI-12423 Update iotdb.sdk.version to 1.2.2
- NIFI-12422 Update excel-streaming-reader to 4.2.1
- NIFI-12421 Update metrics-core to 4.2.22
- NIFI-12414 Update org.apache.commons.lang3.version to 3.14.0
- NIFI-11481 Initial migration to Angular 16
- NIFI-11583 Removed nifi-ignite-nar module from assembly
- CFM-3673 Re-added InvokeGRPC processor to downstream
- CFM-3503 Create PutJiraIssue and JiraRecordSink
- CFM-3712 Align PutIcebergCDC eq-delete strategy with Flink
- CFM-3616 Create Impala JDBC Controller Service
- CFM-3687 Update Commons-text-1.9.jar in nifi-cdf-iceberg-nar
- CFM-3676 Improve documentation about cache
- CFM-3418 Add nifi-cdf-hive-bundle with CLDR-specific Hive components
- CFM-3399 Added Debezium-based processor for SQL Server
- CFM-3397 Added Debezium-based processor for PostgreSQL
- CFM-3395 Added Debezium-based processor for DB2
- CFM-3393 Added Debezium-based processor for Oracle
- CFM-3607 Add provenance event to PutIcebergCDC processor
- CFM-3090 Create RabbitMQ JMS Controller Service
- CFM-3400 Added abstract Debezium-based processor and MySQL implementation
- CFM-3091 Create ActiveMQ JMS Controller Service
- CFM-3605 Add missing Ozone support to UpdateDeltaLakeTable
- CFM-3574 Allow override of fs.defaultFS in UpdateDeltaLakeTable
- CFM-3458 In 'UpdateDeltaLakeTable' handling when a 'DeltaConcurrentModificationException' occurs
- CFM-3329 CML Lookup Service
- CFM-2853 Added CyberarkConjurParameterProvider.
- CFM-3488 Correct padding handling for ListenNetFlow Options Templates
- CFM-3335 PutIcebergCDC processor
- CFM-2853 Added CyberarkConjurParameterProvider.
- CFM-3335 PutIcebergCDC processor
- CFM-3425 Create AzureServiceBusJMSConnectionFactoryProvider to consumer Azure
- CFM-3168 Added EBCDICRecordReader
- CFM-3256 Allowing ClouderFlowLibraryClient to work with unknown fields (will be ignored)
- CFM-3325 Redshift controller service
- CFM-3378 Automatic nar delivery should not die, if 404 happens
- CFM-3328 Add ListenNetFlow Processor
- CFM-3088 Add PostgreSQLConnectionPool
- CFM-3159 Add registry client for Cloudera Flow Library
- CFM-3239 ClouderaSchemaRegistry controller service
- CFM-1822 Handle custom schema in ListenGRPC

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 CEM 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 7.2.18.

Kafka

CDPD-62059: AvroConnectTranslator should handle null values in fromConnectData method

Fix possible NPE exception issues in connector tasks which operate with Avro data format using the Cloudera `AvroConverter`.

Apache patch information

None.

Schema Registry

CDPD-56890: New schemas cannot be created following an upgrade

Schemas can be created again after an upgrade even if the latest version of the schema was deleted before the upgrade.

CDPD-58265: Schema Registry Client incorrectly applies SSL configuration

The Cloudera distributed Schema Registry Java client applies the SSL configurations correctly even with concurrent access in Jersey clients.

CDPD-49470: Schema Registry Client retries requests more than the configured maxAttempts when multiple URLs are used

The Cloudera distributed Schema Registry Java client handles each request as one attempt, and does not attempt more retries based on the number of Schema Registry server URLs anymore.

OPSAPS-68139: Schema Registry does not apply cluster wide Kerberos principal mapping by default

The Schema Registry Kerberos Name Rules property is now empty by default. Schema Registry now automatically applies the cluster-wide auth-to-local (ATL) rules by default. During an upgrade, the previously configured value is preserved. If you have been using the default or a custom value, you must manually clear the property following an upgrade to transition to the new default value.

CDPD-55381: Schema Registry issues authentication cookie for the authorized user, not for the authenticated one

Schema Registry authentication cookie contains the correct authenticated user, even if the authenticated and the authorized users are different. Authenticated and authorized users can be different in scenarios where Schema Registry is used behind Knox.

CDPD-60160: Schema Registry Atlas integration does not work with Oracle databases

The Schema Registry Atlas integration works correctly when Oracle is used as the database of Schema Registry.

CDPD-59015: Schema Registry does not create new versions of schemas even if the schema is changed

The new fingerprinting version introduced in Cloudera Runtime 7.2.18 solves this issue.

CDPD-58990: The getSortedSchemaVersions method should order by version number and not by schemaVersionId

Schemas are ordered correctly by their version number during validation instead of their ID number.

CDPD-58949: Schemas are deduplicated during import

Importing works correctly and Schema Registry does not deduplicate imported schema versions.

Streams Messaging Manager

CDPD-53437: Alert notification messages do not contain an alert cause

In some cases, the resources and the attributes of the resource that triggered the alert were not included in the alert message. Typically, this happened when the alert policy defined a WITH ANY clause. From now on, resources that trigger the alert are always included in the alert message.

OPSAPS-69481: Anchor links are broken on Knox enabled clusters

Links that appear on the **Overview** page when hovering over producers, consumers, or partitions are no longer broken on Knox-enabled clusters.

Streams Replication Manager

CDPD-60426: Configuration changes are lost following a rolling restart of the service

SRM no longer fails to apply configuration changes if it is restarted with a rolling restart.

OPSAPS-67772: SRM Service metrics processing fails when the noexec option is enabled for /tmp

The SRM Service Kafka Streams application now uses the Kafka Streams state directory to extract the RocksDB .so files.

OPSAPS-67738: SRM Service role's Remote Querying feature does not work when the noexec option is enabled for /tmp

The SRM service no longer adds Netty native libraries to /tmp by default. As a result, the Remote Querying feature of SRM no longer fails if the noexec option is enabled for the /tmp directory. In addition, the directory where the Netty native libraries are placed is now user configurable with the SRM Service Netty Native Working Dir property.

OPSAPS-67742: The SRM Service role fails to start if properties are added to Additional Configs For Streams Application Running Inside SRM Service

The SRM Service role no longer fails to start if properties are added to the Additional Configs For Streams Application Running Inside SRM Service configuration. It is also possible to configure the internal Kafka Streams application of the SRM Service role.

Cruise Control

CDPD-47616: Unable to initiate rebalance, number of valid windows (NumValidWindows) is zero

This issue has been fixed as the Cloudera Manager Metrics Reporter is deprecated from Cruise Control.

OPSAPS-68148: Cruise Control rack aware goal upgrade handler

As Cruise Control automatically overrides all occurrences of the deprecated RackAwareGoal with RackAwareDistributionGoal during upgrade, the customized values of the Cruise Control goals will remain the same and there is no need to manually provide the values after an upgrade.

Fixed issues in Streaming Analytics

Review the list of Streaming Analytics issues that are resolved in Cloudera DataFlow for Data Hub 7.2.18.

- CSA-4464 - CSA parcel is built with an interim CDP build
- CSA-4624 - Sensitive field redaction can be circumvented
- CSA-4641 - Mask sensitive information in Environment
- CSA-4650 - Inconsistent sidebar collapse behavior
- CSA-4651 - Wrong padding for arrows on the right at the project homepage
- CSA-4670 - Default operator of MV query ruleset is not set on UI
- CSA-4672 - Fix chart wizard config options
- CSA-4699 - Keytab upload starts failing in SSB after some time, requiring a restart
- CSA-4723 - Adding delete source setting button
- CSA-4799 - Table Metadata is not saved when job is run via sql/execute
- CSA-4800 - ToString of Job can cause stack overflow
- CSA-4939 - Knox Materialized View endpoints are broken

- CSA-4973 - If auto TrustStore is enabled, validation fails on existing Kafka sources
- CSA-4988 - Restrict connector and data format uploads to administrators

Fixed CVEs in Cloudera DataFlow for Data Hub 7.2.18

Review the list of CVEs that are resolved in Cloudera DataFlow for Data Hub 7.2.18.

CVE-2021-45105 & CVE-2021-44832 remediation for CDF 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 CDF for Data Hub.

On February 1, 2022, Cloudera released a hotfix to Public 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 CDF for Data Hub cluster templates are running in the CDP Public Cloud environment powered by Runtime, Cloudera encourages users to upgrade their CDP services running Runtime versions from 7.2.7 so that they include the latest hotfixes. You can update your existing Data Lake 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 fixed in Cloudera Flow Management (CFM) in Data Hub 7.2.18.

All known NiFi CVEs are addressed in both clusters based on NiFi 1.25 and clusters based on NiFi 2.0-M1. See Apache NiFi Security for more information about NiFi's CVEs.

In Flow Management clusters using NiFi 1.25, 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.

- 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)

Behavioral Changes in Cloudera DataFlow for Data Hub 7.2.18

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 7.2.18.

Behavioral Changes in Flow Management

Review the list of Flow Management behavioral changes in Cloudera DataFlow for Data Hub 7.2.18.

Flow Management with NiFi 1

There are no behavioral changes for Flow Management clusters based on NiFi 1.25 in Cloudera DataFlow for Data Hub 7.2.18.

Flow Management with NiFi 2

NiFi 2.0 is a new, major release of Apache NiFi that introduces a lot of significant changes and enhacements, including some breaking changes for Flow Management clusters based on NiFi 2.0. It is important to familiarize yourself with the following points before migrating your existing flows.

⚠️ **Important:** Direct upgrades from NiFi 1.25 to NiFi 2.0 are currently not supported. You can create new Flow Management clusters based on NiFi 2 (Technical Preview) using the new templates and migrate your existing flows to the new clusters.

If you want to migrate a data flow, you need to export the process group as a JSON file from your NiFi 1.x cluster and import this JSON file into your NiFi 2.0 cluster. Tooling to help with upgrades and automatically manage the breaking changes will be provided in the next release.

Java 21

Java 21 is the minimum Java version required with NiFi 2.0. This version is automatically installed and configured on new Data Hub clusters using NiFi 2.0.

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 2.0, 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. The following components are impacted:

- org.apache.nifi.processors.script.ExecuteScript
- org.apache.nifi.processors.script.InvokeScriptedProcessor
- org.apache.nifi.processors.script.ScriptedFilterRecord
- org.apache.nifi.processors.script.ScriptedPartitionRecord
- org.apache.nifi.processors.script.ScriptedTransformRecord
- org.apache.nifi.processors.script.ScriptedValidateRecord
- org.apache.nifi.lookup.script.ScriptedLookupService
- org.apache.nifi.record.script.ScriptedReader
- org.apache.nifi.record.script.ScriptedRecordSetWriter
- org.apache.nifi.record.sink.script.ScriptedRecordSink
- org.apache.nifi.lookup.script.SimpleScriptedLookupService
- org.apache.nifi.reporting.script.ScriptedReportingTask

Removed components and replacement options

The following list contains the list of the components that have been removed between clusters based on NiFi 1.25 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
  - 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 2.0 by replacing the ghost components with the new implementations for each instance of the components listed above.

- Pulsar components

  All Pulsar components have been temporarily removed in this release. They will be reintroduced in upcoming releases. 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 7.2.18.

Kafka

Summary:

The javax.security.auth.useSubjectCredsOnly JVM property is set to true by default. As a result, connectors are required to use their own credentials. This default change is true for newly provisioned clusters. On upgraded clusters, useSubjectCredsOnly remains set to false to ensure backwards compatibility.

Previous behavior:

The javax.security.auth.useSubjectCredsOnly JVM property was set to false

New behavior:

The javax.security.auth.useSubjectCredsOnly JVM property is set to true by default. As a result, connectors are required to use their own credentials. If you are migrating connectors from a cluster running a previous version of Runtime to a new cluster running 7.2.18 or later, you must ensure that credentials are added to the connector configuration when migrated. Otherwise, migrated connectors may not work on the new cluster.

Schema Registry

There are no behavioral changes for Schema Registry in Cloudera DataFlow for Data Hub 7.2.18.

Streams Messaging Manager

There are no behavioral changes for Streams Messaging Manager in Cloudera DataFlow for Data Hub 7.2.18.

Streams Replication Manager

There are no behavioral changes for Streams Replication Manager in Cloudera DataFlow for Data Hub 7.2.18.

Cruise Control

There are no behavioral changes for Cruise Control in Cloudera DataFlow for Data Hub 7.2.18.

Behavioral Changes in Streaming Analytics

Review the list of Streaming Analytics behavioral changes in Cloudera DataFlow for Data Hub 7.2.18.

SQL Stream Builder

Summary:

PROCTIME() shortcut function is removed from LOOKUP JOIN syntax

Previous behavior:

The PROCTIME() function needed to be defined for the SQL query when using LOOKUP JOIN.

New behavior:

The raw PROCTIME() function cannot be used in LOOKUP JOIN queries anymore. For more information about migrating queries that use PROCTIME(), see Updating SQL queries with PROCTIME function documentation.

Summary:

The user-defined function (UDF) tester is not available on Streaming SQL Console

Previous behavior:

New UDFs could be tested on Streaming SQL Console before executing the SQL job.

New behavior:

The UDF test runner is turned off, the added UDFs can be tested with executing the SQL job.

Summary:

Kafka sink JSON serializer is refactored to the Flink implementation

Previous behavior:

An SSB specific implementation was used for the JSON serializer.

New behavior:

The RowJsonSerializer is added to be used for the Kafka sink JSON serializer. The v2 API version is also added for the JSONSampler.

Summary:

Remove user name length limitation

Previous behavior:

Username limitation was set to a minimum four characters.

New behavior:

Limitation is removed for username length.

Summary:

Make API v2 the default selection in the SSB Swagger UI

Previous behavior:

Version could be selected for the REST API when using Swagger UI.

New behavior:

V2 is the default version when using REST API with Swagger UI.

Summary:

ssb-hive-catalog can be configured when creating Iceberg catalog

Previous behavior:

ssb-hive-catalog could only be configured for creating Iceberg tables

New behavior:

ssb-hive-catalog can be used both for creating Iceberg catalogs and tables for seamless Hive integration.

Flink
Summary:

Remove global JobResultStore storage path

Previous behavior:

The JobResultStore was configured as a global HDFS path.

New behavior:

JobResultStore is automatically provided by YARN.