Cloudera Data Flow for Data Hub 7.2.0

Cloudera Data Flow for Data Hub Release Notes

Date published: 2020-05-01
Date modified: 2020-06-30

https://docs.cloudera.com/

Legal Notice

Contents

What's New in Cloudera Data Flow for Data Hub

This section lists major features and updates for Cloudera Data Flow for Data Hub.

June 30, 2020

General availability release for Streams Messaging clusters

This release introduces the GA release of Streams Messaging cluster definitions and templates for installation using Data Hub. The Streams Messaging templates include Kafka, Schema Registry, Streams Messaging Manager, and ZooKeeper. There are four template options, depending on your cloud provider and operational objectives:

- Streams Messaging Heavy Duty for AWS
- Streams Messaging Heavy Duty for Azure
- Streams Messaging Light Duty for AWS
- Streams Messaging Light Duty for Azure

Streams Messaging provides advanced messaging and real-time processing on streaming data using Apache Kafka, centralized schema management using Schema Registry, as well as management and monitoring capabilities powered by Streams Messaging Manager.

These templates set up fault-tolerant standalone deployments of Apache Kafka and supporting Cloudera components (Schema Registry and Streams Messaging Manager), which can be used for Kafka workloads in the cloud or as a disaster recovery instance for on-premises Kafka clusters.

June 16, 2020

Rebase on Kafka 2.4.1

Kafka shipped with this version of Cloudera Runtime is based on Apache Kafka 2.4.1. For more information, see *Apache Kafka Notable Changes* for versions 2.4.0 and 2.4.1, as well as the *Apache Kafka Release Notes* for versions 2.4.0 and 2.4.1 in the upstream documentation.

Schema Registry High Availability

The Streams Messaging Heavy Duty cluster definitions are configured to include a highly available Schema Registry. Schema Registry is now available on two cluster nodes with an external database. For updated cluster layout information, see the cluster layout information in *Planning your Streams Messaging Deployment*.

Schema Registry Ranger integration

Schema Registry is now automatically integrated with Ranger. When you create a Streams Messaging cluster you will see a new corresponding entry in the environment's Ranger instance where you can define authorization policies on schemas. You can define policies on entire schemas, a specific branch or even a specific version of a schema. For more information, see *Securing Schema Registry*.

Updated Streams Messaging: Heavy Duty cluster layout

The cluster nodes have been updated with clearer node definitions and roles. This means that your Streams Messaging: Heavy Duty cluster will consume one more instance, by default. For updated cluster layout information, see *Planning your Streams Messaging Deployment*.

Additionally, the Heavy Duty template now provisions an external HA database to store service information in a fault-tolerant way. Specifically for Streams Messaging clusters, this covers the Schema Registry database and SMM database.

Updated Streams Messaging: Light Duty cluster layout

The Heavy Duty template now provisions an external non-HA database to store service information externally. Specifically for Streams Messaging clusters, this covers the Schema Registry database and SMM database.

Related Information

Apache Kafka 2.4.0 Notable Changes
Apache Kafka 2.4.1 Notable Changes
Securing Schema Registry
Planning your Streams Messaging Deployment

Component support

Cloudera Data Flow for Data Hub includes the following components.

Flow Management clusters

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.5.0

Streams Messaging clusters

- Apache Kafka 2.4.1
- Schema Registry 0.8.1
- Streams Messaging Manager 2.1.0

Unsupported features

Some features exist within this release of Cloudera Data Flow for Data Hub components, but are not supported by Cloudera.

Unsupported Apache NiFi extensions

The following Apache NiFi Processors, Controller Services, and Reporting Tasks were developed and tested by the Cloudera community but are not officially supported by Cloudera. These features are excluded for a variety of reasons, including insufficient reliability or incomplete test case cover, declaration of non-production readiness by the community at large, and feature deviation from Cloudera best practices. Do not use these features in your production environments.

Unsupported Apache NiFi processors

- AttributeRollingWindow
- CompareFuzzyHash
- ConsumeIMAP
- ConsumeKafka_0_11
- ConsumeKafkaRecord_0_11
- ConsumePOP3
- ConvertExcelToCSVProcessor
- CountText
- DebugFlow
- DeleteMongo
- DeleteRethinkDB
- ExecuteSparkInteractive
- ExtractCCDAAttributes
- ExtractEmailAttachments
- ExtractEmailHeaders

- ExtractMediaMetadata
- ExtractTNEFAttachments
- ExecuteFlumeSink
- ExecuteFlumeSource
- FuzzyHashContent
- GetDynamoDB
- GetHDFSEvents
- GetMongo
- GetRethinkDB
- GetSNMP
- ISPEnrichIP
- InferAvroSchema
- ListenBeats
- ListenLumberjack
- ListenSMTP
- ModifyBytes
- MoveHDFS
- ParseNetflowv5
- ParseSyslog5424
- PostSlack
- PublishKafka_0_11
- PublishKafkaRecord_0_11
- PutCassandraRecord
- PutIgniteCache
- PutMongo
- PutMongoRecord
- PutRethinkDB
- PutSlack
- QueryDNS
- RunMongoAggregation
- SetSNMP
- SpringContextProcessor
- StoreInKiteDataset

Note:

Flow Management clusters in CDP Public Cloud do not support Hive 2. As a result, NiFi Processors working with Hive (PutHiveQL, PutHiveStreaming, SelectHiveQL) and the NiFi Controller Service HiveConnectionPool may not support Hive 2.

Unsupported Apache NiFi Controller Services

- ActionHandlerLookup
- AlertHandler
- ConfluentSchemaRegistry
- EasyRulesEngineService
- EasyRulesEngineProvider
- ExpressionHandler
- GraphiteMetricReporterService
- IPLookupService
- JettyWebSocketClient
- JettyWebSocketServer

- LivySessionController
- LogHandler
- MongoDBControllerService
- MongoDBLookupService
- PropertiesFileLookupService
- RecordSinkHandler
- ScriptedActionHandler
- ScriptedRulesEngine
- SimpleCsvFileLookupService
- XMLFileLookupService

Unsupported Apache NiFi Reporting Tasks

Community Driven NiFi Reporting Tasks

- AzureLogAnalyticsProvenanceReportingTask
- AzureLogAnalyticsReportingTask
- DataDogReportingTask
- MetricsReportingTask
- StandardGangliaReporter

Unsupported Streams Messaging features

Some Streams Messaging features exist within this release of Cloudera Data Flow for Data Hub components, but are not supported by Cloudera.

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 based clients are supported. Clients developed with C, C++, Python, .NET and other languages are currently not supported.
- Kafka Connect is not supported. NiFi and Sqoop are proven solutions for batch and real time data loading that complement Kafka's message broker capability. For more information, see *Creating your First Flow Management cluster*.
- The Kafka default authorizer is not supported. This includes setting ACLs and all related APIs, broker functionality, and command-line tools.

Related Information

Cloudera Community Forum

Creating your first Streams Messaging cluster

Unsupported Flow Management features

Some Flow Management features exist within this release of Cloudera Data Flow for Data Hub components, but are not supported by Cloudera.

The following NiFi 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*.

- Encrypted flow file repository
- Encrypted content repository

Unsupported customizations

Cloudera cannot guarantee that default NiFi processors are compatible with proprietary protocol implementations or proprietary interface extensions. For example, we support interfaces like JMS and JDBC that are built around standards, specifications, or open protocols. But we do not support customizations of those interfaces, or proprietary extensions built on top of those interfaces.

Related Information

Cloudera Community Forum

Apache patch information

The following sections list patches in each Cloudera Data Flow in Data Hub component, beyond what was fixed in the base version of the Apache component.

NiFi patches

This release provides Apache NFi 1.11.4 and these additional Apache patches.

- NIFI-7103 – PutAzureDataLakeStorageGen2 processor to provide native support for Azure Data lake Storage Gen 2 Storage
- NIFI-7173
- NIFI-7221 – Add support for protocol v2 and v3 with Schema Registry
- NIFI-7257 – Add Hadoop-based DBCPConnectionPool
- NIFI-7259 – DeleteAzureDataLakeStorage processor to provide native delete support for Azure Data lake Gen 2 Storage
- NIFI-7269 – Solrj in nifi-solr-nar needs upgrade
- NIFI-7278 – Kafka_consumer_2.0 sasl.mechanism SCRAM-SHA-512 not allowed
- NIFI-7279 – When using a DetectDuplicate processor with a RedisDistributedMapCacheClientService cache service, if the Cache Entry Identifier is not in the cache and Cache The Entry Identifier is set to "false", a NullPointerException is thrown rather than the expected non-duplicate relationship.
- NIFI-7281 – Using ListenTCPRecord and CsvReader produces an exception
- NIFI-7286 – ListenTCPRecord does not release port when stopping or terminating while running
- NIFI-7287 – Prometheus NAR missing dependency on SSL classes
- NIFI-7294 – Flows with SolrProcessor configured to use SSLContextService are failing
- NiFi-7314 – HandleHttpRequest should stop Jetty in OnUnscheduled instead of OnStopped
- NiFi-7345 – Multiple entity is created in Atlas for one Hive table if table name contains uppercase characters
- – ScriptedReportingTask gives a NoClassDefFoundError: org/apache/nifi/metrics/jvm/JmxJvmMetrics
- SHA-512

NiFi Registry patches

This release provides Apache NFi Registry 0.5.0 and these additional Apache patches.

- NIFIREG-297 – Upgrade to latest LTS release of Node (and npm)
- NIFIREG-319 – Remove code coverage instrumentation from nifi-fds js modules
- NIFIREG-324 – UI: include hammerJS with unit tests
- NIFIREG-327 – Add a section to docs covering recommended antivirus exclusions
- NIFIREG-329 – Add build profile to test with all supported DBs
- NIFIREG-337 – Support PostGres 10
- NIFIREG-339 – Remove errors in root pom

- NIFIREG-341 – Update README to make use of ASF url for community slack channel
- NIFIREG-348 – Update NiFi logo and icon file
- NIFIREG-352 – Update frontend dependencies
- NIFIREG-354 – Update VersionedFlowSnapshot to work without VersionedFlowSnapshotMetadata
- NIFIREG-357 – Set the autocomplete HTML5 tag to false for username/password login fields
- NIFIREG-361 – Improve logout handling
- NIFIREG-362 – Upgrade out of date dependencies
- NIFIREG-363 – Migrate to Github Actions CI
- NIFIREG-366 – Update NiFi Registry's parent pom version
- NIFIREG-369 – Permissions in lib should be 0644
- NIFIREG-370 – StandardRevisableEntityService returns wrong version on update
- NIFIREG-372 – Downgrade H2 version due to error on upgrade
- NIFIREG-374 – Logging into secured LDAP nifi registry some functionality is incorrectly disabled

Known issues and limitations

This provides a summary of known issues for Cloudera Data Flow in Data Hub.

ReplaceText with multiple concurrency tasks can result in data corruption

ReplaceText, when scheduled to run with multiple Concurrent Tasks, and using a Replacement Strategy of "Regular Expression" or "Literal Replace" can result in content being corrupted.

The issue is far more likely to occur with multiple Concurrent Tasks, but it may be possible to trigger when using a single Concurrent Task.

To get the fix for this issue, file a support case through the Cloudera portal.

Schema Registry is unavailable if Knox has failed

Schema Registry depends on Knox, but Knox is not highly available. If Knox fails, your clients cannot reach Schema Registry.

To work around this issue, switch to an internal Schema Registry configuration. To do this, configure Schema Registry with the host name, port, and direct URL rather than a dynamic Knox endpoint.

Atlas does not connect lineage when NiFi is writing files to S3 or ADLS Gen2

When running a flow that writes data to S3 using `PutHDFS` or `PutS3` processors or when writing files to ADLS Gen2 using `PutHDFS` or `PutADLS`, the Atlas type reported by NiFi is "nifi_dataset" while Atlas is expecting it to be of type "aws_s3_pseudo_dir" or "adls_gen2_directory". As a result, while we can show lineage of NiFi flows, the lineage will not be connected to subsequent processes that use these S3 or ADLS files.

Lineage can be connected manually in Atlas if required.

CFM-1017

`PutAzureDataLakeStorage` has several limitations

- You can add files to read-only buckets
- There is no check for file overwriting. It is possible to overwrite data.
- To add files to a bucket root level, set the destination with an empty string, rather than " / ".

`PutAzureDataLakeStorage` was introduced in CFM 2.0.0, for inclusion in Flow Management clusters in CDP Public Cloud. It is not available in HDF 3.5.x or CFM 1.1.x

You can use the `PutHDFS` processor to write data to Azure Data Lake Storage. See Ingesting Data into Azure Data Lake Storage for details.

Adjust `PublishKafkaProcessor` default timeout value for cloud

The 5000 ms timeout for PublishKafkaRecord processor when "Delivery Guarantee" is set to "all" might not be enough depending on your network setup and workload on the Kafka cluster you are connecting to.

The error message may look similar to:

```
2020-01-22 09:50:12,854 ERROR
org.apache.nifi.processors.kafka.pubsub.PublishKafkaRecord_2_0:
PublishKafkaRecord_2_0[id=cca729a5-016f-1000-ffff-ffffa3429f0c]
Failed to send StandardFlowFileRecord[uuid=03d8a3ab-e1a3-41a4-9
fa1-07af176aeb56,
claim=StandardContentClaim [resourceClaim=StandardResourceClai
m[id=1579683456791-21,
container=default, section=21], offset=844488, length=20],offset
=0,
name=NLgsvfQuAi2aad67b4-b053-4e42-b9e4-a10153941229,size=20] to
Kafka: org.apache.kafka.common.errors.
TimeoutException: Failed to update metadata after 5000 ms.
```

Increase timeout for Max Metadata Wait Time and Acknowledgement Wait Time in the processor configuration.

CFM-661

Terminating a Flow Management cluster does not delete the cluster specific NiFi and NiFi Registry repositories in Ranger

When a new Flow Management cluster is created, the setup process creates new repository entries in Ranger to allow cluster specific Ranger policies. These cluster specific Ranger repositories are not being deleted when a cluster is terminated. This can lead to issues when a cluster is terminated and another cluster with the same name is being created afterwards. The new cluster will re-use the existing Ranger repository but now the NiFi component UUIDs in existing policies do not match the UUIDs of the new cluster.

Manually delete the cluster specific Ranger repositories in Ranger after terminating a Flow Management cluster.

CB-5566

Terminating a Streams Messaging cluster does not delete the cluster specific Kafka repositories in Ranger

When a new Streams Messaging cluster is created, the setup process creates new repository entries in Ranger to allow cluster specific Ranger policies. These cluster specific Ranger repositories are not being deleted when a cluster is terminated.

Manually delete the cluster specific Ranger repositories in Ranger after terminating a Streams Messaging cluster

Scaling Kafka Brokers or NiFi Nodes up/down is not possible

Data Hub does not allow users to resize Kafka broker or NiFi node groups

There is no workaround for this issue.

Technical Service Bulletins

TSB 2022-580: NiFi Processors cannot write to content repository

If the content repository disk is filled more than 50% (or any other value that is set in nifi.properties for nifi.content.repository.archive.max.usage.percentage), and if there is no data in the content repository archive, the following warning message can be found in the logs: "Unable to write

flowfile content to content repository container default due to archive file size constraints; waiting for archive cleanup". This would block the processors and no more data is processed.

This appears to only happen if there is already data in the content repository on startup that needs to be archived, or if the following message is logged: “Found unknown file XYZ in the File System Repository; archiving file”.

Upstream JIRA

- NIFI-10023
- NIFI-9993

Knowledge article

For the latest update on this issue see the corresponding Knowledge article: TSB 2022-580: NiFi Processors cannot write to content repository

Fixed issues

Fixed issues represents 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.

`ReportLineageToAtlas` Reporting task is throwing errors on a new Flow Management cluster

Ranger policies that allow NiFi to publish metadata to Atlas are not created automatically, which prevents NiFi from writing to Atlas.

The FQDNs of the NiFi nodes in a Flow Management cluster are not registered with public DNS

Some NiFi use cases require inbound connectivity to NiFi from external systems using hostnames. Currently the FQDNs of NiFi nodes cannot be resolved over the public internet.

NiFi Registry API endpoint is not displayed in the list of exposed cluster endpoints in Data Hub UI

For all services running within a Flow Management cluster, Knox is set up to proxy requests to the endpoints. While the NiFi Registry API is configured to be proxied by Knox, the endpoint URI is not exposed in the Data Hub cluster management UI.

Latency monitoring tab displays an error if metrics collection is disabled

If latency metrics are disabled and user clicks on the Latency tab in the UI, an error is thrown. This issue has been fixed such that the tab is hidden if metrics collection is disabled.

CSP-778

Common vulnerabilities and exposures

Lists common vulnerabilities and exposures (CVEs) resolved in this CDF for Data Hub release.

There are no CVEs fixed in this release.