Cloudera Data Flow for Data Hub 7.1.0

Cloudera Data Flow for Data Hub Release Notes

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

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.

May 1, 2020

This release introduces the following cluster definitions for Flow Management in CDP Public Cloud:

- Flow Management Light Duty for AWS
- Flow Management Light Duty for Azure
- Flow Management Heavy Duty for AWS
- Flow Management Heavy Duty for Azure

These cluster definitions support installing Flow Management clusters running Apache NiFi and Apache NiFi Registry

Flow Management delivers high-scale data ingestion, transformation, and management to enterprises from any-to-any environment. It addresses key enterprise use cases such as data movement, continuous data ingestion, log data ingestion, and acquisition of all types of streaming data including social, mobile, clickstream, and IoT data.

The Flow Management template includes a no-code data ingestion and management solution powered by Apache NiFi. With NiFi's intuitive graphical interface and 300+ processors, Flow Management enables easy data ingestion and movement between CDP services as well as 3rd party cloud services. NiFi Registry is automatically set up and provides a central place to manage versioned Data Flows.

December 18, 2019

Note:

Streams Messaging clusters are available for Technical Preview. Cloudera encourages you to explore these technical preview features in non-production environments and provide feedback on your experiences through the Cloudera Support Portal.

This release introduces the technical preview release of Streams Messaging cluster definitions for installation using Data Hub. The Streams Messaging templates include Kafka, Schema Registry, Streams Messaging Manager, and ZooKeeper. There are two template options, depending on your 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-prem Kafka clusters.

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

- Schema Registry 0.8.1
- Apache Kafka 2.3.0
- Streams Messaging Manager 2.0.1

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.

Note:

Streams Messaging clusters are available for Technical Preview. Cloudera encourages you to explore these technical preview features in non-production environments and provide feedback on your experiences through the Cloudera Support Portal.

The following Kafka features are not ready for production deployment. Cloudera encourages you to explore these features in non-production environments and rpovide 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.

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.

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.

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

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.

Attempting to write to Atlas may result in an error message similar to:

```
"ReportLineageToAtlas[id=843ce571-0171-1000-ffff-ffffefdc49dd]
Error running task ReportLineageToAtlas[id=843ce571-0171-1000-ff
ff-ffffefdc49dd] due
to java.lang.RuntimeException: Failed to check and create NiFi
 flow type definitions in
Atlas due to org.apache.atlas.AtlasServiceException: Metadata ser
vice API
org.apache.atlas.AtlasClientV2$API_V2@3ea5b832 failed with sta
tus 403 (Forbidden)
Response Body ({"errorCode":"ATLAS-403-00-001","errorMessage":"
nifi is not authorized
to perform create entity-def nifi_output_port"})"
```

As an `environmentAdmin`, go to Ranger and add the `nifi` user to the following pre-existing Ranger policies:

Ranger repository name	Policy name	Action
cm_atlas	all - entity-type, entity-classification, entity	Add nifi user to the allow condition entry where 'Create Entity', 'Update Entity', and similar permissions are allowed.
cm_atlas	all - type-category, type	Add nifi user to the allow condition entry where 'Create Type', 'Update Type', and similar permissions are allowed.
cm_kafka	ATLAS_HOOK	Add nifi user to the allow condition entry where the Publish permissions was allowed.

After adding the `nifi` user to the specified policies, wait up to 5 minutes to confirm that the pre-defined `ReportLineageToAtlas` task can successfully publish lineage to Atlas. You do not need to restart `ReportLineageToAtlas` to pick up this configuration change. To see the newly published information in Atlas, refresh your open Atlas browser windows

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

Add a mapping of FQDN and public IP address of the NiFi nodes to the local hosts file of your external system

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.

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.

If you need to access the NiFi Registry API through Knox, use the following endpoint format: `https://<gateway-nod-fqdn>/<clustername>/cdp-proxy-api/nifi-registry-app/nifi-registry-api/`.

CB-6660

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.

Restarting a Streams Messaging cluster should refresh the DNS entries for the Broker nodes

Data Hub clusters created in a subnet which assigns public IPs now correctly update the DNS entries when a cluster is restarted to point to the newly assigned public IP address. This ensures that Kafka clients will always be able to use the FQDNs of the Kafka Brokers in their connection configuration.

CB-5341

Common vulnerabilities and exposures

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

Lists common vulnerabilities and exposures fixed in Apache NiFi 1.11.4, available in the Flow Management cluster definition for CDP Public Cloud.

CVE-2019-11358

Component: Apache NiFi

Description: Various vulnerabilities existed within the JQuery dependency used by NiFi. See NIST NVD CVE-2019-11358 for more information.

Severity: Medium

Versions Affected: Apache NiFi 1.6.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-11358

CVE-2019-10247, CVE-2019-10246

Component: Apache NiFi

Description: Various vulnerabilities existed within the Jetty dependency used by NiFi. See NIST NVD CVE-2019-10247, NIST NVD CVE-2019-10246 for more information.

Severity: Medium

Versions Affected: Apache NiFi 1.8.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-10247

CVE-2019-16335, CVE-2019-14540, CVE-2019-14439, CVE-2019-12814, CVE-2019-12384, CVE-2019-12086, CVE-2018-1000873, CVE-2018-19362, CVE-2018-19361, CVE-2018-19360

Component: Apache NiFi

Description: Various vulnerabilities existed within the Jackson Core: Databind dependency used by NiFi. See NIST NVD CVE-2019-16335, NIST NVD CVE-2019-14540, NIST NVD CVE-2019-14439, NIST NVD CVE-2019-12814, NIST NVD CVE-2019-12384, NIST NVD CVE-2019-12086, NIST NVD CVE-2018-1000873, NIST NVD CVE-2018-19362, NIST NVD CVE-2018-19361, NIST NVD CVE-2018-19360 for more information.

Severity: Medium

Versions Affected: Apache NiFi 1.0.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-16335

CVE-2019-0193, CVE-2019-0192, CVE-2017-3164

Component: Apache NiFi

Description: Various vulnerabilities existed within the Solr dependency used by NiFi. See NIST NVD CVE-2019-0193, NIST NVD CVE-2019-0192, NIST NVD CVE-2017-3164 for more information.

Severity: Critical

Versions Affected: Apache NiFi 1.0.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-0193

CVE-2017-5637, CVE-2016-5017, CVE-2018-8012

Component: Apache NiFi

Description: Various vulnerabilities existed within the Zookeeper dependency used by NiFi. See NIST NVD CVE-2018-8012, NIST NVD CVE-2017-5637, NIST NVD CVE-2016-5017 for more information.

Severity: Important

Versions Affected: Apache NiFi 1.0.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2017-5637

CVE-2019-10083

Component: Apache NiFi

Description: When updating a Process Group via the API, the response to the request includes all of its contents (at the top most level, not recursively). The response included details about processors and controller services which the user may not have had read access to.

Severity: Low

Versions Affected: Apache NiFi 1.3.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-10083

CVE-2019-12421

Component: Apache NiFi

Description: If NiFi uses an authentication mechanism other than PKI, when the user clicks Log Out, NiFi invalidates the authentication token on the client side but not on the server side. This permits the user's client-side token to be used for up to 12 hours after logging out to make API requests to NiFi.

Severity: Moderate

Versions Affected: Apache NiFi 1.0.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-12421

CVE-2019-10080

Component: Apache NiFi

Description: The XMLFileLookupService allowed trusted users to inadvertently configure a potentially malicious XML file. The XML file has the ability to make external calls to services (via XXE) and reveal information such as the versions of Java, Jersey, and Apache that the NiFI instance uses.

Severity: Low

Versions Affected: Apache NiFi 1.3.0 - 1.9.2

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-10080

CVE-2019-10768

Component: Apache NiFi

Description: An Object.prototype pollution vulnerability existed within the AngularJS dependency used by NiFi. See NIST NVD CVE-2019-10768 for more information.

Severity: Important

Versions Affected: Apache NiFi 1.8.0 - 1.10.0

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2019-10768

CVE-2020-1933

Component: Apache NiFi

Description: Malicious scripts could be injected to the UI through action by an unaware authenticated user in Firefox. Did not appear to occur in other browsers.

Severity: Important

Versions Affected: Apache NiFi 1.0.0 - 1.10.0

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2020-1933

CVE-2020-1928

Component: Apache NiFi

Description: The sensitive parameter parser would log parsed property descriptor values for debugging purposes. This would expose literal values entered in a sensitive property when no parameter was present.

Severity: Moderate

Versions Affected: Apache NiFi 1.10.0

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2020-1928

CVE-2020-1942

Component: Apache NiFi

Description: The flow fingerprint factory generated flow fingerprints which included sensitive property descriptor values. In the event a node attempted to join a cluster and the cluster flow was not inheritable, the flow fingerprint of both the cluster and local flow was printed, potentially containing sensitive values in plaintext.

Severity: Important

Versions Affected: Apache NiFi 0.0.1 - 1.11.0

Apache CVE Report Link: https://nifi.apache.org/security.html#CVE-2020-1942