Cloudera Flow Management 2.1.6

Cloudera Flow Management Release Notes

Date published: 2019-06-26
Date modified: 2023-08-29

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

What's new in this release?

Learn about the new functionalities and improvements in Cloudera Flow Management (CFM) and how these new features benefit you.

CFM 2.1.6 Service Pack 1

On March 11, 2024, Cloudera released CFM 2.1.6 Service Pack 1 (SP1). For more information about the issues fixed in CFM 2.1.6 SP1, see Fixed issues and Fixed CVEs.

If you are using CFM 2.1.6.0, upgrade to **CFM 2.1.6.1000** to access the latest version of the software. You can find the new download links in Download locations.

CFM 2.1.6

The CFM 2.1.6 release is based on Apache NiFi 1.23.1 and it also incorporates a lot of Cloudera exclusive features and improvements. Here is an overview of what is new in this release:

Components documentation

Previously, access to NiFi component documentation was limited to using the Apache NiFi website or the NiFi UI in a running instance. However, with this release, Cloudera makes the documentation of all CFM components (general and Cloudera-specific) available as part of Cloudera Flow Management documentation.

It is important to note that Cloudera Flow Management incorporates over 100 components that are not available in Apache NiFi. To access this comprehensive documentation, see the Cloudera Apache NiFi Components Reference Guide.

Flow Library

Cloudera introduces a new Registry Client implementation, designed to facilitate access to the exclusive Cloudera Flow Library. This library contains predefined flow configurations to expedite the deployment of use cases. While the Flow Library will see ongoing enhancements, it already offers access to all ReadyFlows included in the Cloudera DataFlow for Public Cloud data service.

For additional information and instructions on how to configure the registry client, see Using Flow Library Registry Client .

Cloudera Manager integration

NiFi's integration with Cloudera Manager has been improved by introducing a set of new features:

- Automatic Heap Dump Capture: In the event of an Out Of Memory error, NiFi now automatically captures a heap dump.
- Schema Registry Integration: You can include Schema Registry as a dependency during NiFi installation. Additionally, you can configure a Cloudera Schema Registry controller service to seamlessly connect with the Schema Registry instance of CDP.
- Thread Pool Adjustment: The default size of NiFi's Thread Pool has been updated to align with the available CPU cores on the NiFi host.
- Enhanced Health Checks: This release incorporates additional health checks related to NiFi's memory utilization and configuration.

Support Matrix

In adherence to the End-of-Life (EOL) policies of the supported operating systems, support for Ubuntu 18 is dicontinued and support for SLES 15 and RHEL 9 is introduced. For a comprehensive list of supported operating systems, see the Supported operating systems.

New components added since CFM 2.1.5 SP1 release

New processors:

- ExtractRecordSchema

  You can use this processor to define a Record Reader and have the schema of your data inferred and added as a FlowFile attribute. If schema inference is required, Cloudera recommends to use this processor over the unsupported and deprecated InferAvroSchema processor.
- GenerateRecord
- GetAwsPollyJobStatus
- GetAwsTextractJobStatus
- GetAwsTranscribeJobStatus
- GetAwsTranslateJobStatus
- GetAzureQueueStorage_v12
- ListenNetFlow

  This processor supports receiving NetFlow Export Packets over UDP for the NetFlow versions 1, 5, and 9.
- ModifyCompression
- PutAzureQueueStorage_v12
- PutIcebergCDC [Technical Preview]

  You can use this processor to apply CDC events into Iceberg formatted tables.

  

  Note: The processor supports equality deletes which is not supported yet by other compute engines on CDP. It means that in case of delete operations, the files created by the processor may not be readable by engines like Hive, Spark, etc. Improvements are being made on the compute engines to support equality deletes.
- QueryIoTDBRecord
- RemoveRecordField
- StartAwsPollyJob
- StartAwsTextractJob
- StartAwsTranscribeJob
- StartAwsTranslateJob
- ValidateJson
- VerifyContentMAC

New controller services:

- ADLSCredentialsControllerServiceLookup
- AmazonGlueSchemaRegistry
- AzureServiceBusJMSConnectionFactoryProvider
- AzureStorageCredentialsControllerServiceLookup_v12
- ClouderaSchemaRegistry

  You can use this controller service as a replacement of the HortonworksSchemaRegistry controller service, as it has been deprecated and marked for removal in anticipation of NiFi 2.0 release in the Apache NiFi project.
- CMLLookupService [Technical Preview]

  You can use this controller service in combination with the LookupRecord processor to enrich data by calling Cloudera Machine Learning API to do ML scoring on streams of data.
- EBCDICRecordReader [Technical Preview]

  You can use this reader with Cobol copybook files to read Mainframe data with EBCDIC encoding and convert the data into another structured format such as JSON, Avro, and so on.
- ExcelReader
- PostgreSQLConnectionPool
- RedshiftConnectionPool
- StandardFileResourceService

New parameter provider:

- CyberArkConjurParameterProvider

  You can use this parameter provider to source the values of your parameters from a CyberArk instance.

Related Information

Using Parameter Providers

CFM component versions

Review the official component versions for Cloudera Flow Management (CFM) for compatibility with other applications.

Note: NiFi works with the version of NiFi Registry shipped with your version of CFM or later.

CFM 2.1.6 SP1

- Apache NiFi 1.23.1.2.1.6.1000
- Apache NiFi Registry 1.23.1.2.1.6.1000

CFM 2.1.6

- Apache NiFi 1.23.1.2.1.6.0
- Apache NiFi Registry 1.23.1.2.1.6.0

CFM 2.1.5 SP1

- Apache NiFi 1.18.0.2.1.5.1000
- Apache NiFi Registry 1.18.0.2.1.5.1000

CFM 2.1.5

- Apache NiFi 1.18.0.2.1.5.0
- Apache NiFi Registry 1.18.0.2.1.5.0

CFM 2.1.4 SP1

- Apache NiFi 1.16.0.2.1.4.1000
- Apache NiFi Registry 1.16.0.2.1.4.1000

CFM 2.1.4

- Apache NiFi 1.16.0.2.1.4.0
- Apache NiFi Registry 1.16.0.2.1.4.0

CFM 2.1.3

- Apache NiFi 1.15.2.2.1.3.0
- Apache NiFi Registry 1.15.2.2.1.3.0

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

CFM 2.1.2

- Apache NiFi 1.13.2.2.1.2.0
- Apache NiFi Registry 0.8.0.2.1.2.0

CFM 2.1.1

- Apache NiFi 1.13.2.2.1.1.0
- Apache NiFi Registry 0.8.0.2.1.1.0

CFM 2.0.4

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

CFM 2.0.1

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

Support matrix

Review the support matrix before you start installing Cloudera Flow Management (CFM).

System requirements

Review the system requirements before getting started with installing Cloudera Flow Management (CFM).

Supported versions of CDP

CFM 2.1.6 supports the following versions of CDP Private Cloud Base:

- CDP 7.1.9
- CDP 7.1.8
- CDP 7.1.7 and all Service Packs

Note: CFM provides a set of monitoring features when managed by Cloudera Manager. For these features to be available and working, you need to be using Cloudera Manager 7.6.1 or above.

Supported JAVA Development Kits (JDK)

Supported JDKs:

- Oracle Java™ SE Development Kit 8, Update 252 (JDK 8u252) and later
- OpenJDK 1.8, Update 252 (JDK 8u252) and later
- OpenJDK 11
- OpenJDK 17
- Azul Zulu JDK 1.8, Update 252 (JDK 8u252) and later
- Azul Zulu JDK 11
- Azul Zulu JDK 17

Other system requirements

ZooKeeper

You must install the ZooKeeper service available with your CDP Private Cloud Base cluster.

Python

When deploying CFM on RHEL 8 and using Cloudera Manager with Python 2, you need to specify a symbolic link to python2.

```
ln -s /usr/bin/python2 /usr/bin/python
```

Number of cores

Four cores per NiFi node is the minimum number of cores required by Cloudera to be supported. Cloudera recommends eight cores per NiFi node as it usually provides the best starting point for the most common use cases.

Supported operating systems

Review the list of operating systems supported by Cloudera Flow Management (CFM).

Operating system	Versions
CentOS	• 7.6 • 7.7 • 7.8 • 7.9 • 8.2 • 8.4
RHEL	• 7.6 • 7.7 • 7.8 • 7.9 • 8.2 • 8.4 • 8.6 • 8.7 • 8.8 • 9.1
SLES	• 12 SP5 • 15 SP4
Ubuntu	• 20.04
Windows	• 10 • Server 2016 • Server 2019

Note:

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

NiFi Registry is not supported on Windows.

Supported NiFi Registry databases

Review the list of databases supported by NiFi Registry.

- H2
- PostgreSQL 11.x
- PostgreSQL 12.x

- PostgreSQL 13.x
- PostgreSQL 14.x
- PostgreSQL 15.x
- MySQL 8.x

Related Information

Supported NiFi processors
Supported NiFi controller services
Supported NiFi reporting tasks
Components supported by partners

Supported NiFi processors

Cloudera Flow Management (CFM) is shipped with Apache NiFi and includes a set of processors, most of which are supported by Cloudera. You should be familiar with the available supported processors, and avoid using any unsupported processors in production environments.

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 or incomplete test case coverage, declaration of non-production readiness by the community at large, and feature deviation from Cloudera best practices.

AttributesToCSV	GetGcpVisionAnnotateImagesOperationStatus	PutElasticsearchHttpRecord1
AttributesToJSON	GetHBase	PutElasticsearchJson
Base64EncodeContent	GetHDFS	PutElasticsearchRecord1
CalculateRecordStats	GetHDFSFileInfo	PutEmail
CaptureChangeMySQL	GetHDFSSequenceFile	PutFile
CompressContent1,2	GetHTMLElement	PutFTP1
ConnectWebSocket	GetHTTP	PutGCSObject
ConsumeAMQP	GetHubSpot	PutGoogleDrive
ConsumeAzureEventHub	GetIgniteCache	PutGridFS
ConsumeEWS	GetJiraIssue	PutHBaseCell
ConsumeGCPubSub	GetJMSQueue	PutHBaseJSON
ConsumeGCPubSubLite	GetJMSTopic	PutHBaseRecord1
ConsumeJMS	GetMongoRecord	PutHDFS
ConsumeKafka_1_0	GetSFTP	PutHive3QL
ConsumeKafka_2_0	GetShopify	PutHive3Streaming
ConsumeKafka_2_6	GetSNMP	PutHiveQL
ConsumeKafka2CDP	GetSnowflakeIngestStatus	PutHiveStreaming
ConsumeKafka2RecordCDP	GetSolr	PutHTMLElement
ConsumeKafkaRecord_1_0	GetSplunk	PutIceberg
ConsumeKafkaRecord_2_0	GetSQS	PutIcebergCDC [Technical Preview]
ConsumeKafkaRecord_2_6	GetTCP	PutInfluxDB
ConsumeKinesisStream	GetTwitter	PutJMS1
ConsumeMQTT1	GetWorkdayReport	PutKinesisFirehose
ConsumeTwitter	GetZendesk	PutKinesisStream

ConsumeWindowsEventLog	HandleHttpRequest	PutKudu
ControlRate	HandleHttpResponse	PutLambda
ConvertAvroSchema	HashAttribute	PutMongoRecord
ConvertAvroToJSON	HashContent	PutORC1
ConvertAvroToORC	IdentifyMimeType	PutParquet
ConvertAvroToParquet	InvokeAWSGatewayApi	PutRecord
ConvertCharacterSet	InvokeGRPC	PutRedisHashRecord [Technical Preview]
ConvertCSVToAvro	InvokeHTTP	PutRiemann
ConvertJSONToAvro	InvokeScriptedProcessor	PutS3Object
ConvertJSONToSQL	JoinEnrichment	PutSalesforceObject
ConvertProtobuf	JoltTransformJSON	PutSFTP
ConvertRecord	JoltTransformRecord	PutSmbFile
CreateHadoopSequenceFile	JSLTTransformJSON	PutSnowflakeInternalStage [Technical Preview]
CryptographicHashAttribute	JsonQueryElasticsearch	PutSNS
CryptographicHashContent	ListAzureBlobStorage	PutSolrContentStream
DecryptContent	ListAzureBlobStorage_v12	PutSolrRecord
DecryptContentCompatibility	ListAzureDataLakeStorage	PutSplunk
DecryptContentPGP	ListBoxFile	PutSplunkHTTP
DeduplicateRecord	ListCDPObjectStore	PutSQL
DeleteAzureBlobStorage	ListDatabaseTables	PutSQS1
DeleteAzureBlobStorage_v12	ListDropbox	PutSyslog
DeleteAzureDataLakeStorage	ListenBeats	PutTCP
DeleteByQueryElasticsearch	ListenFTP	PutUDP
DeleteCDPObjectStore	ListenGRPC*	PutWebSocket1
DeleteDynamoDB	ListenGRPC*	QueryAirtableTable
DeleteGCSObject	ListenHTTP	QueryCassandra
DeleteGridFS	ListenNetFlow	QueryDatabaseTable1
DeleteHBaseCells	ListenRELP	QueryDatabaseTableRecord
DeleteHBaseRow	ListenSyslog	QueryElasticsearchHttp
DeleteHDFS	ListenTCP	QueryRecord
DeleteS3Object	ListenTCPRecord	QuerySalesforceObject
DeleteSQS	ListenTrapSNMP	QuerySolr
DetectDuplicate	ListenUDP	QuerySplunkIndexingStatus
DistributeLoad	ListenUDPRecord	QueryWhois
DuplicateFlowFile	ListenWebSocket	RemoveRecordField
EncodeContent	ListFile	ReplaceText
EncryptContent2	ListFTP	ReplaceTextWithMapping
EncryptContentPGP	ListGCSBucket	ResizeImage1
EnforceOrder	ListGoogleDrive	RetryFlowFile
EvaluateJsonPath	ListHDFS	RouteHL7

EvaluateXPath	ListS3	RouteOnAttribute
EvaluateXQuery	ListSFTP	RouteOnContent
ExecuteGroovyScript	ListSmb	RouteText
ExecuteInfluxDBQuery	LogAttribute	SampleRecord
ExecuteProcess	LogMessage	ScanAccumulo
ExecuteScript	LookupAttribute	ScanAttribute1
ExecuteSQL	LookupRecord	ScanContent
ExecuteSQLRecord	MergeContent	ScanHBase
ExecuteStateless1,2	MergeRecord1	ScriptedFilterRecord
ExecuteStreamCommand	ModifyCompression	ScriptedPartitionRecord
ExtractAvroMetadata	ModifyHTMLElement	ScriptedTransformRecord
ExtractGrok	MonitorActivity	ScriptedValidateRecord
ExtractHL7Attributes	MoveAzureDataLakeStorage	ScrollElasticsearchHttp
ExtractImageMetadata	MoveHDFS	SearchElasticsearch
ExtractRecordSchema	Notify	SegmentContent
ExtractText	PaginatedJsonQueryElasticsearch	SelectHive3QL1
FetchAzureBlobStorage	ParseCEF1	SelectHiveQL
FetchAzureBlobStorage_v12	ParseEvtx	SendTrapSNMP
FetchAzureDataLakeStorage	ParseSyslog	SetSNMP
FetchBoxFile	PartitionRecord	SignContentPGP
FetchCDPObjectStore	PostHTTP	SplitAvro
FetchDistributedMapCache	PublishAMQP	SplitContent
FetchDropbox	PublishGCPubSub1	SplitJson1
FetchElasticsearchHttp	PublishGCPubSubLite1	SplitRecord1
FetchFile	PublishJMS1	SplitText1
FetchFTP	PublishKafka_1_0	SplitXml
FetchGCSObject	PublishKafka_2_0	StartAwsPollyJob
FetchGoogleDrive	PublishKafka_2_6	StartAwsTextractJob
FetchGridFS	PublishKafka2CDP	StartAwsTranscribeJob
FetchHBaseRow	PublishKafka2RecordCDP	StartAwsTranslateJob
FetchHDFS	PublishKafkaRecord_1_0	StartGcpVisionAnnotateFilesOperation
FetchParquet	PublishKafkaRecord_2_0	StartGcpVisionAnnotateImagesOperation
FetchS3Object	PublishKafkaRecord_2_6	StartSnowflakeIngest [Technical Preview]
FetchSFTP	PublishMQTT	TagS3Object
FetchSmb	PutAccumuloRecord1	TailFile
FlattenJson	PutAzureBlobStorage	TransformXml
ForkEnrichment	PutAzureBlobStorage_v12	TriggerHiveMetaStoreEvent
ForkRecord	PutAzureCosmosDBRecord	UnpackContent
GenerateFlowFile	PutAzureDataLakeStorage1	UpdateAttribute
GenerateRecord	PutAzureEventHub	UpdateByQueryElasticsearch

GenerateTableFetch	PutAzureQueueStorage1	UpdateCounter
GeoEnrichIP	PutAzureQueueStorage_v12	UpdateDatabaseTable
GeoEnrichIPRecord	PutBigQuery	UpdateDeltaLakeTable [Technical Preview]
GeohashRecord	PutBigQueryBatch	UpdateHive3Table
GetAsanaObject	PutBigQueryStreaming	UpdateHiveTable
GetAwsPollyJobStatus	PutBoxFile	UpdateRecord
GetAwsTextractJobStatus	PutCassandraQL1	ValidateCsv
GetAwsTranscribeJobStatus	PutCassandraRecord1	ValidateJson
GetAwsTranslateJobStatus	PutCDPObjectStore	ValidateRecord
GetAzureEventHub	PutCloudWatchMetric	ValidateXml
GetAzureQueueStorage	PutCouchbaseKey	VerifyContentMAC
GetAzureQueueStorage_v12	PutDatabaseRecord1	VerifyContentPGP
GetCouchbaseKey1	PutDistributedMapCache	Wait
GetElasticsearch	PutDropbox	YandexTranslate
GetFile	PutDynamoDB	
GetFTP	PutDynamoDBRecord	
GetGcpVisionAnnotateFilesOperationStatus	PutElasticsearchHttp1	

Footnotes

- 1 – indicates a memory intensive processor
- 2 – indicates a CPU intensive processor
- * – there are two ListenGRPC processors available, one is provided by Apache and the other is provided by Cloudera

Related Information

Supported NiFi Registry databases
Supported NiFi controller services
Supported NiFi reporting tasks
Components supported by partners

Supported NiFi controller services

Cloudera Flow Management (CFM) is shipped with Apache NiFi and includes a set of controller services, most of which are supported by Cloudera. You should be familiar with the available supported controller services, and avoid using any unsupported controller services in production environments.

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 or incomplete test case coverage, declaration of non-production readiness by the community at large, and feature deviation from Cloudera best practices.

AccumuloService	HiveConnectionPool
ActionHandlerLookup	HortonworksSchemaRegistry
ADLSCredentialsControllerService	IPFIXReader
ADLSCredentialsControllerServiceLookup	IPLookupService
ADLSIDBrokerCloudCredentialsProviderControllerService	JASN1Reader
AlertHandler	JMSConnectionFactoryProvider

AmazonGlueSchemaRegistry	JndiJmsConnectionFactoryProvider
AvroReader	JsonConfigBasedBoxClientService
AvroRecordSetWriter	JsonPathReader
AvroSchemaRegistry	JsonRecordSetWriter
AWSCredentialsProviderControllerService	JsonTreeReader
AWSIDBrokerCloudCredentialsProviderControllerService	KafkaRecordSink_1_0
AzureBlobIDBrokerCloudCredentialsProviderControllerService	KafkaRecordSink_2_0
AzureCosmosDBClientService	KafkaRecordSink_2_6
AzureEventHubRecordSink	KerberosKeytabUserService
AzureServiceBusJMSConnectionFactoryProvider	KerberosPasswordUserService
AzureStorageCredentialsControllerService	KerberosTicketCacheUserService
AzureStorageCredentialsControllerService_v12	KeytabCredentialsService
AzureStorageCredentialsControllerServiceLookup	KuduLookupService
AzureStorageCredentialsControllerServiceLookup_v12	LoggingRecordSink
CassandraDistributedMapCache	LogHandler
CassandraSessionProvider	MongoDBControllerService
CdpCredentialsProviderControllerService	MongoDBLookupService
CdpOauth2AccessTokenProviderControllerService	ParquetReader
CEFReader	ParquetRecordSetWriter
CiscoEmblemSyslogMessageReader	PostgreSQLConnectionPool
ClouderaSchemaRegistry	PrometheusRecordSink
CMLLookupService [Technical Preview]	ReaderLookup
ConfluentSchemaRegistry	RecordSetWriterLookup
CouchbaseClusterService	RecordSinkHandler
CouchbaseKeyValueLookupService	RecordSinkServiceLookup
CouchbaseMapCacheClient	RedisConnectionPoolService
CouchbaseRecordLookupService	RedisDistributedMapCacheClientService
CSVReader	RedshiftConnectionPool
CSVRecordLookupService	RestLookupService
CSVRecordSetWriter	ScriptedActionHandler
DatabaseRecordLookupService	ScriptedLookupService
DatabaseRecordSink	ScriptedReader
DBCPConnectionPool	ScriptedRecordSetWriter
DBCPConnectionPoolLookup	ScriptedRecordSink
DistributedMapCacheClientService	ScriptedRulesEngine
DistributedMapCacheLookupService	SimpleDatabaseLookupService
DistributedMapCacheServer	SimpleKeyValueLookupService
DistributedSetCacheClientService	SimpleScriptedLookupService
DistributedSetCacheServer	SiteToSiteReportingRecordSink
EasyRulesEngineProvider	SmbjClientProviderService

EasyRulesEngineService	SnowflakeComputingConnectionPool
EBCDICRecordReader [Technical Preview]	StandardAsanaClientProviderService
ElasticSearchClientServiceImpl	StandardAzureCredentialsControllerService
ElasticSearchLookupService	StandardDropboxCredentialService
ElasticSearchStringLookupService	StandardFileResourceService
EmailRecordSink	StandardHashiCorpVaultClientService
EmbeddedHazelcastCacheManager	StandardHttpContextMap
ExcelReader	StandardOauth2AccessTokenProvider
ExpressionHandler	StandardPGPPrivateKeyService
ExternalHazelcastCacheManager	StandardPGPPublicKeyService
FreeFormTextRecordSetWriter	StandardPrivateKeyService
GCPCredentialsControllerService	StandardProxyConfigurationService
GrokReader	StandardRestrictedSSLContextService
HadoopDBCPConnectionPool	StandardS3EncryptionService
HadoopCatalogService	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	

Related Information

Supported NiFi Registry databases
Supported NiFi processors
Supported NiFi reporting tasks
Components supported by partners

Supported NiFi reporting tasks

Cloudera Flow Management (CFM) is shipped with Apache NiFi and includes a set of reporting tasks, most of which are supported by Cloudera. You should be familiar with the available supported reporting tasks, and avoid using any unsupported reporting tasks in production environments.

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

- ReportLineageToAtlas
- ScriptedReportingTask
- SiteToSiteBulletinReportingTask
- SiteToSiteMetricsReportingTask
- SiteToSiteProvenanceReportingTask
- SiteToSiteStatusReportingTask

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 or incomplete test case coverage, 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.

Related Information

Supported NiFi Registry databases
Supported NiFi processors
Supported NiFi controller services
Components supported by partners

Supported NiFi parameter providers

Cloudera Flow Management (CFM) is shipped with Apache NiFi and includes a set of parameter providers, most of which are supported by Cloudera. You should be familiar with the available supported parameter providers, and avoid using any unsupported parameter providers in production environments.

- AwsSecretsManagerParameterProvider
- AzureKeyVaultSecretsParameterProvider
- CyberArkConjurParameterProvider (Cloudera exclusive)
- DatabaseParameterProvider
- EnvironmentVariableParameterProvider
- FileParameterProvider
- GcpSecretManagerParameterProvider
- HashiCorpVaultParameterProvider

Additional parameter providers are developed and tested by the Cloudera community but are not officially supported by Cloudera. Parameter providers are excluded for a variety of reasons, including insufficient reliability or incomplete test case coverage, 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.

Components supported by partners

Learn about the processors and controller services built and supported by Cloudera partners.

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

Processors supported by partners

- ConsumePulsar (v1.18.0)
- ConsumePulsarRecord (v1.18.0)
- PublishPulsar (v1.18.0)
- PublishPulsarRecord (v1.18.0)

Controller services supported by partners

- PulsarClientAthenzAuthenticationService (v1.18.0)
- PulsarClientJwtAuthenticationService (v1.18.0)
- PulsarClientOauthAuthenticationService (v1.18.0)

- PulsarClientTlsAuthenticationService (v1.18.0)
- StandardPulsarClientService (v1.18.0)

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.

Related Information

Supported NiFi Registry databases
Supported NiFi processors
Supported NiFi controller services
Supported NiFi reporting tasks

Download locations

You can download the Cloudera Flow Management (CFM) software artifacts from the Cloudera Archive. There are different CFM artifacts for different operating systems, standalone components, and Windows files.

Use the following tables to identify the Cloudera Flow Management (CFM) repository location for your operating system and operational objectives.

Note:

You must have credentials to download CFM files. Your download credential is not the same as the credential you use to access the Cloudera Support Portal.

You can get download credentials in the following ways:

- Contact your Cloudera sales representative.
- Check the Welcome email you have received for your Flow Management account.
- File a non-technical case on the Cloudera Support Portal for the Cloudera Support team to assist you.

Table 1: RHEL/CentOS 7

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/parcel/CFM-2.1.6.1000-47-el7.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/parcel/CFM-2.1.6.1000-47-el7.parcel.sha
CSD	**NiFi** https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/parcel/NIFI-1.23.1.2.1.6.1000-47.jar **NiFi Registry** https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/parcel/NIFIREGISTRY-1.23.1.2.1.6.1000-47.jar

Table 2: RHEL/CentOS 8

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat8/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat8/yum/tars/parcel/CFM-2.1.6.1000-47-el8.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat8/yum/tars/parcel/CFM-2.1.6.1000-47-el8.parcel.sha

File	Location
CSD	**NiFi:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat8/yum/tars/parcel/NIFI-1.23.1.2.1.6.1000-47.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat8/yum/tars/parcel/NIFIREGISTRY-1.23.1.2.1.6.1000-47.jar

Table 3: RHEL 9

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat9/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat9/yum/tars/parcel/CFM-2.1.6.1000-47-el9.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat9/yum/tars/parcel/CFM-2.1.6.1000-47-el9.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat9/yum/tars/parcel/NIFI-1.23.1.2.1.6.1000-47.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat9/yum/tars/parcel/NIFIREGISTRY-1.23.1.2.1.6.1000-47.jar

Table 4: SLES 12

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles12/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles12/yum/tars/parcel/CFM-2.1.6.1000-47-sles12.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles12/yum/tars/parcel/CFM-2.1.6.1000-47-sles12.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles12/yum/tars/parcel/NIFI-1.23.1.2.1.6.1000-47.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles12/yum/tars/parcel/NIFIREGISTRY-1.23.1.2.1.6.1000-47.jar

Table 5: SLES 15

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles15/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles15/yum/tars/parcel/CFM-2.1.6.1000-47-sles15.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles15/yum/tars/parcel/CFM-2.1.6.1000-47-sles15.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles15/yum/tars/parcel/NIFI-1.23.1.2.1.6.1000-47.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/sles15/yum/tars/parcel/NIFIREGISTRY-1.23.1.2.1.6.1000-47.jar

Table 6: Ubuntu 20

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.6.1000/ubuntu20/apt/tars/parcel/manifest.json

File	Location
Parcel	https://archive.cloudera.com/p/cfm2/2.1.6.1000/ubuntu20/apt/tars/parcel/CFM-2.1.6.1000-47-focal.parcel
Parcel SHA file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/ubuntu20/apt/tars/parcel/CFM-2.1.6.1000-47-focal.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/ubuntu20/apt/tars/parcel/NIFI-1.23.1.2.1.6.1000-47.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm2/2.1.6.1000/ubuntu20/apt/tars/parcel/NIFIREGISTRY-1.23.1.2.1.6.1000-47.jar

Table 7: Standalone components (OS agnostic)

File	Location
NiFi (.tar.gz)	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/cdf_extensions/nifi-1.23.1.2.1.6.1000-47-bin.tar.gz
NiFi (.tar.gz.sha256)	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/cdf_extensions/nifi-1.23.1.2.1.6.1000-47-bin.tar.gz.sha256
NiFi Registry (.tar.gz)	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/nifi/nifi-registry-1.23.1.2.1.6.1000-47-bin.tar.gz
NiFi Registry (.tar.gz.sha256)	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/nifi/nifi-registry-1.23.1.2.1.6.1000-47-bin.tar.gz.sha256
NiFi Toolkit (.tar.gz)	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/nifi/nifi-toolkit-1.23.1.2.1.6.1000-47-bin.tar.gz
NiFi Toolkit (.tar.gz.sha256)	https://archive.cloudera.com/p/cfm2/2.1.6.1000/redhat7/yum/tars/nifi/nifi-toolkit-1.23.1.2.1.6.1000-47-bin.tar.gz.sha256

Table 8: Windows files

File	Location
NiFi MSI	https://archive.cloudera.com/p/cfm2/2.1.6.1000/windows/nifi-2.1.6.1000-47.msi
NiFi MSI SHA file	https://archive.cloudera.com/p/cfm2/2.1.6.1000/windows/nifi-2.1.6.1000-47.msi.sha256

Unsupported features

The following features are 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 coverage, 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 customizations

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

Technical preview features

The following features are available in Cloudera Flow Management (CFM) 2.1.6 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.

Processors in technical preview:

- PutIcebergCDC processor

  Note: The processor supports equality deletes which is not supported yet by other compute engines on CDP. It means that in case of delete operations, the files created by the processor may not be readable by engines like Hive, Spark, and so on. Improvements are being made on the compute engines to support equality deletes.
- PutRedisHashRecord
- PutSnowflakeInternalStage
- StartSnowflakeIngest
- UpdateDeltaLakeTable

Controller services in technical preview:

- CMLLookupService controller service
- EBCDICRecordRecord controller service

Behavioral changes

Learn about behavioral changes in Cloudera Flow Management (CFM) 2.1.6.

Summary:

The Neo4JCypher3ClientService Controller Service has been completely removed in favor of the Neo4JCypherClientService controller service, which uses a more recent version of the underlying library.

Summary:

As part of NIFI-11614 and to ensure better security, some restrictions around the JndiJmsConnectionFactoryProvider controller service have been implemented.

New behavior:

The default validation for the JNDI Provider URL property only allows the following URL schemes:

- file
- jgroups
- ssl
- t3
- t3s
- tcp
- udp
- vm

If an additional URL scheme is required to interact with a specific JMS solution, a NiFi admin has to configure the following Java system property in the application bootstrap.conf file to override the default list: java.arg.jndiJmsUrlSchemesAllowed=-Dorg.apache.nifi.jms.cf.jndi.provider.url.schemes.allowed=ssl tcp

Note: The property must contain a space-separated list of URL schemes.

Known issues

Review the list of known issues in Cloudera Flow Management (CFM) 2.1.6.

Known issues in CFM 2.1.6 SP1

CVEs not fixed in CFM 2.1.6 SP1

A fix is required on HBase and/or Hive side, or alternatively, the use of a different version of the same is recommended:

CVE-2014-0114

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

CVE-2014-3643

jersey: XXE via parameter entities not disabled by the jersey SAX parser

CVE-2017-9735

Jetty through 9.4.x is prone to a timing channel in util/security/Password.java, which makes it easier for remote attackers to obtain access by observing elapsed times before rejection of incorrect passwords.

CVE-2018-1320

Apache Thrift Java client library versions 0.5.0 through 0.11.0 can bypass SASL negotiation isComplete validation in the org.apache.thrift.transport.TSaslTransport class. An assert used to determine if the SASL handshake had successfully completed could be disabled in production settings making the validation incomplete.

CVE-2020-27216

In Eclipse Jetty versions 1.0 thru 9.4.32.v20200930, 10.0.0.alpha1 through 10.0.0.beta2, and 11.0.0.alpha1 through 11.0.0.beta2O, on Unix like systems, the system's temporary directory is shared between all users on that system. A collocated user can observe the process of creating a temporary sub directory in the shared temporary directory and race to complete the creation of the temporary subdirectory. If the attacker wins the race then they will have read and write permission to the subdirectory used to unpack web applications, including their WEB-INF/lib jar files and JSP files. If any code is ever executed out of this temporary directory, this can lead to a local privilege escalation vulnerability.

CVE-2022-37865

Apache Ivy allows creating/overwriting any file on the system

With Apache Ivy 2.4.0 an optional packaging attribute has been introduced that allows artifacts to be unpacked on the fly if they used pack200 or zip packaging. For artifacts using the "zip", "jar" or "war" packaging Ivy prior to 2.5.1 does not verify the target path when extracting the archive. An archive containing absolute paths or paths that try to traverse "upwards" using ".." sequences can then write files to any location on the local file system that the user executing Ivy has write access to. Ivy users of version 2.4.0 to 2.5.0 should upgrade to Ivy 2.5.1.

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

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

CVE-2023-34610

An issue was discovered json-io thru 4.14.0 allows attackers to cause a denial of service or other unspecified impacts via crafted object that uses cyclic dependencies.

Not fixed for other reasons

CVE-2018-17196

In Apache Kafka versions between 0.11.0.0 and 2.1.0, it is possible to manually craft a Produce request which bypasses transaction/idempotent ACL validation. Only authenticated clients with Write permission on the respective topics are able to exploit this vulnerability. Users should upgrade to 2.1.1 or later where this vulnerability has been fixed.

Reason: Cloudera recommends you to use nifi-kafka-2-6-nar*

CVE-2018-20225

An issue was discovered in pip (all versions) because it installs the version with the highest version number, even if the user had intended to obtain a private package from a private index. This only affects use of the --extra-index-url option, and exploitation requires that the package does not already exist in the public index (and thus the attacker can put the package there with an arbitrary version number). NOTE: it has been reported that this is intended functionality and the user is responsible for using --extra-index-url securely

Reason: The used 2.7.3 version is the latest from jython-standalone.

CVE-2019-20916

The pip package before 19.2 for Python allows Directory Traversal when a URL is given in an install command, because a Content-Disposition header can have ../ in a filename, as demonstrated by overwriting the /root/.ssh/authorized_keys file. This occurs in _download_http_url in _internal/download.py.

Reason: The used 2.7.3 version is the latest from jython-standalone.

CVE-2020-9040

Couchbase Server Java SDK before 2.7.1.1 allows a potential attacker to forge an SSL certificate and pose as the intended peer. An attacker can leverage this flaw by crafting a cryptographically valid certificate that will be accepted by Java SDK's Netty component due to missing hostname verification.

Reason: Cloudera recommends customers to remove nifi-couchbase*.nar

CVE-2022-42889: Apache Commons Text prior to 1.10.0 allows RCE when applied to untrusted input due to insecure interpolation defaults

Apache Commons Text performs variable interpolation, allowing properties to be dynamically evaluated and expanded. The standard format for interpolation is "${prefix:name}", where "prefix" is used to locate an instance of org.apache.commons.text.lookup.StringLookup that performs the interpolation. Starting with version 1.5 and continuing through 1.9, the set of default Lookup instances included interpolators that could result in arbitrary code execution or contact with remote servers. These lookups are: - "script" - execute expressions using the JVM script execution engine

(javax.script) - "dns" - resolve dns records - "url" - load values from urls, including from remote servers Applications using the interpolation defaults in the affected versions may be vulnerable to remote code execution or unintentional contact with remote servers if untrusted configuration values are used. Users are recommended to upgrade to Apache Commons Text 1.10.0, which disables the problematic interpolators by default.

Reason: Dependency not fixed in nifi-kite-processors, it is recommended to remove nifi-kite-processor.

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

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

Reason: The requested versions of org.apache.derby require Java 9 or higher.

CVE-2023-24998: Apache Commons FileUpload, Apache Tomcat: FileUpload DoS with excessive parts

Apache Commons FileUpload before 1.5 does not limit the number of request parts to be processed resulting in the possibility of an attacker triggering a DoS with a malicious upload or series of uploads. Note that, like all of the file upload limits, the new configuration option (FileUploadBase#setFileCountMax) is not enabled by default and must be explicitly configured.

Reason: It is not possible to change atlas.version to other than 2.1.0.7.1.7.1000-141. The dependency was excluded from the atlas jar, the required dependency version was added.

CVE-2023-34062

In Reactor Netty HTTP Server, versions 1.1.x prior to 1.1.13 and versions 1.0.x prior to 1.0.39, a malicious user can send a request using a specially crafted URL that can lead to a directory traversal attack. Specifically, an application is vulnerable if Reactor Netty HTTP Server is configured to serve static resources.

Reason : The reactor-netty-http was addressed in NIFI-12393, only the relevant part of the change was applied.

CVE-2023-36478

HTTP/2 HPACK integer overflow and buffer allocation

Eclipse Jetty provides a web server and servlet container. In versions 11.0.0 through 11.0.15, 10.0.0 through 10.0.15, and 9.0.0 through 9.4.52, an integer overflow in `MetaDataBuilder.checkSize` allows for HTTP/2 HPACK header values to exceed their size limit. `MetaDataBuilder.java` determines if a header name or value exceeds the size limit, and throws an exception if the limit is exceeded. However, when length is very large and huffman is true, the multiplication by 4 in line 295 will overflow, and length will become negative. `(_size+length)` will now be negative, and the check on line 296 will not be triggered. Furthermore, `MetaDataBuilder.checkSize` allows for user-entered HPACK header value sizes to be negative, potentially leading to a very large buffer allocation later on when the user-entered size is multiplied by 2. This means that if a user provides a negative length value (or, more precisely, a length value which, when multiplied by the 4/3 fudge factor, is negative), and this length value is a very large positive number when multiplied by 2, then the user can cause a very large buffer to be allocated on the server. Users of HTTP/2 can be impacted by a remote denial of service attack. The issue has been fixed in versions 11.0.16, 10.0.16, and 9.4.53. There are no known workarounds.

Reason: The latest 9.x version is used, 10+ versions ofjetty require Java 9 or higher.

CVE-2023-39410: Apache Avro Java SDK: Memory when deserializing untrusted data in Avro Java SDK

When deserializing untrusted or corrupted data, it is possible for a reader to consume memory beyond the allowed constraints and thus lead to out of memory on the system. This issue affects Java applications using Apache Avro Java SDK up to and including 1.11.2. Users should update to apache-avro version 1.11.3 which addresses this issue.

Reason: Dependency not fixed in nifi-kite-processors and nifi-hive-1-1, it is recommended to remove those processors or use nifi-hive3.nar

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

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

Reason: Version from 5.17.x require Java 11, it is not possible to upgrade to 6.x.

CVE-2023-6378: Logback "receiver" DOS vulnerability

A serialization vulnerability in the logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.

Reason: Versions from 1.4.x require Java 11.

CVEs excluded based on the NiFi exclusion list

You can find the exclusion list here.

CVE-2023-25194: Apache Kafka Connect API: Possible RCE/Denial of service attack via SASL JAAS JndiLoginModule configuration using Kafka Connect

A possible security vulnerability has been identified in Apache Kafka Connect API. This requires access to a Kafka Connect worker, and the ability to create/modify connectors on it with an arbitrary Kafka client SASL JAAS config and a SASL-based security protocol, which has been possible on Kafka Connect clusters since Apache Kafka Connect 2.3.0. When configuring the connector via the Kafka Connect REST API, an authenticated operator can set the `sasl.jaas.config` property for any of the connector's Kafka clients to "com.sun.security.auth.module.JndiLoginModule", which can be done via the `producer.override.sasl.jaas.config`, `consumer.override.sasl.jaas.config`, or `admin.override.sasl.jaas.config` properties. This will allow the server to connect to the attacker's LDAP server and deserialize the LDAP response, which the attacker can use to execute java deserialization gadget chains on the Kafka connect server. Attackers can cause unrestricted deserialization of untrusted data (or) RCE vulnerability when there are gadgets in the classpath. Since Apache Kafka 3.0.0, users are allowed to specify these properties in connector configurations for Kafka Connect clusters running with out-of-the-box configurations. Before Apache Kafka 3.0.0, users may not specify these properties unless the Kafka Connect cluster has been reconfigured with a connector client override policy that permits them. Since Apache Kafka 3.4.0, asystem property ("-Dorg.apache.kafka.disallowed.login.modules") has been added to disable the problematic login modules usage in SASL JAAS configuration. Also by default "com.sun.security.auth.module.JndiLoginModule" is disabled in Apache Kafka Connect 3.4.0. All Kafka Connect users are advised to validate connector configurations and only allow trusted JNDI configurations. Also examine connector dependencies for vulnerable versions and either upgrade their connectors, upgrading that specific dependency, or removing the connectors as options for remediation. Finally, in addition to leveraging the "org.apache.kafka.disallowed.login.modules" system property, Kafka Connect users can also implement their own connector client config

override policy, which can be used to control which Kafka client properties can be overridden directly in a connector config and which cannot.

CVE-2023-4759: Improper handling of case insensitive filesystems in Eclipse JGit allows arbitrary file write

Arbitrary File Overwrite in Eclipse JGit <= 6.6.0 In Eclipse JGit, all versions <= 6.6.0.202305301015-r, a symbolic link present in a specially crafted git repository can be used to write a file to locations outside the working tree when this repository is cloned with JGit to a case-insensitive filesystem, or when a checkout from a clone of such a repository is performed on a case-insensitive filesystem. This can happen on checkout (DirCacheCheckout), merge (ResolveMerger via its WorkingTreeUpdater), pull (PullCommand using merge), and when applying a patch (PatchApplier). This can be exploited for remote code execution (RCE), for instance if the file written outside the working tree is a git filter that gets executed on a subsequent git command. The issue occurs only on case-insensitive filesystems, like the default file systems on Windows and macOS. The user performing the clone or checkout must have the rights to create symbolic links for the problem to occur, and symbolic links must be enabled in the git configuration. Setting the git configuration option core.symlinks = false before checking out avoids the problem. The issue was fixed in Eclipse JGit version 6.6.1.202309021850-r and 6.7.0.202309050840-r, available via Maven Central https://repo1.maven.org/maven2/org/eclipse/jgit/ and repo.eclipse.org https://repo.eclipse.org/content/repositories/jgit-releases/ . A backport is available in 5.13.3 starting from 5.13.3.202401111512-r. The JGit maintainers would like to thank RyotaK for finding and reporting this issue.

CVE-2024-21634: Ion Java StackOverflow vulnerability

Amazon Ion is a Java implementation of the Ion data notation. Prior to version 1.10.5, a potential denial-of-service issue exists in `ion-java` for applications that use `ion-java` to deserialize Ion text encoded data, or deserialize Ion text or binary encoded data into the `IonValue` model and then invoke certain `IonValue` methods on that in-memory representation. An actor could craft Ion data that, when loaded by the affected application and/or processed using the `IonValue` model, results in a `StackOverflowError` originating from the `ion-java` library. The patch is included in `ion-java` 1.10.5. As a workaround, do not load data which originated from an untrusted source or that could have been tampered with.

Known issues in CFM 2.1.6

Snowflake - NoSuchMethodError

Due to the issue documented in NIFI-11905, you may encounter the following error when utilizing Snowflake components:

```
java.lang.NoSuchMethodError: 'net.snowflake.client.jdbc.telemetr
y.Telemetry net.snowflake.client.jdbc.telemetry.TelemetryClient.
createSessionlessTelemetry(net.snowflake.client.jdbc.internal.ap
ache.http.impl.client.CloseableHttpClient, java.lang.String)'
    at net.snowflake.ingest.connection.TelemetryService.<init>
(TelemetryService.java:68)
```

This issue has been resolved with the introduction of NIFI-12126, which will be available in CFM 2.1.6 SP1. In the interim, Cloudera recommends to use the NARs from the CFM 2.1.5 SP1 release:

- nifi-snowflake-processors-nar
- nifi-snowflake-services-api-nar
- nifi-snowflake-services-nar

Per Process Group Logging

The Per Process Group logging feature is currently not working even when you specify a log suffix in the configuration of a process group. As a result, you may not observe the expected logging behavior.

No fix or workaround is available until a Service Pack is released for CFM 2.1.6. However, if you encounter this problem, contact Cloudera to request a fix.

Configuration of java.arg.7

A property has been added for defining java.arg.7 to provide the ability to override the default location of the temporary directory used by JDK. By default this value is empty in Cloudera Manager. If you use this argument for another purpose, change it to a different, unused argument number (or use letters instead: java.arg.*mycustomargument*). Not changing the argument can impact functionalities after upgrades/migrations.

JDK error

JDK 8 version u252 is supported. Any lower version may result in this error when NiFi starts:

```
SHA512withRSAandMGF1 Signature not available
```

When using Java 8, only version u252, and above are supported.

JDK limitation

JDK 8u271, JDK 8u281, and JDK 8u291 may cause socket leak issues in NiFi due to JDK-8245417 and JDK-8256818. Verify the build version of your JDK. Later builds are fixed as described in JDK-8256818.

When using Java 8, only version u252, and above are supported.

Kudu Client

All the records are sent as a single Kafka message containing an array of records.

There is an issue in the Kudu client preventing the creation of a new tables using the NiFi processors. The table needs to exist before NiFi tries to push data into it. You may see this error when this issue arises:

```
Caused by: org.apache.kudu.client.NonRecoverableException: failed
 to wait for Hive Metastore notification log listener to catch
 up: failed to retrieve notification log events: failed to open
 Hive Metastore connection: SASL(-15): mechanism too weak for
 this user
```

Verify the necessary table exists in Kudu.

NiFi Node Connection test failures

In CFM 2.1.3, Cloudera Manager includes a new health check feature. The health check alerts users if a NiFi instance is running but disconnected from the NiFi cluster. For this health check to be successful, you must update a Ranger policy. There is a known issue when the NiFi service is running but the NiFi Node(s) report `Bad Health` due to the NiFi Node Connection test.

Update the policy:

1. From the Ranger UI, access the Controller policy for the NiFi service.
2. Verify the `nifi` group is set in the policy.
3. Add the `nifi` user, to the policy, with READ permissions.

NiFi UI Performance considerations

A known issue in Chrome 92.x causes significant slowness in the NiFi UI and may lead to high CPU consumption.

For more information, see the *Chrome Known Issues documentation* at 1235045.

Use another version of Chrome or a different browser.

SSHJ version change and key negotiation issue with old SSH servers

ListSFTP and PutSFTP processors fail when using the legacy ssh-rsa algorithm for authentication with the following error:

```
UserAuthException: Exhausted available authentication methods
```

Set Key Algorithms Allowed property in PutSFTP to ssh-rsa.

KeyStoreException: placeholder not found

After an upgrade, NiFi may fail to start with the following error:

```
WARN org.apache.nifi.web.server.JettyServer: Failed to start web
 server... shutting down.
java.security.KeyStoreException: placeholder not found
```

The error is caused by missing configuration for the type of the keystore and truststore files.

1. Go to Cloudera Manager -> NiFi service -> Configuration.
2. Add the below properties for NiFi Node Advanced Configuration Snippet (Safety Valve) for staging/nifi.properties.xml.

   ```
   nifi.security.keystoreType=**[value]**
   nifi.security.truststoreType=**[value]**
   ```

   Where value must be PKCS12, JKS, or BCFKS. JKS is the preferred type, BCFKS and PKCS12 files are loaded with BouncyCastle provider.
3. Restart NiFi.

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 to generate 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 to infer the schema of your data with an appropriate reader and add the schema as a FlowFile attribute.

Fixed issues

Review the list of resolved issues.

Issues fixed in CFM 2.1.6 SP1

- NIFI-12827: Upgraded PostgreSQL JDBC test driver from 42.6.0 to 42.7.2
- NIFI-12745: Fixed AvroReader silently dropping malformed records
- NIFI-12732: ListS3 resets its tracking state after configuration change
- NIFI-12731: Ensure state is updated in GetHBase whenever the session is committed
- NIFI-12705: Updated metrics-jvm to 4.2.25 Additionally update metrics-graphite and metrics-core to 4.2.22
- NIFI-12682: Fixed MiNiFi agent manifest hash swaps
- NIFI-12677: Removed documentation of non-existent strategy for ExcelReader

- NIFI-12650: Upgraded json-path from 2.8.0 to 2.9.0
- NIFI-12612: In asn1 bundle handle OBJECT IDENTIFIER type as string.
- NIFI-12596: PutIceberg is missing case-insensitive Record type handling in List and Map types
- NIFI-12594: ListS3 - observe min/max object age when entity state tracking is used
- NIFI-12592: Upgraded Apache Curator from 5.5.0 to 5.6.0
- NIFI-12567: Prevent NPE in CuratorLeaderElectionManager.getLeadershipChangeCount
- NIFI-12562: Upgraded json-schema-validator from 1.0.87 to 1.1.0
- NIFI-12561: Fixed MergeContent DELIMITER_STRATEGY_NONE Handling
- NIFI-12559: Upgraded SSHJ from 0.37.0 to 0.38.0
- NIFI-12535: Fixed documentation for 'PadRight Examples' table name in the Expression Language Guide as well as the last example in the table
- NIFI-12526: Fixed handling of Fetch Size in QueryCassandra, added fragment attributes
- NIFI-12520: ExtractHL7Attributes processor ignores repeatable field values
- NIFI-12517: Updated isJson function to improve space handling
- NIFI-12516: Corrected Cluster Replicated Response Headers for HTTP/2
- NIFI-12506: Added Threading for Status Analytics Retrieval
- NIFI-12481: Filtering out unauthorized registry clients to avoid unhandled error
- NIFI-12470: Fixed forEach callback for usage with Object.entries() to address layout issue in Status History
- NIFI-12462: Upgraded Logback from 1.3.13 to 1.3.14
- NIFI-12441: Added No Tracking Strategy to ListS3
- NIFI-12438: Upgraded Logback from 1.3.11 to 1.3.13
- NIFI-12418: Corrected Provider Groups Missing in Refreshed Tokens
- NIFI-12416: Relocated the additionalDetails.html to the appropriate bundle in order for it to be seen when generating the documentation.
- NIFI-12412: Support Proxies for Blob Checkpoints in ConsumeAzureEventHub
- NIFI-12403: Improved Jolt UI Parameter Processing
- NIFI-12387: Initialize Controller Service Comments with Empty String
- NIFI-12383: Replication client should handle accept encoding with lowercase
- NIFI-12376: Fixed logic error with bitwise operator in AvroReader
- NIFI-12373: Added LICENSE and NOTICE for nifi-standard-shared-nar
- NIFI-12370: Fixed Distributed Map Cache Client Service Shutdown
- NIFI-12368: Clear versionedComponentId for copied Snippets
- NIFI-12363: Upgraded JLine from 3.23.0 to 3.24.1
- NIFI-12358: Fixed NPE in HostHeaderHandler
- NIFI-12355: Upgraded AMQP Client from 5.19.0 to 5.20.0
- NIFI-12346: Upgraded Apache Calcite from 1.35.0 to 1.36.0
- NIFI-12323: Removed String Length Limits from JSON Flow Configuration
- NIFI-12319: Upgraded ActiveMQ to 5.15.16
- NIFI-12318: Fixed byte array generation in GenerateRecord
- NIFI-12314: Fixed EL for SQL Query Property in QueryNiFiReportingTask
- NIFI-12276: Addressed Dependency Check Findings
- NIFI-12273: Fixed command.argument references in ExecuteStreamCommand docs
- NIFI-12271: Fixed PutAzureBlobStorage_v12 rollback on failure with FileResourceService
- NIFI-12265: Fixed OpenPGP Hexadecimal Key Formatting with leading 0
- NIFI-12254: Clarified Bulk operation header documentation for PutElasticsearchRecord and PutElasticsearchJson processors
- NIFI-12238: Fixed SplitText endline trimming with max fragment size
- NIFI-12237: Changed label height and width from POSITION to SIZE difference
- NIFI-12232: Corrected Group Component ID Handling for Clustered Flows

- NIFI-12228: This closes #7882. Fixed issue with FlowFile Concucrrency that can occasionally bring in more data than it should
- NIFI-12222: Fixed StandardVersionedComponentSynchronizerTest for support branch
- NIFI-12222: Protect against missing parameter context when syncing a PG in component synchronizer
- NIFI-12207: Upgraded Netty from 4.1.99 to 4.1.100
- NIFI-12194: Added Yield on Exceptions in Kafka Processors
- NIFI-12170: Upgraded snappy-java to 1.1.10.5
- NIFI-12165: Changed the properties "Custom Transformation Class Name" and "Custom Module Directory" to depend on the "Jolt Transformation DSL" property when its value is "Custom"
- NIFI-12160: Kafka Connect Check for NAR unpacking before starting
- NIFI-12158: MockProcessSession write methods preserves attributes
- NIFI-12154: Upgraded Apache Avro from 1.11.2 to 1.11.3
- NIFI-12151: Fixed StandardPrivateKeyService fails due to missing BouncyCastleProvider
- NIFI-12134: Disable Directory Listing property is duplicated on PutSFTP processor
- NIFI-12127: Allow Jackson's max string length to be configured on SplitJson and EvaluateJsonPath
- NIFI-12126: Downgrade snowflake-jdbc to 3.13.33 snowflake-ingest-sdk:2.0.3 is not compatible with snowflake-jdbc:3.14.x
- NIFI-12122: Fixed persistence of Parameter Context descriptions
- NIFI-12118: Refactored RemoveRecordPath member variable that was caching values, and improve performance with Pattern.matcher().find() instead of .match().
- NIFI-12117: Allow configuring Jackson's max string length in JoltTransformJSON
- NIFI-12107: Fixed sorting in the cluster table
- NIFI-12089: Fixed typo in additionalDetails of CSVReader
- NIFI-12084: Fixed per Process Group logging
- NIFI-12083: Upgraded Jetty from 9.4.51 to 9.4.52
- NIFI-12067: mock process session keeps track of flowfiles created during the session and removes them on rollback rather than putting them on the input queue
- NIFI-12063: Clarified Elasticsearch Query Documentation
- NIFI-12037: Changed List.of to Collections.singletonList for Java 8
- NIFI-12037: Updated AzureUserGroupProvider to allow configuration of the graph endpoint and API scope to support regional clouds
- NIFI-12034: Upgraded Apache Commons Compress from 1.23.0 to 1.24.0
- NIFI-12019: Improved reliability of TestSynchronousFileWatcher
- NIFI-12019: Bugfix for SynchronousFileWatcher time check interval
- NIFI-12014: NullPointerException in PutSQL when adding error attributes
- NIFI-12010: Handle auto-commit and commit based on driver capabilities in SQL components
- NIFI-11987: Set read buffer size in PutAzureBlobStorage_v12
- NIFI-11981: PublishGCPubSub failure / Record-based processing / AVRO
- NIFI-11980: Bump org.apache.ivy:ivy from 2.5.1 to 2.5.2
- NIFI-11976: Removed error log check in TestListenTCPRecord
- NIFI-11959: Corrected single-line comment handling for Jolt JSON
- NIFI-11909: Cleared Password field after login
- NIFI-11899: Corrected Bulletin Metrics Registry to return latest Bulletins
- NIFI-11782: Resolved NPE when moving snippet with label to process group
- NIFI-11739: Added ability to ignore missing fields in PutIceberg
- NIFI-11677: Removed non required yield in DeleteHDFS
- NIFI-11595: Backported StateMap.getStateVersion() for StateProviders
- NIFI-11595: StateProvider.replace() supports creating the initial state
- NIFI-11519: Fixed DBCPConnectionPool Sensitive Dynamic Properties
- NIFI-11389: Fixed controller services's link to referencing controller
- NIFI-11288: Added AWS STS dependency for AssumeRoleWithWebIdentity method

- NIFI-11177: Added defensive code for null values for Iceberg
- NIFI-9677: Fixed issue that an empty JSON array causes flow file to be considered unmatched even though it should be considered as a match.
- NIFI-9464: Fixed race condition between "Timer-Driven" threads when running SiteToSiteProvenanceReportingTask.onTrigger and "Compress Provenance Log" threads running EventFileCompressos.run that can cause the SiteToSiteProvenanceReportingTask.onTrigger to pair an already compressed .prov.gz file with a .toc file that corresponds to the uncompressed .prov file.
- NIFI-8135: Allow CHOICE data types in conversion of Records to Java Maps
- NIFI-5137: Fixed the path to Controller Service grid item's state
- CFM-3775:
  - Updated scala-library version to 2.13.12 to mitigate CVE-2022-36944 and Reactor Netty client to 1.0.34 to mitigate CVE-2023-34062
  - Removed support for old Postgres versions
  - Updated org.json:json version to 20231013 to mitigate CVE-2023-5072
  - Updated snappy-java to 1.1.10.5 (CVE-2023-43642) excluded bcprov-ext references (CVE-2018-1000180, CVE-2018-1000613) updated snakeyaml to 2.2 (CVE-2017-18640)
  - Updated commons-fileupload to 1.5 and graal-sdk to 23.1.2 to fix CVE-2023-24998 and CVE-2024-20932
  - Updated jetty.version to 9.4.54.v20240208 to mitigate CVE-2023-36478
  - Updated box-java-sdk to 4.6.1 and reactor-netty-http to 1.1.15
- CFM-3687: Updated Commons-text-1.9.jar in nifi-cdf-iceberg-nar

Issues fixed in CFM 2.1.6

CFM 2.1.6 uses Apache NiFi 1.23.1 with additional commits on top of it. It includes all fixed issues of this Apache NiFi release as well as the below list:

- CFM-1822: Added nifi-cdf-grpc-nar to assembly
- CFM-1822: Added ARM64 protoc binaries in nifi-cdf-grpc-bundle
- CFM-2853: Added CyberarkConjurParameterProvider
- CFM-3022: Excluded Content-Type header in custom S3 signer
- CFM-3088: Added PostgreSQLConnectionPool
- CFM-3159: Added registry client for Cloudera Flow Library
- CFM-3159: Fixed StringUtils dependency in nifi-cdf-flow-library modules
- CFM-3168: Added EBCDICRecordReader
- CFM-3239: ClouderaSchemaRegistry controller service
- CFM-3325: Redshift controller service
- CFM-3328: Added ListenNetFlow Processor
- CFM-3328: Added nifi-cdf-netflow-nar to nifi-assembly
- CFM-3329: CML Lookup Service
- CFM-3335: PutIcebergCDC processor
- CFM-3378: Automatic nar delivery should not die, if 404 happens
- CFM-3425: Created AzureServiceBusJMSConnectionFactoryProvider
- CFM-3471: Allowing ClouderFlowLibraryClient to work with unknown fields
- CFM-3488: Corrected padding handling for ListenNetFlow Options Templates
- NIFI-11817: Fixed ListCDPObjectStore after ListHDFS refactor
- NIFI-11889: Added Record-oriented Transmission to PutTCP
- NIFI-11916: Iceberg processor extensibility improvement
- NIFI-11971: Ensured that if no bytes are written to a file after calling ProcessSession.write() that content claim's length is set to 0 when closing OutputStream
- NIFI-11971: Ensured that when creating a new content claim that an existing claim length of -1 is always treated as 0 to ensure that -1 is never added to the offset

For a summary of improvements, bug fixes and new features delivered with Apache NiFi, see the following release notes:

- Apache NiFi 1.19.0
- Apache NiFi 1.19.1
- Apache NiFi 1.20.0
- Apache NiFi 1.21.0
- Apache NiFi 1.22.0
- Apache NiFi 1.23.0
- Apache NiFi 1.23.1

Fixed Common Vulnerabilities and Exposures

Review the list of fixed Common Vulnerabilities and Exposures (CVE).

CVEs fixed in CFM 2.1.6 SP1

CVE-2017-15095

A deserialization flaw was discovered in the jackson-databind in versions before 2.8.10 and 2.9.1. This flaw could allow an unauthenticated user to run arbitrary code by sending the maliciously crafted input to the readValue method of the ObjectMapper. This issue extends the previous flaw CVE-2017-7525 by blacklisting more classes that could be used maliciously.

CVE-2017-18640

The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564.

CVE-2017-7686

Apache Ignite 1.0.0-RC3 to 2.0 uses an update notifier component to update the users about new project releases that include additional functionality, bug fixes and performance improvements. To do that the component communicates to an external PHP server (http://ignite.run) where it needs to send some system properties like Apache Ignite or Java version. Some of the properties might contain user sensitive information.

CVE-2018-1000180

Bouncy Castle BC 1.54 - 1.59, BC-FJA 1.0.0, BC-FJA 1.0.1 and earlier have a flaw in the Low-level interface to RSA key pair generator, specifically RSA Key Pairs generated in low-level API with added certainty may have less M-R tests than expected. This issue has been fixed in BC 1.60 beta 4 and subsequent versions, and in BC-FJA 1.0.2 and subsequent versions.

CVE-2018-1000613

Legion of the Bouncy Castle Legion of the Bouncy Castle Java Cryptography APIs 1.58 up to but not including 1.60 contains a CWE-470: Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') vulnerability in XMSS/XMSS^MT private key deserialization that can result in Deserializing an XMSS/XMSS^MT private key can result in the execution of unexpected code. This attack appears to be exploitable via A handcrafted private key can include references to unexpected classes which will be picked up from the class path for the executing application. This vulnerability appears to have been fixed in 1.60 and later.

CVE-2018-10936

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

CVE-2018-1295

In Apache Ignite 2.3 or earlier, the serialization mechanism does not have a list of classes allowed for serialization/deserialization, which makes it possible to run arbitrary code when 3-rd party vulnerable classes are present in Ignite classpath. The vulnerability can be exploited if the one sends a specially prepared form of a serialized object to one of the deserialization endpoints of some Ignite components - discovery SPI, Ignite persistence, Memcached endpoint, socket steamer.

CVE-2018-8018

In Apache Ignite before 2.4.8 and 2.5.x before 2.5.3, the serialization mechanism does not have a list of classes allowed for serialization/deserialization, which makes it possible to run arbitrary code when 3-rd party vulnerable classes are present in Ignite classpath. The vulnerability can be exploited if the one sends a specially prepared form of a serialized object to GridClientJdkMarshaller deserialization endpoint.

CVE-2020-13692

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

CVE-2021-37714: Crafted input may cause the jsoup HTML and XML parser to get stuck, timeout, or throw unchecked exceptions

jsoup is a Java library for working with HTML. Those using jsoup versions prior to 1.14.2 to parse untrusted HTML or XML may be vulnerable to DOS attacks. If the parser is run on user supplied input, an attacker may supply content that causes the parser to get stuck (loop indefinitely until canceled), to complete more slowly than usual, or to throw an unexpected exception. This effect may support a denial of service attack. The issue is patched in version 1.14.2. There are a few available workarounds. Users may rate limit input parsing, limit the size of inputs based on system resources, and/or implement thread watchdogs to cap and timeout parse runtimes.

CVE-2022-21724: Unchecked Class Instantiation when providing Plugin Classes

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

CVE-2022-31197: SQL Injection in ResultSet.refreshRow() with malicious column names in pgjdbc

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

CVE-2022-36944

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

CVE-2023-31582

jose4j before v0.9.3 allows attackers to set a low iteration count of 1000 or less.

CVE-2023-43642: Missing upper bound check on chunk length in snappy-java

snappy-java is a Java port of the snappy, a fast C++ compresser/decompresser developed by Google. The SnappyInputStream was found to be vulnerable to Denial of Service (DoS) attacks when decompressing data with a too large chunk size. Due to missing upper bound check on chunk length, an unrecoverable fatal error can occur. All versions of snappy-java including the latest released version 1.1.10.3 are vulnerable to this issue. A fix has been introduced in commit `9f8c3cf74` which is included in the 1.1.10.4 release. Users are advised to upgrade. Users unable to upgrade should only accept compressed data from trusted sources.

CVE-2023-45860

In Hazelcast Platform through 5.3.4, a security issue exists within the SQL mapping for the CSV File Source connector. This issue arises from inadequate permission checking, which could enable unauthorized clients to access data from files stored on a member's filesystem.

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

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

CVE-2023-48795

The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

CVE-2023-5072: DoS Vulnerability in JSON-Java

Denial of Service in JSON-Java versions up to and including 20230618. A bug in the parser means that an input string of modest size can lead to indefinite amounts of memory being used.

CVE-2023-51074

json-path v2.8.0 was discovered to contain a stack overflow using the Criteria.parse() method.

CVE-2024-20932

Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Security). Supported versions that are affected are Oracle Java SE: 17.0.9; Oracle GraalVM for JDK: 17.0.9; Oracle GraalVM Enterprise Edition: 21.3.8 and 22.3.4. Easily exploitable vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition. Successful attacks of this vulnerability can result in unauthorized creation, deletion or modification access to critical data or all Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition accessible data. Note: This vulnerability applies to Java deployments, typically in clients running sandboxed Java Web Start applications or sandboxed Java applets, that load and run untrusted code (for example, code that comes from the internet) and rely on the Java sandbox for security. This vulnerability does not apply to Java deployments, typically in servers, that load and run only trusted code (for example, code installed by an administrator). CVSS 3.1 Base Score 7.5 (Integrity impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N).

CVE-2024-1597: pgjdbc SQL Injection via line comment generation

pgjdbc, the PostgreSQL JDBC Driver, allows an attacker to inject SQL if using PreferQueryMode=SIMPLE. Note this is not the default. In the default mode there is no vulnerability. A placeholder for a numeric value must be immediately preceded by a minus. There must be a second placeholder for a string value after the first placeholder; both must be on the same line. By constructing a matching string payload, the attacker can inject SQL to alter the query,bypassing the protections that parameterized queries bring against SQL Injection attacks. Versions before 42.7.2, 42.6.1, 42.5.5, 42.4.4, 42.3.9, and 42.2.8 are affected.

CVEs fixed in CFM 2.1.6

In addition to the CVEs mentioned in CFM 2.1.5, the below CVEs are fixed in CFM 2.1.6:

CVE-2023-34212: Potential Deserialization of Untrusted Data with JNDI in JMS Components

The JndiJmsConnectionFactoryProvider Controller Service along with the ConsumeJMS and PublishJMS Processors, in Apache NiFi 1.8.0 through 1.21.0 allow an authenticated and authorized user to configure URL and library properties that enable deserialization of untrusted data from a remote location. The resolution validates the JNDI URL and restricts locations to a set of allowed schemes.

CVE-2023-34468: Potential Code Injection with Database Services using H2

The DBCPConnectionPool and HikariCPConnectionPool Controller Services in Apache NiFi 0.0.2 through 1.21.0 allow an authenticated and authorized user to configure a Database URL with the H2 driver that enables custom code execution. The resolution validates the Database URL and rejects H2 JDBC locations.

CVE-2023-36542: Potential Code Injection with Properties Referencing Remote Resources

Apache NiFi 0.0.2 through 1.22.0 include Processors and Controller Services that support HTTP URL references for retrieving drivers, which allows an authenticated and authorized user to configure a location that enables custom code execution. The resolution introduces a new Required Permission for referencing remote resources, restricting configuration of these components to privileged users. The permission prevents unprivileged users from configuring Processors and Controller Services annotated with the new Reference Remote Resources restriction.

CVE-2023-40037: Incomplete Validation of JDBC and JNDI Connection URLs

Apache NiFi 1.21.0 through 1.23.0 support JDBC and JNDI JMS access in several Processors and Controller Services with connection URL validation that does not provide sufficient protection

against crafted inputs. An authenticated and authorized user can bypass connection URL validation using custom input formatting. The resolution enhances connection URL validation and introduces validation for additional related properties.

Related Information

Support matrix