Cloudera Flow Management 2.1.3

Cloudera Flow Management Release Notes

Date published: 2019-06-26
Date modified: 2022-02-07

https://docs.cloudera.com/

Legal Notice

Contents

What's new in this release?

You must be aware of the additional functionalities and improvements to features of components in Cloudera Flow Management (CFM) 2.1.3. Learn how the new features and improvements benefit you.

Important:
log4j vulnerability updates

This release removes any transitive dependencies to the log4j libraries. NiFi relies on SLF4J for logging with Logback as the runtime implementation.

Important:
Java 8/JDK Considerations

When using Java 8, only JDK 8, version u252 (8u252) and above are supported. This gives access to strong cryptographic algorithms.

CFM 2.1.3 is based on the latest version of Apache NiFi and includes significant improvements. Here are some important new features:

Improved NiFi UI responsiveness

The experience when using NiFi (UI) with flows containing thousands or tens of thousands of components is improved. (NIFI-9309)

Parameter Contexts Inheritance

Provides the ability to leverage the Parameter Contexts in a more flexible way in multi-tenant environments. See *Parameter Contexts Inheritance* for more information. (NIFI-8490)

Improved Hadoop-related processors class loading

Improves how class loading isolation works around the `UserGroupInformation` class for Hadoop-related components. This improves startup time for NiFi when using thousands of these components. (NIFI-9382)

ExecuteStateless processor

The option to execute flows in NiFi while using the Stateless implementation. Improves performance for specific flows as well as providing exactly-once operations in some cases. See *Related Links* for more information about this feature. (NIFI-9239)

Components Validation

Provides a way to validate and get insights about a component's configuration in NiFi's (UI) configuration view. (NIFI-9009)

Related Information

SAML Authentication
NiFi Restricted Components
FIPS 140-2 Compliance
Status History Repository
CFM upgrade and migration paths

Component support

Lists the official component versions for Cloudera Flow Management. To know the component versions for compatibility with other applications, you must be familiar with the latest component versions in CFM.

Note:

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

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 version 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

For more information, see the *CFM 2.0.4 Release Notes*.

CFM 2.0.1

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

For more information, see the *CFM 2.0.1 Release Notes*.

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.

Technical Preview Features

The following features are available within CFM 2.1.3 but are not ready for production deployment. Cloudera encourages you to explore these technical preview features in non-production environments and provide feedback on your experiences through the Cloudera Community Forums.

- The following rules engine and handlers controller services:
  - EasyRulesEngineService
  - EasyRulesEngineProvider
  - ScriptedRulesEngine
  - ActionHandlerLookup
  - AlertHandler
  - ExpressionHandler
  - LogHandler
  - RecordSinkHandler
  - ScriptedActionHandler

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.

Behavioral Changes

Learn about the change in behavior in this version of CFM 2.1.3.

There are no behavior changes introduced in this release.

Related Information

Apache NiFi Migration Guidance

NiFi Restricted Components

Known Issues

Summarizes known issues for this release.

Special characters in Keystore/Truststore passwords

If there are special characters in the passwords of the truststores/keystores, the normal operation of NiFi and its integration with Cloudera Manager (command and control, monitoring, etc) is affected.

Update the passwords using only [A-Z a-z 0-9] characters or upgrade to CFM 2.1.5. You can also file a support case to get a hotfix from Cloudera Support.

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.

KafkaRecordSink puts multiple records in one message

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

For more information, see NIFI-8326.

There is no workaround for this issue.

Kudu Client

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

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.

KeyStoreException: placeholder not found

After an upgrade, NiFi may fail to start, displaying 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.

Technical Service Bulletins

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

If the content repository disk is filled more than 50% (or any other value that is set in nifi.properties for nifi.content.repository.archive.max.usage.percentage), and if there is no data in the content repository archive, the following warning message can be found in the logs: "Unable to write flowfile content to content repository container default due to archive file size constraints; waiting for archive cleanup". This would block the processors and no more data is processed.

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

Upstream JIRA

- NIFI-10023
- NIFI-9993

Knowledge article

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

TSB 2022-589: CVE-2022-33140 Apache NiFi ShellUserGroupProvider Vulnerability

The optional ShellUserGroupProvider in Apache NiFi 1.10.0 to 1.16.2 and Apache NiFi Registry 0.6.0 to 1.16.2 does not neutralize arguments for group resolution commands, allowing injection of operating system commands on Linux and macOS platforms. The ShellUserGroupProvider is not included in the default configuration. Command injection requires ShellUserGroupProvider to be one of the enabled User Group Providers (UGP) in the Authorizers configuration. Command injection also requires an authenticated user with elevated privileges. Apache NiFi requires an authenticated user with authorization to modify access policies in order to execute the command. Apache NiFi Registry requires an authenticated user with authorization to read user groups in order to execute the command. The resolution removes command formatting based on user-provided arguments.

Knowledge article

For the latest update on this issue see the corresponding Knowledge article: TSB 2022-589: CVE-2022-33140 Apache NiFi ShellUserGroupProvider Vulnerability

Fixed Issues

Review the list of Cloudera Flow Management issues that are resolved in CFM 2.1.3.

CFM 2.1.3 is based on Apache NiFi 1.15.2.2.1.3.0 and includes additional commits:

- NIFI-830: Added FlowFile Naming Strategy to InvokeHTTP
- NIFI-3328: SendTrapSNMP and ListenTrapSNMP processors added
- NIFI-8806: Refactored ListenTCP using Netty
- NIFI-9205: Update prioritizer configuration
- NIFI-9291: Added NiFi HTTP request logging
- NIFI-9304: Added Google Cloud Pub/Sub Lite processors
- NIFI-9308: Added EmailRecordSink

- NIFI-9334: Add support for upsert in 'PutMongoRecord'. Use 'bulkWrite' for both insert and upsert
- NIFI-9338: Add Azure Blob processors using Azure Blob Storage client library v12 for Java
- NIFI-9355: Upgraded Apache Curator from 4.2.0 to 5.2.0
- NIFI-9371: Removed synchronized keyword from Active Threads methods
- NIFI-9379: Add dependent properties and resource definitions to manifest model
- NIFI-9382: Improve startup time when loading flow that uses many HDFS related processors
- NIFI-9384: Corrected usage and generics in ListenTCP
- NIFI-9385: Add Flow Metrics producer for Cloudera Manager
- NIFI-9394: Removed RequestLogger and TimerFilter
- NIFI-9524: exclude commons-logging and log4j-core banned dependencies for other build profiles as well
- NIFI-9534: Upgraded Log4j 2 BOM from 2.17.0 to 2.17.1
- NIFI-9552: Make sure cl-over-slf4j is included under ext/ranger/install/lib directory

Common Vulnerabilities and Exposures

Lists common vulnerabilities and exposures fixed in CFM 2.1.3.

- CVE-2021-42550: In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configuration files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.
- CVE-2021-44832: Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2.
- CVE-2021-45105: Apache Log4j2 versions 2.0-alpha1 through 2.16.0 (excluding 2.12.3 and 2.3.1) did not protect from uncontrolled recursion from self-referential lookups. This allows an attacker with control over Thread Context Map data to cause a denial of service when a crafted string is interpreted. This issue was fixed in Log4j 2.17.0, 2.12.3, and 2.3.1.
- CVE-2021-45046: It was found that the fix to address CVE-2021-44228 in Apache Log4j 2.15.0 was incomplete in certain non-default configurations. This could allows attackers with control over Thread Context Map (MDC) input data when the logging configuration uses a non-default Pattern Layout with either a Context Lookup (for example, $${ctx:loginId}) or a Thread Context Map pattern (%X, %mdc, or %MDC) to craft malicious input data using a JNDI Lookup pattern resulting in an information leak and remote code execution in some environments and local code execution in all environments. Log4j 2.16.0 (Java 8) and 2.12.2 (Java 7) fix this issue by removing support for message lookup patterns and disabling JNDI functionality by default.
- CVE-2021-44228: Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.
- CVE-2021-34429: For Eclipse Jetty versions 9.4.37-9.4.42, 10.0.1-10.0.5 & 11.0.1-11.0.5, URIs can be crafted using some encoded characters to access the content of the WEB-INF directory and/or bypass some security constraints. This is a variation of the vulnerability reported in CVE-2021-28164/GHSA-v7ff-8wcx-gmc5.
- CVE-2021-44145: In the TransformXML processor of Apache NiFi before 1.15.1 an authenticated user could configure an XSLT file which, if it included malicious external entity calls, may reveal sensitive information.

Important:

NiFi contains the vulnerable log4j2 libraries and should be updated. For information about the available hotfix, contact your Cloudera Support representative.

Cloudera is not aware of known exploitation paths within CFM for Private Cloud Base, however upgrading protects against the possibility of any exploitation.

Download from the CFM Repository

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 support portal.

You can get download credentials in the following ways:

- Contact your Cloudera sales representative.
- View the Welcome email for your Flow Management account.
- File a non-technical case within the Cloudera support portal for our Support team to assist you.

Table 1: RHEL/CentOS 7

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/parcel/CFM-2.1.3.0-125-el7.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/parcel/CFM-2.1.3.0-125-el7.parcel.sha

Table 2: RHEL/CentOS 8

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat8/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat8/yum/tars/parcel/CFM-2.1.3.0-125-el8.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat8/yum/tars/parcel/CFM-2.1.3.0-125-el8.parcel.sha

Table 3: SLES 12

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.3.0/sles12/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.3.0/sles12/yum/tars/parcel/CFM-2.1.3.0-125-sles12.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.3.0/sles12/yum/tars/parcel/CFM-2.1.3.0-125-sles12.parcel.sha

Table 4: Ubuntu 18

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.3.0/ubuntu18/apt/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.3.0/ubuntu18/apt/tars/parcel/CFM-2.1.3.0-125-bionic.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.3.0/ubuntu18/apt/tars/parcel/CFM-2.1.3.0-125-bionic.parcel.sha

Table 5: Ubuntu 20

File	Location
Manifest	https://archive.cloudera.com/p/cfm2/2.1.3.0/ubuntu20/apt/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm2/2.1.3.0/ubuntu20/apt/tars/parcel/CFM-2.1.3.0-125-focal.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm2/2.1.3.0/ubuntu20/apt/tars/parcel/CFM-2.1.3.0-125-focal.parcel.sha

Table 6: CSD files (OS agnostic)

File	Location
NiFi	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/parcel/NIFI-1.15.2.2.1.3.0-125.jar
NiFi Registry	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/parcel/NIFIREGISTRY-1.15.2.2.1.3.0-125.jar

Table 7: Standalone components (OS agnostic)

File	Location
NiFi (.tar.gz)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-1.15.2.2.1.3.0-125-bin.tar.gz
NiFi (.zip)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-1.15.2.2.1.3.0-125-bin.zip
NiFi (.zip.sha256)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-1.15.2.2.1.3.0-125-bin.zip.sha256
NiFi Registry (.tar.gz)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-registry-1.15.2.2.1.3.0-125-bin.tar.gz
NiFi Toolkit (.tar.gz)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-toolkit-1.15.2.2.1.3.0-125-bin.tar.gz
NiFi Toolkit (.zip)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-toolkit-1.15.2.2.1.3.0-125-bin.zip
NiFi Toolkit (.zip.sha256)	https://archive.cloudera.com/p/cfm2/2.1.3.0/redhat7/yum/tars/nifi/nifi-toolkit-1.15.2.2.1.3.0-125-bin.zip.sha256

Table 8: Windows files

File	Location
NiFi MSI	https://archive.cloudera.com/p/cfm2/2.1.3.0/windows/nifi-2.1.3.0-125.msi
NiFi MSI sha file	https://archive.cloudera.com/p/cfm2/2.1.3.0/windows/nifi-2.1.3.0-125.msi.sha