Cloudera Flow Management - Kubernetes Operator 2.9.0

CFM Operator Release Notes

Date published: 2024-06-11
Date modified: 2024-10-31

https://docs.cloudera.com/

Legal Notice

Contents

What's new

Learn about the new features and notable changes throughout releases of Cloudera Flow Management - Kubernetes Operator.

Release 2.9.1

Learn about the new features and notable changes in release 2.9.1 of Cloudera Flow Management - Kubernetes Operator.

A Cloudera Flow Management - Kubernetes Operator 2.9.1 bundle for RedHat OpenShift OperatorHub is released. This is not a functional release, deployed images are still at 2.9.0-b96.

Fixed issues

- Cloudera Flow Management - Kubernetes Operator running out of memory when deploying NiFis
- Missing role permissions

Release 2.9.0

Learn about the new features and notable changes in release 2.9.0 of Cloudera Flow Management - Kubernetes Operator.

Improvements

- Cluster domains other than the default 'cluster.local' are now supported.
- Kubernetes replaced ZooKeeper as the default state management and leader election option.
- JVM memory settings are now calculated based on Pod memory.
- A NiFi CR config for Single User Authentication is now available.
- Pod and Node affinity are now configurable.
- The cfmctl CLI utility lists resources that block uninstallation of a cluster.

Fixed issues

- Node Cert alt names for proper SNI resolution
- NiFi Registry StatefulSet not updated on spec change
- OIDC did not use NiFi truststore
- CFM Operator continually overwriting default sensitive properties key
- Incorrect port configuration for non secure NiFi

Release 2.8.0

Learn about the new features and notable changes in release 2.8.0 of Cloudera Flow Management - Kubernetes Operator.

Cloudera Flow Management - Kubernetes Operator 2.8.0 is the first release of the CFM Kubernetes operator, which provides a way to deploy, manage, and operate NiFi clusters on Kubernetes application platforms. This release comes with container images based on Apache NiFi 1.25 and Apache NiFi 2.0 (milestone release). To learn more about the Cloudera Flow Management - Kubernetes Operator and its typical deployment architecture, see the Cloudera Flow Management - Kubernetes Operator Overview. To get started with installing the operator, see Installation overview.

Apache Parquet CVE-2025-30065

A critical vulnerability (CVE-2025-30065) in Apache Parquet's parquet-avro module allows arbitrary code execution through schema manipulation and crafted files. Cloudera advises upgrading to supported versions with fixes once they become available and implementing mitigations in the meantime.

Background:

On April 1, 2025, a critical vulnerability in the parquet-avro module of Apache Parquet (CVE-2025-30065, CVSS score 10.0) was announced.

Cloudera has determined the list of affected products, and is issuing this TSB to provide details of remediation for affected versions.

Upgraded versions are being released for all currently affected supported releases of Cloudera products. Customers using older versions are advised to upgrade to a supported release that has the remediation, once it becomes available.

Vulnerability Details:

Exploiting this vulnerability is only possible by modifying the accepted schema used for translating Parquet files and subsequently submitting a specifically crafted malicious file.

CVE-2025-30065 | Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code.

CVE:

NVD - CVE-2025-30065

Severity (Critical):

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

Impact:

Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code. Attackers may be able to modify unexpected objects or data that was assumed to be safe from modification. Deserialized data or code could be modified without using the provided accessor functions, or unexpected functions could be invoked.

Deserialization vulnerabilities most commonly lead to undefined behavior, such as memory modification or remote code execution.

Mitigation:

Until Cloudera has released a product version with the Apache Parquet vulnerability fix, please continue to use the mitigations listed below:

Customers with their own FIM Solution:

1. Utilize a File Integrity Monitoring (FIM) solution. This allows administrators to monitor files at the filesystem level and receive alerts on any unexpected or suspicious activity in the schema configuration.

General advisory:

1. Use network segmentation and traffic monitoring with a device capable of deep packet inspection, such as a network firewall or web application firewall, to inspect all traffic sent to the affected endpoints.

2. Configure alerts for any suspicious or unexpected activity. You may also configure sample analysis parameters to include:
   - Parquet file format “magic bytes” = PAR1
   - Connections from sending hosts that are not expected source IP ranges.
3. Be cautious with Parquet files from unknown or untrusted sources. If possible, do not process files with uncertain origins or that can be ingested from outside the organization.
4. Ensure that only authorized users have access to endpoints that ingest Parquet files.

For the latest updates on this issue, see the corresponding Knowledge article.

Kubernetes Ingress NGINX Controller vulnerabilities

Five vulnerabilities affecting the Ingress Nginx Controller for Kubernetes were publicly disclosed on March 24, 2025, and were given the nickname 'IngressNightmare'.

The ‘IngressNightmare’ vulnerabilities may allow Remote Code Execution (RCE) and potentially expose Kubernetes clusters to malicious configuration modifications. Exploitation requires specially crafted HTTP requests that bypass security measures, such as a Web Application Firewall (WAF). Successful exploitation may lead to complete cluster compromise, data exfiltration, and denial of service.

Details of the CVEs:

- CVE-2025-1974 (CVSS score: 9.8) – An unauthenticated attacker with access to the pod network can achieve arbitrary code execution in the context of the ingress-nginx controller under certain conditions
- CVE-2025-24514 (CVSS score: 8.8) – The auth-url Ingress annotation can be used to inject configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller
- CVE-2025-1097 (CVSS score: 8.8) – The auth-tls-match-cn Ingress annotation can be used to inject configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller
- CVE-2025-1098 (CVSS score: 8.8) – The mirror-target and mirror-host Ingress annotations can be used to inject arbitrary configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller
- CVE-2025-24513 (CVSS score: 4.8) – An improper input validation vulnerability that could result in directory traversal within the container, leading to denial-of-service (DoS) or limited disclosure of secret objects from the cluster when combined with other vulnerabilities

How do these vulnerabilities affect on cloud?

For mitigating CVE-2025-1974 on on cloud, refer to the information below.

Note: recommends limiting direct access to cluster hosts to only authorized administrators and auditing all activity as a security best practice.

Mitigation of CVE-2025-24514, CVE-2025-1097, CVE-2025-1098, and CVE-2025-24513 is secondary to the previous CVE. They require no immediate action, as attackers can only exploit these with direct access to cluster hosts and privileges to create arbitrary ingress objects via the Kubernetes API.

Note: Cloudera has tested these mitigation steps only on the currently supported releases. Customers using older versions are advised to upgrade to a supported release before attempting the mitigation actions.

For the latest updates on this issue, see the corresponding Knowledge articles:

- TSB 2025-839: Critical Kubernetes Ingress NGINX Controller Vulnerability Allows RCE Without Authentication
- TSB 2025-839 Mitigation steps for Cloudera DataFlow on cloud

Instructions

1. Check your current version

To determine which version your cluster is running, use either the UI or the CLI:

DataFlow Service version discovery from the UI:

1. In Cloudera DataFlow, from the **Environments** page, select the environment you want to inspect.
2. Click Manage DataFlow from the **Environment Details** pane.
3. Look for “DATAFLOW VERSION”.

DataFlow Service version discovery from the CLI:

1. Run the cdp df list-services command to display a list of the environments with their workload versions included.
2. Run cdp df describe-service --service-crn <DataFlow CRN> to display more detailed information about an environment and display the DataFlow version.

Note: For instructions on how to install and use the Cloudera CLI, refer to the CLI documentation.

2. Mitigation steps for Cloudera DataFlow on Cloud

Upgrade the “ingress-nginx” controller component deployed by Cloudera DataFlow to version v1.11.5 for all environments running Cloudera DataFlow versions 2.9.0 and lower. To apply the patch to the environment:

1. Go to Cloudera DataFlow in the Cloudera Management Console.
2. Select the target environment for the patch from the “Environments” list.
3. Click “Manage Cloudera Data Flow service” in the upper right corner.
4. Click the “Actions” dropdown menu, and select “Download Kubeconfig”.
5. A KubeConfig yaml file will be downloaded to your local machine. (Named “kubeconfig.yaml” by default.)
6. Depending on your operating system, follow the instructions at https://kubernetes.io/docs/tasks/tools/#kubectl to install the “kubectl” binary to your local machine.
7. Create a text file called bump-controller-version.yaml, add the following text, and save the file.

```
# kubectl -n nginx-ingress patch helmreleases.helm.toolkit.fluxcd.io ing
ress-nginx --patch-file admission-webhook-off.yaml --type merge
spec:

  values:

    ingress-nginx:

      controller:

        image:

          tag: v1.11.5
```

8. Open a terminal or command prompt, and run (from the directory you saved the yaml files) the following command to test your access to the Kubernetes API server:

   kubectl --kubeconfig kubeconfig.yaml get nodes

   The command prints out information about the list of worker nodes.

9. Execute the following command to apply the patch:

   kubectl --kubeconfig kubeconfig.yaml -n nginx-ingress patch helmreleases ingress-nginx --patch-file bump-controller-version.yaml --type merge

   It prints out helmrelease.helm.toolkit.fluxcd.io/ingress-nginx patched on successful execution.
10. Execute the following command and verify that the printout in the command line ends with v1.11.5:

    kubectl --kubeconfig kubeconfig.yaml -n nginx-ingress get deployment dfx-nifi-nginx-ingress-controller -o jsonpath='{.spec.template.spec.containers[?(@.name=="controller")].image}'
11. Execute to watch the progress of the upgrade as the ingress-nginx controller pod is restarted in the background:

    kubectl --kubeconfig kubeconfig.yaml -n nginx-ingress get deployments dfx-nifi-nginx-ingress-controller -w
12. Wait until the columns show the following:
    a. “READY” column showing “1/1”,
    b. “UP-TO-DATE” column showing “1”,
    c. “AVAILABLE” column showing “1”
13. Press “Ctrl-C” to return to the command line.

Note: While executing the upgrade, for a short duration (usually a second or two)users may briefly lose connectivity to the workload side UI for deploying new flows or accessing the NiFi Web UI of existing flows, but there’s no interruption to data processing in any existing flows.

14. Verify that the new “ingress-nginx” controller is working properly by accessing the NiFi Web UI of any existing flow.
15. Repeat the process for each environment where the patch needs to be applied.

Important:

1. The mitigation steps apply a patch to the DataFlow environments. Before upgrading the patched environment of Cloudera DataFlow to another version, the patch must be reverted temporarily. To revert the patch, execute the following command:

   ```
   kubectl --kubeconfig kubeconfig.yaml -n nginx-ingress patch helmreleases ingress-nginx -p '{"spec":{"values":null}}' --type=merge
   ```
2. When creating a new deployment of Cloudera DataFlow version 2.9.0 and below, customers need to repeat the mitigation steps outlined above once the DataFlow environment is successfully created.

Known issues

Learn about the known issues in this release of Cloudera Flow Management - Kubernetes Operator

Apache Parquet CVE-2025-30065

A critical vulnerability (CVE-2025-30065) in Apache Parquet's parquet-avro module allows arbitrary code execution through schema manipulation and crafted files. Cloudera advises upgrading to supported versions with fixes once they become available and implementing mitigations in the meantime.

Until Cloudera has released a product version with the Apache Parquet vulnerability fix, please continue to use the mitigations listed below:

Customers with their own FIM Solution:

1. Utilize a File Integrity Monitoring (FIM) solution. This allows administrators to monitor files at the filesystem level and receive alerts on any unexpected or suspicious activity in the schema configuration.

General advisory:

1. Use network segmentation and traffic monitoring with a device capable of deep packet inspection, such as a network firewall or web application firewall, to inspect all traffic sent to the affected endpoints.
2. Configure alerts for any suspicious or unexpected activity. You may also configure sample analysis parameters to include:
   - Parquet file format “magic bytes” = PAR1
   - Connections from sending hosts that are not expected source IP ranges.
3. Be cautious with Parquet files from unknown or untrusted sources. If possible, do not process files with uncertain origins or that can be ingested from outside the organization.
4. Ensure that only authorized users have access to endpoints that ingest Parquet files.

For the latest updates on this issue, see the corresponding Knowledge article.

CDPDFX-10225: Cloudera Flow Management - Kubernetes Operator crashes once when creating a NiFi Registry (Standalone)

When first creating a NifiRegistry resource, the Cloudera Flow Management - Kubernetes Operator may crash once before recovering. No impact to functionality.

None.

Supported component versions

Cloudera Flow Management - Kubernetes Operator components and their versions delivered in this release of the product.

Table 1: Cloudera Flow Management - Kubernetes Operator component versions

Component	Version
Cloudera Flow Management - Kubernetes Operator and cfmctl	2.9.0
NiFi	1.27.0 / 2.0.0 [Technical Preview]
NiFi Registry	1.27.0

System requirements

To install and use Cloudera Flow Management - Kubernetes Operator and its components, your Kubernetes cluster environment must meet the following system requirements and prerequisites.

- Kubernetes cluster
  - Version 1.23 or later
  - OpenShift 4.10 or later

  Note: Cloudera Flow Management - Kubernetes Operator complies with Cloud Native Computing Foundation (CNCF) standards and is compatible with CNCF-compliant Kubernetes distributions. For supporting your specific Kubernetes distribution, contact Cloudera.
- Administrative rights on the Kubernetes cluster
- Access to kubectl or oc, configured to connect to your running cluster
- Access to helm
- cert-manager installed on the Kubernetes cluster
- Log collection enabled for the Kubernetes cluster
- Cloudera requires that the logs of Cloudera Flow Management - Kubernetes Operator components are stored long term for diagnostic and supportability purposes.

- Persistent storage class configured on the Kubernetes cluster that satisfies the durability and low-latency requirements for operating NiFi. The ideal storage class configuration can vary depending on the environment and use case, and it is determined by the Kubernetes platform where the product is deployed.
- (Optional): Prometheus installation running in the same Kubernetes cluster where you install Cloudera Flow Management - Kubernetes Operator. Prometheus is used for collecting and monitoring NiFi metrics.