

## Planning Your Flow Management cluster

Date published: 2019-06-26

Date modified: 2023-11-02



# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>Data flow design.....</b>	<b>4</b>
<b>Cluster layout.....</b>	<b>4</b>
<b>Disk configuration.....</b>	<b>5</b>
<b>Resource-intensive processors.....</b>	<b>5</b>
<b>Sizing recommendations.....</b>	<b>6</b>

## Data flow design

When planning how to size and configure Flow Management clusters, it is important to keep in mind the flow design factors that might impact your cluster sizing needs and the performance of your data flow.

**Note:**

Flow design is the most important factor influencing the expected data flow performance, as well as the type and amount of resources used by NiFi.

You can use NiFi for a wide array of use cases, and the resource requirements are greatly determined by data flow design. Depending on the actions it performs in a data flow, each processor may or may not be required to read or to write the processed data from or on disks.

For example, a flow ingesting 100 MB of data per second with its first processor, may need to read and write this data on disks multiple times before the result is sent to the final destination. If the data flow has four processors writing the content on disks before being sent to the final destination, the disks used for the content repositories in the NiFi cluster should be able to handle 400 MB per second at the cluster level.

### NiFi design principles

NiFi is designed to use all the available resources of the nodes where it is running. It takes advantage of:

- All available cores
- All network capacity
- All disk speed and capacities

**Note:**

It is crucial to understand that NiFi data flow behavior depends on three major factors:

- Data flow source
- Flow operations during data transfer
- Data flow target

For more information about principles involved in the NiFi design, see the *Apache NiFi Overview*.

## Cluster layout

When planning how to size and configure Flow Management clusters, it is important to keep in mind the flow design factors that might impact your cluster sizing needs and the performance of your data flow.

At least three nodes are recommended for a Flow Management cluster used in production environments. NiFi must be running on dedicated nodes.

Starting with CFM 2.1.3.0, security has been improved in NiFi and it changed the internal authentication token handling to use HttpOnly Session Cookies. Cookies are specific to a hostname and path, but the browser does not differentiate between different ports on the same host. In case multiple NiFi instances are running on the same node, but listening on different ports, it is required to configure an HTTPS load balancer and setup different context paths for each NiFi instance.



**Important:** Running multiple NiFi instances on the same node is not recommended by Cloudera.

When NiFi is clustered, you should use an external ZooKeeper cluster. In production deployments, the ZooKeeper instances should not be co-located with the NiFi instances.

## Disk configuration

When planning how to size and configure Flow Management clusters, it is important to keep in mind the flow design factors that might impact your cluster sizing needs and the performance of your data flow.

For most modern systems, the disk throughput is lower than the network throughput, so the network is usually not a bottleneck. For most data movement use cases, the CPU usage is much lower than the disk I/O, but it is still important to monitor the CPU and tune the number of threads per processor. See *Tuning your Data Flow* for recommendations about fine tuning threads usage.

NiFi has three repositories on disk and the disk configuration is a very important performance factor:

### Content Repository

- Contains the content of each FlowFile
- Sequential disk I/O (ideally leveraging the OS cache)

### FlowFile (metadata) Repository

- Contains the FlowFile attributes and current FlowFile state (which queue it is in) for each FlowFile
- Sequential and random disk I/O

### Provenance (metadata) Repository

- Contains a provenance log with entries for every action performed on a FlowFile (merge, drop, and so on)
- Sequential and random disk I/O

Every FlowFile that NiFi receives or creates is immediately written to disk in the content repository for fault tolerance. Subsequent FlowFile content modifications (decompression, format conversion, and so on) are also written to the content repository. Processors that do not modify the content, such as `RouteOnAttribute`, do not impact the content repository. Instead, the FlowFile repository keeps a pointer for each FlowFile showing its state, such as which queue it is located in. This optimization eliminates the need for redundant writes to the content repository.

For higher performance, configure multiple disks for both the content and provenance repositories.

For more information, see *Configuration Best Practices* and *File System Content Repository Properties* and *Write Ahead Provenance Repository Properties* in the NiFi System Properties documentation.

### Related Information

[File System Content Repository Properties](#)

[Write Ahead Provenance Repository Properties](#)

[Configuration Best Practices](#)

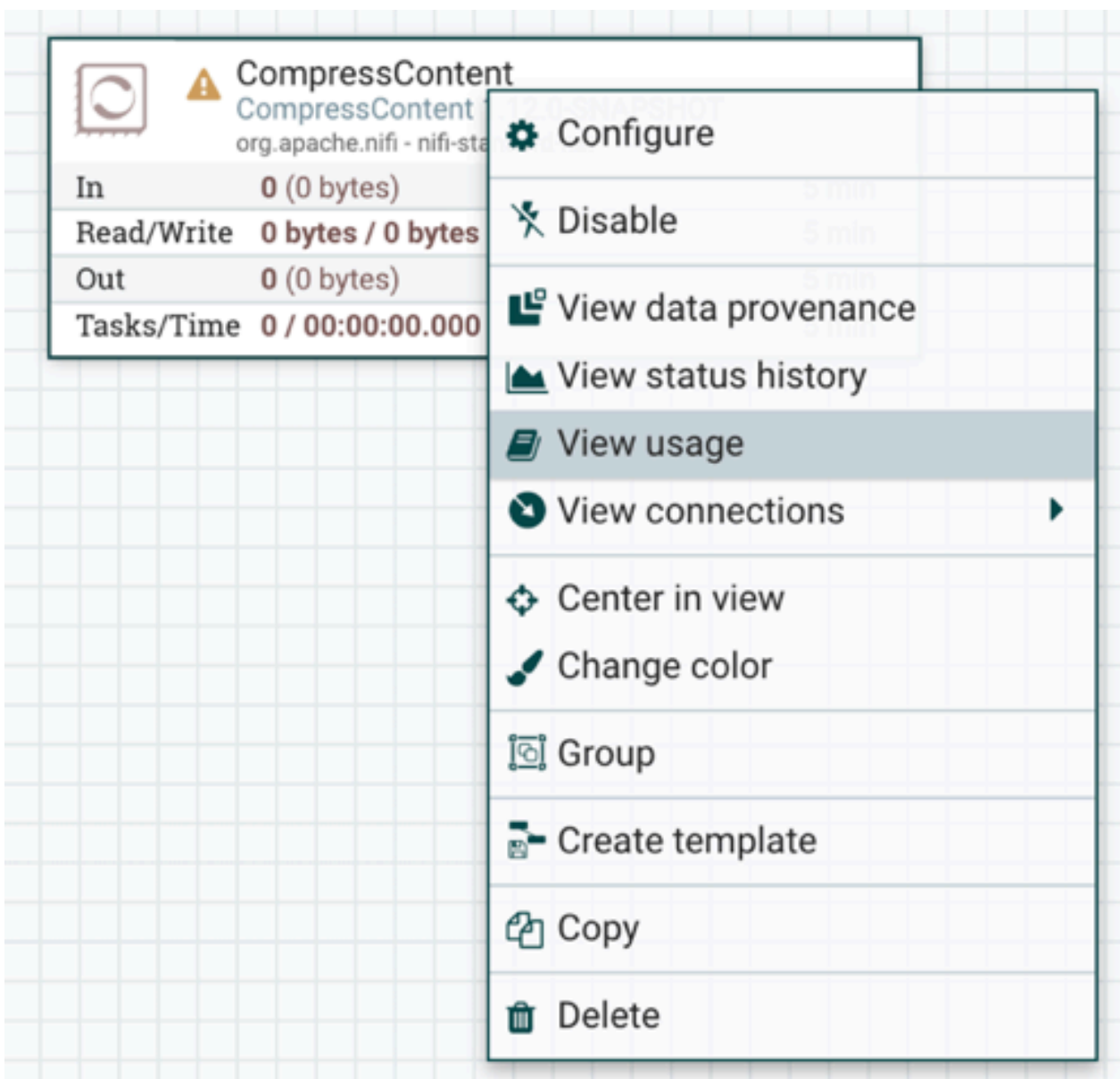
[Tuning your Data Flow](#)

## Resource-intensive processors

Learn how to identify resource-intensive processors.

You can check in the *Apache NiFi documentation* whether a processor is CPU intensive or memory intensive and use this information when planning your data flows.

For example, if you right-click the `CompressContent` processor to open its documentation, you can see that this processor can be both memory and CPU intensive.



The screenshot shows the 'CompressContent' component in the Cloudera Flow Management interface. The component's status is 'Warning' (indicated by a yellow triangle icon). The component's name is 'CompressContent' and its org is 'org.apache.nifi - nifi-sta'. The component's statistics are: In: 0 (0 bytes), Read/Write: 0 bytes / 0 bytes, Out: 0 (0 bytes), and Tasks/Time: 0 / 00:00:00.000. A context menu is open over the component, showing the following options: Configure, Disable, View data provenance, View status history, View usage (highlighted), View connections, Center in view, Change color, Group, Create template, Copy, and Delete.

**System Resource Considerations:**

Resource	Description
CPU	An instance of this component can cause high usage of this system resource. Multiple instances or high concurrency settings may result a degradation of performance.
MEMORY	An instance of this component can cause high usage of this system resource. Multiple instances or high concurrency settings may result a degradation of performance.

## Sizing recommendations

Learn how to configure your Flow Management cluster with sizing considerations in mind.

Cloudera recommends the following setup for on-premises, bare metal installations:

- 1 RAID 1 or 10 array for the OS
- 1 RAID 1 or 10 array for the FlowFile repository
- 1 or many RAID 1 or 10 array(s) for the content repository

- 1 or many RAID 1 or 10 array(s) for the provenance repository

For high performance setup, Cloudera recommends SSDs over spinning disks.

For cloud environments, larger disks usually provide better throughputs. Review your cloud provider documentation for more information.

In terms of memory, NiFi is optimized to support FlowFiles of any size. This is achieved by never materializing the file into memory directly. Instead, NiFi uses input and output streams to process events (there are a few exceptions with some specific processors). This means that NiFi does not require significant memory even if it is processing very large files. Most of the memory on the system should be left available for the OS cache. By having a large enough OS cache, many of the disk reads are skipped completely. Consequently, unless NiFi is used for very specific memory oriented data flows, setting the Java heap to 8 GB or 16 GB is usually sufficient.

The performance you can expect directly depends on the hardware and the flow design. For example, when reading compressed data from a cloud object store, decompressing the data, filtering it based on specific values, compressing the filtered data, and sending it to a cloud object store, you can achieve the following results:

Nodes	Data rate per second	Events per second	Data rate per day	Events per day
1	192.5 MB	946,000	16.6 TB	81.7 billion
5	881 MB	4.97 million	76 TB	429.4 billion
25	5.8 GB	26 million	501 TB	2.25 trillion
100	22 GB	90 million	1.9 PB	7.8 trillion
150	32.6 GB	141.3 million	2.75 PB	12.2 trillion

Data rates and event rates were captured running the flow described above on Google Kubernetes Engine. Each node has 32 cores, 15 GB RAM, and a 2 GB heap. The Content Repository is a 1 TB Persistent SSD (400 MB per second write, 1200 MB second read).

NiFi scales well, both vertically and horizontally. Depending on the number of data flows running in the NiFi cluster and your operational requirements, you can add nodes to the NiFi cluster over time to meet your needs.

With this information in mind, Cloudera recommends:

- 4 cores per NiFi node is the minimum number of cores required by Cloudera to be supported. Cloudera recommends 8 cores per NiFi node as it usually provides the best starting point for the most common use cases.
- At least 6 disks per NiFi node to ensure dedicated disks for repositories
- At least 4GB of RAM for the NiFi heap

Now that you have finished reviewing the Flow Management cluster sizing considerations, see *Processing one billion events per second with NiFi* for additional information and a use case walk through.

### Related Information

[Processing one billion events per second with Apache NiFi](#)