

Cloudera Edge Management 1.6.0

Using Cloudera Edge Management

Date published: 2019-04-15

Date modified: 2023-07-27

CLOUDERA

<https://docs.cloudera.com/>

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

Cloudera Edge Management user interface.....	5
Dashboard in CEM.....	5
Edge events in CEM.....	5
Design in CEM.....	6
Agent manager in CEM.....	8
Monitoring deployments in CEM.....	8
Monitoring events in CEM.....	14
Deploying agents in CEM [Technical Preview].....	17
Using the Agent Deployer.....	17
Configuring the Agent Deployer for securing agents.....	25
Managing agents in CEM.....	27
Edit table layout in CEM.....	29
Agent details in CEM.....	32
Debug agent in CEM.....	35
Expression language in CEM.....	37
Parameters in CEM.....	38
Adding parameters in CEM.....	38
Referencing parameters in CEM.....	40
Using parameters in CEM.....	41
Setting agent parameter values.....	48
Building a dataflow in CEM.....	49
Adding components to the canvas in CEM.....	49
Configuring a processor in CEM.....	51
Configuring a remote process group in CEM.....	57
Adding services in CEM.....	58
Connecting components in CEM.....	60
Configuring a connection in CEM.....	61
Bending connections in CEM.....	62
Refreshing a dataflow in CEM.....	63
Publishing a dataflow in CEM.....	63
Reverting a dataflow in CEM.....	64

Example dataflow in CEM.....	65
Monitoring metrics in CEM with Grafana.....	66

Cloudera Edge Management user interface

The Cloudera Edge Management (CEM) UI highlights the different segments of the UI. Learn how to use the CEM UI to monitor deployments and events. Also learn how to build dataflows, use parameters, publish dataflows, and revert dataflows.

The CEM UI provides mechanisms for building, editing and publishing dataflows, as well as monitoring C2 server and agent events.

When you start the application, you can navigate to the UI by going to the default address of `http://<hostname>:10090/efm/ui` in a web browser.



Note: Anyone who has access to the CEM UI can view events, and view or modify dataflows.

Dashboard in CEM

The Dashboard screen enables you to monitor C2 server and agent deployments.

When you navigate to the Cloudera Edge Management (CEM) UI for the first time, the Dashboard screen appears as shown in the following image:

Status	Class Name ↑	Number of Agents	Last Flow Updated	Updated Agents
Good Health	minifi-cpp-1.21.02.0-19	1	No flow has been published	>
Good Health	minifi-cpp-latest	3	No flow has been published	>
Good Health	minifi-java-1.2.2.2	1	No flow has been published	>
Good Health	minifi-java-latest	3	No flow has been published	>
Unknown Health	Unassigned	0	No flow has been published	>

Items per page: 10 1 - 5 of 5 |< < > >|

For more information on how to monitor deployments, see *Monitoring deployments in CEM*.

Related Information

[Monitoring deployments in CEM](#)

Edge events in CEM

The Edge Events screen enables you to monitor C2 server and agent events.

When you navigate to the Edge Events page in the Cloudera Edge Management (CEM) UI, the Edge Events screen appears as shown in the following image:

CEM

CLUSTER

Edge Management

Monitor

Edge Events

Design

Agent Manager

anonymous

1.4.0-SNAPSHOT

Edge Events

Filter

bySeverity

Date/Time	Severity	Event Type	Message	Class Name	Source Type	Event Source ID
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	4a955dd1-a687-44...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	5f43eba2-72c2-4ee...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	ffb5baaa-02d3-455...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-1.2.2.2	Agent	a3217be3-bee8-41...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39e114ee-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39d18a6a-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39741e70-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-1.21.02...	Agent	48694392-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	4a955dd1-a687-44...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	5f43eba2-72c2-4ee...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	ffb5baaa-02d3-455...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-1.2.2.2	Agent	a3217be3-bee8-41...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39e114ee-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39d18a6a-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39741e70-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-1.21.02...	Agent	48694392-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	4a955dd1-a687-44...

Time range: All Rows per page: 20 1 - 20 of 80

For more information on how to monitor events, see *Monitoring events in CEM*.

Related Information
[Monitoring events in CEM](#)

Design in CEM

Learn how to build and publish dataflows by using designer in the Cloudera Edge Management (CEM) UI.

To navigate to the designer section, click the Design icon ( Design). The Open Flow screen initially appears, as shown in the following image, and displays information about the classes in the system:

CEM

CLUSTER

Edge Management

Monitor

Edge Events

Design

Agent Manager

anonymous

1.4.0-SNAPSHOT

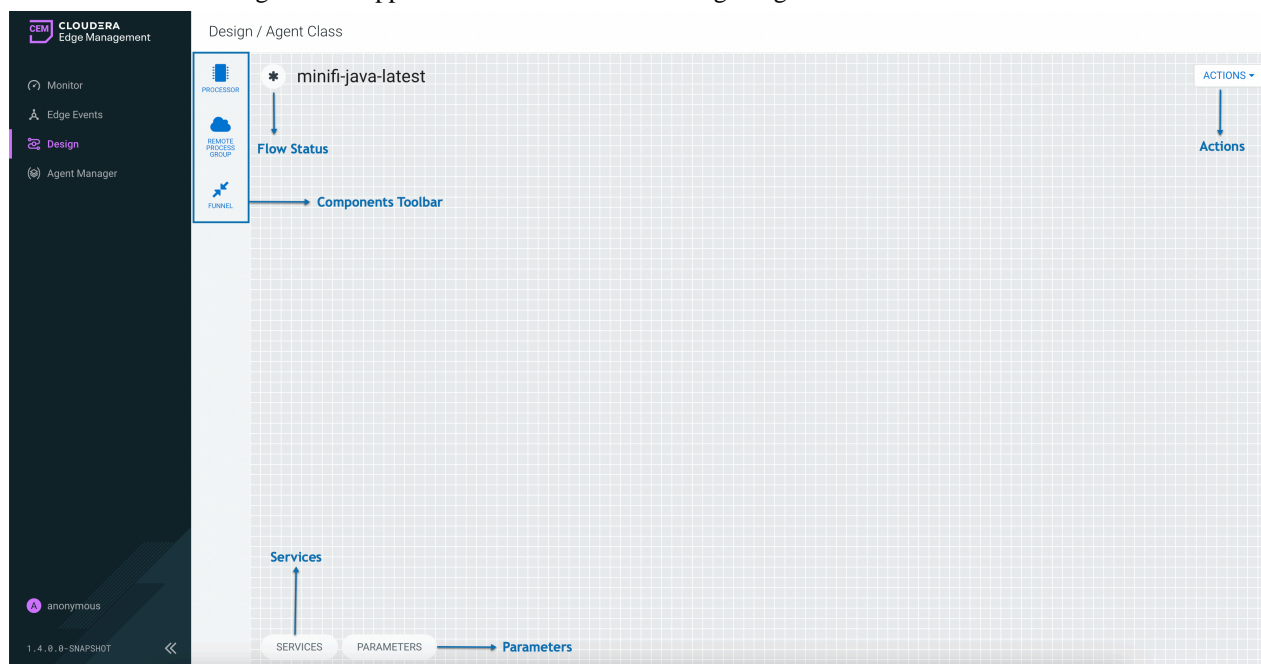
Design

Filter items

Status	Class	Published Version	Published On
*	minifi-cpp-1.21.02.0-19		
*	minifi-cpp-latest		
*	minifi-java-1.2.2.2		
*	minifi-java-latest		

Open

To build a dataflow, select the desired class from the table and click Open. Alternatively, you can double-click on the desired class. The Design screen appears as shown in the following image:



For more information on how to build a dataflow, see *Building a dataflow in CEM*. For more information on publishing a dataflow, see *Publishing a dataflow in CEM*.

Open Flow

The Open Flow screen appears initially when you navigate to the designer. It displays the following information about the dataflows and classes in the system:

- Status
- Class
- Published Version
- Published On

You can sort data in each column in ascending or descending order by clicking the column name. For example, you can sort the dataflows based on which class they are associated with by clicking the Class column name.

You can also filter the data by class by using the Filter items search bar.

Design

The Design screen contains the following segments, each responsible for different functionalities of the application:

- Components Toolbar. The components toolbar consists of the components that you can drag onto the canvas to build your dataflow.
- Services. SERVICES are shared services that can be used by processors and other services to utilize for configuration or task execution.
- Parameters. Provides the ability to parameterize the values of processor and service properties in the flow.
- Actions. Use the ACTIONS drop-down to open a different dataflow, publish a dataflow, or revert to the last published version of a dataflow.

- Flow Status. The Flow Status displays information about the flow including when the flow was published, what was the last published version of the flow, and the status of the flow.

•



Modified (). The flow has been modified since the flow has been published.

•



Current (). The current published version of the flow.



Note: Once you open a dataflow, it is the default flow that displays in the Design screen. To build or edit another dataflow, select Open from the ACTIONS menu.

Related Information

[Building a dataflow in CEM](#)

[Publishing a dataflow in CEM](#)

Agent manager in CEM

The Agent Manager screen enables you to monitor health of the agents and view, filter, and initiate commands on specific agents.

When you navigate to the Agent Manager page in the Cloudera Edge Management (CEM) UI, the Agent Manager screen appears as shown in the following image:

Status	Class Name ↑	Flow Version	Agent ID	Last Seen
✓	minifi-cpp-1.21.02.0-19		48694392-ba7a-11ec-89e7-de74b8d...	2022-04-13 05:56 CEST
✓	minifi-cpp-latest		39741e70-ba7a-11ec-b554-365471e...	2022-04-13 05:56 CEST
✓	minifi-cpp-latest		39d18a6a-ba7a-11ec-a1c4-be3556c7...	2022-04-13 05:56 CEST
✓	minifi-cpp-latest		39e114ee-ba7a-11ec-8ffc-2ed00cd6...	2022-04-13 05:56 CEST
✓	minifi-java-1.2.2.2		a3217be3-bee8-4187-b13e-4f1e50f...	2022-04-13 05:56 CEST
✓	minifi-java-latest		4a955dd1-a687-44a6-9371-102c50c...	2022-04-13 05:57 CEST
✓	minifi-java-latest		5f43eba2-72c2-4ee8-8d4d-a4dbaa44...	2022-04-13 05:57 CEST
✓	minifi-java-latest		ffb5aaa-02d3-4555-be61-3bc0dc0bf...	2022-04-13 05:57 CEST


For more information on how to monitor agents, see *Managing agents in CEM*.

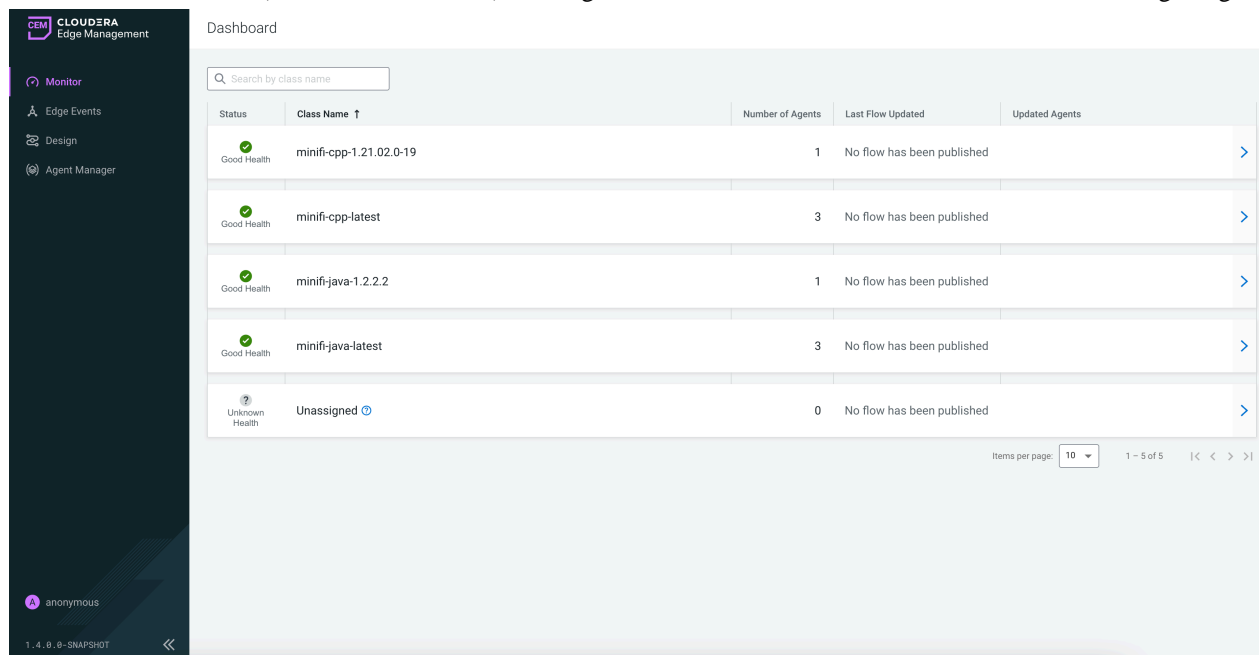
Related Information

[Managing agents in CEM](#)

Monitoring deployments in CEM

Learn the options, available in the Dashboard screen, that enable you to monitor C2 server and agent deployments. The Dashboard is the initial page in the Cloudera Edge Management (CEM) UI.

Click the Monitor icon ( Monitor) to navigate to the Dashboard screen as shown in the following image:



CEM provides the following details for deployments:

- **Status**
 - **Good Health:** Recent heartbeats are received from all of the agents in the agent class.
 - **Concerning Health:** Some of the agents did not send heartbeat for the time period defined in the `efm.monitor.maxHeartbeatInterval` property.
 - **Bad Health:** None of the agents sent heartbeats in the given class for the time period defined in the `efm.monitor.maxHeartbeatInterval` property.
 - **Unknown Health:** Agents for this class may have not yet registered or heartbeat intervals cannot be determined.
- **Class Name**
- **Number of Agents**

The count of assigned agents (Online + Missing) to the given agent class.
- **Last Flow Updated**

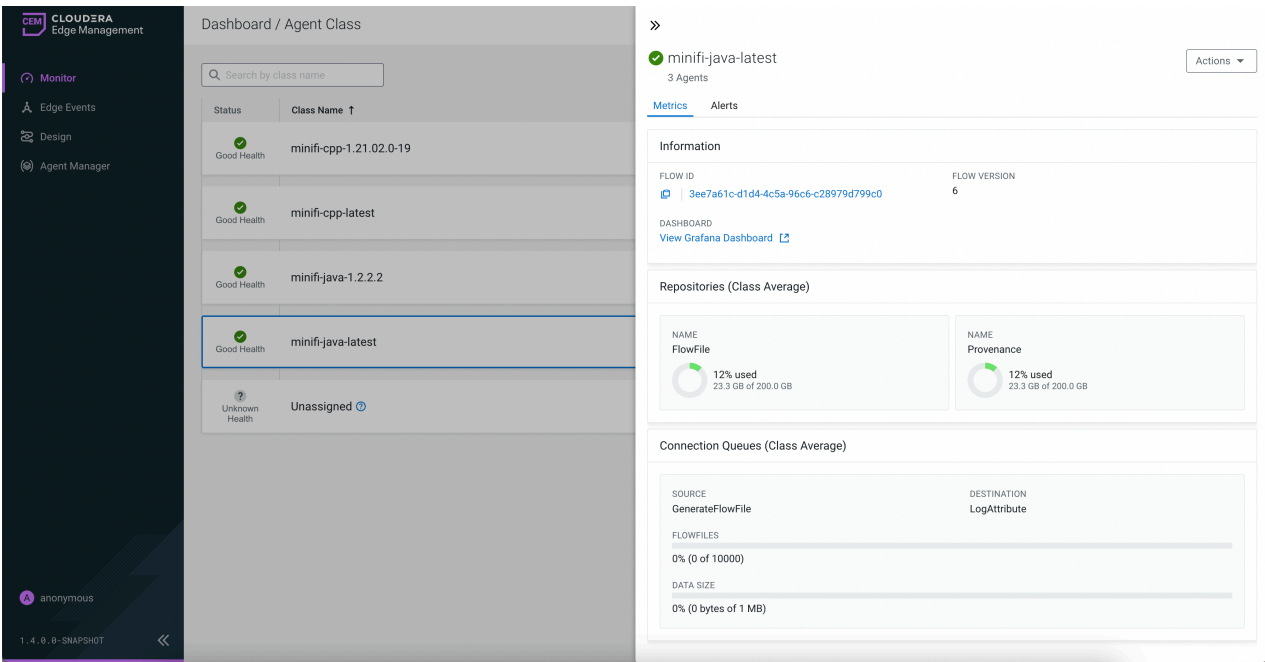
The time of the latest flow update of the given agent class.
- **Updated Agents**

Shows the current status of the last batched command, for example, flow publish, property update. While there are missing agents in a class, batched commands can not fully complete.
- **Arrow icon**

Click the arrow icon to fetch more details about the deployment. You can monitor metrics and alerts details for your deployment in two different tabs.

Monitoring class details in the Metrics tab

Click the arrow icon in the Dashboard page to fetch more details for your deployment. The Metrics tab appears as shown in the following image:



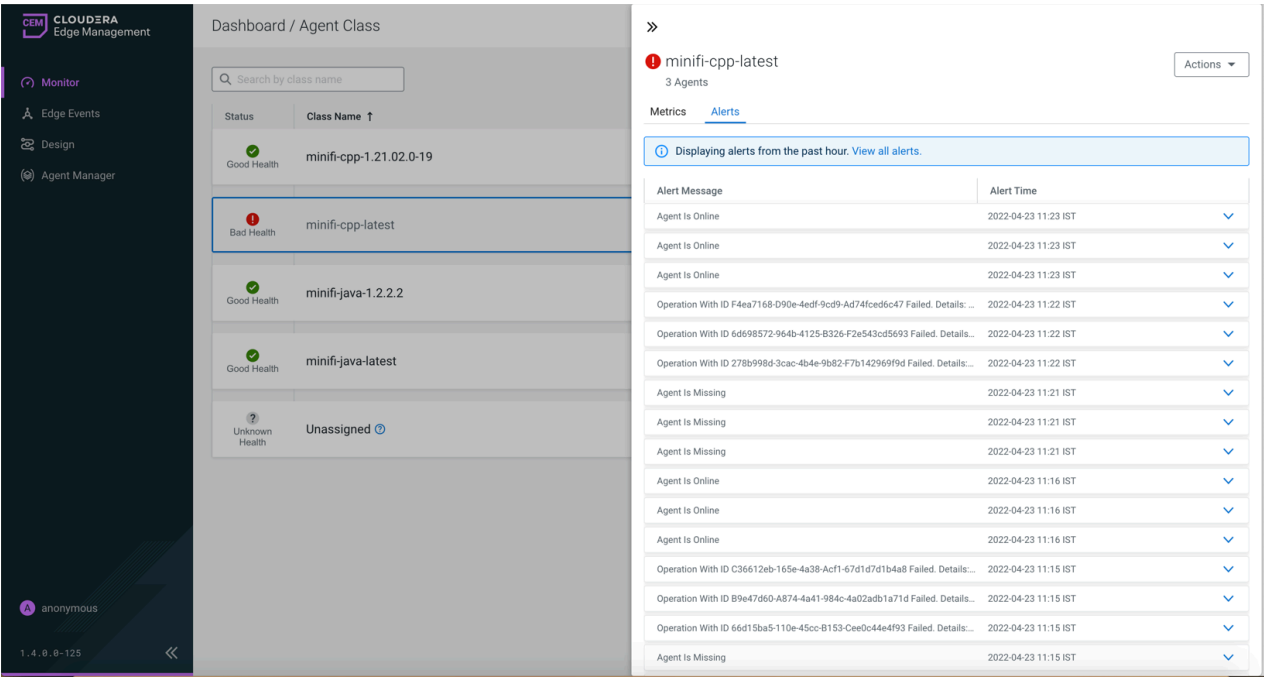
The Metrics tab displays the following details:

- Information
This section displays details about the class including flow ID, latest flow version, and Grafana dashboard link if you enabled it. For more information, see *Monitoring Metrics with Grafana*.
- Repositories (Class Average)
Usage details of the repositories of the agents in the agent class.
- Connection Queues (Class Average)
Aggregate views of all the connections throughout the agents in the selected class.

Monitoring alert details in the Alerts tab

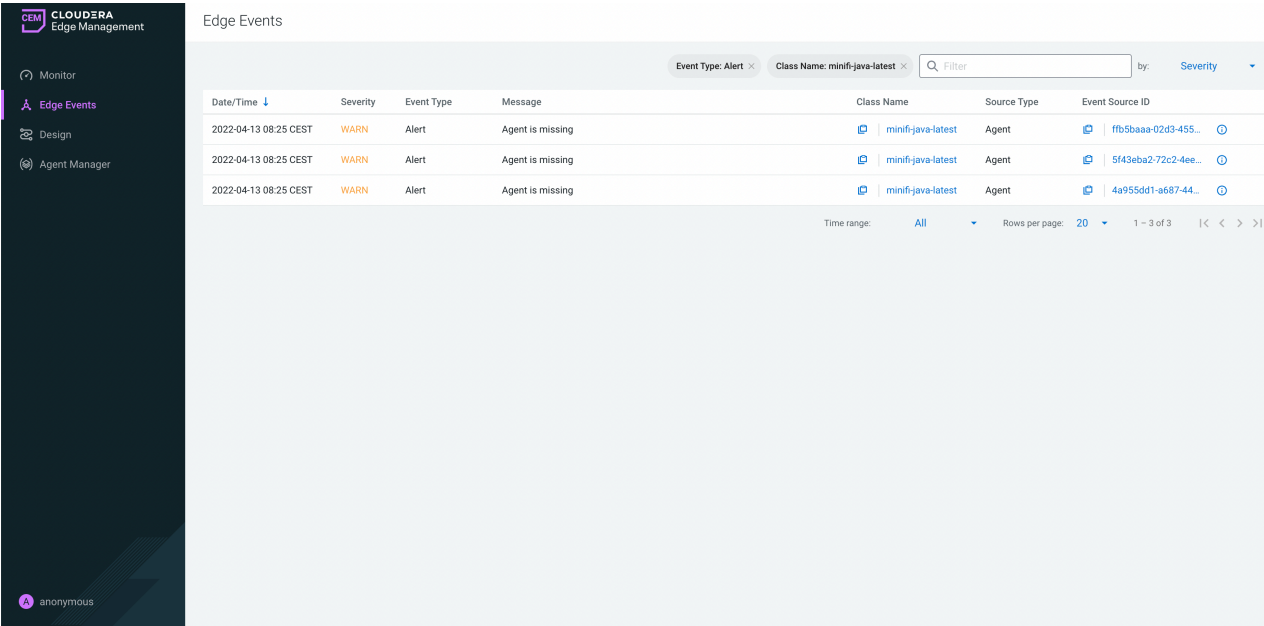
The Alerts tab displays the alert messages associated with the agent class.

Click the arrow icon in the Dashboard page, and then go to the Alerts tab to fetch alert messages along with the time they are raised at for your deployment, as shown in the following image:



You can scroll through the alerts to see more alerts in the system. Each alert has an arrow beside it, and you can click it to fetch more details about that alert.

Click the View all alerts link to navigate to the Edge Events page which is filtered by alert event type, as shown in the following image:



Actions button

Agent class level commands can be initiated through this option.

CEM

CLOUDERA

Edge Management

Monitor

Edge Events

Design

Agent Manager

anonymous

1.4.0-125

Dashboard / Agent Class

Search by class name

Status	Class Name ↑
Good Health	minifi-cpp-1.21.02.0-19
Bad Health	minifi-cpp-latest
Good Health	minifi-java-1.2.2.2
Good Health	minifi-java-latest
Unknown Health	Unassigned

minifi-cpp-latest

3 Agents

Metrics

Alerts

Information

FLOW ID

5c964a49-86b1-427b-b9d3-466b56605a9c

FLOW VERSION

1

DASHBOARD

View Grafana Dashboard

Repositories (Class Average)

NAME

FlowFile

Unknown

NAME

Provenance

Unknown

Connection Queues (Class Average)

No connection queues to display.

Actions

Edit Agent Configuration

This operation will apply property changes to all agents in the class.

Manage Agents

- Edit Agent Configuration

The Edit Agent Configuration action enables you to change a range of agent properties using EFM.



Note: In a secure environment, the Edit Agent Configuration action is only available to users that have the Operator role assigned to them on the agent class. For more information, see *Policies for agent class roles*. Additionally, the action is only supported with the following agent versions:

- minifi-cpp 1.22.04 or higher
- minifi-java 1.23.02 or higher

Click **Actions** **Edit Agent Configuration** . The Edit Agent Configuration dialog appears:

Edit Agent Configuration

×

⚠

All agents in this class must be restarted for a new configuration to take effect. New property values are applied to all agents.

CLASS NAME	NUMBER OF AGENTS
<div> <div>minifi-cpp-latest</div> </div>	3

Property Name

minifi.disk.space.watchdog.enable

Value

true

⊕ Add

🔍

No Properties to display.

Cancel

Apply

Select a property which you want to update in the Property Name field, provide the desired value for that property in the Value field, and click Apply.

You can also update multiple properties at the same time. To do so, use the Add button and repeat the previous steps.



Note: If you edit the agent class name, take into consideration that leading and trailing whitespaces are accepted for Agent Class names.

The command is sent to the agents only after you click the Apply button. All agents in the agent class are notified about the required updates in their next heartbeat, so updating the values depends on the heartbeat interval configured. The status of the ongoing property update command can be followed in the Dashboard view.

You can see that property update is possible only on class level (not individual agent level). This is because Cloudera wants to enforce configuration consistency between the agents under the same agent class. To read about how to view a specific agent's configuration, see *Configuration tab* section in *Agent details in CEM*.

- Manage Agents

To get the list of agents assigned to the current agent class, click the Manage Agents option from the Actions dropdown. For more details about how to manage agents, see *Managing agents in CEM*.

Related Information

[Monitoring metrics in CEM with Grafana](#)


[Policies for agent class roles](#)

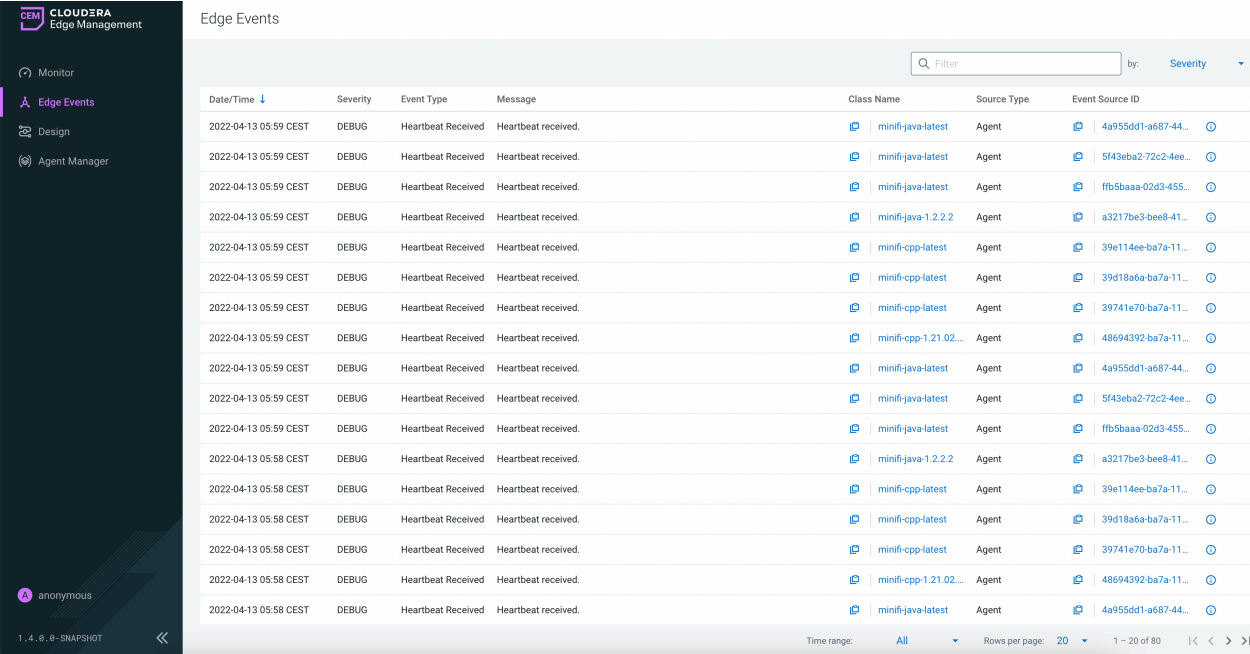
[Agent details in CEM](#)

[Managing agents in CEM](#)

Monitoring events in CEM

Learn the options, available in the Edge Events screen, that enable you to monitor C2 server and agent events.

Click the Edge Events icon ( Edge Events) to navigate to the Edge Events screen.



Date/Time	Severity	Event Type	Message	Class Name	Source Type	Event Source ID
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	4a955dd1-a687-44...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	5f43eba2-72c2-4ee...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	ffb5baaa-02d3-455...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-1.2.2.2	Agent	a3217be3-bee8-41...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39e114ee-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39d18a5a-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39741e70-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-1.21.02...	Agent	48694392-ba7a-11...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	4a955dd1-a687-44...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	5f43eba2-72c2-4ee...
2022-04-13 05:59 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	ffb5baaa-02d3-455...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-1.2.2.2	Agent	a3217be3-bee8-41...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39e114ee-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39d18a5a-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-latest	Agent	39741e70-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-cpp-1.21.02...	Agent	48694392-ba7a-11...
2022-04-13 05:58 CEST	DEBUG	Heartbeat Received	Heartbeat received.	minifi-java-latest	Agent	4a955dd1-a687-44...

The Edge Events screen provides the following details for events:

- Date/Time
- Severity
- Event Type
- Message
- Class Name
- Source Type
- Event Source ID

The Class Name and Event Source ID fields also act as links. If you click a class name link, the Metrics tab for that class appears and helps you to track details and alerts for that class, as described in the *Monitoring deployments in CEM*. If you click an event source ID link, the Metrics tab for that event appears and helps you to track details, alerts, commands, and configurations for that event, as described in the *Managing agents in CEM*.

You can display events based on time range (All, Last Hour, Last 4 Hours, Last 24 Hours, Last 7 Days) by selecting the desired value in the Time range drop-down. The number of rows displayed (20, 50, 100) can be configured by selecting the desired value from the Rows per page drop-down. The following image shows the Time range and Rows per page options in the UI:

Sorting and filtering

You can sort data in each column in ascending or descending order by clicking the column name. For example, you can sort the events based on class by clicking the Class Name column.

You can also filter the events by Date/Time, Severity, Event Type, Message, Class Name, Source Type, and Event Source ID. Select the column name in the drop-down box at the top-right corner of the UI, enter the filter value, and select RETURN on the keyboard to apply the filter.

You can use multiple column names and filter values to filter your data. Here is an example of filtering by Event Type and Source Type:


After you filter the event details as per your requirement, you can share the URL with other users who can then view your filtered event list.

Reload and show latest

You can view new events in the system by selecting either the Reload or Show Latest link which appears after you spend some time in the Edge Events page. Reload refreshes events using the existing search and sort criteria. Show Latest reloads events but sorts by the latest events using the existing search criteria.

The following image shows the Reload and Show Latest links in the Edge Events screen:

Event details

You can find the details of events by clicking the edge event details icon () beside each event. The following image shows event details of the event selected in the Edge Events section:

Related Information

[Monitoring deployments in CEM](#)

[Managing agents in CEM](#)

Deploying agents in CEM [Technical Preview]

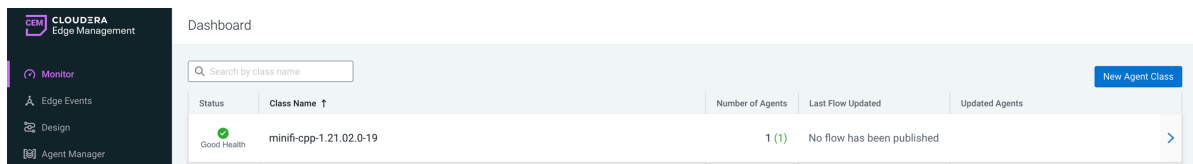
Using the Agent Deployer

Learn how you can deploy a MiNiFi application using the Agent Deployer in the Edge Flow Manager (EFM). With the help of a generated one-liner command, you can download and autoconfigure agents directly.

Procedure

1. Create a new agent class.

- a) Click Monitor on the left menu bar to access the Dashboard page.
- b) Click New Agent Class.



- c) Enter a name in the Agent Class Name text box and click Create.

The screenshot shows a 'New Agent Class' dialog box. It has a title bar 'New Agent Class' and a label 'Agent Class Name' above a text input field. The input field contains the text 'new-agent-class'. At the bottom right of the dialog are two buttons: 'Cancel' and 'Create'.

The new class is created and is added to the list of agent classes.



Note:

If you already have an existing agent class, you may skip this step. Proceed with caution, when adding a new agent to an existing class. It is recommended to only add agents with the same version as the ones in the existing agent class to ensure the compatibility of agent manifests. If you are uncertain about the versions, start with a new agent class.

2. Create an agent repository.

Before deploying an agent, you need to create an agent repository under EFM to use the Deploy Agent functionality. The default binaries base directory is `${EFM_HOME_DIRECTORY}/agent-deployer/binaries`. You can customize the path by setting the `efm.agent-deployer.binariesRootPath` property in the `efm.properties` file.

Ensure your directory structure follows the convention `[{agentType}]/{osArch}/{agentVersion}`

Accepted values:

- `agentType`: java or cpp
- `osArch`: linux*, windows*, macos* (This allows using different binaries for different OS versions.)
- `agent version`: Cloudera version of the agent (for example: 1.23.02)



Note:

MacOS is only supported for testing purposes.

Each directory should contain only one file. While the file name can be any valid name, the extension must adhere to the following constraints:

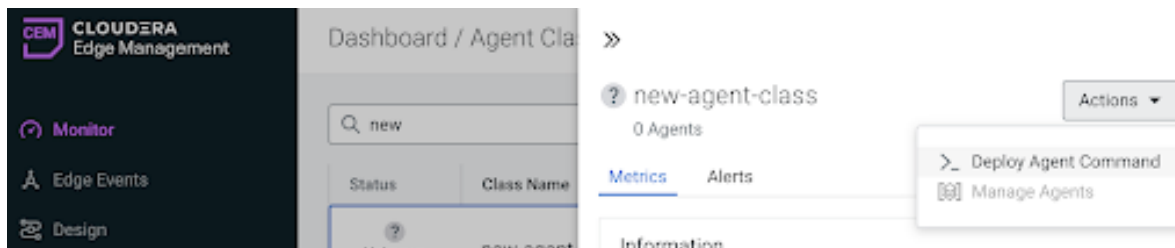
- On Linux: Only the tar.gz file type should be used.
- On Windows:
 - For the Java agent type, the tar.gz format should be used.
 - For the CPP agent type, the MSI format should be used.

A few examples:

```
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/java/linux/1.23.02/minifi.
tar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/java/linux/1.23.04/minifi.t
ar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/java/windows/1.23.02/minifi.
tar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/java/windows/1.23.04/minifi
.tar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/cpp/linux/1.23.03/minifi.t
ar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/cpp/linux/1.23.06/minifi.tar
.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/cpp/windows32bti/1.23.03/m
inifi.msi
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/cpp/windows64bit/1.23.06/min
ifi.msi
```

3. Generate Deploy Agent CLI command.

- a) Click Monitor on the left menu bar to open the Dashboard with the list of agent classes.
- b) Click the created agent class to display the agent class details.
- c) Click Actions Deploy Agent Command .



If the agent repository has been created successfully in the previous step, the Agent Type, Agent Version, and OS drop-downs are pre-populated.

- d) Choose the desired options and click Generate.

Deploy Agent CLI Command

Binaries are read from root [/var/s3fs/agent-deployer/binaries] with the expected folder structure of [{agentType}]/[osArch]/[agentVersion]

CLASS NAME
new-agent-class

Agent Type

Agent Version and OS

java

v1.23.04 - Linux

☐ Show Advanced Configurations

Cancel

Generate

The generated command appears in the same window.

Deploy Agent CLI Command

Binaries are read from root [/var/s3fs/agent-deployer/binaries] with the expected folder structure of [{agentType}/{osArch}/{agentVersion}]

CLASS NAME

new-agent-class

Agent Type

java

Agent Version and OS

v1.23.04 - Linux

☐ Show Advanced Configurations

The generated command can only be used once, for a single agent deployment. Once executed, it can not be reused. Generate as many commands as agents to deploy.

Deploy Agent CLI Command

```

1 curl -L \
2   -d agentClass=new-agent-class \
3   -d agentIdentifier=007ac386-c406-4323-81e3-3cc85334cc74 \
4   -d agentType=java \
5   -d agentVersion=1.23.04 \
6   -d autoConfigureSecurity=false \
7   -d baseUrl=http%3A%2F%2Funsecure.cemcldr.link%2Fefm%2Fapi \
8   -d hbPeriod=5000 \
9   -d osArch=Linux \
10  -d serviceName=minifi \
11  -d serviceUser=minifi \
12  -d trustSelfSignedCertificates=false \
13  http://unsecure.cemcldr.link/efm/api/agent-deployer/script | bash -

```

Cancel

Generate

e) To install the agent, copy and paste the generated CLI command into the host where the agent will be installed.

Note:

Ensure the agent host has a network connection to EFM to successfully run the command.

20

4. Set the advanced options for the Deploy Agent CLI command.

In specific scenarios, the default configuration may not meet your specific requirements. Use advanced configurations to customize parameters.

To make the advanced options visible, select the Show Advanced Configurations checkbox.

Deploy Agent CLI Command ✕

☒ Show Advanced Configurations

EFM Base URL

Agent ID

Heartbeat Period

Service User (applicable for Linux)

Service Name (applicable for Linux)

Autoconfigure Security

☐ True ☒ False

Self-Signed Certificates

☐ Trust Self-Signed Certificates [?](#)

CA Cert PEM File Location (Optional)

Dynamic Properties (Optional)

Property Name	Value
<input type="text"/>	<input type="text"/>

EFM Base URL

The full URL of the EFM REST API base. The generated command uses this URL to access EFM from the remote host. If EFM is behind a load balancer or a proxy, you can override this URL.



Note: For a more sophisticated way to handle EFM behind a proxy, set the `efm.proxy.c2ProxyPath` property in the `efm.properties` file. This automatically sets the EFM Base URL to the correct value.

Agent ID

The agent ID is automatically generated, but you can also set a custom value. If a custom value is set, make sure that each command generation has a unique identifier.

Heartbeat Period

The agent sends heartbeats periodically using this value.

Service User

- On Linux: A user is created with this name and the agent process runs under this user.
- On Windows: This is not applicable at the moment because the Java agent is not started as a service and the C++ agent is parameterized with the Service User.

Autoconfigure Security

If security is enabled in EFM (for example any authentication method is turned on), the agent needs to connect to EFM in a secured manner. If Autoconfigure Security is enabled, EFM generates the necessary certificates for the agent, and the command downloads the certificates through a secured channel, and configures the agent automatically.

You can provide your own Certificate Authority (CA) for signing certificates. If a custom CA is not provided, EFM generates one during startup, which will be used for the certificates.



Note: To enable this feature, the following values must be set in the `efm.properties` file:

```
efm.agent-deployer.security.autoConfiguration=true
efm.agent-deployer.security.ca.privateKeyPassword=<strong_password>
```

For more information on Autoconfigure Security features, see *Configuring Agent Deployer for securing agents*.

Self-Signed Certificates and CA Cert PEM File Location

If EFM is set up using a self-signed certificate (where EFM generated the CA and it was not provided externally), the request issued by the Agent Download command may be rejected since the issuing host will not trust EFM.

You have two options:

- Select the Trust Self-Signed Certificates checkbox to trust the self-signed certificate and proceed with the Agent Download command.
- Use the CA Cert PEM File Location to reference a CA Cert on the agent file system. This allows the agent to trust EFM by using the specified CA certificate.

Dynamic properties:

You have the flexibility to override any arbitrary MiNiFi properties.

- For MiNiFi Java agents: The properties are located in the `bootstrap.conf` file.
- For MiNiFi C++ agents: The properties are located in the `minifi.properties` file.

5. Run the command.

Copy and paste the generated CLI command on the target host's shell or command line.



Note: For Windows, PowerShell version 5.1 or above is required.

The command downloads the agent binary into the directory where the script was run. After that, the command extracts, configures, and runs the agent in the background.

For Java agents on Linux, you should see a log message similar to the following.



Note: Log messages may vary slightly based on the specific agent and OS type.

[illegible]

```
-- Configuring MiNiFi...
-- Starting MiNiFi as a simple background process...
-- Waiting until MiNiFi is up...
-- OK
#Mon Jul 24 15:40:29 CEST 2023
port=64281
pid=87120
secret.key=451f4e40-15a5-4a58-8d18-63983fec927c

MiNiFi is now started as a background process.

You can stop it by issuing the following commands:

%> cd "minifi-1.23.04-b15"
%> bin/minifi.sh stop

To start again:
%> cd "minifi-1.23.04-b15"
%> bin/minifi.sh start

-- Installation has successfully completed.
   In addition of the existing (default) configuration values, the following have been applied:

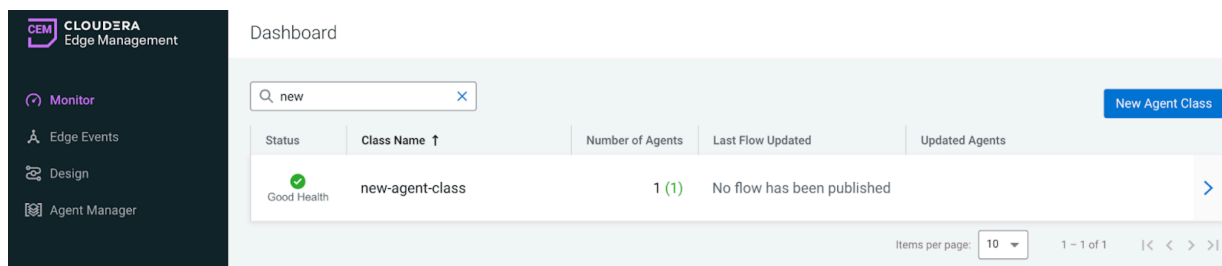
---
c2.agent.identifier=061c6cf6-1922-41b7-8239-ad25a0f5be9a
c2.rest.path.heartbeat=/c2-protocol/heartbeat
c2.rest.path.acknowledge=/c2-protocol/acknowledge
c2.rest.url=http://unsecure.cemcldr.link/efm/api/c2-protocol/heartbeat
c2.rest.url.ack=http://unsecure.cemcldr.link/efm/api/c2-protocol/acknowledge
c2.agent.class=new-agent-class
c2.agent.heartbeat.period=5000
c2.enable=true
c2.rest.path.base=http://unsecure.cemcldr.link/efm/api
---

If you would like to modify this configuration, you need to perform these steps:

1) Stop MiNiFi
2) Edit the files located in "minifi-1.23.04-b15/conf"
```

3) Start MiNiFi

After successful installation, the agent sends heartbeats, appearing as an active agent in EFM.



Note:

- On Linux: both Java and C++ agents are started as a service using the Service User and Service Name parameters.
- On Windows:
 - Java agents run as a background process. The agents need to be manually restarted after an OS restart.
 - C++ agents are started as a service, but configuring the Service Name and Service User is not possible.

Related Information

[Configuring Agent Deployer for securing agents](#)

Configuring the Agent Deployer for securing agents

Learn how you can establish a secure connection between the Edge Flow Manager (EFM) and the agents. EFM's agent deployer functionality provides a convenient approach for security configuration. By enabling auto-configuration, the deployment script automatically downloads a security bundle during agent setup, containing the agent's certificates along with its binary. This streamlined process ensures a robust and secure environment for seamless communication between EFM and agents.

There are two ways to sign the agent certificates:

- Using your own Intermediate CA key and certificate to sign the agent certificate

In this scenario, you need to place your own private key and certificate with the names `efm-cert.pem` and `efm-key.key` respectively under the folder specified in the `efm.agent-deployer.security.ca.location` property.

- Letting EFM generate a self-signed Root CA certificate

During EFM startup, the application checks if `efm-cert.pem` and `efm-key.key` are present in the configured folder. If they are missing, EFM generates them using the `efm.agent-deployer.security.ca.*` properties. In clustered mode, these certificates are synchronized between the EFM nodes as needed.

Minimal configuration

The default configuration is suitable for basic use cases. It enables security auto-configuration and generates a self-signed Root CA certificate, which is utilized for signing the agent certificates.

To enable this default configuration, ensure the following settings are in place:

```
efm.agent-deployer.security.autoConfiguration=true
efm.agent-deployer.security.ca.privateKeyPassword=password
```

Where:

- `efm.agent-deployer.security.autoConfiguration`
Indicates the automatic generation of the Root CA and agent certificates.
- `efm.agent-deployer.security.ca.privateKeyPassword`
Specifies the password for the private key of the root certificate.

Advanced configuration

To enable this configuration, ensure the following settings are in place:

```
efm.agent-deployer.security.autoConfiguration=true
efm.agent-deployer.security.trustSelfSignedCertificates=true
efm.agent-deployer.security.accessTokenTtlMin=60
```

Where:

- `efm.agent-deployer.security.autoConfiguration`
Enables the automatic generation of the root CA and agent certificates.
- `efm.agent-deployer.security.trustSelfSignedCertificates`
Allows configuration to trust self-signed certificates by default or not. You can override this parameter anytime under Show Advanced Configurations in the UI.
- `efm.agent-deployer.security.accessTokenTtlMin`
For security reasons, this parameter allows you to configure how long the generated command can be used after its generation.

Root Certificate properties

These properties are used to generate the Root CA certificate. If you wish to use your own key and certificate, just set the `efm.agent-deployer.security.ca.privateKeyPassword` and `efm.agent-deployer.security.ca.location` properties and place the `efm-cert.pem` and `efm-key.key` files in this location.

```
efm.agent-deployer.security.ca.dn=CN=generated-efm-root-ca
efm.agent-deployer.security.ca.privateKeyAlgorithm=RSA
efm.agent-deployer.security.ca.privateKeyEncryptionAlgorithm=AES-256-CBC
efm.agent-deployer.security.ca.privateKeySize=4096
efm.agent-deployer.security.ca.privateKeyPassword=password
efm.agent-deployer.security.ca.certificateSigningAlgorithm=SHA256WITHRSA
efm.agent-deployer.security.ca.certificateValidityInDays=7305
efm.agent-deployer.security.ca.location=conf
```

Where:

- `efm.agent-deployer.security.ca.dn`
Specifies the Distinguished Name (DN) of the self-signed root certificate (if you are not providing your own certificate).
- `efm.agent-deployer.security.ca.privateKeyAlgorithm`
Sets the algorithm of the root certificate's private key (default: RSA).
- `efm.agent-deployer.security.ca.privateKeyEncryptionAlgorithm`
Specifies the algorithm used for encrypting the root certificate's private key (default: AES-256-CBC).
- `efm.agent-deployer.security.ca.privateKeySize`
Determines the number of bits for generated keys (default: 4096).
- `efm.agent-deployer.security.ca.privateKeyPassword`
The password for the private key.

- `efm.agent-deployer.security.ca.certificateSigningAlgorithm`
Sets the algorithm for root certificate generation (default: SHA256WITHRSA).
- `efm.agent-deployer.security.ca.certificateValidityInDays`
Specifies the validity period of the root certificate in days. The default is 20 years, which can be adjusted based on the security requirements.
- `efm.agent-deployer.security.ca.location`
The directory path where the `efm-key.key` and `efm-cert.pem` files are located.

Agent Certificate properties

Agent keys and certificates are generated using these properties.

```
efm.agent-deployer.security.agent.privateKeyAlgorithm=RSA
efm.agent-deployer.security.agent.privateKeyEncryptionAlgorithm=AES-256-CBC
efm.agent-deployer.security.agent.privateKeySize=4096
efm.agent-deployer.security.agent.certificateSigningAlgorithm=SHA256WITHRSA
efm.agent-deployer.security.agent.certificateValidityInDays=7305
efm.agent-deployer.security.agent.keystoreType=JKS
efm.agent-deployer.security.agent.truststoreType=JKS
efm.agent-deployer.security.agent.location=conf
```

Where:

- `efm.agent-deployer.security.agent.privateKeyAlgorithm`
Specifies the algorithm of the agent's private key (default: RSA).
- `efm.agent-deployer.security.agent.privateKeyEncryptionAlgorithm`
Sets the algorithm for encrypting the agent's private key (default: AES-256-CBC).
- `efm.agent-deployer.security.agent.privateKeySize`
Determines the number of bits for generated keys (default: 4096).
- `efm.agent-deployer.security.agent.certificateSigningAlgorithm`
Sets the algorithm for signing agent certificates (default: SHA256WITHRSA).
- `efm.agent-deployer.security.agent.certificateValidityInDays`
Specifies the validity period for agent certificates in days. The default is 20 years, which can be adjusted based on the security requirements.
- `efm.agent-deployer.security.agent.keystoreType`
For Java agents, you can configure the keystore type (default: JKS, other options are BCFKS, PKCS12).
- `efm.agent-deployer.security.agent.truststoreType`
For Java agents, you can configure the truststore type (default: JKS, other options are BCFKS, PKCS12).
- `efm.agent-deployer.security.agent.location`
The folder in the agent's device where the keystore/truststore/key/cert is placed (default: conf).

Managing agents in CEM

Learn about the Agent Manager and how to manage agents in Cloudera Edge Management (CEM).


The Agent Manager view provides you better understanding and more control over the agents in the system. The health of the agents can be easily monitored. Richer details about the agents can be retrieved. With the debug command option, live logs and configuration can be gathered without leaving EFM. With the property update functionality, the agent configuration can be changed through EFM.

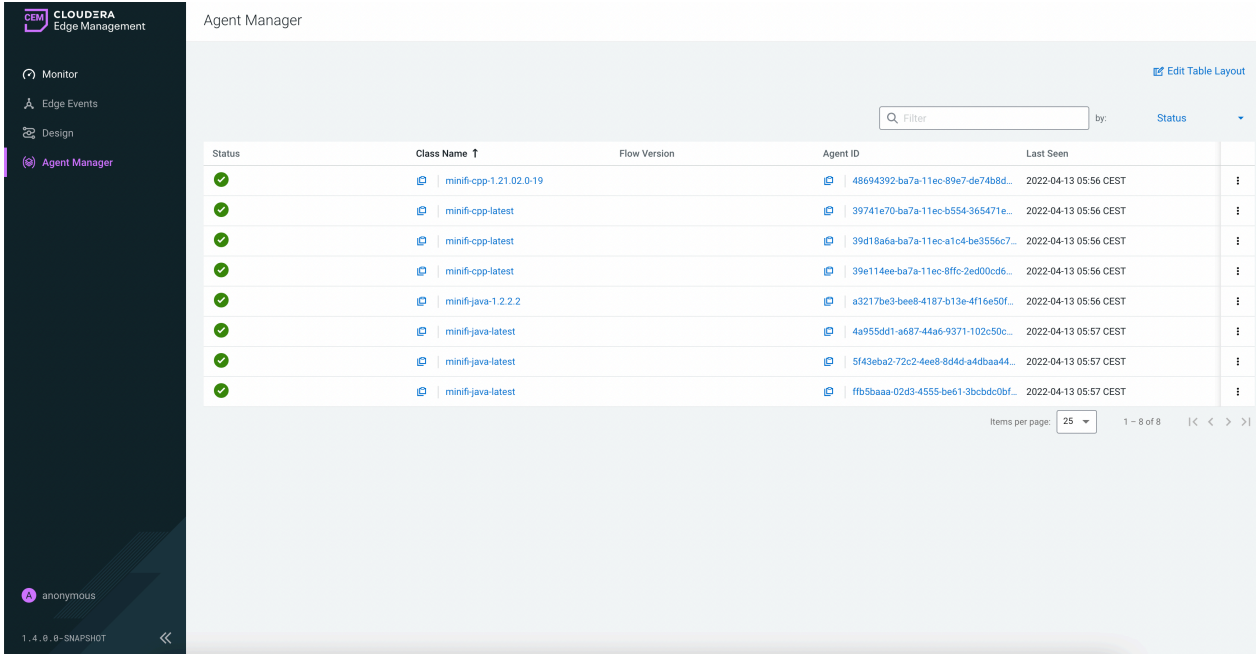


Note: Not all agent versions are supporting remote command executions.

Learn about the options, available in the Agent Manager screen, that enable you to view, filter, and initiate commands on specific agents.

For better understanding and more control over the agents in the system, check out the video on the Cloudera Edge Management YouTube playlist:

To access the Agent Manager view, select the Agent Manager () option from the menu. The Agent Manager screen appears, as shown in the following image:



Status	Class Name ↑	Flow Version	Agent ID	Last Seen
✓	minifi-cpp-1.21.02.0-19		48694392-ba7a-11ec-89e7-de74b8d...	2022-04-13 05:56 CEST
✓	minifi-cpp-latest		39741e70-ba7a-11ec-b554-365471e...	2022-04-13 05:56 CEST
✓	minifi-cpp-latest		39d18a6a-ba7a-11ec-a1c4-be3556c7...	2022-04-13 05:56 CEST
✓	minifi-cpp-latest		39e114ee-ba7a-11ec-8ffc-2ed00cd6...	2022-04-13 05:56 CEST
✓	minifi-java-1.2.2.2		a3217be3-bee8-4187-b13e-4f16e50f...	2022-04-13 05:56 CEST
✓	minifi-java-latest		4a955dd1-a687-44a6-9371-102c50c...	2022-04-13 05:57 CEST
✓	minifi-java-latest		5f43eba2-72c2-4ee8-8d4d-a4dbaa44...	2022-04-13 05:57 CEST
✓	minifi-java-latest		ffb5baaa-02d3-4555-be61-3bcbdc0bf...	2022-04-13 05:57 CEST

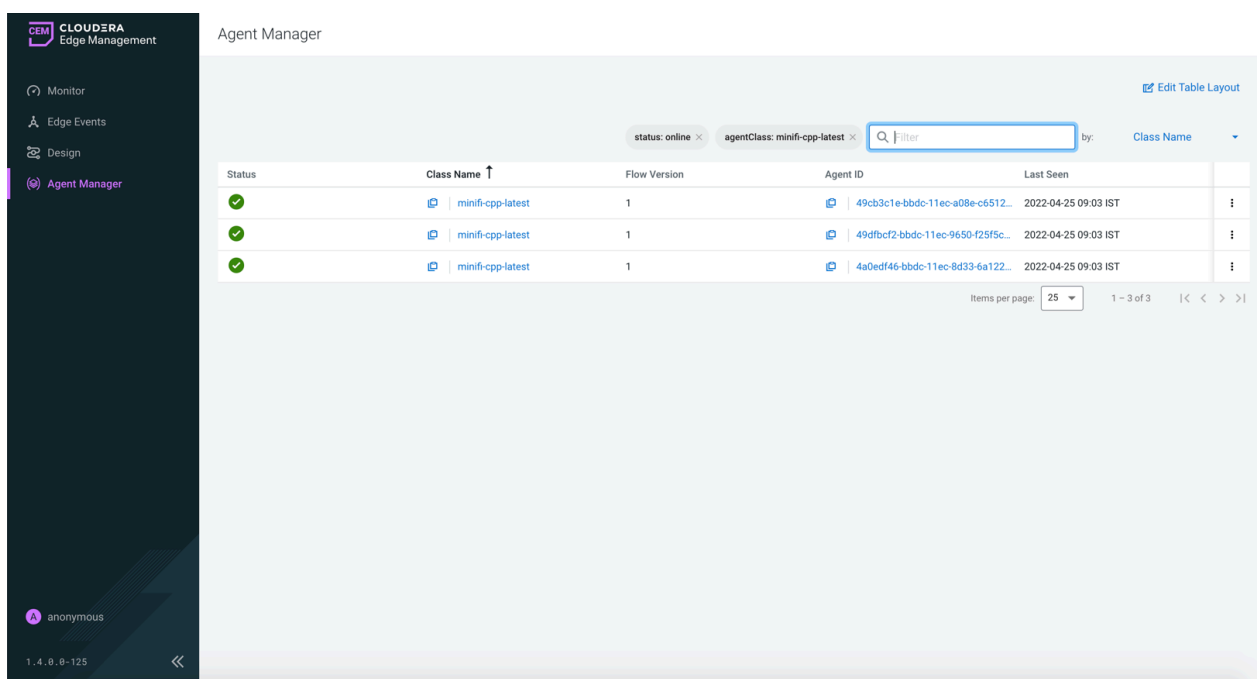
Edit Table Layout

CEM provides a dynamically editable table view to make the listing view as customizable as possible depending on your needs. For more information about how to customize the table view in Agent Manager screen, see *Edit table layout in CEM*.

Sorting and filtering

You can sort data by most of the columns in ascending or descending order by clicking the column name. You can also filter the agents. To do this select the column name in the drop-down box at the top-right corner of the UI, enter the filter value, and press RETURN on the keyboard to apply the filter.

You can use multiple column names and filter values to filter your data. Here is an example of filtering by Status and Class Name:



Items per page

You can set the displayed item count per page by using the Items per page dropdown at the bottom of the page near the pagination options. If there is more than one page available, you can use that navigation panel to jump to the first, previous, next, or straight to the last page.

View agent details

The agent details view is accessible through the View Agent Details icon in the extreme right of the given agent's row. For more information about viewing agent details, see *Agent details in CEM*.

For more information about Debug Command and Property Update Command, check out the video on the Cloudera Edge Management YouTube playlist:

Related Information

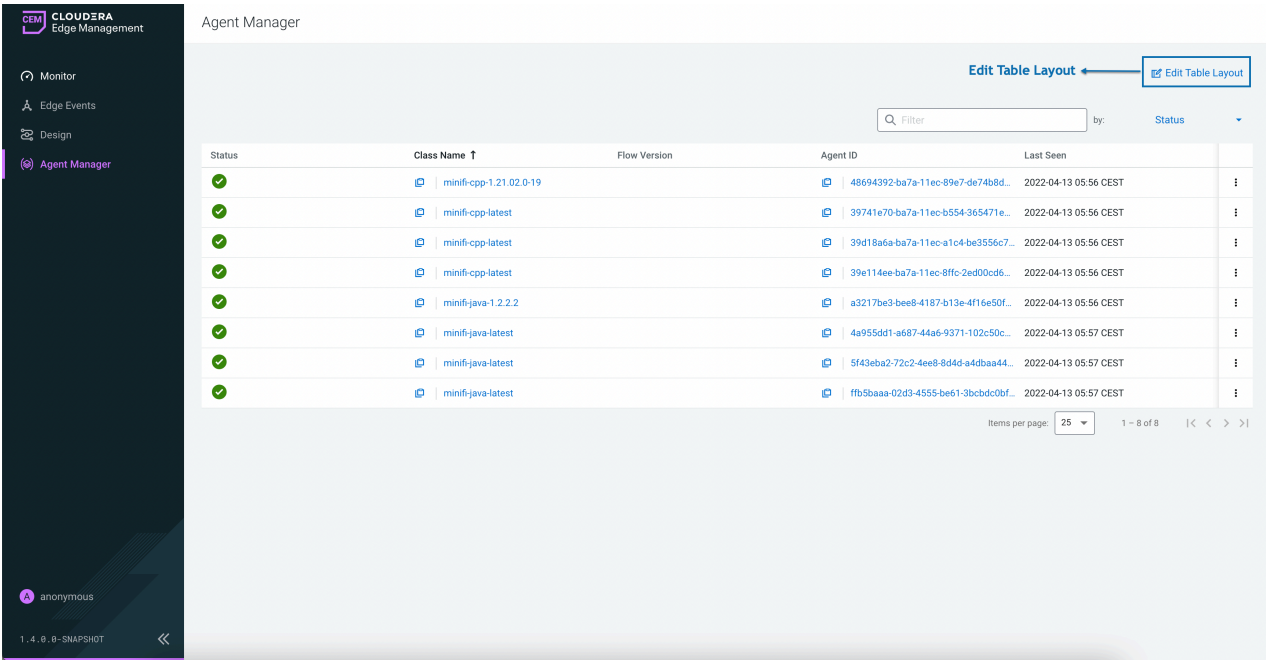
[Edit table layout in CEM](#)

[Agent details in CEM](#)

Edit table layout in CEM

Learn how to make the agent listing view as customizable as possible depending on your needs.

With the table layout editor, you can change which attribute you want to see on the agent listing table. In addition you can also modify the order of the columns. To open the editor, click the Edit Table Layout button at the top-right corner of the screen, as shown in the following image:



The Edit Table Layout dialog appears, as shown in the following image:

Edit Table Layout ✕

Columns to Display

☐ All Columns

☒ Status

☒ Class Name

☒ Flow Version

☒ Agent ID

☒ Last Seen

☐ Last Flow Update

☐ Hostname

☐ IP

☐ Registered On

☐ Device Operating System

☐ Device Architecture

☐ Device Cores

☐ Device Memory

☐ Agent Type

☐ Agent Version

Displayed Column Order

☐ Status

☐ Class Name

☐ Flow Version

☐ Agent ID

☐ Last Seen

Cancel

Apply

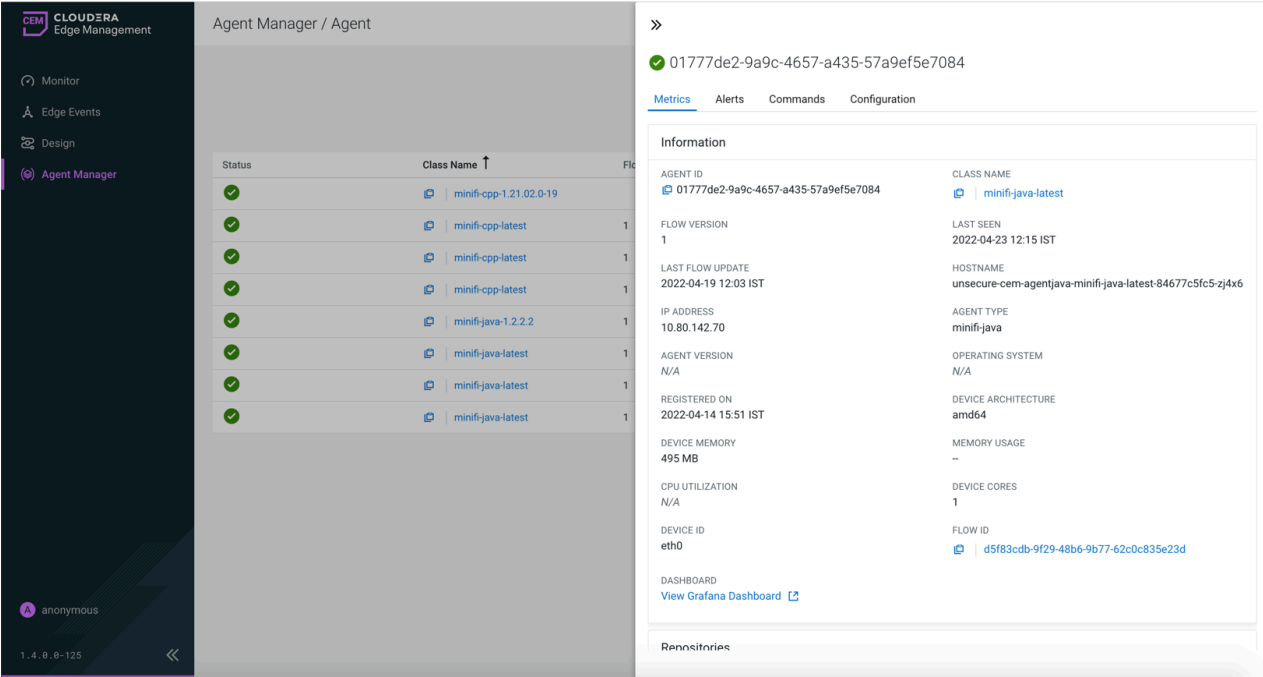
You can select columns for display in the Columns to Display section on the left. You can change the order of display in the Displayed Column Order section through dragging and dropping items to the desired order.

You can bookmark or share the configured table layout by using the URL.

Agent details in CEM

Learn how to check individual agent details, view the alerts, view the configurations, check the status, and follow the history of the triggered commands.

To access the agent details, click the View Agent Details icon (⋮) in the extreme right of the given agent’s row. A dialog appears with the Metrics tab as the first tab, as shown in the following image:



Metrics tab

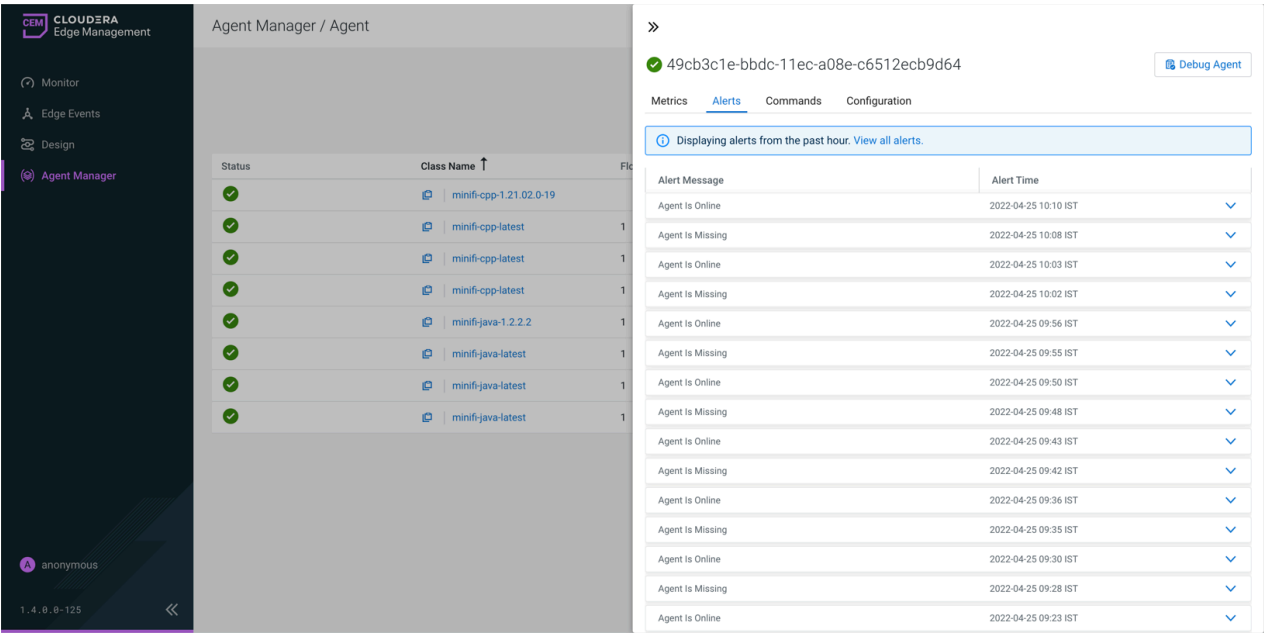
In the Information panel, you get a general overview of the agent status, deployed flow version, and other useful device information. You can reach the agent class details or the designer of the given agent class if you click on the given element.

The Repositories panel provides usage details of the repositories of the selected agent (where applicable).

The Connection Queues panel displays all of the connections used in the given agent along with its metrics.

Alerts tab

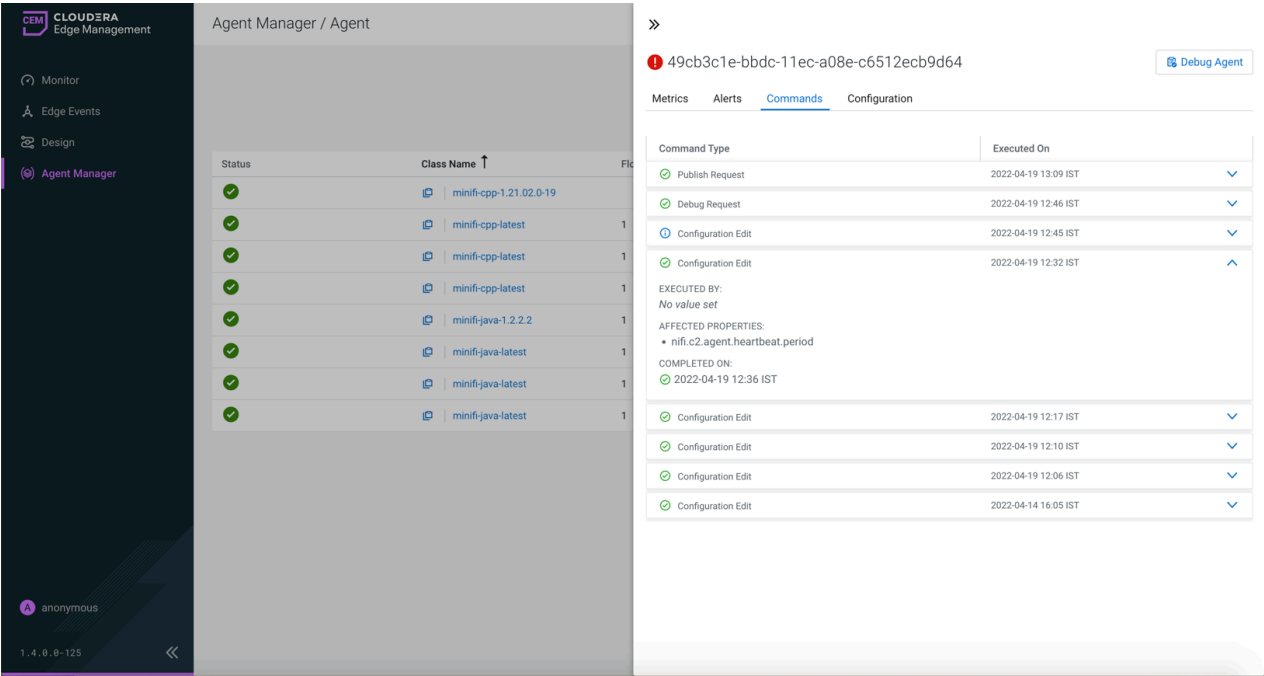
The Alerts tab display alerts from the past hours. Here you can monitor, for example, if there was no heartbeat from the agent for a specific time, or when it became available again. To view all alerts, click the View all alerts link.



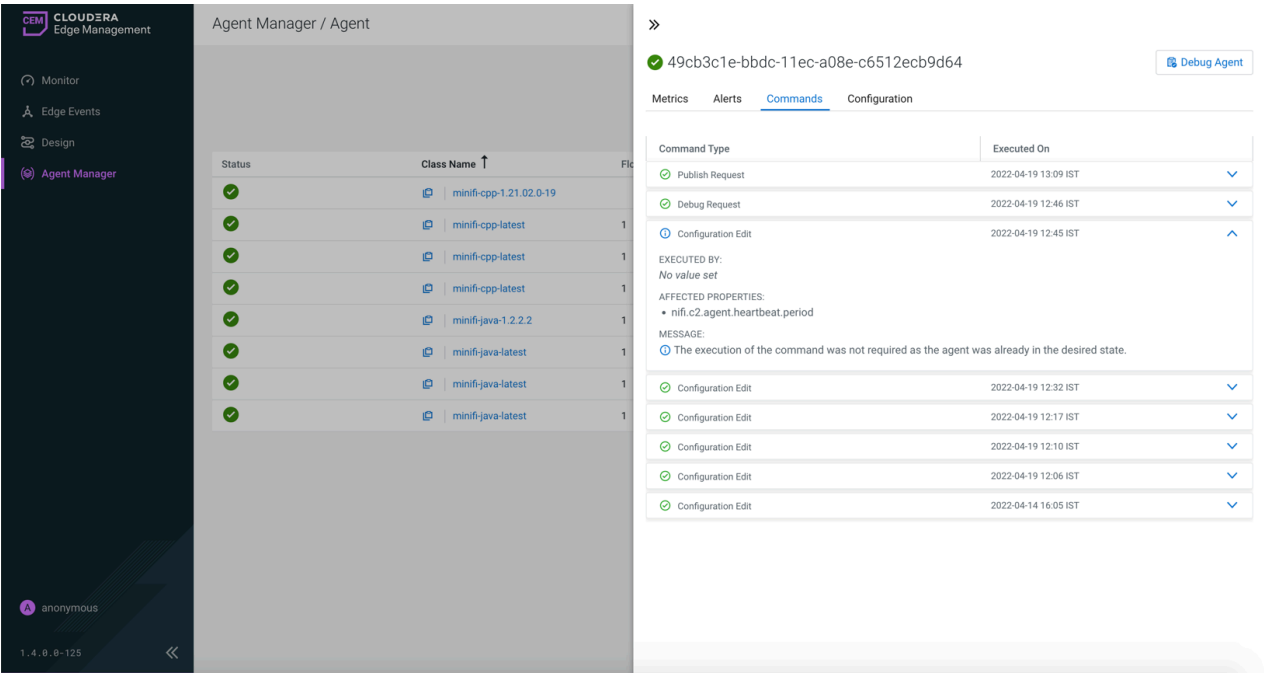
Commands tab

In the Commands tab, you can check the last 20 commands and their statuses which were sent to the agent. You can configure the number of displayed items through the `efm.agentManager.commands.displayLimit` property in the `efm.properties` file.

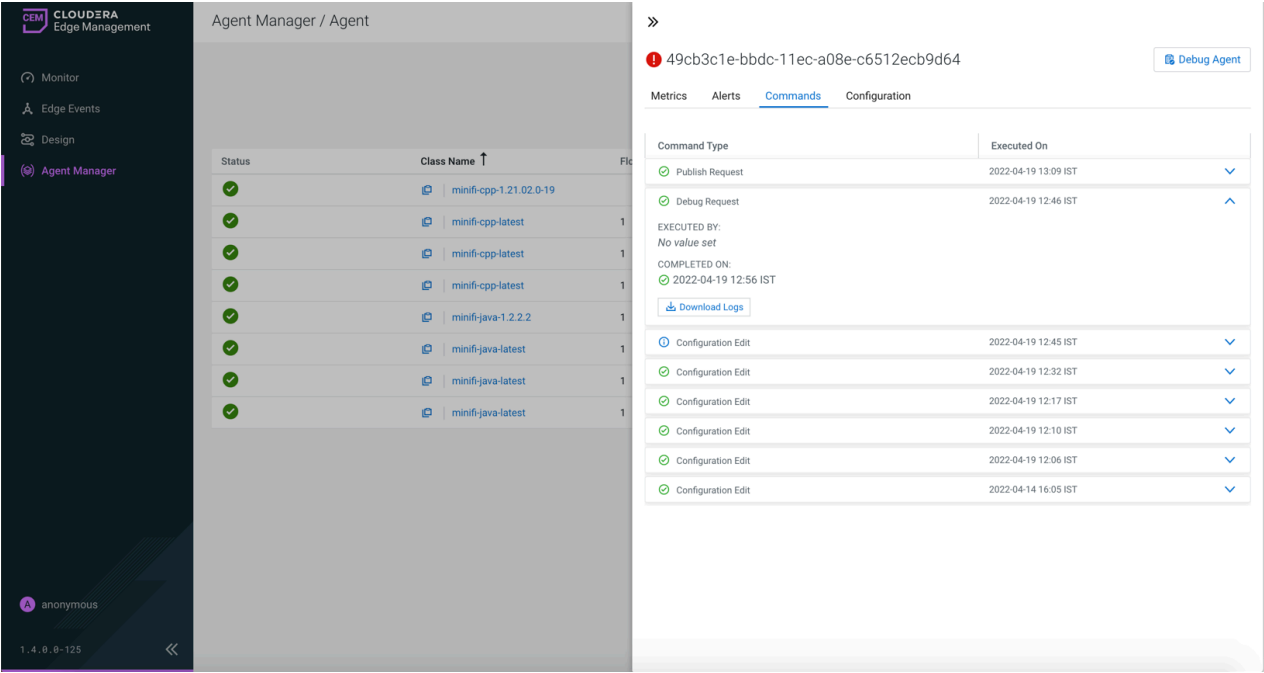
The details view may vary depending on the type of the command. For example, for Configuration Edit command, you can check the affected property name, as shown in the following image:



If no update was required on the agent and the given agent had the new value already, then you see the The execution of the command was not required as the agent was already in the desired state. message, as shown in the following image:

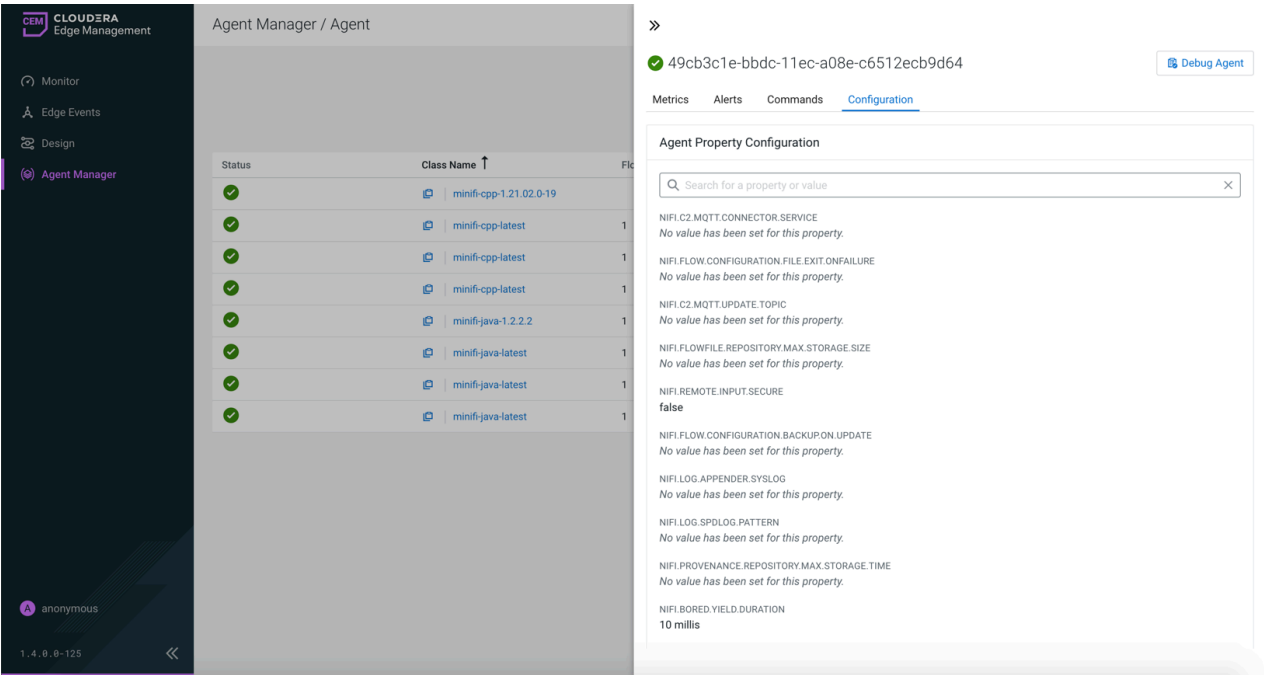



For debug requests, you are prompted with a Download Logs button to download the debug bundle retrieved from the agent, as shown in the following image:



Configuration tab

The property configuration of an agent can be reviewed on the Configuration tab if the selected agent supports this functionality. Properties displayed can be filtered by both name and value. The following image shows the Configuration tab:



 **Note:** This feature is only supported with the following agent versions:

- minifi-cpp 1.22.04 or higher
- minifi-java 1.23.02 or higher

CEM also supports editing properties at agent class level. For more information, see *Monitoring deployments in CEM*.

Debug Agent button

The debug command functionality allows you to collect debug information from agents utilizing the C2 protocol. For more information, see *Debug agent in CEM*.

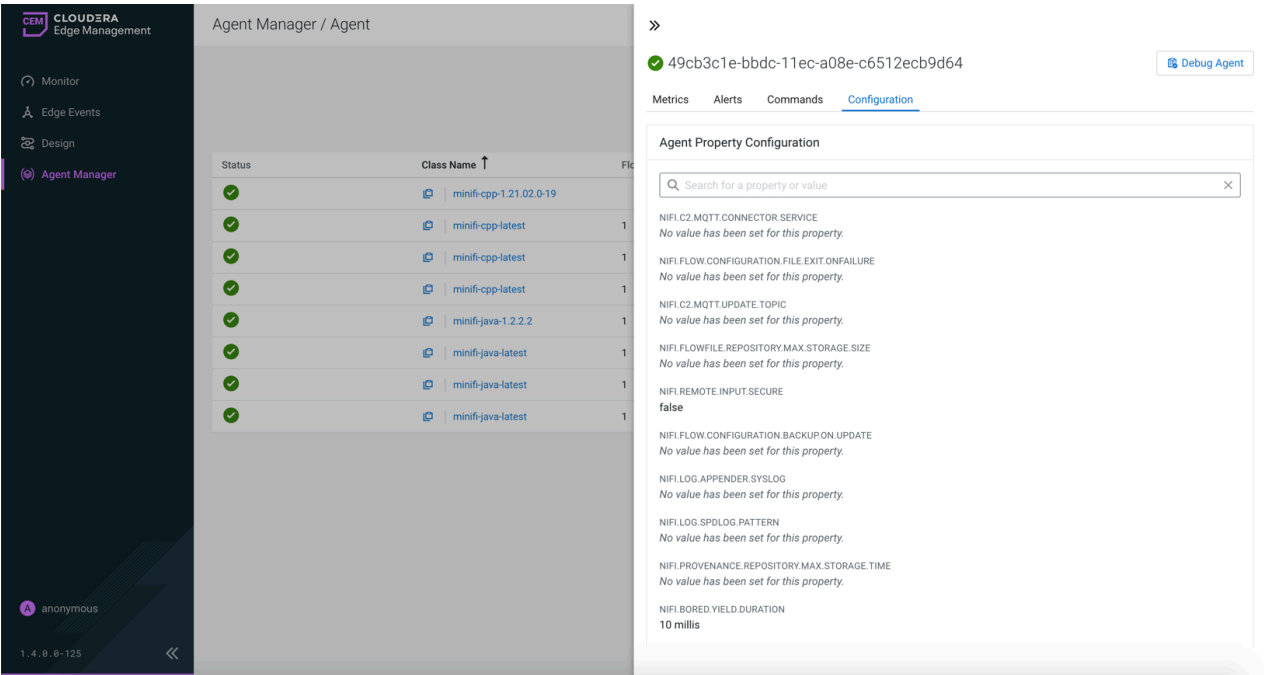
Related Information

- [Debug agent in CEM](#)
- [Monitoring deployments in CEM](#)

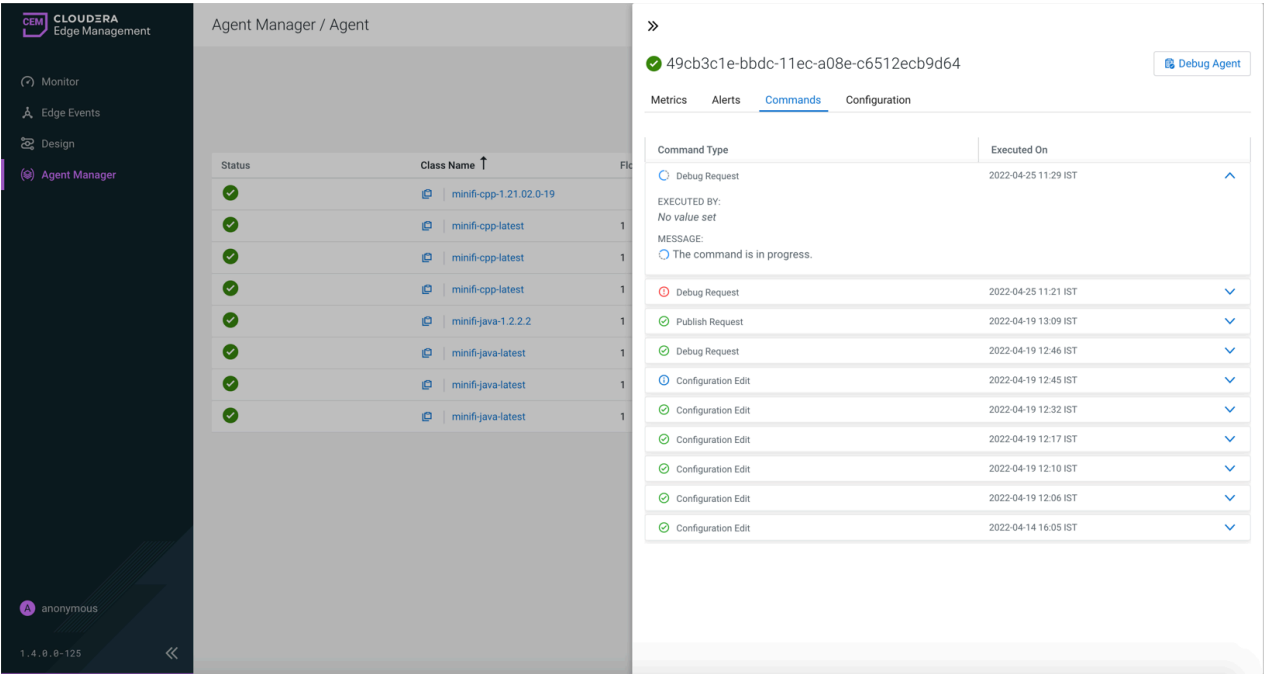
Debug agent in CEM

Learn how to use the debug command functionality which allows you to collect debug information from agents utilizing the C2 protocol.

Execute the debug command to initiate the operation to the agent by using the Debug Agent button, as shown in the following image:



This operation command initiates the debug request within the next heartbeat to the agent, and the view shifts to the Commands tab. The following image shows that the debug request is in progress:



When the agent receives it in the next heartbeat, it starts to upload the files to the EFM server.

When the files are available in the EFM server, a download button appears in the command details.

You can control the maximum uploadable file size by changing the `efm.data.transfer.maxFileSize` parameter. The default value of the parameter is 16 MB which is the maximum value supported by the current storage implementation.

**Note:**

- The debug operation is supported only for CPP agents from a minimum version of 1.22.01 and for Java agents from a minimum version of 1.22.10.
- The agent debug option is available only for the operators in a secure environment. For more information, see *Policies for agent class roles*.




Related Information

[Policies for agent class roles](#)

Expression language in CEM

The NiFi Expression Language provides the ability to reference the attributes of flow files, compare them to other values, and manipulate their values.

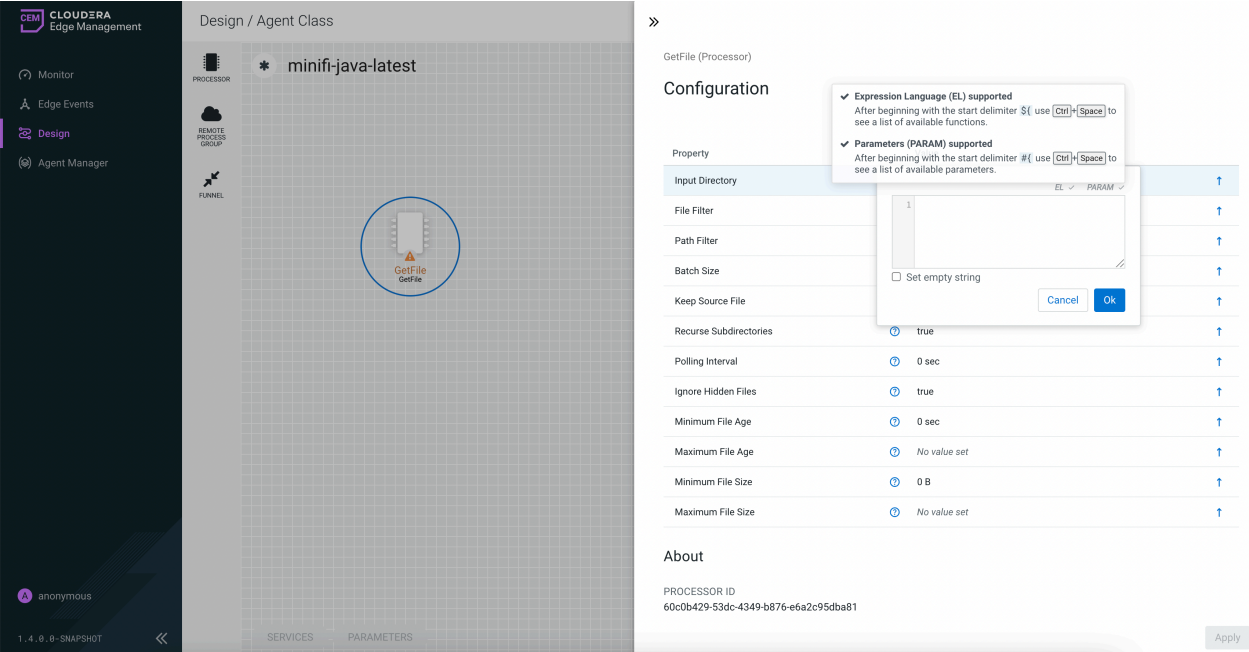
As you extract attributes from content of the flow files and add user-defined attributes, they do not help much as an operator unless you have some mechanism by which you can use them. The NiFi Expression Language allows you to access and manipulate FlowFile attribute values as you configure your flows. Not all processor properties allow the Expression Language to be used, but many do. In order to determine whether or not a processor or service property supports the Expression Language, click in the property value field. The eligibility indicators show if Expression

Language is supported () or unsupported (). Additionally, you can hover over the help icon () next to the property name. The help icon provides a tooltip that displays the Expression Language scope.

To configure an eligible property to utilize Expression Language, use the \$ symbol as the start, with the expression enclosed in curly braces:

```
${Expression}
```

If you start your new entry with the start delimiter \${, selecting the keystroke control+space shows a list of available functions. Help text describing this process appears when you hover over the Expression Language eligibility indicator.



Configuration

Property	Value	EL	PARAM
Input Directory		✓	✓
File Filter		✓	✓
Path Filter		✓	✓
Batch Size		✓	✓
Keep Source File		✓	✓
Recurse Subdirectories	<input checked="" type="checkbox"/> true	✓	✓
Polling Interval	<input type="text" value="0 sec"/>	✓	✓
Ignore Hidden Files	<input checked="" type="checkbox"/> true	✓	✓
Minimum File Age	<input type="text" value="0 sec"/>	✓	✓
Maximum File Age	<input type="text" value="No value set"/>	✓	✓
Minimum File Size	<input type="text" value="0 B"/>	✓	✓
Maximum File Size	<input type="text" value="No value set"/>	✓	✓

About

PROCESSOR ID
60c0b429-53dc-4349-b876-e6a2c95dba81

Apply

An expression can be as simple as an attribute name. For example, to reference the uuid attribute, you can simply use the value `${uuid}`. If the attribute name begins with any character other than a letter, or if it contains a character other than a number, a letter, a period (.), or an underscore (_), you need to quote the attribute name. For example, `${My Attribute Name}` is not valid, but `${'My Attribute Name'}` refers to the My Attribute Name attribute.

Parameters in CEM

Parameters provide the ability to parameterize the values of processors and service properties in the flow including sensitive properties. You can create and configure parameters within the Cloudera Edge Management (CEM) UI.

To determine whether a parameter can be used for a property, click in the property value field. The eligibility

indicators show if parameters are supported (✓) or unsupported (⊘).

The screenshot displays the CEM UI interface. On the left, a sidebar shows navigation options: Monitor, Edge Events, Design, and Agent Manager. The main area is titled 'Design / Agent Class' and shows a 'minifi-java-latest' processor. A 'GetFile' processor is highlighted in the center. To the right, the 'Configuration' panel for the 'GetFile' processor is visible. It lists various properties with their current values and eligibility indicators for 'EL' (Expression Language) and 'PARAM' (Parameters). A modal dialog is open over the 'Input Directory' property, showing that both 'EL' and 'PARAM' are supported. The dialog also includes a search bar and a 'Set empty string' checkbox.



Note: Properties that reference services can not use parameters.

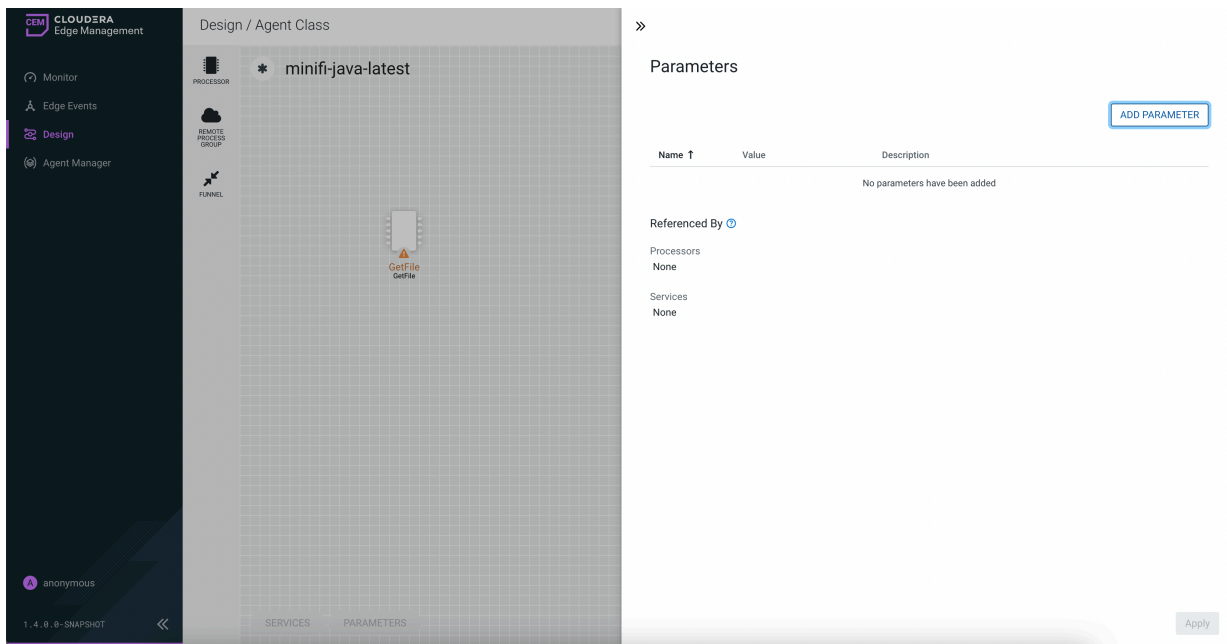
Adding parameters in CEM

Learn how to add parameters using the Cloudera Edge Management (CEM) UI.

Procedure

1. To add a parameter, click the Parameters button at the bottom-left corner of the canvas, or simply right-click on the canvas and select Parameters.

The Parameters window appears as shown in the following image:



2. Click the ADD PARAMETER button.

The Add Parameter window dialog appears as shown in the following image:

The 'Add Parameter' dialog box is shown. It has a title bar with a close button (X). The form contains the following fields and controls:

- Name***: A text input field with a placeholder 'Name'.
- Value**: A large text input field with a placeholder 'VALUE'.
- ☐ **Set empty string**: A checkbox.
- Sensitive Value**: Two radio buttons, 'Yes' and 'No'. The 'No' radio button is selected.
- Description**: A text input field with a placeholder 'Description' and a character count '0/1000' on the right.
- Buttons**: 'Cancel' and 'Add' buttons at the bottom right.

3. Configure the following properties:

Property	Description
Name	Enter a name that is used to denote a parameter. Only alpha-numeric characters (a-z, A-Z, 0-9), hyphens (-), underscores (_), periods (.), and spaces are allowed.
Value	Enter a value that is used when the parameter is referenced. Parameter values do not support Expression Language or embedded parameter references.
Set empty string	Select the checkbox to set parameter value to an empty string.
Sensitive Value	Set whether the parameter value should be considered sensitive. If so, the value of the parameter is not shown in the UI once applied. Once you create a parameter, its sensitivity flag cannot be changed.
Description	Enter a description that explains what the parameter is for. This field is optional.

4. Click Add button to add the parameter.

5. Click Apply button to save the changes.

Referencing parameters in CEM

Learn how to reference parameters using the Cloudera Edge Management (CEM) UI.

To configure an eligible property to reference a parameter, use the # symbol as the start, with the name of the parameter enclosed in curly braces:

```
# {Parameter.Name}
```

This can be escaped using an additional # character at the beginning. To illustrate this, assume that the parameter abc has a value of xxx and parameter def has a value of yyy. Then, the following user-defined property values will evaluate to these effective values:

User-Entered Literal Property Value	Effective Property Value	Explanation
# {abc}	xxx	Simple substitution
# {abc}/data	xxx/data	Simple substitution with additional literal data
# {abc}/# {def}	xxx/yyy	Multiple substitution with additional literal data
# {abc	# {abc	No { } for parameter replacement
#abc	#abc	No { } for parameter replacement
## {abc}	# {abc}	Escaped # for literal interpretation
### {abc}	#xxx	Escaped # for literal interpretation, followed by simple substitution
#### {abc}	## {abc}	Escaped # for literal interpretation, twice
##### {abc}	##xxx	Escaped # for literal interpretation, twice, followed by simple substitution
# {abc/data}	Exception thrown on property set operation	/ not a valid parameter name character

When referencing a parameter from within expression language, the parameter reference is evaluated first. As an example, to replace xxx with zzz for the abc parameter:

```
${ # {abc}:replace('xxx', 'zzz') }
```

Using parameters in CEM

Learn how to use parameters in Cloudera Edge Management (CEM).
Parameters can be easily referenced or created as you configure the components in your flow.

Referencing existing parameters

- Existing parameters can be referenced for a processor or service property value.
- 1. To reference an existing parameter, select the property value field and clear the default value if one exists.
 - 2. Enter the start delimiter #{.

>>

FetchFile (Processor)

Configuration

Property	Value	
File to Fetch	<div>EL ✓ PARAM ✓ 1 #{</div>	↑
Completion Strategy		↑
Move Destination Directory		↑
Move Conflict Strategy	<input type="checkbox"/> Set empty string	↑
Log level when file not found	<div>Cancel Ok</div>	↑
Log level when permission denied	<div>ⓘ ERROR</div>	↑

About

PROCESSOR ID
c0745798-2968-4b02-a4fe-52e27999585a

PROCESSOR TYPE
FetchFile 1.3.1

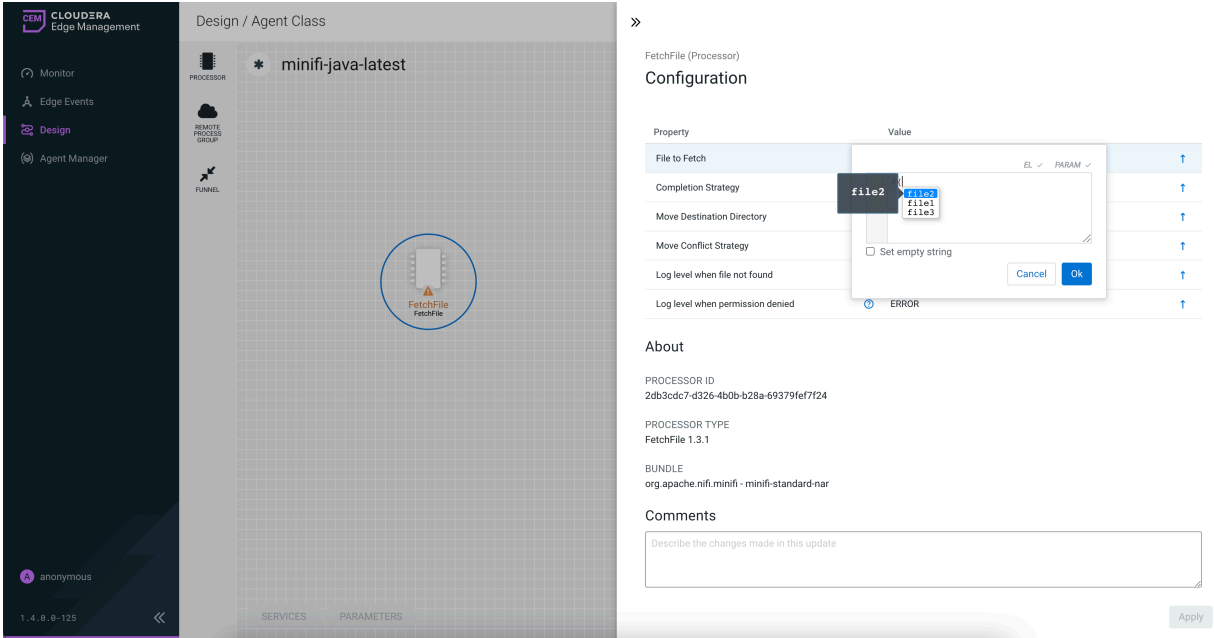
BUNDLE
org.apache.nifi.minifi - minifi-standard-nar

Comments

Describe the changes made in this update

Apply

3. If you know the parameter you want to reference, enter its name. Alternatively, select the keystroke control+space to show the list of available parameters:



4. Complete the entry with a closing curly brace } and select Ok.

>>

FetchFile (Processor)

Configuration

Property	Value	
File to Fetch	<div>EL ✓ PARAM ✓ 1 #file1-paramter</div>	↑
Completion Strategy		↑
Move Destination Directory		↑
Move Conflict Strategy		↑
Log level when file not found		↑
Log level when permission denied	ERROR	↑

About

PROCESSOR ID
c0745798-2968-4b02-a4fe-52e27999585a

PROCESSOR TYPE
FetchFile 1.3.1

BUNDLE
org.apache.nifi.minifi - minifi-standard-nar

Comments

Describe the changes made in this update

Apply

5. Click Apply to save the changes.

Help text describing this process appears when you hover over the Expression Language and Parameters eligibility indicators.

>>

FetchFile (Processor)

Configuration

Property		
File to Fetch	EL ✓ PARAM ✓	↑
Completion Strategy		↑
Move Destination Directory		↑
Move Conflict Strategy		↑
Log level when file not found		↑
Log level when permission denied	ⓘ ERROR	↑

✓ Expression Language (EL) supported

After beginning with the start delimiter `${` use `Ctrl+Space` to see a list of available functions.

✓ Parameters (PARAM) supported

After beginning with the start delimiter `#` use `Ctrl+Space` to see a list of available parameters.

1 `#file1-paramter`

☐ Set empty string

Cancel

Ok

About

PROCESSOR ID

c0745798-2968-4b02-a4fe-52e27999585a

PROCESSOR TYPE

FetchFile 1.3.1

BUNDLE

org.apache.nifi.minifi - minifi-standard-nar

Comments


Describe the changes made in this update

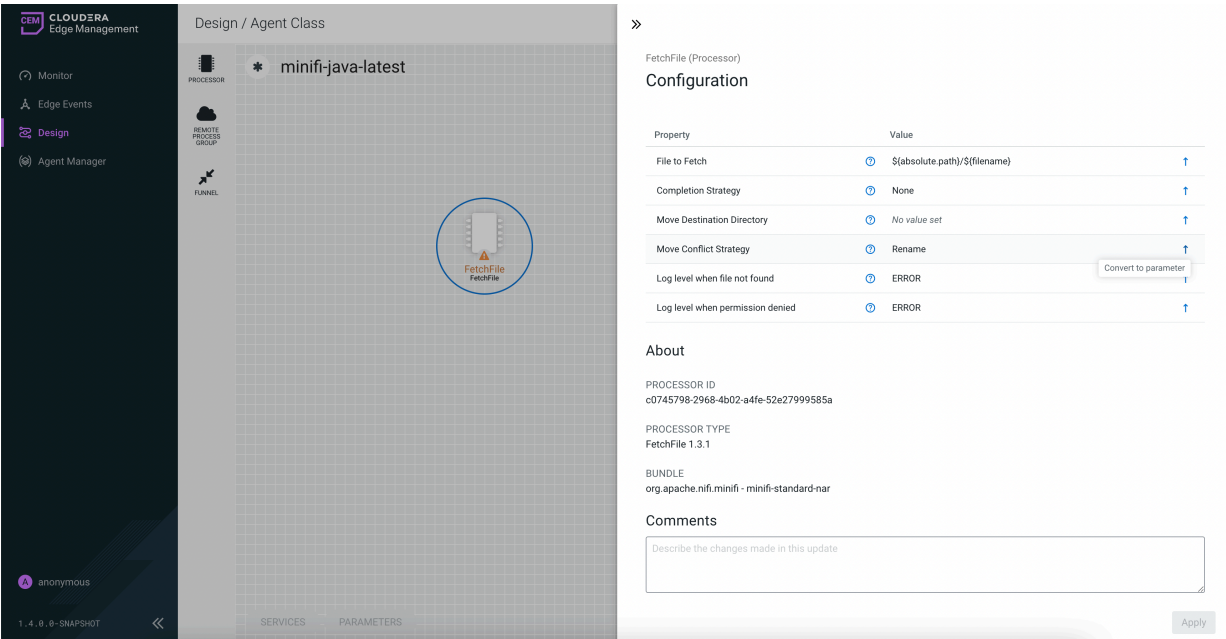
Apply

Converting property values to parameters

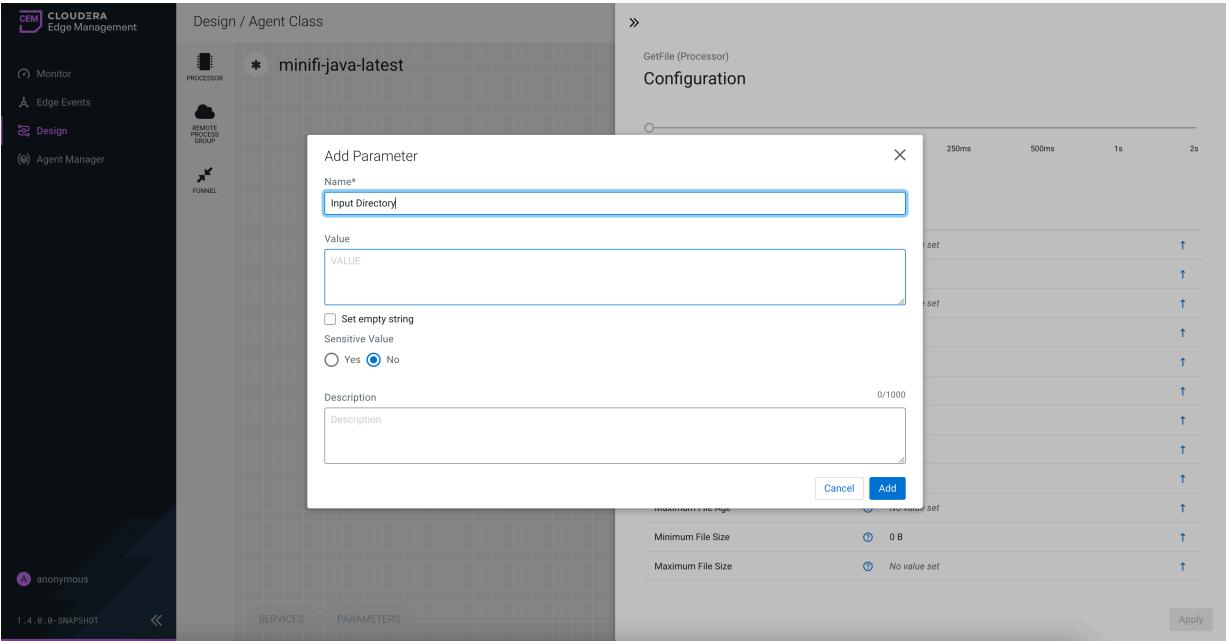
Parameters can be created during process or service configuration. Instead of entering a property value, you can convert the property value to a parameter.

1.

To create a parameter for a property value, select the Convert to parameter icon () in the row of that property.

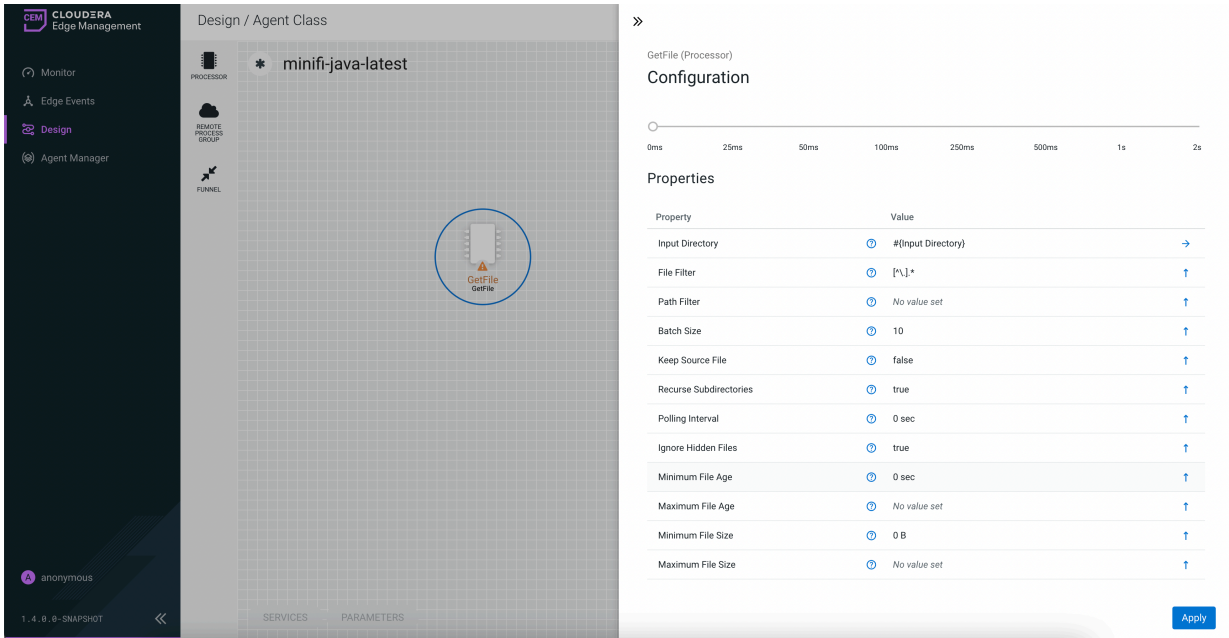


The Create Parameter dialog appears as shown in the following image:



2. Configure the parameter properties.

3. Click Add to add the parameter. The property references the new parameter with the proper syntax applied automatically.

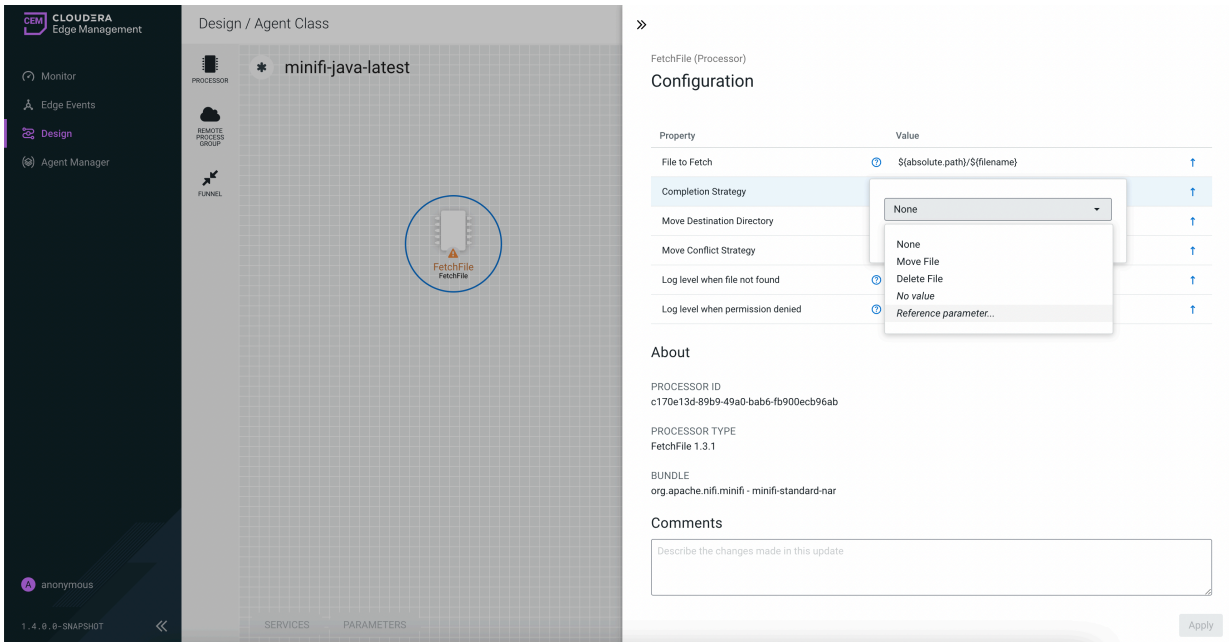


4. Click Apply to save the changes.

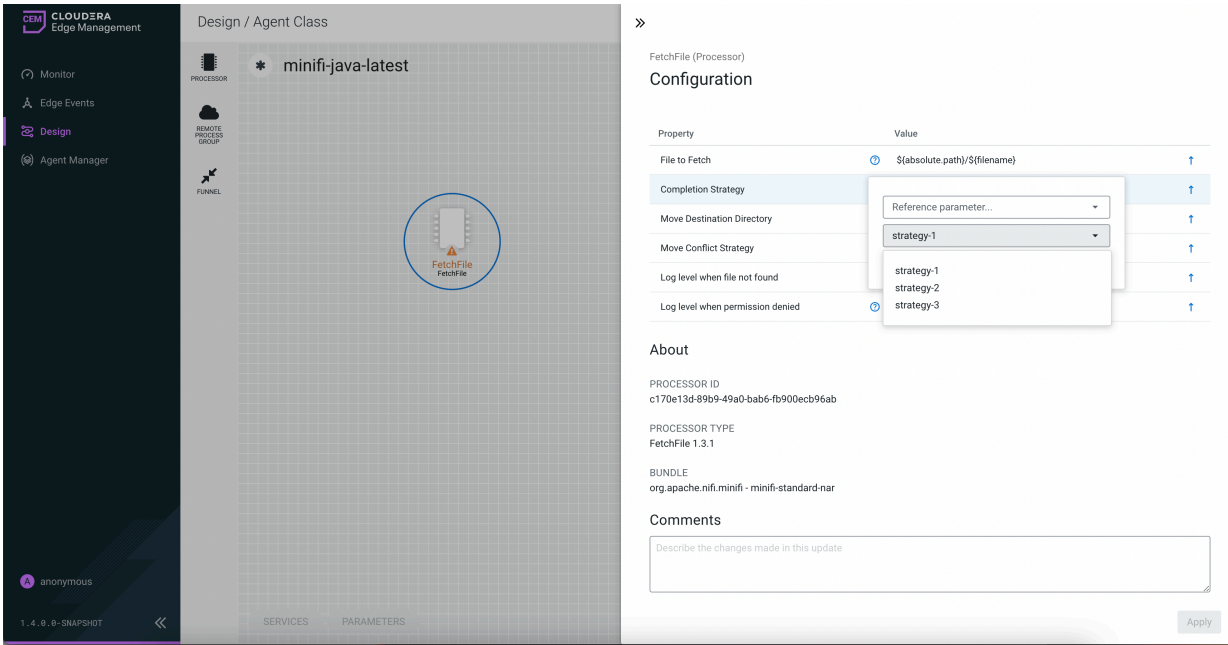
Converting selectable property values to parameters

Property values that are selectable can also reference parameters.

1. Select the property value drop-down. The option Reference parameter... is available for eligible properties.



2. Select Reference parameter... option. A list of parameters to choose appears.

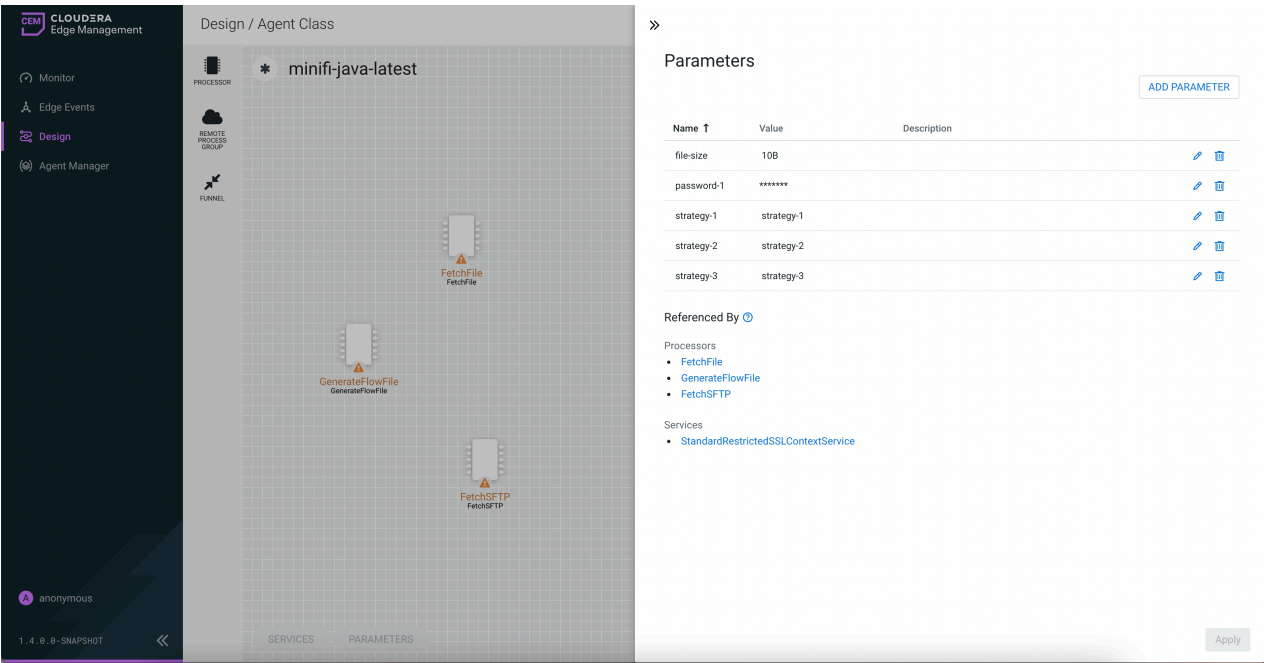


3. Select a parameter and click OK.

4. Click Apply to save the changes.

Referenced parameters

When parameters have been referenced in the flow, the Parameters window lists all the components that use them in the Referenced By section.

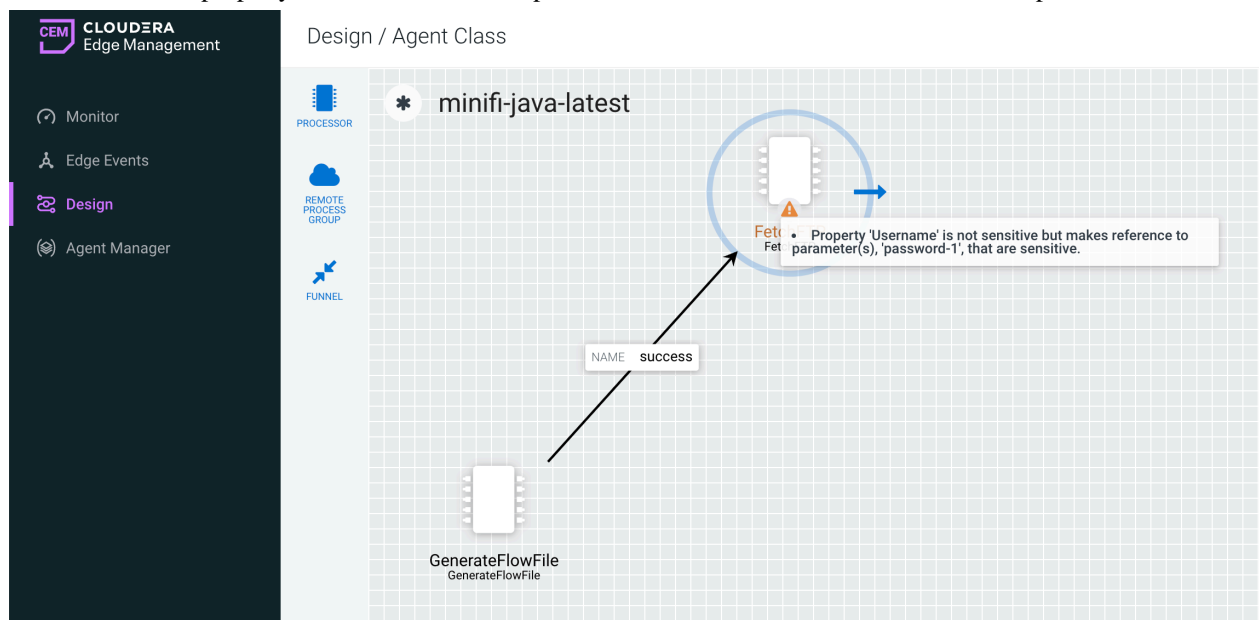


Select a specific parameter to see the processors and services that reference just that parameter.

Using parameters with sensitive properties

Non-sensitive properties should only reference non-sensitive parameters. Sensitive properties should only reference sensitive parameters. This is important when publishing versioned flows. The value of the sensitive parameter itself is not sent to the flow registry, but only the fact that the property references the sensitive parameter.

If a non-sensitive property references a sensitive parameter, or vice-versa, the UI marks the component as invalid:



Setting agent parameter values

Learn about parameter context and how to set agent parameter values in Cloudera Edge Management (CEM).

Any collection of replacement key-value pairs is known as a parameter context. Parameters never exist on their own. They only exist in a parameter context. Edge Flow Manager (EFM) allows setting parameter contexts at the following levels:

Levels	Description
Flow	The flow level parameter context acts as a default for the entire flow. If there are no agent class or agent level parameter context mappings set, the flow parameter context is used. You can manage the values for this level in the UI in the Flow Designer by clicking Parameters in the lower left corner. Every time a flow is published, a copy of the current parameter context from the Flow Designer gets copied to the flow level parameter context for the published flow. Therefore, changing the values for parameters in the Flow Designer does not update previously published flows.
Agent Class	An agent class level parameter context, if present, overwrites the flow level context. This is to support multiple versions of a flow definition to an agent class, but allowing agent class configuration values that override whatever is set in the Flow Designer UI as default values. Agent class level parameter contexts are set using the EFM REST API, and can be set once to impact all future flows published to that agent class.
Agent	An agent level parameter context, if present, overwrites the agent class and flow level contexts. This is to support deploying a flow to multiple agents in an agent class but allowing per-agent configuration values. Agent level parameter contexts are set using the EFM REST API, and can be set once to affect all future flows deployed to that agent.

Resolving parameter contexts

When an agent requests a flow from EFM, EFM substitutes parameter values just-in-time for placeholders by resolving parameter values in this hierarchy of contexts. In order to accomplish this, the flow URI that EFM sends to agents as part of a flow update operation looks like the following:

```
GET /efm/api/flows/{flowId}?aid={agentId}
```

When a flow is fetched, the agent, agent class, and flow level parameter contexts are retrieved and applied to the flow in the hierarchy listed above.

REST API endpoints for parameters

For information about creating parameter contexts and mapping them to flows, agent classes, and agents, see the following:

- Parameters section in *Cloudera Edge Flow Manager REST API*.
- Parameter Mappings section in *Cloudera Edge Flow Manager REST API*.

For tutorials of specific use cases that leverage this feature, including examples interacting with the REST API using curl, see *Using Agent Parameters*.

Related Information

[REST API Reference](#)

[Using agent parameters](#)

Building a dataflow in CEM

You can build an automated dataflow using the Cloudera Edge Management (CEM) UI. Simply drag components from the toolbar to the canvas, configure the components to meet specific needs, and connect the components together.

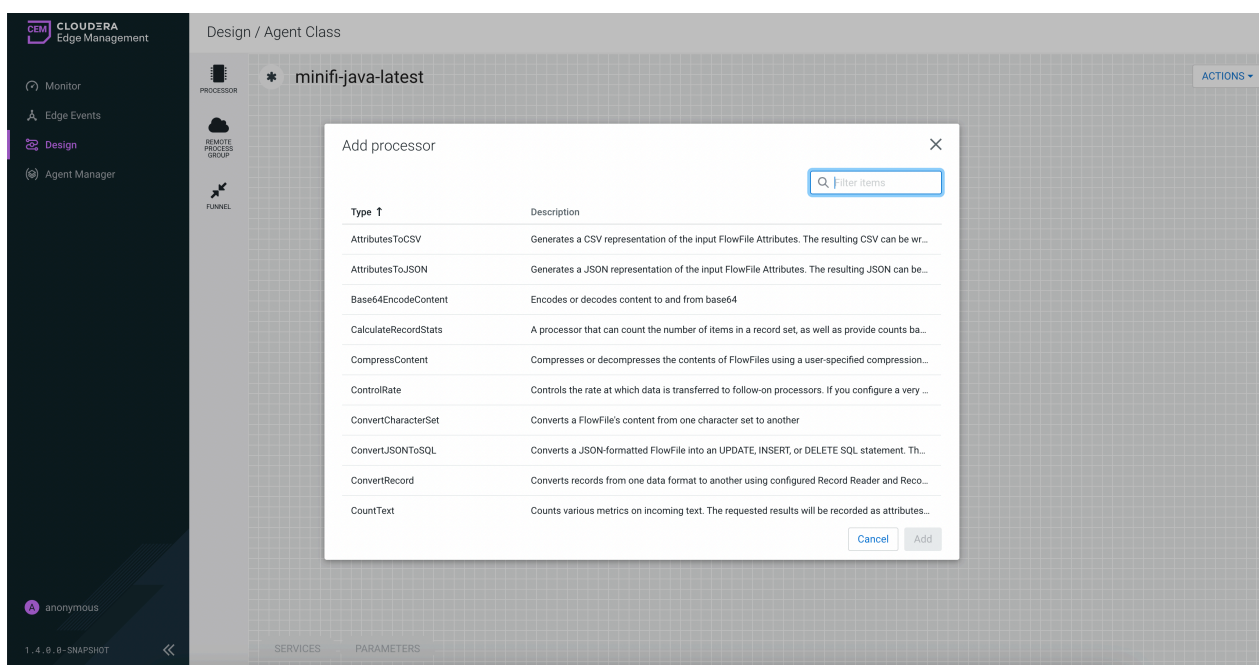
For additional information about flow creation and related concepts, check out the video on the Cloudera Edge Management YouTube playlist:

Adding components to the canvas in CEM

Learn how to add each of the components available in the Components Toolbar in the Cloudera Edge Management (CEM) UI. You can add processors, remote process groups, and funnels.

Processor

The processor is the most commonly used component, as it is responsible for data ingress, egress, routing, and manipulating. There are many different types of processors. When you drag a processor onto the canvas, the Add Processor dialog appears, as shown in the following image, which allows you to choose which type of processor to use:

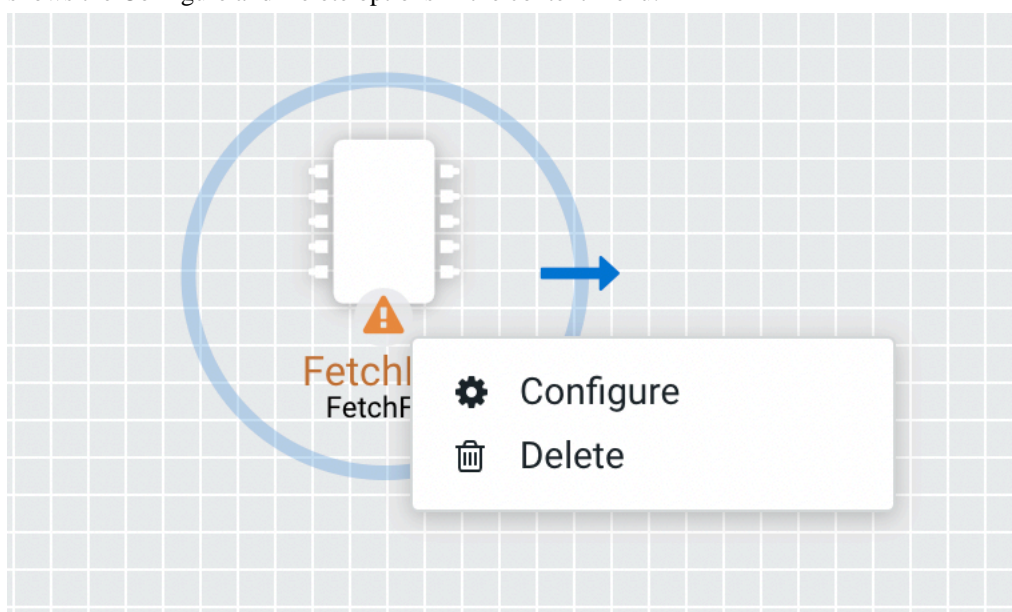


You can filter the list based on the processor type by using the Filter items field at top-right corner of the Add Processor dialog. After selecting a processor, you can click the Add button to add the selected processor to the canvas at the location that it was dropped. Alternatively, you can double-click on a processor type.



Note: For any component added to the canvas, it is possible to select it with the mouse and move it anywhere on the canvas. Also, it is possible to move all items at once by clicking and dragging the mouse on the canvas.

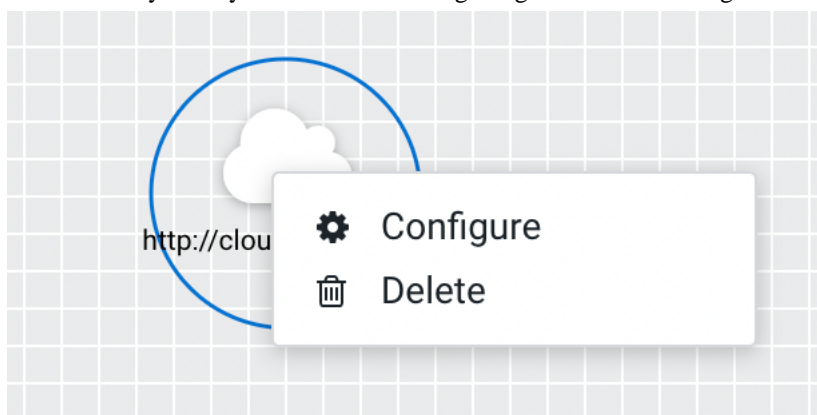
After you drag a processor onto the canvas, you can configure properties of the processor, parameterize processor property values, or delete the processor. To configure properties, double-click on the processor, or right-click on the processor and select Configure from the context menu. To delete a processor, right-click on the processor and select Delete from the context menu, or highlight the processor and select DELETE on your keyboard. The following image shows the Configure and Delete options in the context menu:



Remote process group

A Remote Process Group (RPG) references a remote instance of NiFi. When you drag an RPG onto the canvas, rather than being prompted for a name, you are prompted for the URL of the remote NiFi instance. If the remote NiFi is clustered, you need to provide at least one URL of any NiFi instance in that cluster. When data is transferred from an RPG running in MiNiFi, the RPG first connects to the remote instance whose URL is configured to determine which nodes are in the cluster and how busy each node is. This information is then used to load balance the data that is pushed to each node. The remote instances are then interrogated periodically to determine information about any nodes that are dropped from or added to the cluster and to recalculate the load balancing based on the load of each node. If the cluster node specified in the URL is down, the RPG cannot establish a connection with the cluster. To mitigate this scenario, you can enter multiple URLs, allowing the RPG to establish a connection with more than one node.

After you drag an RPG onto the canvas, you can configure settings of the RPG or delete the RPG. To configure settings, double-click on the RPG, or right-click on the RPG and select Configure from the context menu. To delete an RPG, right-click on the processor and select Delete from the context menu, or highlight the RPG and select DELETE on your keyboard. The following image shows the Configure and Delete options in the context menu:



Funnel

Funnels are used to combine data from many connections into a single connection. If many connections are created with the same destination, the canvas can become cluttered if those connections have to span a large space. By funneling these connections into a single connection, that single connection can then be drawn to span that large space instead.

To delete a funnel, right-click on the funnel and select Delete from the context menu, or highlight the funnel and select DELETE on your keyboard.

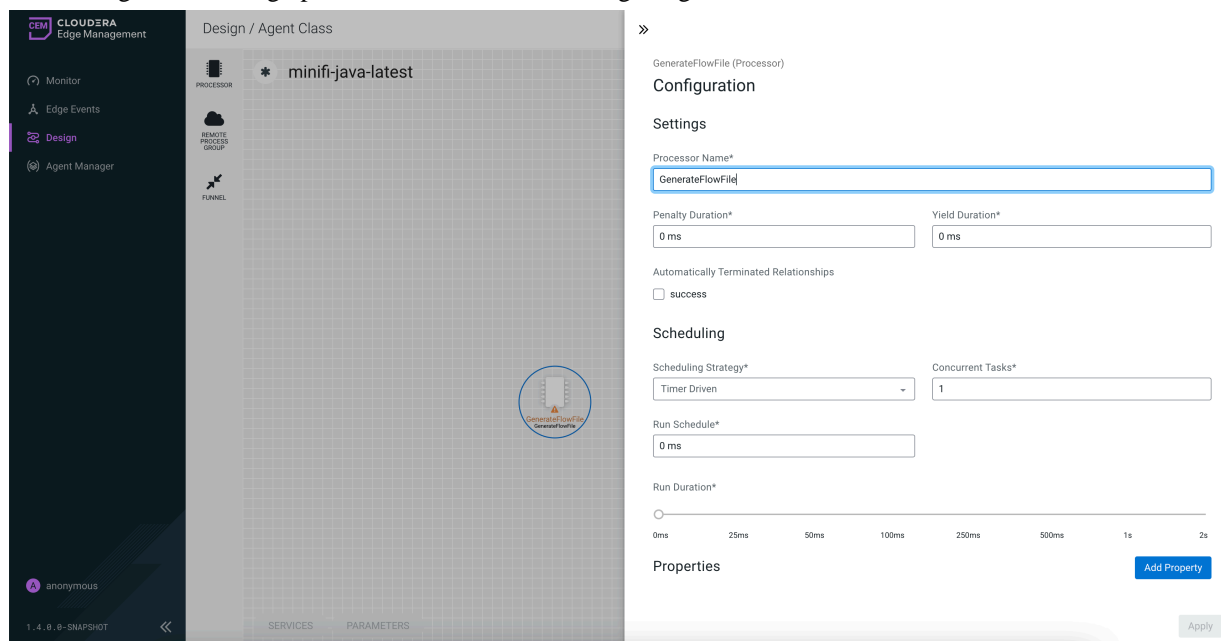
Configuring a processor in CEM

Learn how to configure a processor using the Cloudera Edge Management (CEM) UI.

Procedure

1. To configure a processor, right-click on the processor and select the Configure option.
Alternatively, just double-click on the processor.

The Configuration dialog opens as shown in the following image:



The Configuration dialog contains the following sections:

- Settings. The Settings section contains the following configuration items:

Properties	Description
Processor Name	Allows you to change the name of the processor. The name of a processor by default is the same as the processor type.
Penalty Duration	The amount of time used when a processor penalizes a FlowFile. During the normal course of processing a piece of data (a FlowFile), an event might occur that indicates that the data cannot be processed at this time but the data might be processable at a later time. When this occurs, the processor might choose to penalize the FlowFile. This prevents the FlowFile from being processed for some period of time. For example, if the processor needs to push the data to a remote service, but the remote service already has a file with the same name as the filename that the processor is specifying, the processor might penalize the FlowFile. The penalty duration allows you to specify how long the FlowFile must be penalized. The default value is 30,000 milliseconds.
Yield Duration	When a processor yields, the amount of time that elapses before the processor is re-scheduled is the yield duration. A processor might determine that some situation exists such that the processor can no longer make any progress, regardless of the data that it is processing. For example, if a processor needs to push data to a remote service and that service is not responding, the processor cannot make any progress. As a result, the processor must yield, which prevents the processor from being scheduled to run for some period of time. The default value is 1,000 milliseconds.

Properties	Description
Automatically Terminated Relationships	Each of the relationships that is defined by the processor is listed here. In order for a processor to be considered valid, each relationship defined by the processor must be either connected to a downstream component or auto-terminated. If a relationship is auto-terminated, any FlowFile that is routed to that relationship is removed from the flow and its processing is considered as complete.

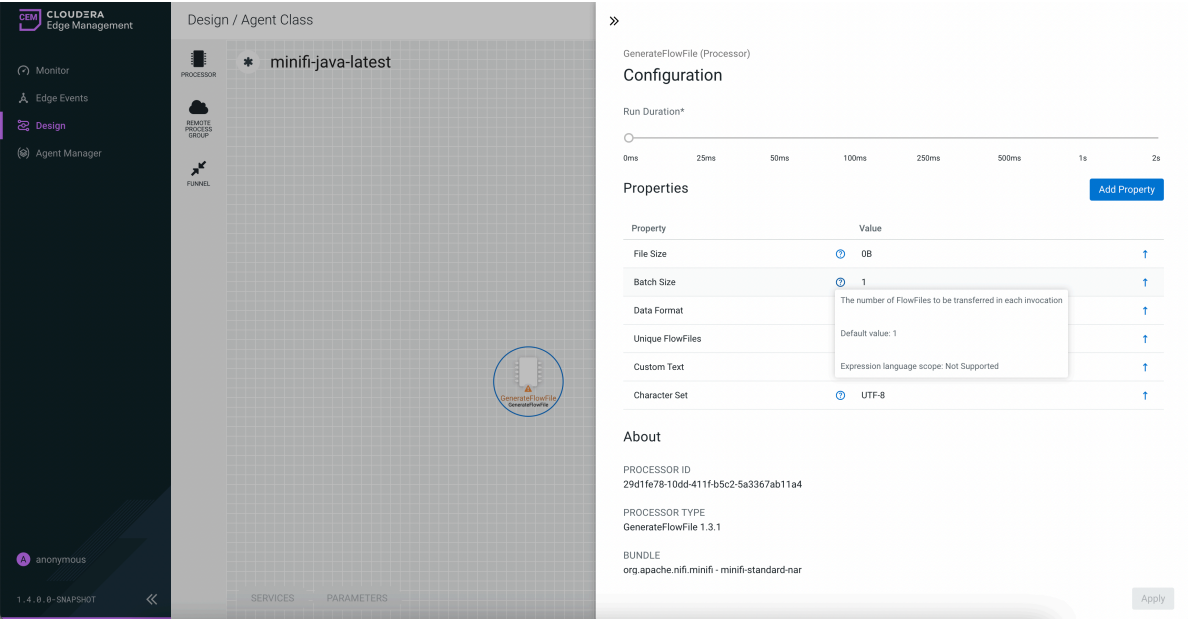
- **Scheduling.** The Scheduling section contains the following configuration items:

Properties	Description
Scheduling Strategy	There are two options for scheduling components: <ul style="list-style-type: none"> • Timer Driven. This is the default mode. The processor is scheduled to run on a regular interval. The interval at which the processor runs is defined by the Run Schedule option (see below). • Event Driven. When this mode is selected, the processor is triggered to run by an event, and that event occurs when FlowFiles enter connections feeding this processor. This mode is currently considered experimental and is not supported by all processors. When this mode is selected, the Run Schedule option is not configurable, as the processor is not triggered to run periodically but as the result of an event.
Concurrent Tasks	This controls how many threads the processor uses or how many FlowFiles must be processed by this processor at the same time. Increasing this value allows the processor to handle more data in the same amount of time. However, it does this by using system resources that then are not usable by other processors. This essentially provides a relative weighing of processors. For example, it controls how much resources of the system must be allocated to this processor instead of other processors. This field is available for most processors. There are, however, some types of processors that can only be scheduled with a single concurrent task.
Run Schedule	This dictates how often the processor must be scheduled to run. The valid values for this field depend on the selected scheduling strategy (see above). When you select the Event Driven scheduling strategy, this field is not available. When you select the Timer Driven scheduling strategy, this value is a time duration specified by a number followed by a time unit, for example, 1 second or 5 mins. A value of 0 second means that the processor must run as often as possible as long as it has data to process. This is true for any time duration of 0, regardless of the time unit (for example, 0 sec, 0 mins, 0 days).
Run Duration	This slider controls how long the processor must be scheduled to run each time it is triggered. When a processor finishes running, it must update the repository in order to transfer the FlowFiles to the next connection. Updating the repository is expensive, so the more work that can be done at once before updating the repository, the more work the processor can handle (higher throughput). However, this means that the next processor cannot start processing those FlowFiles until the previous process updates this repository. As a result, the latency (the time required to process the FlowFile from beginning to end) becomes longer. As a result, the slider provides a spectrum from which you can choose to favor Lower Latency or Higher Throughput.

- **Properties.** The Properties section provides a mechanism to configure processor-specific behavior. There are no default properties. Each type of processor must define which properties make sense for its use case.

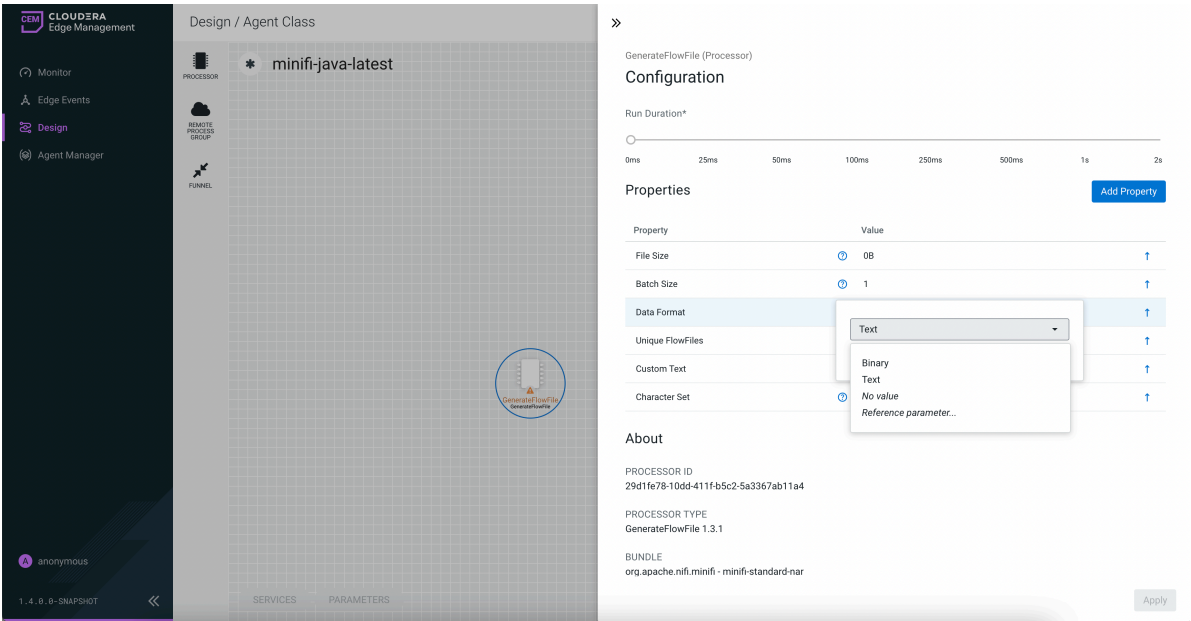
A GenerateFlowFile processor, by default, has four properties including Batch Size, Data Format, File Size, and Unique FlowFiles. Next to the name of each property, there appears a small question-mark symbol (•) indicating that additional information is available. Hovering over this symbol with the mouse provides

additional details about the property, the default value and whether Expression Language is supported. Here is an example of GenerateFlowFile processor with additional information for the Batch Size property:

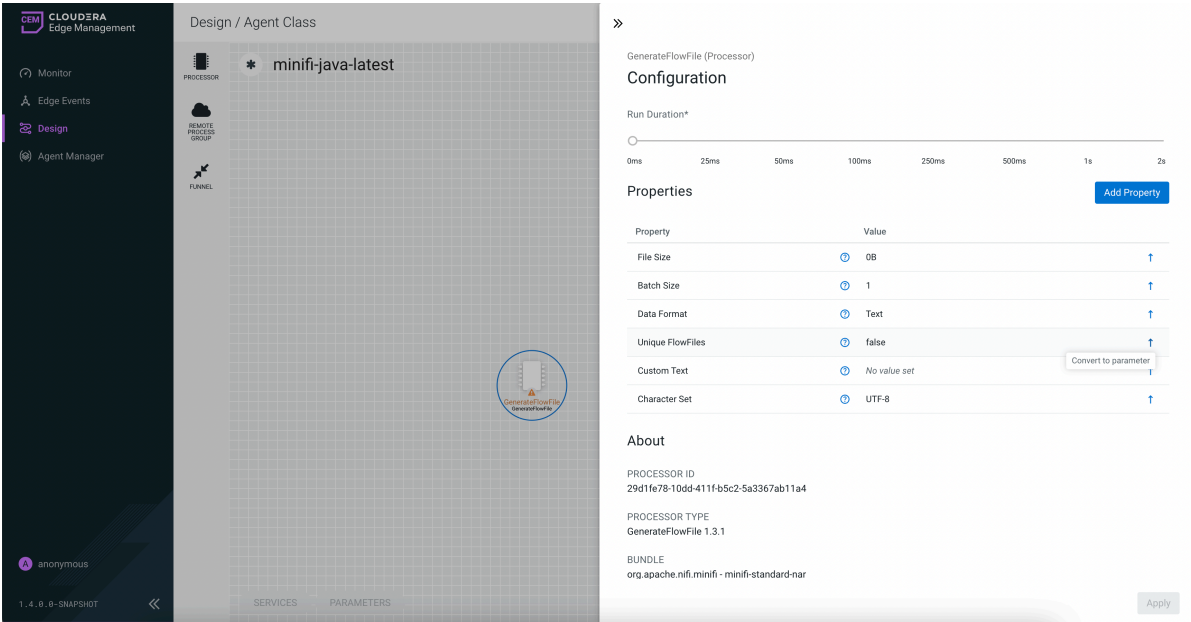


Clicking on the value for the property allows you to change the value. Depending on the values that are allowed for the property, you are either provided a drop-down from which to choose a value, or a text area

to type a value. Here is an example of GenerateFlowFile processor with the drop-down for the Data Format property:

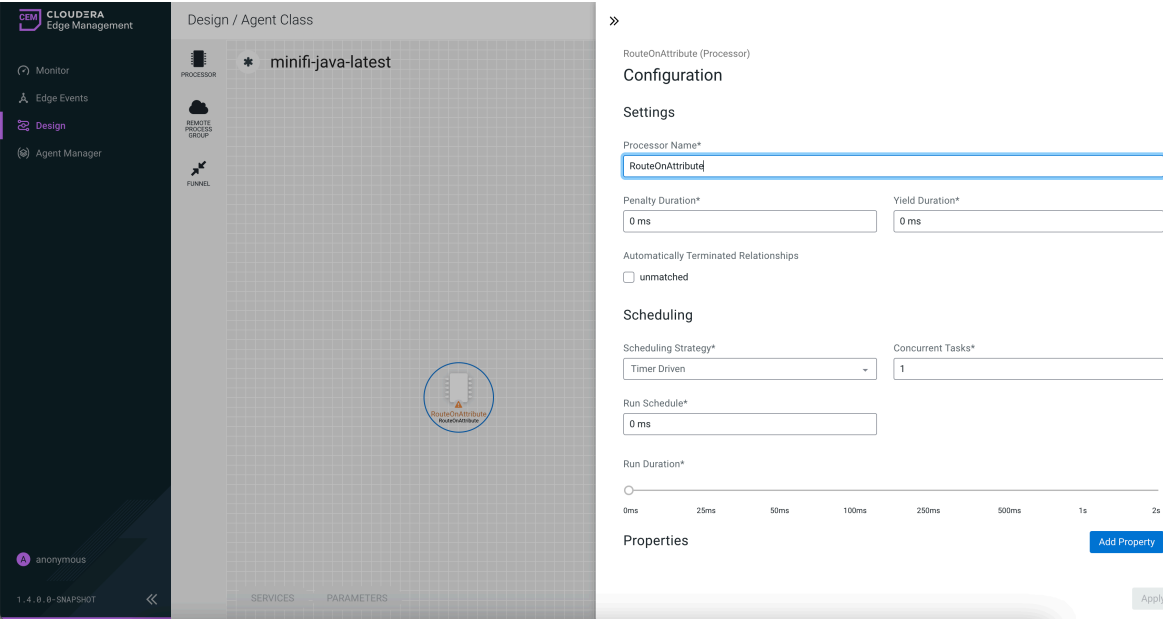



Each of the properties has an arrow in the row showing that they can be converted to parameters. The following image shows the Convert to parameter option for the Unique FlowFiles property:



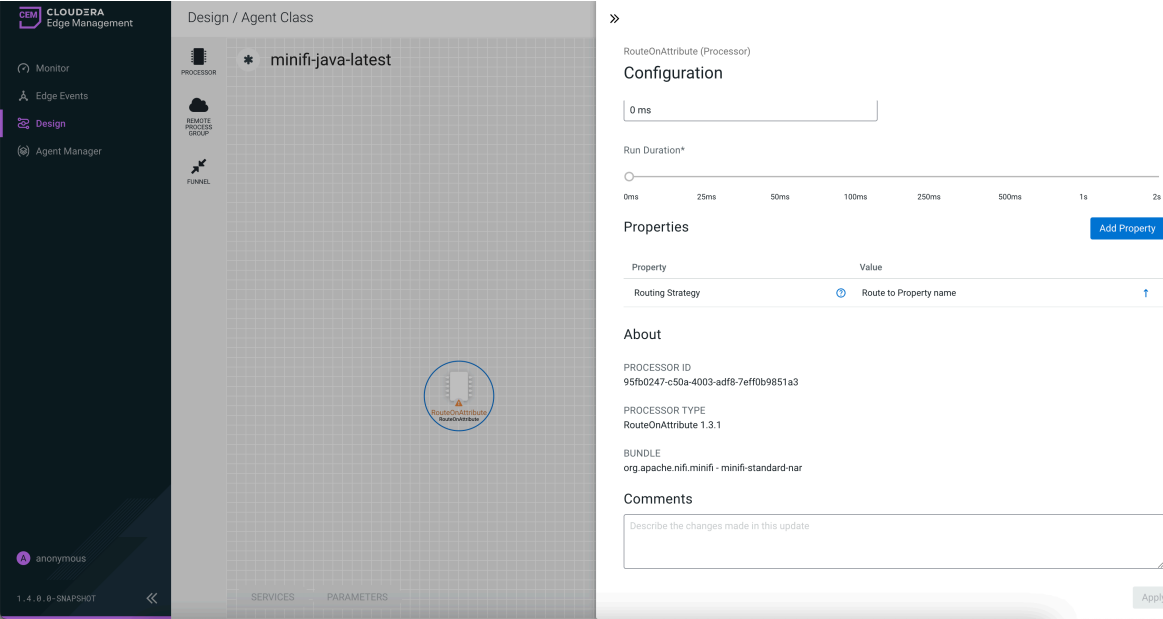
For some processors, there appears an Add Property button, beside the Properties section, for adding a user-defined property. When you click this button, a dialog opens, which allows you to enter the name and value of a new property. Not all processors allow user-defined properties. The RouteOnAttribute processor, however,

allows user-defined properties. In fact, this Processor will not be valid until you add a property. The following image shows the Add Property button in the Configuration dialog of the RouteOnAttribute processor:



Note: After a user-defined property has been added, a trash icon () appears on the right-hand side of that row. You can remove the user-defined property from the processor by clicking the trash icon.

- About. The About section provides the Processor ID, Processor Type, and Bundle details of the processor, as shown in the following image:



- Comments. This tab simply provides an area for you to include whatever comments are appropriate for this component.
2. After you configure a processor, click the Apply button to apply the changes.

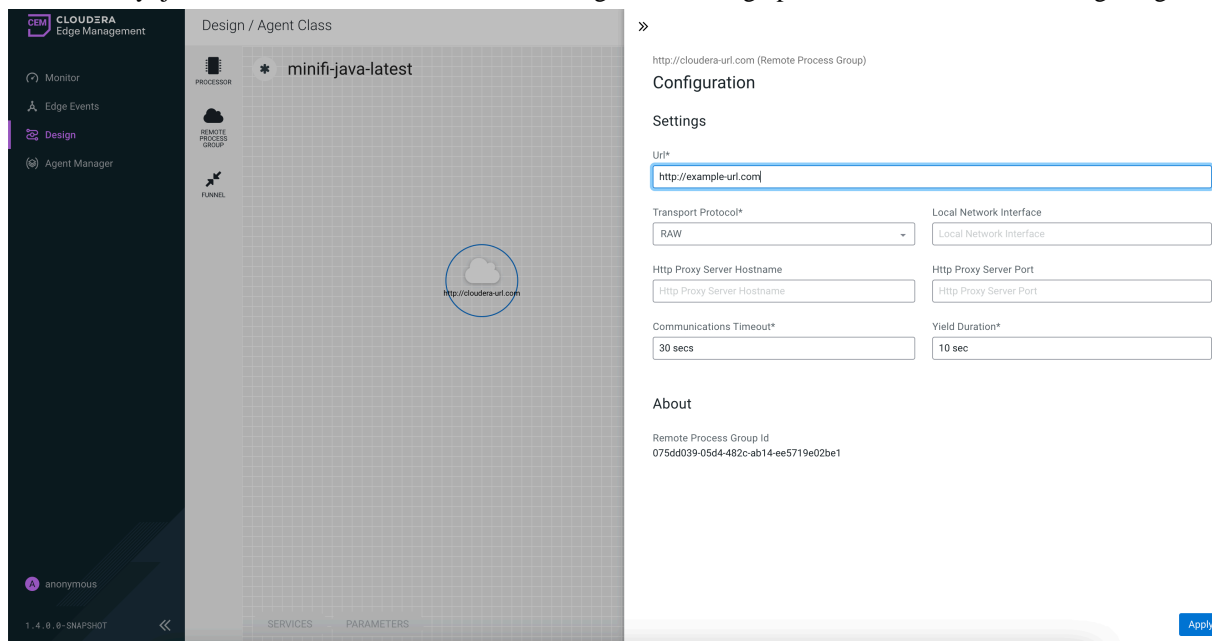
Configuring a remote process group in CEM

Learn how to configure a remote process group using the Cloudera Edge Management (CEM) UI.

Procedure

1. To configure an RPG, right-click on the RPG and select the Configure option.

Alternatively, just double-click on the RPG. The Configuration dialog opens as shown in the following image:



The Configuration dialog contains the following two sections:

- Settings
 - About. The About section provides the Remote Process Group ID.
2. Configure the following properties in Settings section:

Properties	Description
URL	Allows you to change the URL of the RPG.
Transport Protocol	There are two options for transport protocol: <ul style="list-style-type: none"> • RAW. This is the default protocol which uses raw socket communication by using a dedicated port. • HTTP. The HTTP transport protocol is useful if the remote NiFi instance is in a restricted network that only allows access through HTTP(S) protocol or only accessible from a specific HTTP Proxy server.
Local Network Interface	In some cases, it might be desirable to prefer one network interface over another. For example, if a wired interface and a wireless interface exist, the wired interface might be preferred. This can be configured by specifying the name of the network interface to use in this box. If the value entered is not valid, the Remote Process Group will not be valid and will not communicate with other NiFi instances until this is resolved.
HTTP Proxy Server Hostname	Specify the host name of the proxy server, if you select HTTP transport protocol.
HTTP Proxy Server Port	Specify the port number of the proxy server, if you select HTTP transport protocol.

Properties	Description
Communications Timeout	When communication with the RPG takes longer than this amount of time, it will timeout. The default value is 30 seconds.
Yield Duration	When communication with the RPG fails, it will not be scheduled again until this amount of time elapses. The default value is 10 seconds.

- After you configure an RPG, apply the changes by clicking the Apply button.

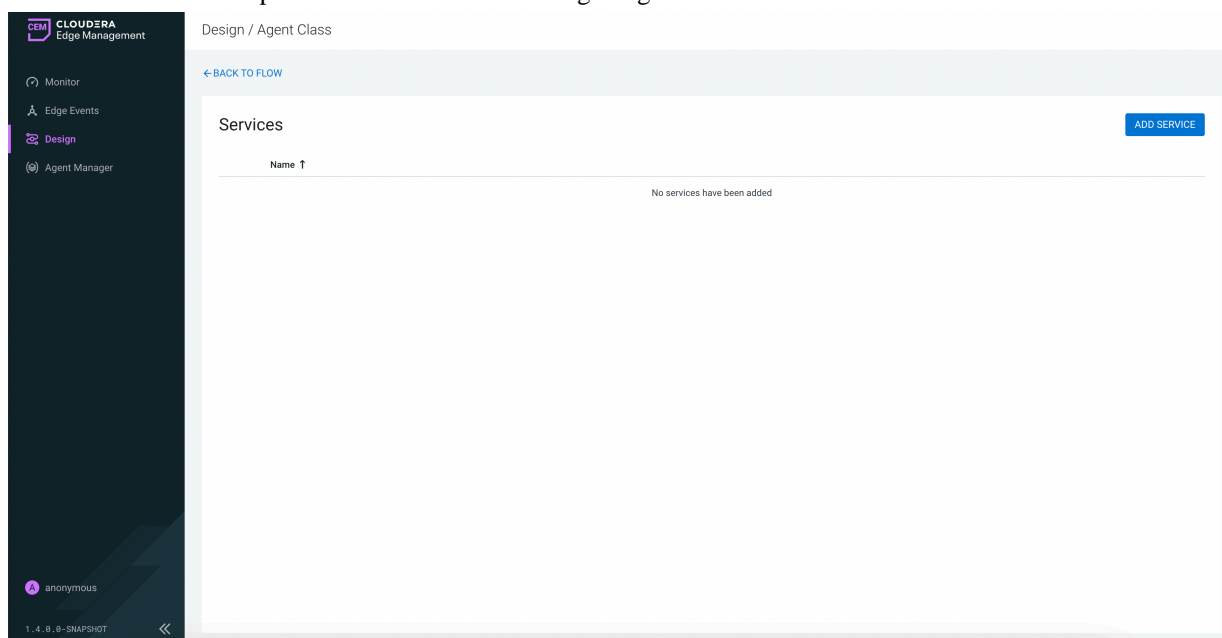
Adding services in CEM

Services are shared services that can be used by processors and other services to utilize for configuration or task execution. Learn how to add services using the Cloudera Edge Management (CEM) UI.

Procedure

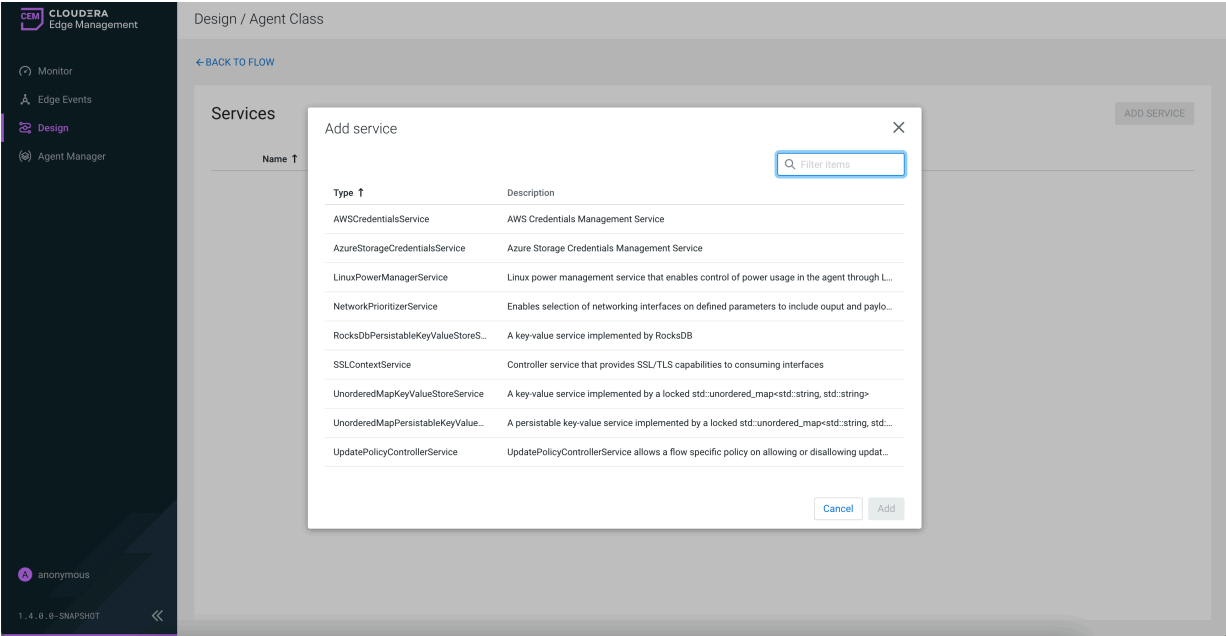
- To add a service, click the SERVICES button at the bottom-left corner of the canvas, or simply right-click on the canvas and select Services.

The Services window opens as shown in the following image:




- 2. Click the ADD SERVICE button.

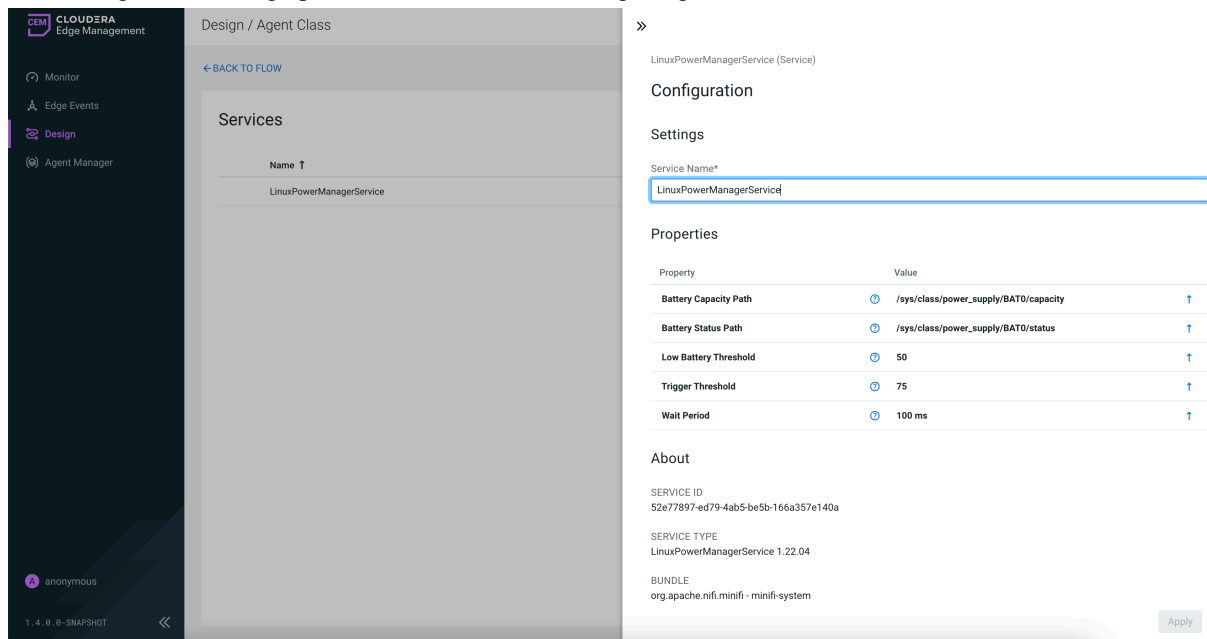
The Add Service dialog opens. It provides a list of the available services as shown in the following image:



- 3. Select the service you want to add and click Add, or simply double-click on the name of the service to add it. You can also use the Filter items field at the top-right corner of the window to search for the desired service by name.

4.

After you add a service, configure it by clicking the Configure icon () in the far-right column. The Configuration dialog opens as shown in the following image:




The Configuration dialog contains the following sections:

- **Settings.** The Settings section provides a place for you to give the service a unique name. The name of a service by default is the same as the service type.
- **Properties.** The Properties section lists the various properties that apply to the particular service. You can hover over the question mark icons with the mouse to see more information about each property.
- **About.** The About section provides the Service ID, Service Type, and Bundle details of the service.
- **Comments.** The Comments section is just an open-text field, where you can include comments about the service.

5. After you configure a service, click the Apply button to apply the configuration



If you want to delete a service, click the trash icon () in the far-right column. To return to the canvas, click the BACK TO FLOW link.

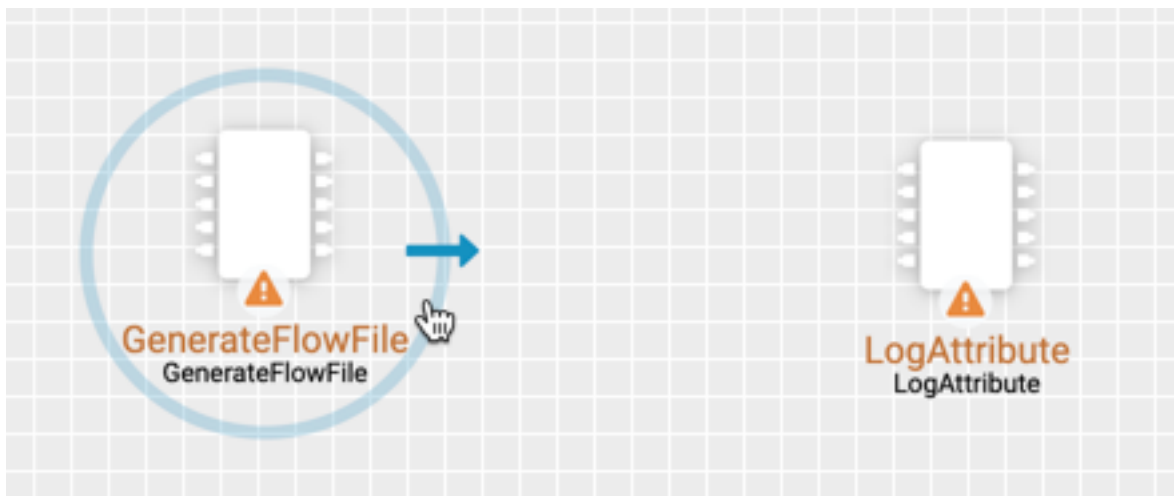
Connecting components in CEM

After you add processors and other components to the canvas of the Cloudera Edge Management (CEM) UI and configure them, the next step is to connect them to one another. This is accomplished by creating a connection between each component.

Procedure

1. Hover the mouse over a component.

An arrow appears as shown in the following image:



2. Drag the arrow from one component to another until the second component is highlighted, then release the mouse.

A Create Connection dialog appears as shown in the following image:



The dialog allows you to choose the Source Relationship that must be included in the connection. At least one relationship must be selected. If only one relationship is available, it is automatically selected.

3. Select Add to create the connection.



Note: It is possible to draw a connection so that it loops back on the same processor. This can be useful if you want the processor to try to re-process flow files if the flow files go down a failure relationship. To create this type of looping connection, simply drag the connection arrow away and then back to the same processor until it is highlighted. Then release the mouse and the same Create Connection dialog, referenced earlier, appears.

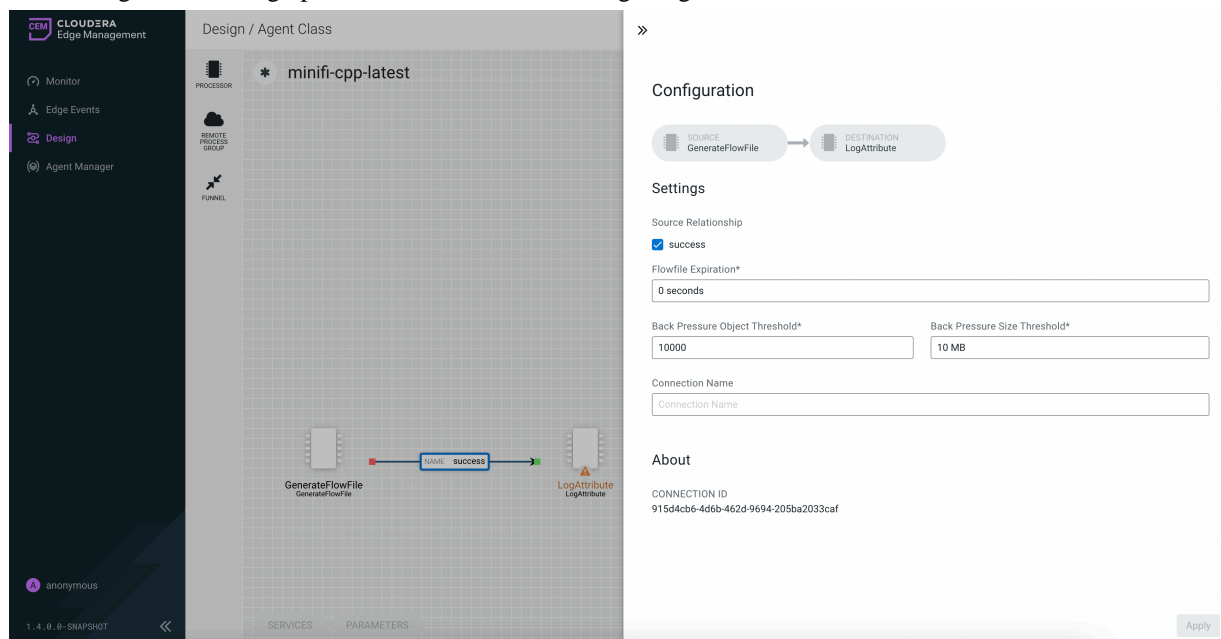
Configuring a connection in CEM

After you create a connection, you can change the configuration properties of the connection or move the connection using the Cloudera Edge Management (CEM) UI.

Procedure

1. To change the configuration of a connection, right-click on the connection and select the Configure option, or double-click on the connection.

The Configuration dialog opens as shown in the following image:



The Configuration dialog contains the following two sections:

- Settings
 - About. The About section provides the Connection ID.
2. Configure the following properties in the Settings section:

Property	Description
Source Relationship	Allows you to change the Source Relationships of the connection.
Flowfile Expiration	FlowFile expiration is a concept by which data that cannot be processed in a timely fashion can be automatically removed from the flow. This is useful, for example, when the volume of data is expected to exceed the volume that can be sent to a remote site. The expiration period is based on the time that the data entered the MiNiFi instance. In other words, if the file expiration on a given connection is set to 1 hour, and a file that has been in the MiNiFi instance for one hour reaches that connection, it will expire. The default value is 60 seconds. A value of 0 seconds indicates that the data will never expire.
Back Pressure Object Threshold	This is the number of FlowFiles that can be in the queue before back pressure is applied. The default value is 0.
Back Pressure Size Threshold	This specifies the maximum amount of data (in size) that must be queued up before applying back pressure. The default value is 10,000 Bytes.
Connection Name	This field allows you to change the name of the connection. It is blank by default.

3. After you configure a connection, click the Apply button to apply the changes.

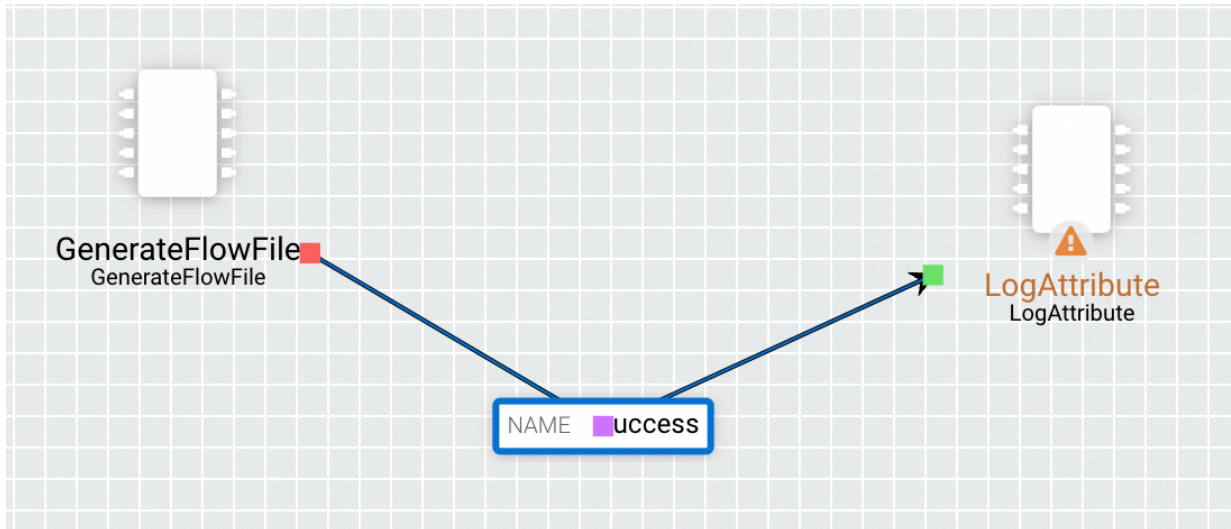
Bending connections in CEM

Learn how to bend a connection using the Cloudera Edge Management (CEM) UI.

Procedure

1. To add a bend point (or elbow) to an existing connection, simply double-click on the connection in the spot where you want the bend point to be.
2. Use the mouse to grab the bend point and drag it so that the connection is bent in the desired way.

The following image shows a bend point in the connection between GenerateFlowFile and LogAttribute processors:



You can add as many bend points as you want. You can also use the mouse to drag and move the label on the connection to any existing bend point. To remove a bend point, simply double-click on it again.

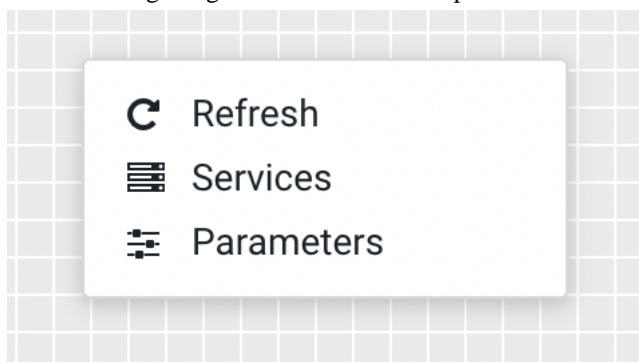
Refreshing a dataflow in CEM

Learn how to refresh a dataflow using the Cloudera Edge Management (CEM) UI.

Procedure

To refresh a flow that you monitor, right-click on the canvas and select Refresh.

The following image shows the Refresh option:



Publishing a dataflow in CEM

Learn how to publish a dataflow using the Cloudera Edge Management (CEM) UI.

About this task



Note: Publishing is an asynchronous process and agents update their flow as they periodically heartbeat to the CEM server.

Procedure

1. To publish a dataflow and make it available to all agents associated with its class, select Publish from the ACTIONS drop-down menu on the canvas.

The Publish Flow dialog appears as shown in the following image:

Publish Flow



Publishing this flow will make it available to all agents associated with minifi-cpp-latest.

CHANGE COMMENTS

CHANGE COMMENTS

Cancel Publish

2. Enter comments if desired, and click Publish.

The flow status changes from modified () to current ().

Reverting a dataflow in CEM

Learn how to revert a dataflow using the Cloudera Edge Management (CEM) UI.

Procedure

1. To remove all changes that you made since a flow was published, select Revert to last published from the ACTIONS drop-down menu on the canvas.



The Revert to last published version dialog appears as shown in the following image:

⚠ Revert to last published version

All changes made to this flow since it was published will be lost.

Cancel Revert

2. Select Revert to complete the process.

The flow status changes from modified () to current ().

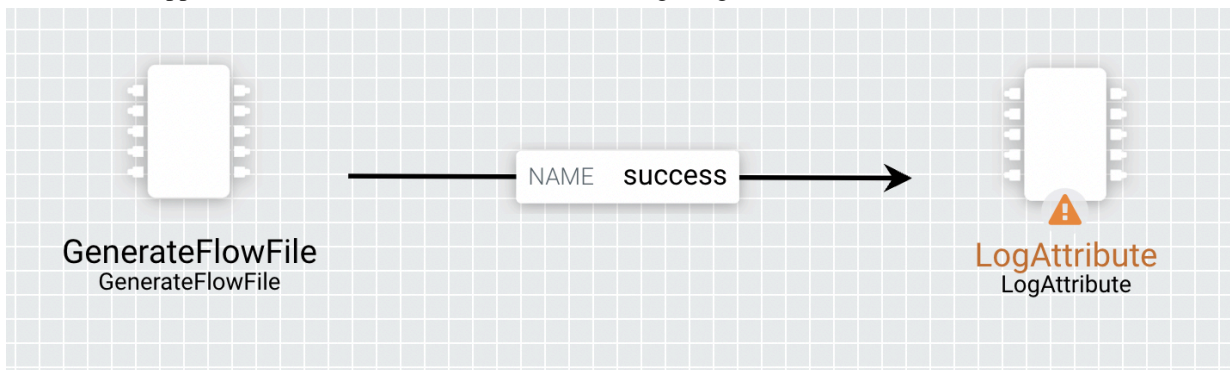
Example dataflow in CEM

The example dataflow consists of just two processors: GenerateFlowFile and LogAttribute. These processors are normally used for testing, but they can also be used to build a ReadyFlow and see Cloudera Edge Management (CEM) in action.

Procedure

1. Drag the GenerateFlowFile and LogAttribute processors to the canvas and connect them (using the guidelines provided above).

The dataflow appears on the canvas as shown in the following image:



2. Configure the GenerateFlowFile processor and click Apply to apply the changes.
 - In the Scheduling section, set Run Schedule to: 5 sec. Note that the GenerateFlowFile processor can create many FlowFiles very quickly. Therefore, setting the Run Schedule is important, so that the flow does not overwhelm the system CEM is running on.
 - In the Properties section, set File Size to 10 KB.

The following image shows the configuration properties of the GenerateFlowFile processor:

Configuration

Settings

Processor Name*
GenerateFlowFile

Penalty Duration* 0 ms Yield Duration* 0 ms

Automatically Terminated Relationships
☐ success

Scheduling

Scheduling Strategy* Timer Driven Concurrent Tasks* 1

Run Schedule* 0 ms

Run Duration* 0 ms 25ms 50ms 100ms 250ms 500ms 1s 2s

Properties

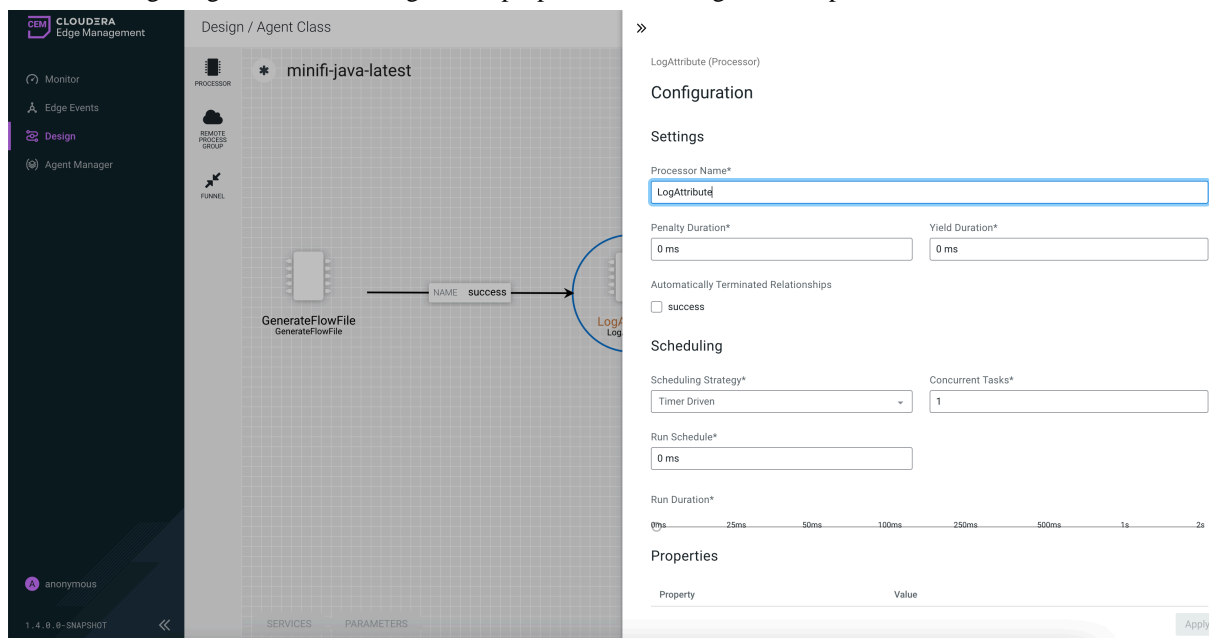
Property	Value

[Add Property](#) [Apply](#)

3. Configure the LogAttribute processor and click Apply to apply the changes.

- In the Settings section, under Automatically Terminated Relationships, select the checkbox next to success. This terminates flow files after the processor successfully processes them.
- In the Properties section, set the Log Payload property to true. This way, the payload of the FlowFile is logged, in addition to its attributes. Otherwise, if set to false, just the attributes are logged.

The following image shows the configuration properties of the LogAttribute processor:



4. Publish the flow by selecting Publish from the ACTIONS drop-down.

Monitoring metrics in CEM with Grafana

Cloudera Edge Management (CEM) can export time series metrics to several metric storage providers. The recommended metrics store service is Prometheus. Prometheus integrates with Grafana for time series metric visualization. With Prometheus and Grafana, you can store and visualize metrics for CEM.

You need to perform the following tasks before you start visualizing CEM metrics with Prometheus and Grafana.

Enabling Prometheus metrics in CEM

Ensure that the following metrics exporting property is enabled in CEM in the `efm.properties` file:

```
management.metrics.export.prometheus.enabled=true
```

You need to customize the following `efm.dashboard.*` properties:

```
efm.dashboard.base-url=http://grafana.example.com:3000
efm.dashboard.url.agentclass=/d/efm-agent-class/?var-agentClass={agentClass}
```

The base-url must reflect the location where you host Grafana. For details, see the *Setting up Grafana* section.

The dashboard URLs must point to the locations where you have set up agent and agent class specific URLs (see below).

Setting up Prometheus

1. Install Prometheus on a host that has network connectivity to CEM. For instructions about how to install Prometheus, see <https://prometheus.io>.
2. Configure your prometheus.yml file to scrape the CEM instance. For example:

Global config

```
global:
  scrape_interval: 1m
  evaluation_interval: 1m
```

The following is a scrape configuration for CEM. Add this to any other scrape configurations you desire. In this example, it is Prometheus.

```
scrape_configs:
- job_name: 'cem-efm'
  metrics_path: '/efm/actuator/prometheus'
  scrape_interval: 15s
  static_configs:
  - targets: ['efm.example.com:10090']
```

For additional scrape configuration properties, such as TLS settings, see the Prometheus configuration guide: https://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config.

3. Verify Prometheus configuration.

Setting up Grafana

1. Install Grafana on a host that has network connectivity to Prometheus. For instructions about how to install Grafana, see <https://grafana.com>.
2. Configure Grafana to use Prometheus as a datasource. This can be done through the Grafana UI or through a data sources.yml provisioning file in conf/provisioning/datasources. For example:

```
# config file version
apiVersion: 1
```

List of datasources to insert or update depending on what is available in the database:

```
apiVersion: 1

datasources:
- name: EFM Prometheus
  type: prometheus
  access: proxy
  orgId: 1
  url: http://prometheus.example.com:9090
  password:
  user:
  database:
  basicAuth: false
  basicAuthUser:
  basicAuthPassword:
  withCredentials: false
  isDefault: true
  jsonData:
    graphiteVersion: "1.1"
    tlsAuth: false
    tlsAuthWithCACert: false
  secureJsonData:
    tlsCACert: "..."
    tlsClientCert: "..."
```

```
tlsClientKey: "..."  
version: 1  
editable: true
```

For more information about configuring a Prometheus datasource in Grafana, see <https://grafana.com/docs/features/datasources/prometheus/>.

3. Download the CEM Grafana dashboard templates.

You can download the Grafana dashboard templates from <https://github.com/cloudera/cloudera-edge-management-grafana-boards>.

4. Import each JSON dashboard through the Grafana UI. Follow the instructions provided in the following Grafana document: https://grafana.com/docs/reference/export_import/#importing-a-dashboard.

Alternatively, you can create a dashboards provider config file in `conf/provisioning/dashboards` to import dashboards.

```
apiVersion: 1  
  
providers:  
- name: 'efm-prometheus'  
  orgId: 1  
  folder: 'EFM Prometheus'  
  type: file  
  disableDeletion: false  
  editable: true  
  options:  
    path: /tmp/dashboards
```

Update the path property as per your requirements and also create the directory in the local filesystem. Place the downloaded dashboard definitions in the created directory and start Grafana.

CEM dashboards should now be available in the Grafana UI.

Navigating to Grafana dashboard in CEM

To navigate to class specific Grafana dashboards, select a class in the Dashboard screen. The Metrics tab appears for the class. Select the View Grafana Dashboard link in the Information section.

For more information about the integration with Prometheus and Grafana, check out the video on the Cloudera Edge Management YouTube playlist: