Cloudera Edge Management 2.4.0

Using Cloudera Edge Management

Date published: 2019-04-15
Date modified: 2026-05-27

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Edge Flow Manager user interface

The Edge Flow Manager user interface (UI) provides a centralized platform to design, deploy, and monitor dataflows and edge agents.

You can use the Edge Flow Manager to:

- Build, edit, publish, and revert dataflows
- Monitor deployments and agent activity
- Manage agents, resources, and access control

To access the Edge Flow Manager UI:

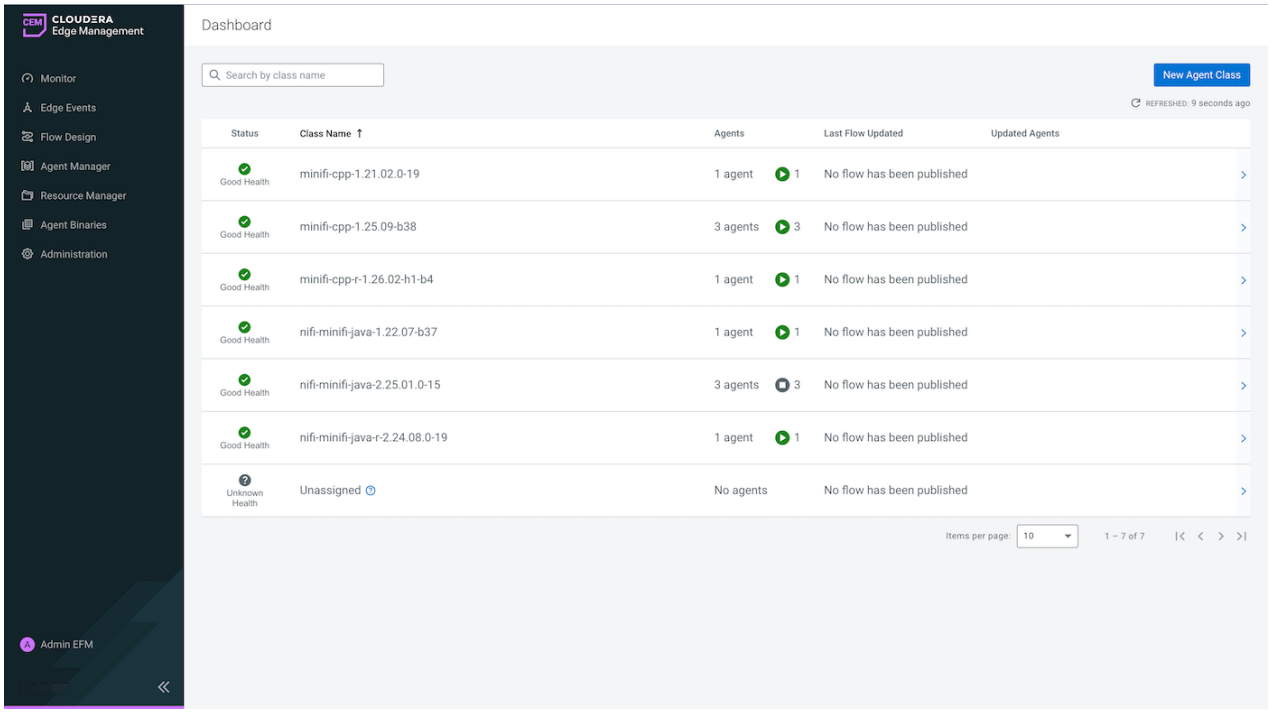
1. Start the application.
2. Open a web browser and navigate to http://<hostname>:10090/efm/ui.

Note: Replace <hostname> with the hostname of your environment.

Access to different features depends on assigned roles. Users can view or modify dataflows and monitor events based on their permissions.

Monitor

Select Monitor in the left navigation to open the **Dashboard**. This page provides an overview of C2 server status and agent deployments.



For more information on how to monitor deployments and other related actions, see *Managing deployments in Cloudera Edge Management*.

Related Information

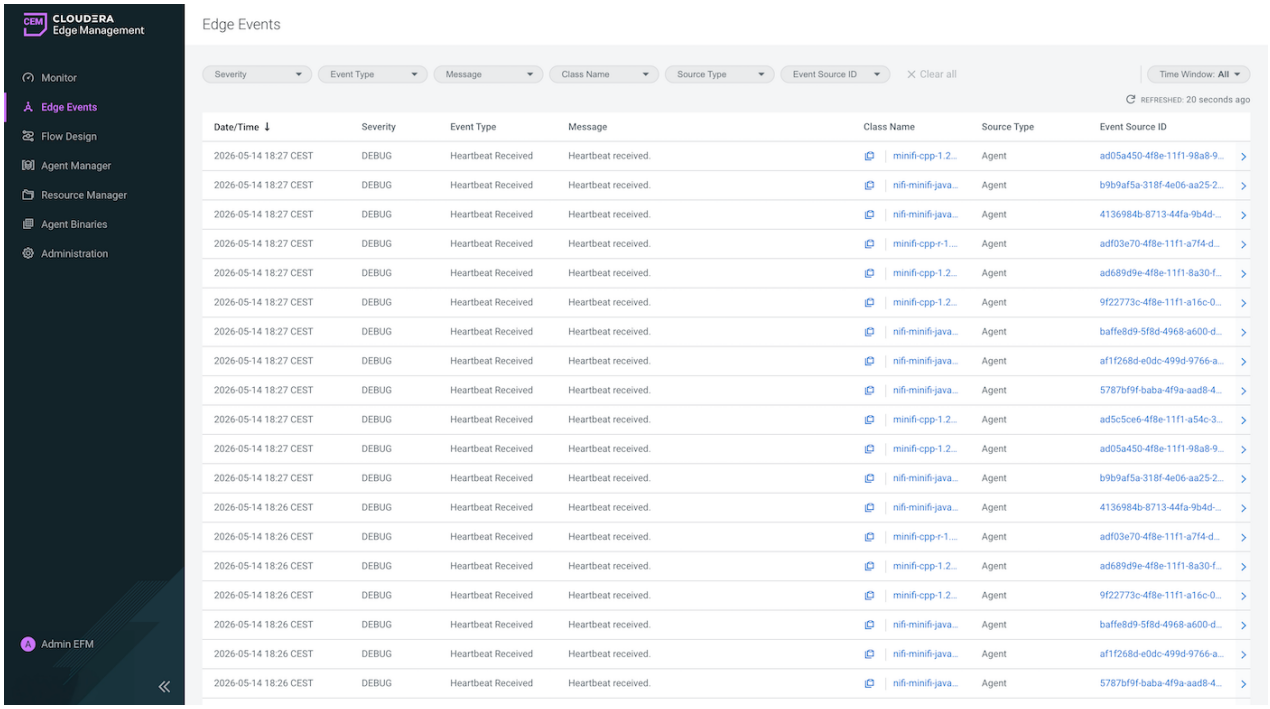
Monitoring deployments in Cloudera Edge Management

Edge Events

Select Edge Events in the left navigation to open the **Edge Events** view, which displays the list of events generated by the C2 server and agent events.

Use this view to:

- Track agent activity
- Investigate issues and system events



For more information on how to monitor server and agent events, see *Monitoring events in Cloudera Edge Management*.

Related Information

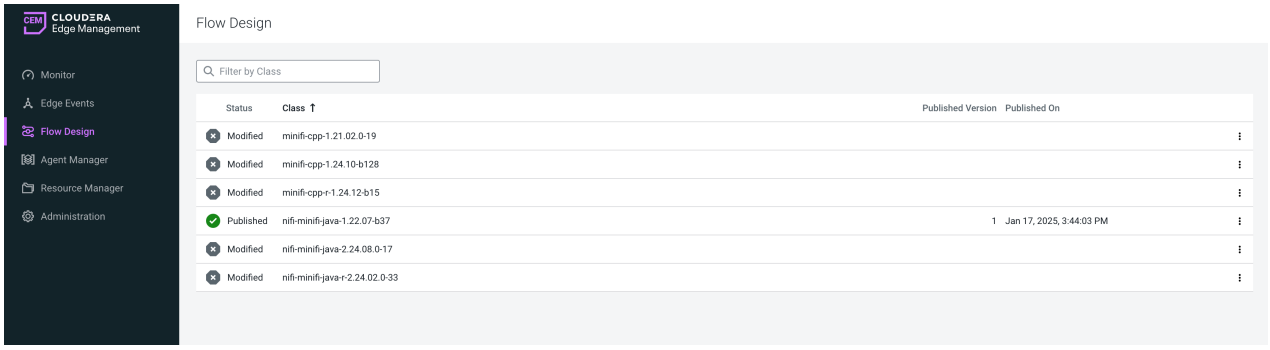
Monitoring events in Cloudera Edge Management

Flow Design

Select Flow Design in the left navigation to open the **Flow Design** view that allows you to build and publish dataflows in Cloudera Edge Management.

This view lists all available dataflows and agent classes with the following information:

- Status
- Class
- Published Version
- Published On

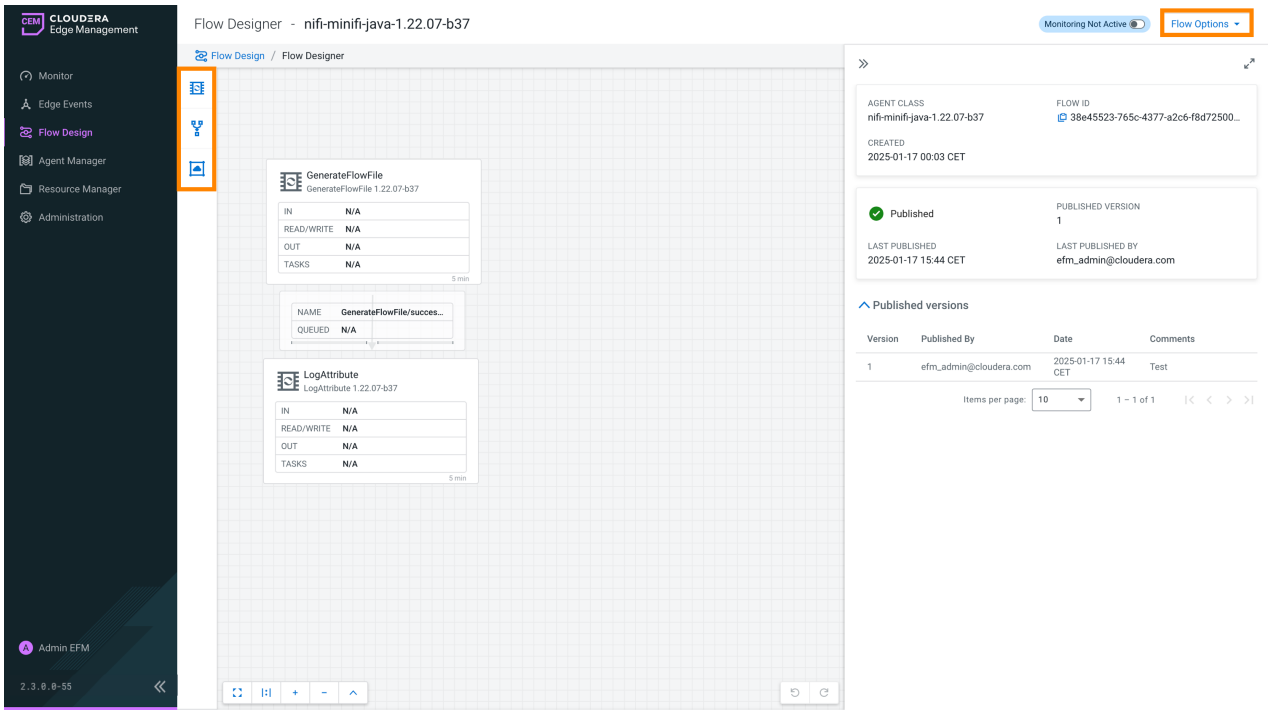


You can sort data in each column in ascending or descending order by clicking the column header.

You can filter flows by class by using the Filter by Class field.

Flow Designer

To edit a dataflow, click the arrow at the end of a class row or double-click the class name. This opens the **Flow Designer**.



Components toolbar

Located on the left side of the canvas, the toolbar contains the main components (processor, funnel, remote processor group) you need to design your dataflow. You can drag these components onto the canvas to build your flow.

Flow options menu

Located in the top-right corner of the screen, the Flow Options menu provides actions for managing your dataflow:

- Services: Access shared services for processor configurations or task operations
- Parameters: Manage parameters for configuring processor and service properties
- Publish: Publish the current flow version
- Revert to Last Published: Restore the last published version of the flow
- Refresh Manifest: Synchronize manifest changes
- Import Flow: Import an existing flow

- Export Flow: Export the current flow for reuse or backup purposes
- Back to Flow Designs: Return to the main Flow Design page

For more information, see:

- *Building a dataflow in Cloudera Edge Management*
- *Publishing a dataflow in Cloudera Edge Management*

Monitoring dataflows

The monitoring view provides a read-only, near real-time view of a running flow.

Use it to:

- Observe processor, connection, and queue behavior
- Identify bottlenecks or issues

While this view is similar in appearance to the **Flow Designer** interface and can be accessed from the **Flow Designer**, it operates in a read-only mode, ensuring that no modifications can be made to the flow during monitoring.

Important: Monitoring is only available for published flows. If your flow has not been published, the toggle for monitoring is inactive. If you switch to monitoring while editing a flow, the system will display the last published version of the flow. Changes in progress will not be reflected in the monitoring view.

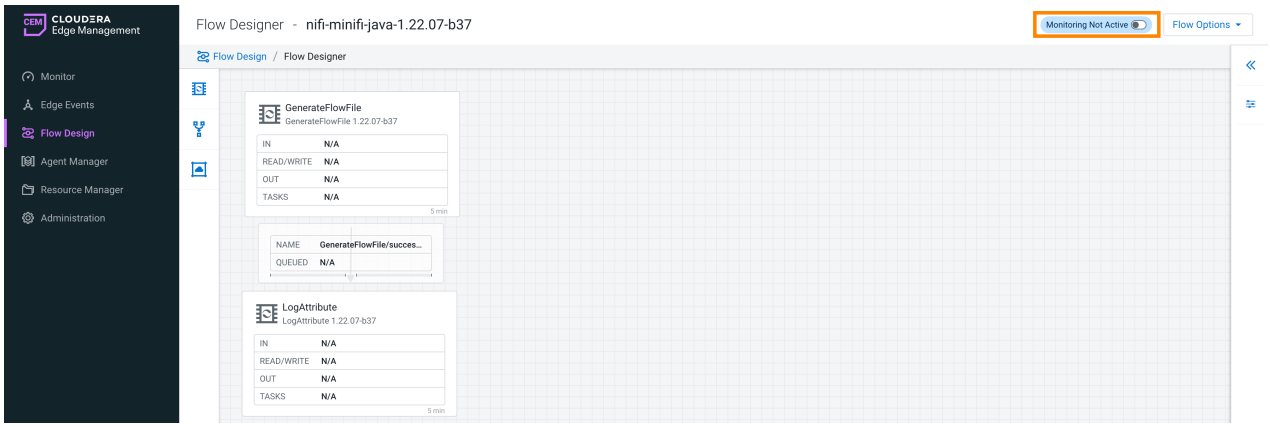
Use the Monitoring Not Active / Active toggle in the upper-right corner to switch between:

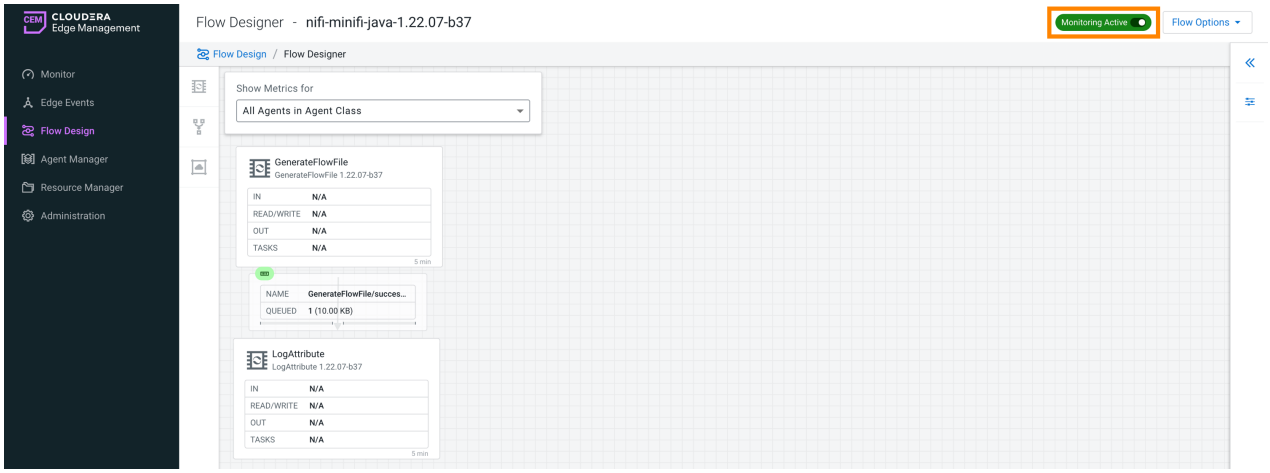
- Editing mode: Modify the flow
- Monitoring mode: View runtime behavior

You can monitor:

- All agents in a class (aggregated view)
- A specific agent (detailed view)

Important: Processor-related statistics are supported only with specific agent versions. Java Agent support starts from version 2.24.08.0, and C++ Agent support starts from version 1.25.03. If you use earlier agent versions, the Monitoring view displays data for queues only, with processor statistics unavailable.





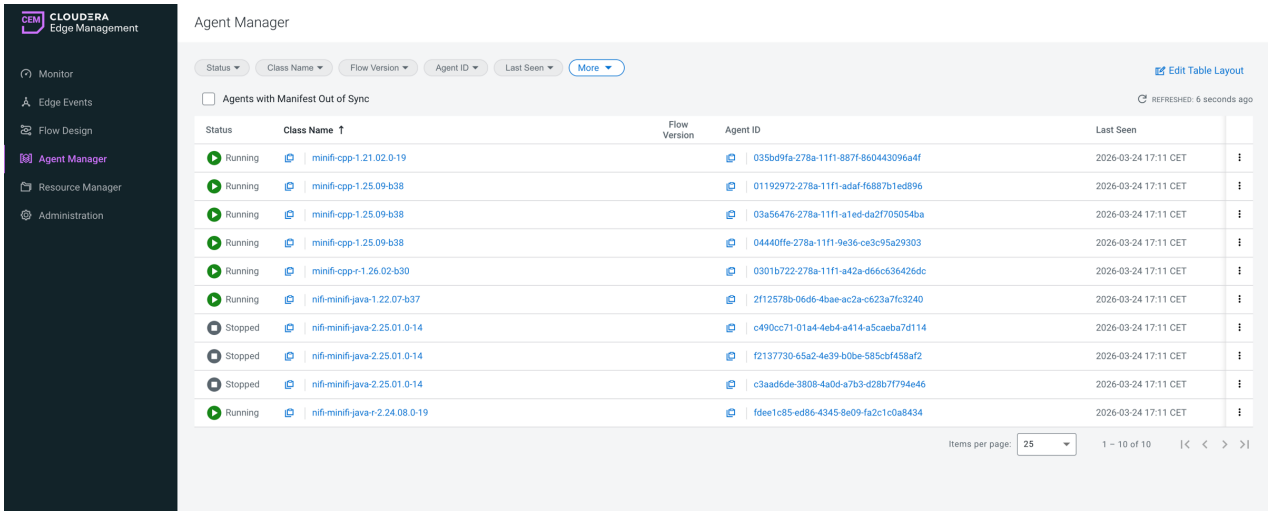
Related Information

Building a dataflow in Cloudera Edge Management

Publishing a dataflow in Cloudera Edge Management

Agent Manager

Select **Agent Manager** in the left navigation to open a centralized view to monitor the health and status of agents.



The Agent Manager allows you to:

- View and filter agents based on specific criteria
- Monitor agent health and activity
- Initiate commands on one or more agents

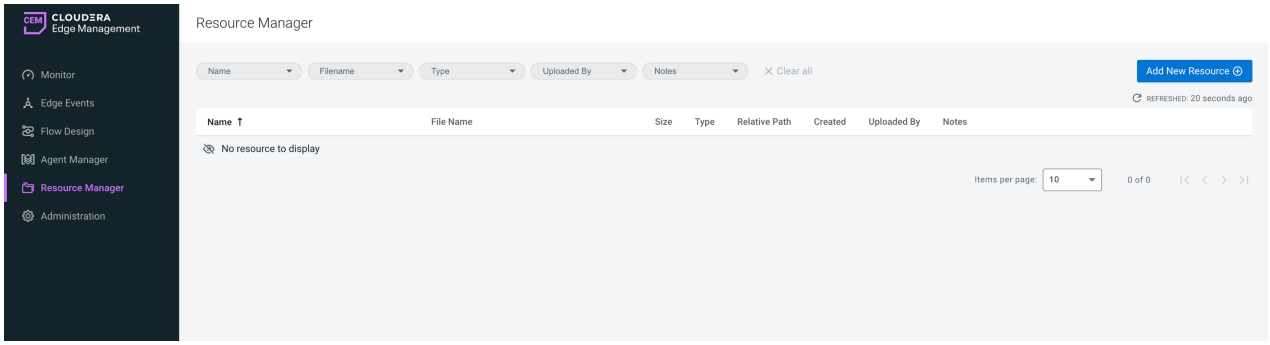
For more information, see *Managing agents in Cloudera Edge Management*.

Related Information

Managing agents in Cloudera Edge Management

Resource Manager

Select **Resource Manager** in the left navigation to open a view that enables you to manage assets and extensions in Cloudera Edge Management.



You can:

- Upload new resources
- Manage existing resources

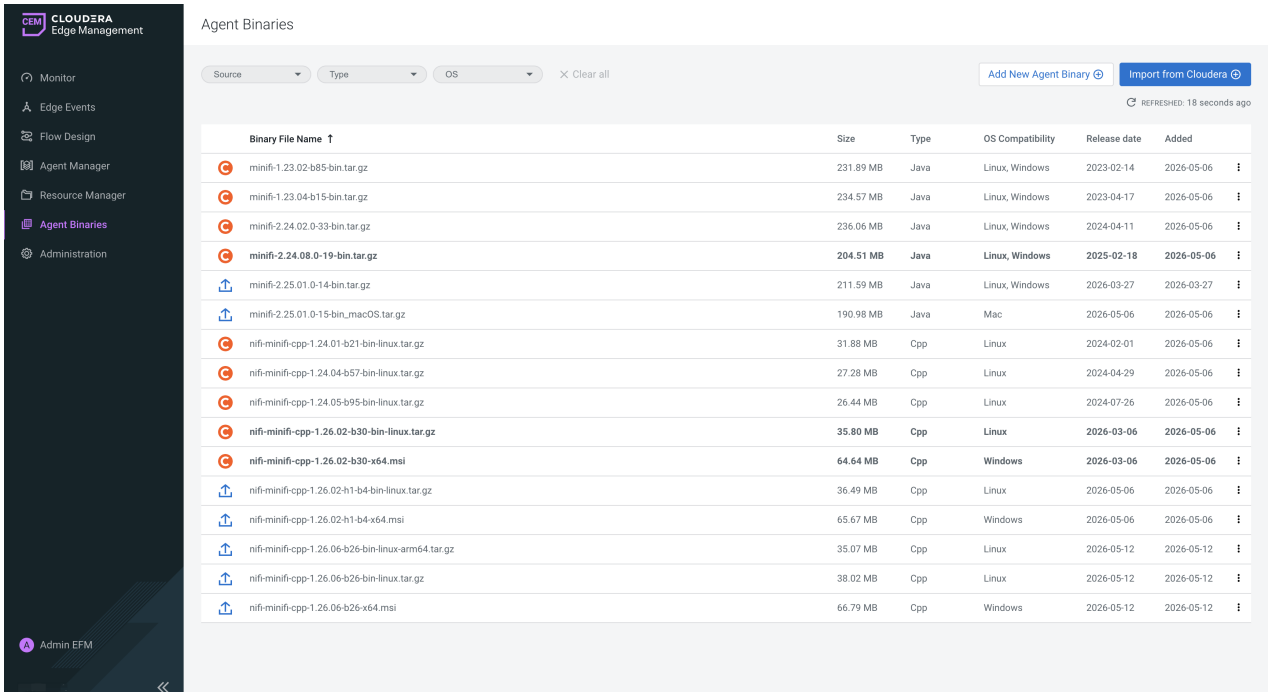
For more information on how to work with resources, see *Managing resources in Cloudera Edge Management*.

Related Information

Managing resources in Cloudera Edge Management

Agent Binaries

Select **Agent Binaries** in the left navigation to open a centralized view where you can manage agent binaries. This page simplifies agent deployment workflows by providing a single location to manage all agent binaries required for your edge environments.

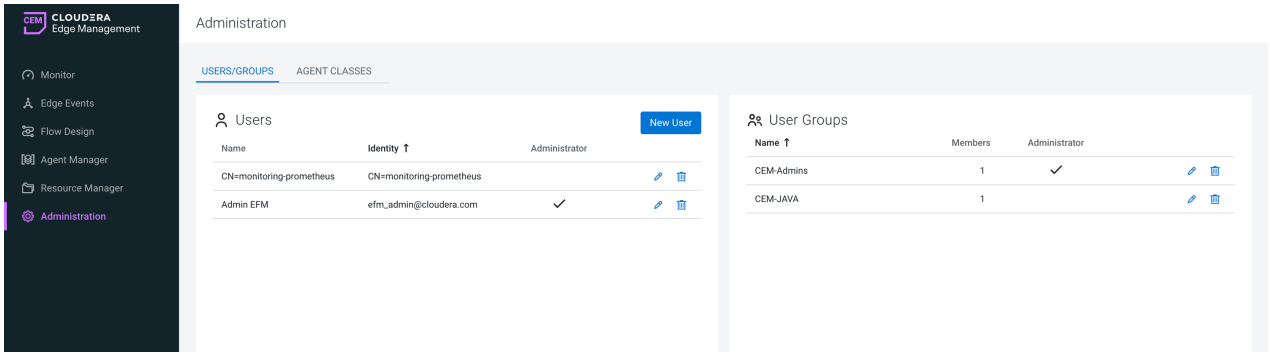


For more information, see *Managing agent binaries in Cloudera Edge Management*.

Administration

Select Administration in the left navigation to open the **Administration** view for managing access control using role-based authorization.

Role-based authorization means that access is granted based on roles assigned to agent classes, determining what users can view and modify. These roles must be associated with authenticated users, ensuring secure and role-specific operations. The **Administration** screen allows you to manage user permissions and define access controls.



You can:

- Assign roles to users based on their access requirements
- Manage permissions for each user to ensure secure and controlled operations
- Associate roles with agent classes for fine-grained access control

For more information, see *Access control policies*.

Related Information

Access control policies

Managing agent binaries in Cloudera Edge Management

The **Agent Binaries** page provides centralized binary management. Instead of manually placing binaries on the file system, you can use the UI to upload, import, and manage binaries across all Edge Flow Manager nodes.

Agent binaries represent the installable packages (Java or C++) used when deploying agents to edge devices.

On the **Agent Binaries** page, you can:

- Upload custom agent binaries
- Import official binaries from Cloudera
- View available binaries and their metadata
- Download binaries
- Delete unused binaries

All binaries are stored in a shared repository and are automatically synchronized across Edge Flow Manager nodes, and deletions are also propagated. This ensures all nodes have a consistent set of binaries.

Agent binaries are stored in the directory defined by efm.agent-deployer.binariesRootPath.

By default: ${EFM_HOME_DIRECTORY}/agent-deployer/binaries

This location is shared with the Agent Deployer functionality. The Agent Binaries repository is directly used by the Agent Deployer.

- All binaries available on this page are selectable when generating a deploy command
- Newly uploaded or imported binaries automatically become available for deployment
- No manual file system copying is required

Directory structure

New binaries follow this structure: {agentType}/{agentVersion}/<binary-file>

For example:

```
java/1.23.02/minifi.tar.gz
cpp/1.24.05/nifi-minifi-cpp.tar.gz
```

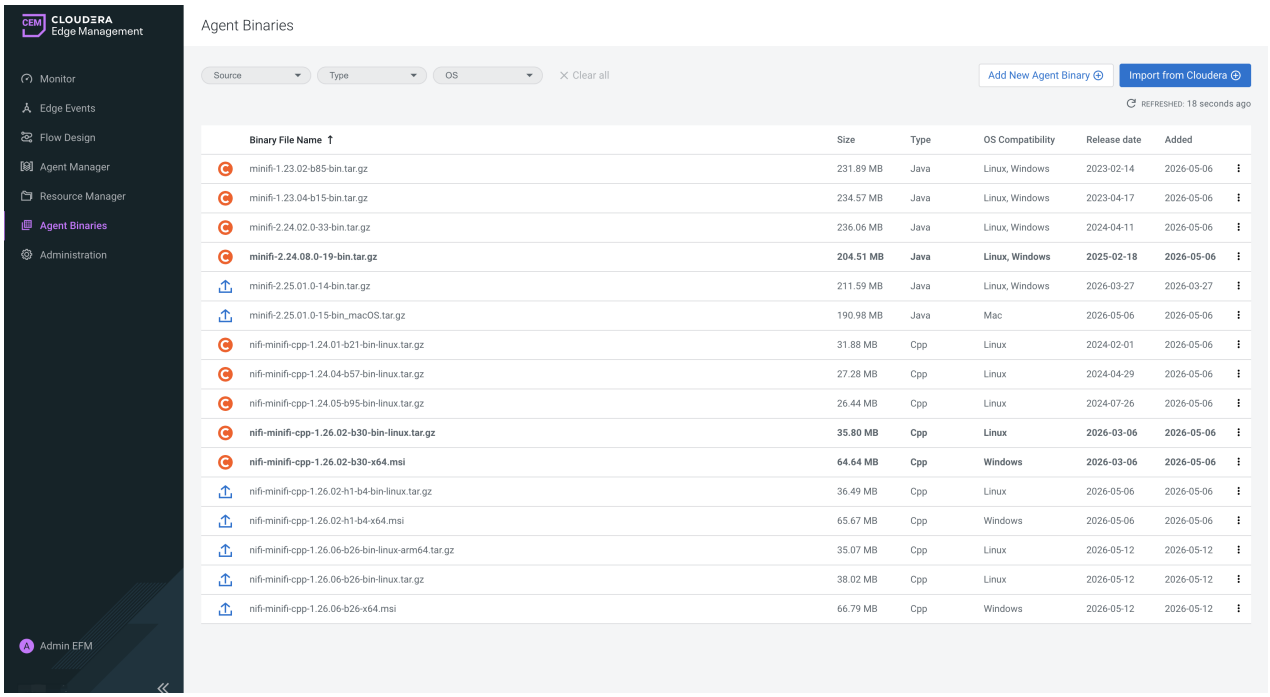
Note: New uploads use the updated structure, but older directory structures are still supported for backward compatibility.

Accessing the Agent Binaries page

To open the **Agent Binaries** page, in the Edge Flow Manager UI, select Agent Binaries from the left navigation. The page displays a list of available binaries along with details such as:

- File name
- Size
- Type (Java or C++)
- OS compatibility
- Release date
- Date added

You can filter the list using the available filters (Source, Type, OS).

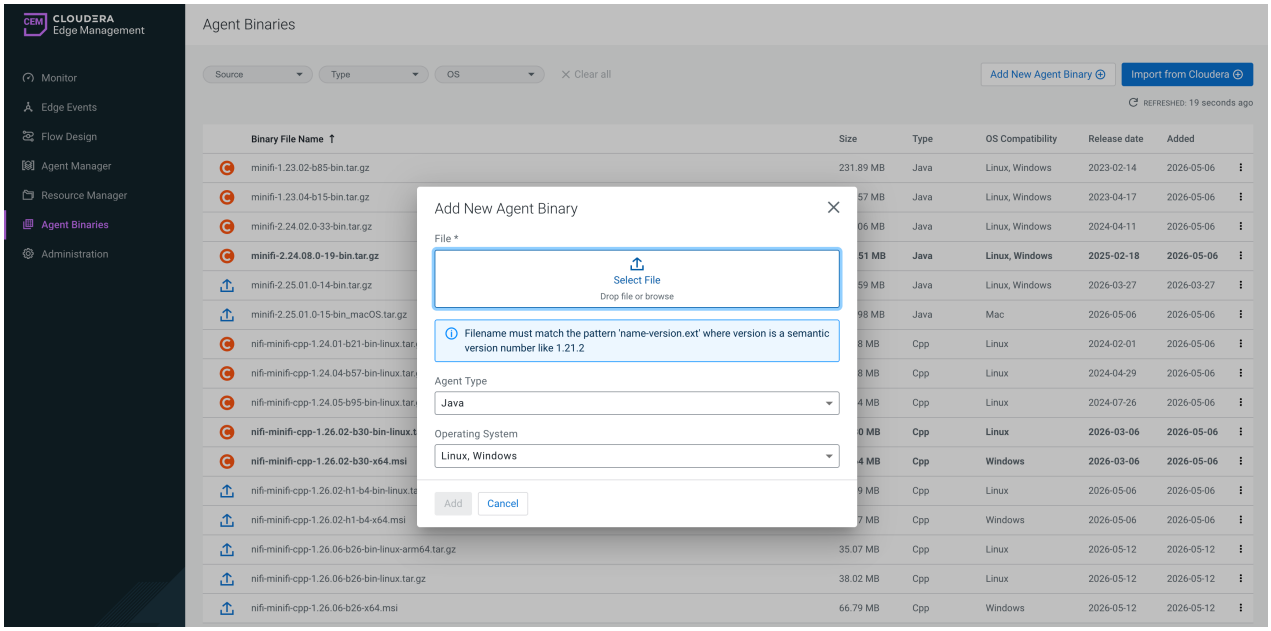


Uploading a custom agent binary

You can upload your own agent binaries, for example if you use custom builds.

To upload a binary:

1. Click Add New Agent Binary.
2. Select the binary file.
3. Specify:
   - Agent Type (Java or C++)
   - Operating System
4. Click Add.



The binary is uploaded and appears in the list.

Note: The filename must follow this format: name-version.ext, where version is a semantic version (for example, 1.21.2).

Importing agent binaries from Cloudera Archive

You can import officially supported agent binaries directly from Cloudera Archive.

Important: To enable this feature, you must configure Cloudera archive credentials before starting Edge Flow Manager.

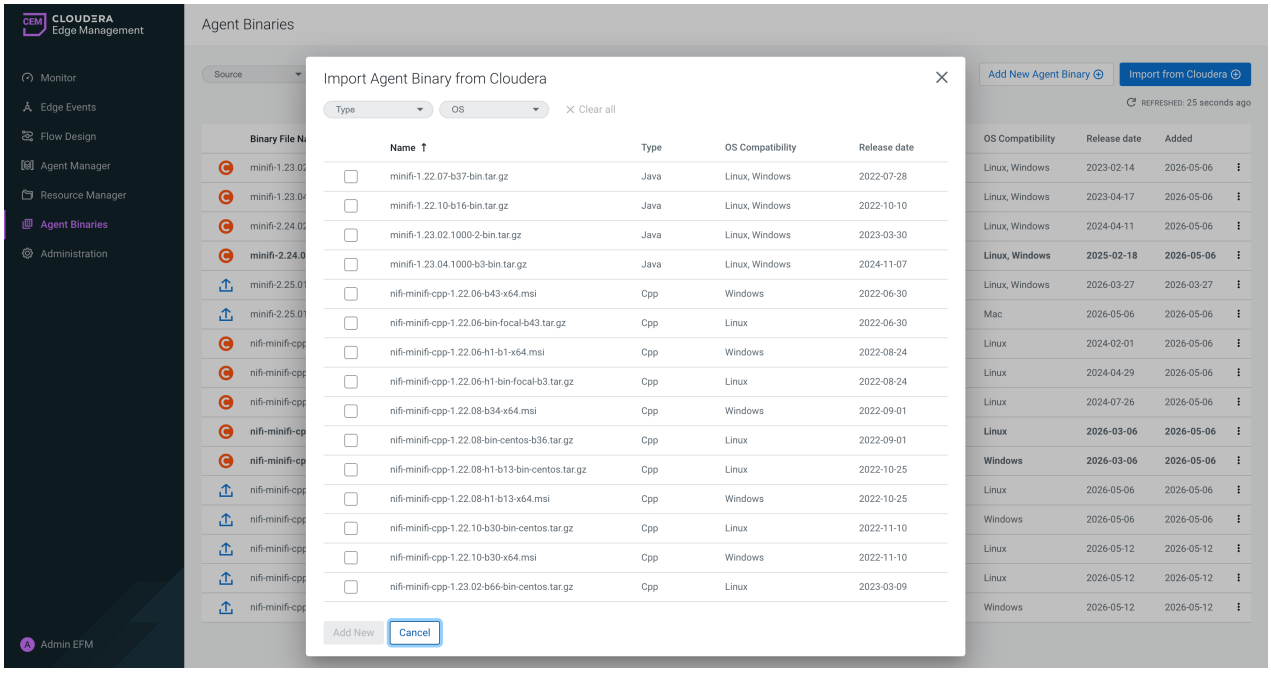
Set the following properties on each node:

- efm.paywall.username
- efm.paywall.password

If these properties are not set, the Import from Cloudera option is disabled and only manual upload is available.

To import binaries from the Cloudera Archive:

1. Click Import from Cloudera.
2. Select one or more binaries from the list.
    - Edge Flow Manager periodically queries the Cloudera archive for available versions
    - The list is dynamically updated
    - Latest versions are highlighted in the UI
3. Click Add New.



The selected binaries are downloaded and added to the repository.

Viewing and managing binaries

Viewing binaries

The list shows all available binaries in the repository. Icons indicate the source:

- Custom uploaded binaries
- Cloudera-imported binaries

Downloading a binary

To download a binary, click the Download icon next to the binary.

Deleting a binary

To remove a binary:

1. Open the actions menu next to the binary.
2. Select Delete.

Deletion is synchronized across all Edge Flow Manager nodes.

Configuring agent binary management

The Agent Binaries feature can be configured using properties in the efm.properties file.

These properties control:

- Access to Cloudera-hosted binaries
- Storage location of binaries
- Synchronization behavior across Edge Flow Manager nodes
- Refresh and polling intervals

Cloudera paywall integration

These properties enable importing agent binaries from the Cloudera Archive.

efm.paywall.username / efm.paywall.password

They specify the credentials used to access Cloudera-hosted agent binaries.

If these properties are not set, the Import from Cloudera action is disabled on the UI.

efm.paywall.java.baseUrl / efm.paywall.cpp.baseUrl

They define the base URLs used to retrieve Java and C++ agent binaries from Cloudera.

Binary storage
efm.agent-deployer.binariesRootPath

It defines the root directory where agent binaries are stored.

This directory is shared with the Agent Deployer and is used as the central repository for all uploaded and imported binaries.

Synchronization and cache behavior

These properties control how binaries are synchronized and tracked across Edge Flow Manager nodes.

efm.binarymanager.nodeAliveTtl

It defines how long a node is considered active in the cluster cache. After this time, the node is treated as unavailable.

efm.binarymanager.deletedBinaryTtl

It specifies how long metadata for deleted binaries is retained before expiring.

efm.binarymanager.resourceSynchronizationTriggerInterval

It controls how frequently the system synchronizes binaries across nodes.

efm.binarymanager.nodeResourceCacheUpdateTriggerInterval

It determines how often the local node refreshes its binary cache state.

Refresh and polling intervals

These properties control how often Edge Flow Manager checks for updates from the local file system and from Cloudera archive.

efm.binary-manager.agent-binary-sync.fixedDelay

It defines how frequently the system scans for new binaries in the local storage and new binary versions available from Cloudera.

efm.binary-manager.agent-binary-sync.initialDelay

It specifies the delay after startup before the first synchronization is triggered.

Note:

In most deployments, the default values are sufficient and do not require modification. Changes to these properties require restarting the system to take effect.

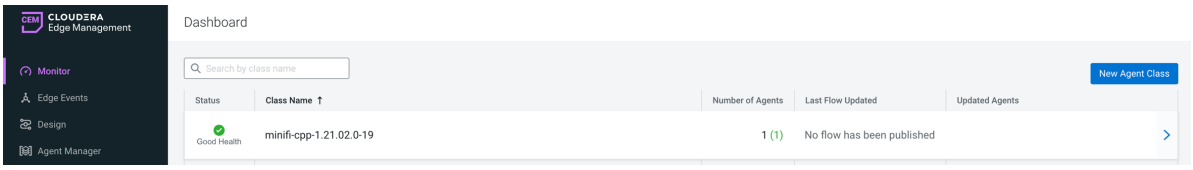
Deploying agents in Cloudera Edge Management

Using the Agent Deployer

Learn how you can deploy a MiNiFi application using the Agent Deployer in the Edge Flow Manager, including generating and running deployment commands.

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 and click Create.

   The new aget class is added to the list.

   Note:

   If you already have an agent class, you can skip this step. When adding agents to an existing class, use the same agent version to ensure compatibility. If you are uncertain about the versions, create a new class.

2. Prepare the agent binaries.

   Before deploying an agent, ensure the required binaries are available so that you can use the Deploy Agent functionality.

   You can:

   - Upload or import binaries using the **Agent Binaries** page (recommended).

     For more information, see *Managing agent binaries in Cloudera Edge Management.*

     or
   - Create an agent repository and place binaries manually in the directory defined by efm.agent-deployer.binariesRootPath.

     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}/{agentVersion}]

     Accepted values:

     - agentType: java or cpp
     - agent version: Cloudera version of the agent (for example: 1.23.02)

Note:

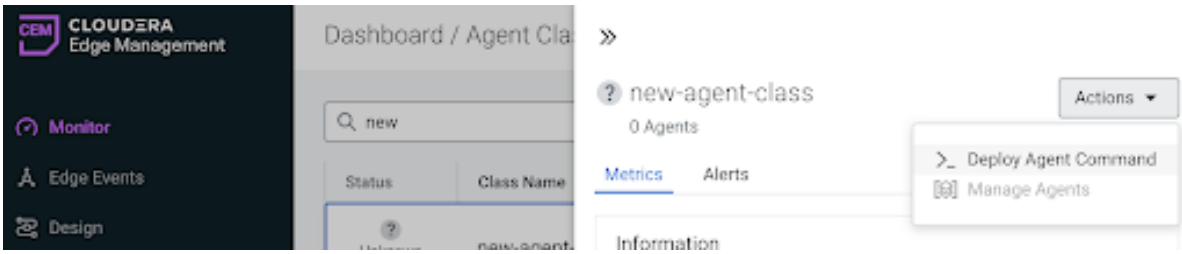
MacOS is only supported for testing purposes. Each directory should contain only one file. While the file name can be any valid name, the extension must adhere to the following constraints:

- 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 C++ agent type, the MSI format should be used.
- On Mac: The tar.gz should be used, but the filename must end with "_macos" before the tar.gz.

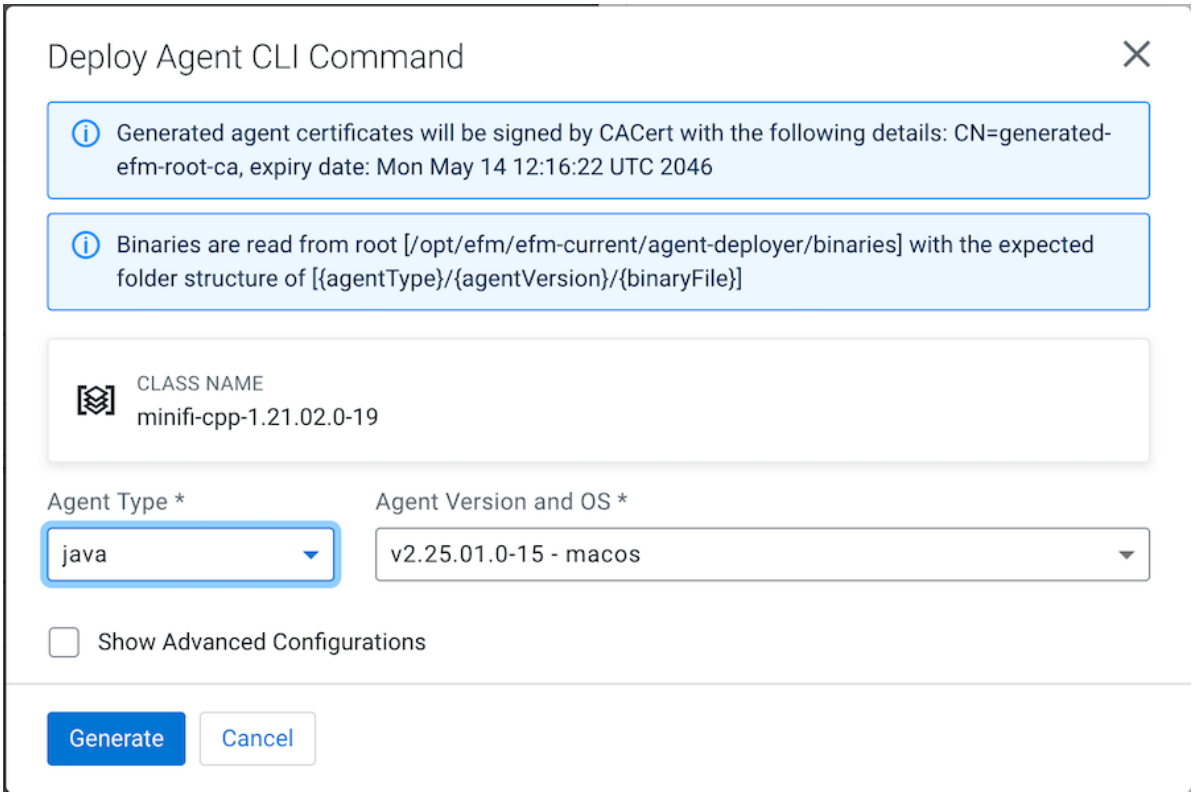
A few examples:

```
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/java/1.23.02/minifi.tar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/java/1.23.04/minifi_macos
.tar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/cpp/1.23.03/minifi.tar.gz
${EFM_HOME_DIRECTORY}/agent-deployer/binaries/cpp/1.23.03/minifi.msi
```

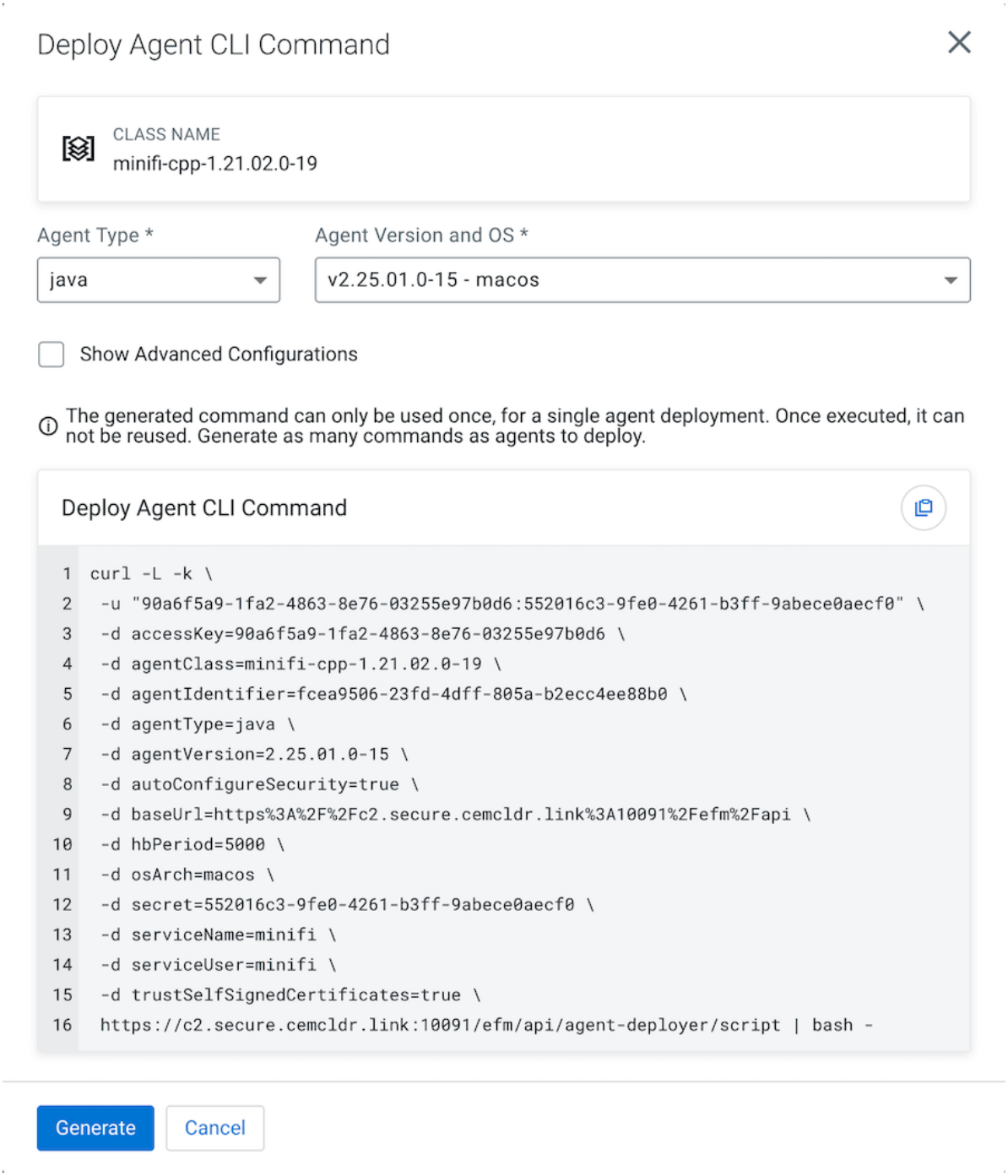
3. Generate a deploy command.
   a) Click Monitor in the left navigation to open the Dashboard.
   b) Click an agent class to display the agent class details pane.
   c) Click Actions Deploy Agent Command .



   If binaries are available, the Agent Type, Agent Version, and OS fields are pre-populated.
   d) Select the desired options.
   e) Click Generate.



The generated deployment command is displayed.



4. Run the deployment command.
   a) To install the agent, copy the generated CLI command into the host where the agent will be installed.

      Note:

      Ensure the agent host has a network connection to Edge Flow Manager to successfully run the command.

   b) Run the command on the target host.

   The command:

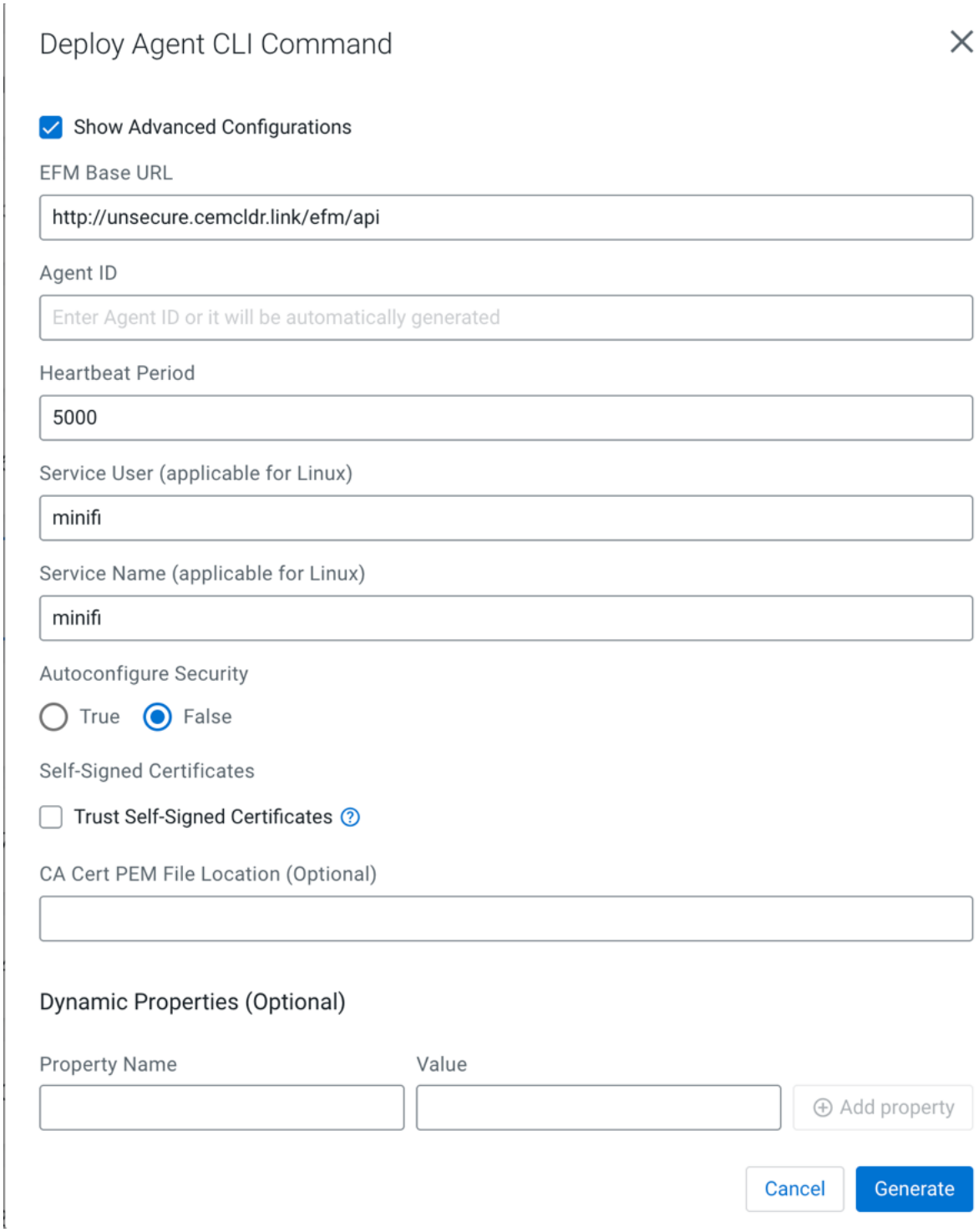
   - Downloads the agent binary
   - Extracts and configures it

- Starts the agent

After successful execution, the agent appears in Edge Flow Manager and begins sending heartbeats.

5. Configure the advanced options for the `Deploy Agent CLI` command.

In specific scenarios, the default configuration may not meet your specific requirements. In this case, use the advanced configurations to customize parameters. To make the advanced options visible, select Show Advanced Configurations.



Edge Flow Manager Base URL

The full URL of the EFM REST API base. The generated command uses this URL to access Edge Flow Manager 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 Edge Flow Manager behind a proxy, set the efm.proxy.c2ProxyPath property in the efm.properties file. This automatically sets the Edge Flow Manager 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

Defines how often the agent sends heartbeat messages.

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 Edge Flow Manager (for example any authentication method is turned on), the agent needs to connect to Edge Flow Manager in a secured manner. If Autoconfigure Security is enabled, Edge Flow Manager 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, Edge Flow Manager 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=<st
rong_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 Edge Flow Manager is set up using a self-signed certificate (where Edge Flow Manager 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 Edge Flow Manager.

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 Edge Flow Manager 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.

6. 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.

Note: Installing MiNiFi C++ 1.26.02 and higher on Windows using this command installs the application to the default installation directory (typically C:\Program Files\ApacheNiFiMiNiFi\nifi-minifi-cpp) and not to the current working directory.

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.

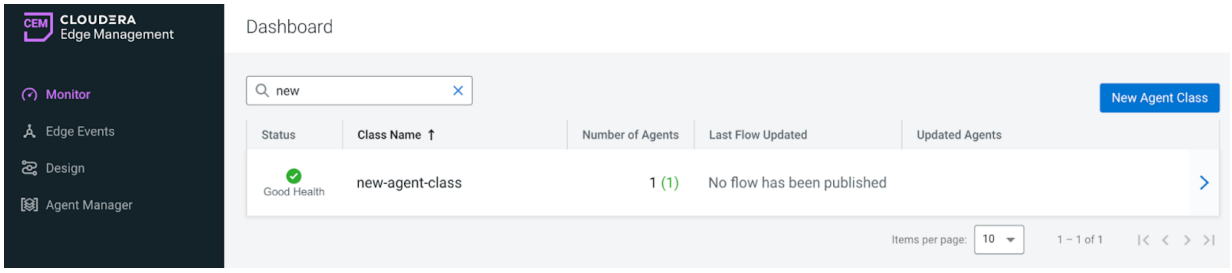
For Java agents on Linux, you should see a log message similar to the following.

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

```
user@user-host /opt/cloudera/minifi curl -L \
 -d agentClass=new-agent-class \
 -d agentIdentifier=061c6cf6-1922-41b7-8239-ad25a0f5be9a \
 -d agentType=java \
 -d agentVersion=1.23.04 \
 -d autoConfigureSecurity=false \
 -d baseUrl=http%3A%2F%2Funsecure.cemcldr.link%2Fefm%2Fapi \
 -d hbPeriod=5000 \
 -d osArch=MacOs \
 -d serviceName=minifi \
 -d serviceUser=minifi \
 -d trustSelfSignedCertificates=false \
 http://unsecure.cemcldr.link/efm/api/agent-deployer/script | bash -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  C
urrent
                                 Dload  Upload   Total   Spent    Left  S
peed
100 17505  100 17208  100   297  18755    323 --:--:-- --:--:-- --:--:--
 19152
 _______ __ _______ __ _______ __      __                 __         __ __
|   |   |__|    |  |__|    ___|__|    |__.-----.-----.  |_.----.-|  |  .--
---.----.
|       |  |       |  |    ___|  |    |  |     |__ --|   _|  _  |  |  |  -
__|   _|
|__|_|__|__|__|____|__|___|   |__|    |__|__|__|_____|____|___._|__|__|
_____|__|

-- Verifying if the following commands exist: tar gzip cat grep sed seq m
kdir sleep
-- Verifying if java is installed on the system...
-- Found java at: /Users/fkis/.sdkman/candidates/java/current/bin/java
openjdk version "1.8.0_362"
OpenJDK Runtime Environment (Zulu 8.68.0.19-CA-macos-aarch64) (build 1.
8.0_362-b08)
OpenJDK 64-Bit Server VM (Zulu 8.68.0.19-CA-macos-aarch64) (build 25.362-
b08, mixed mode)
-- Verifying if the system is capable of handling tar.gz archives...
-- Looking for a download utility on the system...
-- > curl
-- curl version: 7.88.1
-- Downloading MiNiFi...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Cu
rrent
                                 Dload  Upload   Total   Spent    Left  Sp
eed
100  234M  100  234M    0     0  1701k      0  0:02:21  0:02:21 --:--:--
 1123k
```

```
-- MiNiFi has been downloaded to directory: minifi-1.23.04-b15
-- 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 follo
wings 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/acknowl
edge
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 th
ese 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 Edge Flow Manager.



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 configure secure communication between the Edge Flow Manager and the agents using the Agent Deployer.

The Agent Deployer supports automatic security configuration during deployment. When enabled, it downloads a security bundle along with the agent binary, including the required certificates. This simplifies agent provisioning and ensures secure, trusted communication between the Edge Flow Manager and the 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 Edge Flow Manager generate a self-signed Root CA certificate

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

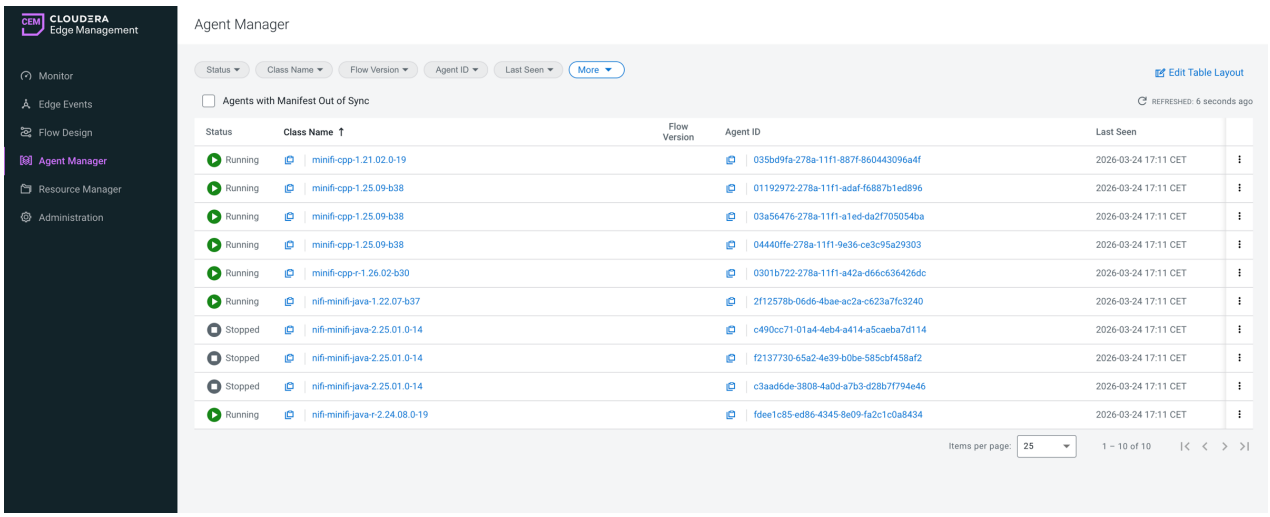
Managing agents in Cloudera Edge Management

Learn about the Agent Manager, which provides a centralized view for monitoring and managing individual agents across all agent classes..

The Agent Manager view provides you with 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 Edge Flow Manager. With the property update functionality, the agent configuration can be changed through Edge Flow Manager.

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

To access the Agent Manager view, click Agent Manager in the left navigation.



For more information about the Agent Manager, see Managing agents in Edge Flow Manager on the Cloudera Edge Management YouTube playlist:.

Agent list

The Agent Manager displays a list of agents across agent classes and provides filtering and customization options to help you focus on relevant agents.

You can:

- Filter agents using the available filters (for example, Status, Class Name, Flow Version, Agent ID, Last Seen)
- Customize the table layout to display relevant columns
- Identify agents with configuration inconsistencies (for example, manifest out of sync)

The Agent Manager provides the following information for each agent:

Status

Indicates the current state of the agent:

- Running – The agent is active and sending heartbeats
- Stopped – The agent is registered but not currently running
- Missing – The agent is not sending heartbeats and is considered unavailable

Status helps you quickly identify agents that may require attention.

The Agent Manager helps you detect configuration inconsistencies at the agent level. Agents that are not aligned with the expected configuration (for example, using an outdated manifest) are indicated in the UI. You can also use the Agents with Manifest Out of Sync filter to quickly identify affected agents.

Configuration inconsistencies may occur when:

- Updates are still in progress
- Some agents did not receive the latest configuration
- Synchronization across nodes is delayed

Use this view together with the Dashboard alerts to identify and resolve inconsistencies.

Class Name

Displays the agent class the agent belongs to.

Flow Version

Shows the version of the flow currently assigned to the agent.

Agent ID

Shows the unique identifier of the agent.

Last Seen

Indicates the last time the agent sent a heartbeat to Edge Flow Manager.

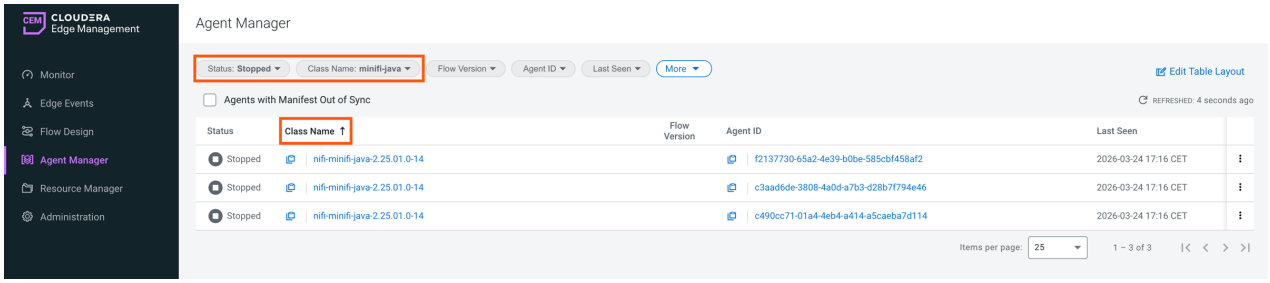
Sorting and filtering

You can sort data by most of the columns in ascending or descending order by clicking the column header.

To filter agents:

1. Select a column from the filter drop-down at the top of the table.
2. Enter a filter value.
3. Click ⊕ to apply the filter.

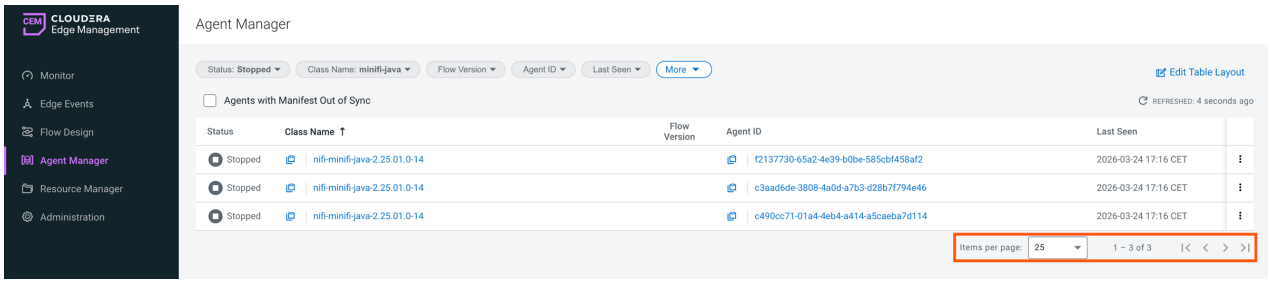
You can apply multiple filters across different columns to refine the results.



Items per page

Use the Items per page dropdown at the bottom of the table to control how many agents are displayed on a page.

If multiple pages are available, use the pagination controls to navigate between pages (first, previous, next, or last).



More options

Click # View Agent Details at the end of an agent row to open the agent details pane, which provides detailed information about a selected agent, including its status, configuration, system properties, and available actions.

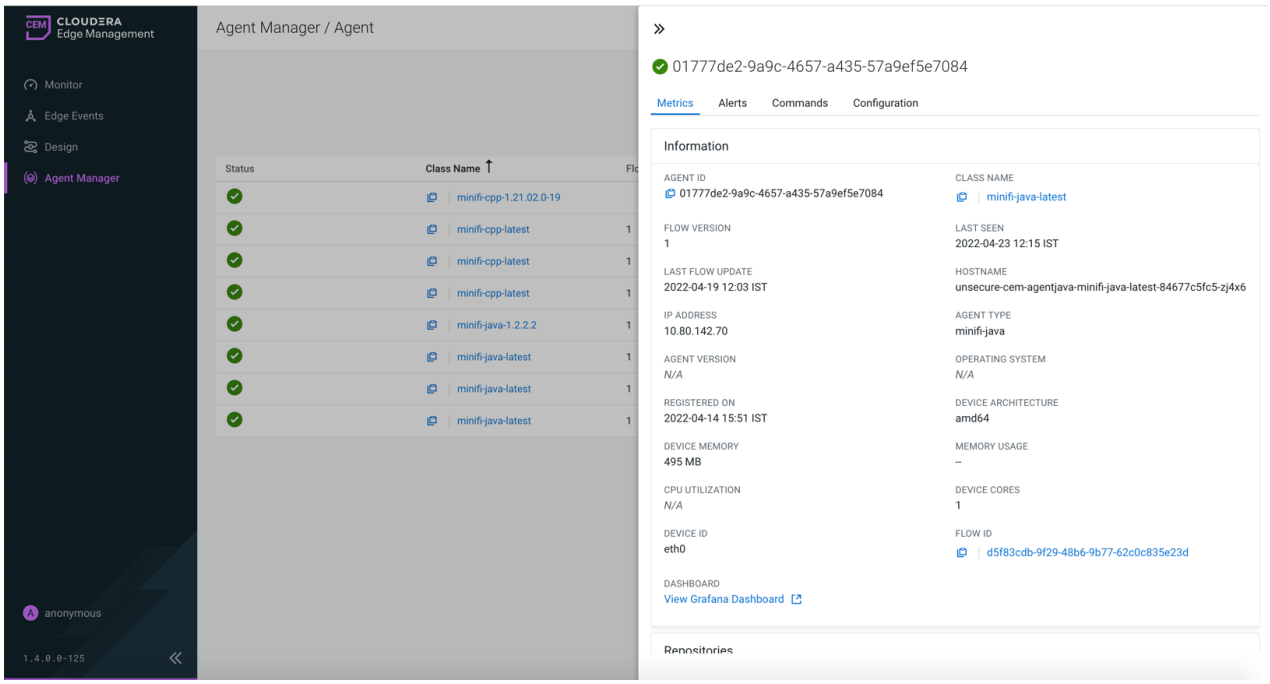
From here, you can:

- View metrics
- View alerts
- View and assign resources
- View and modify configurations

Viewing agent details in Cloudera Edge Management

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

From the Agent Manager, click # View Agent Details at the end of an agent row to open the agent details pane.



Metrics tab

It displays runtime metrics for the agent.

Agent Details

Provides basic information about the agent.

- Agent ID – Unique identifier of the agent
- Class name – Agent class the agent belongs to
- Agent Type – Type of agent (for example, Java or C++)
- Agent Version – Version of the agent
- Last Seen – Last time the agent sent a heartbeat
- Registered On – Timestamp when the agent was registered

Device Details

Provides information about the host system running the agent.

- Hostname – Name of the host machine
- IP address – Network address of the host
- Operating system – OS of the host
- Device architecture – CPU architecture (for example, amd64)
- Device memory – Total memory of the host (if available)
- Agent Memory Usage – Memory currently used by the agent
- Agent CPU Utilization – CPU usage of the agent (if available)
- Device CPU Load Average – System load average
- Device CPU Utilization – Overall CPU usage (if available)
- Device Cores – Number of CPU cores

- Device ID – Identifier of the network interface

Some metrics may not be available depending on the agent type and version.

Flow Definition

Provides information about the flow assigned to the agent.

- Flow ID – Identifier of the assigned flow
- Published Flow Version – Currently deployed flow version
- Last Flow Update – Timestamp of the last update

If no flow is published, the UI indicates that no version is available.

Repositories

Displays storage usage for agent repositories.

- FlowFile repository – Storage used for FlowFiles
- Provenance repository – Storage used for provenance data

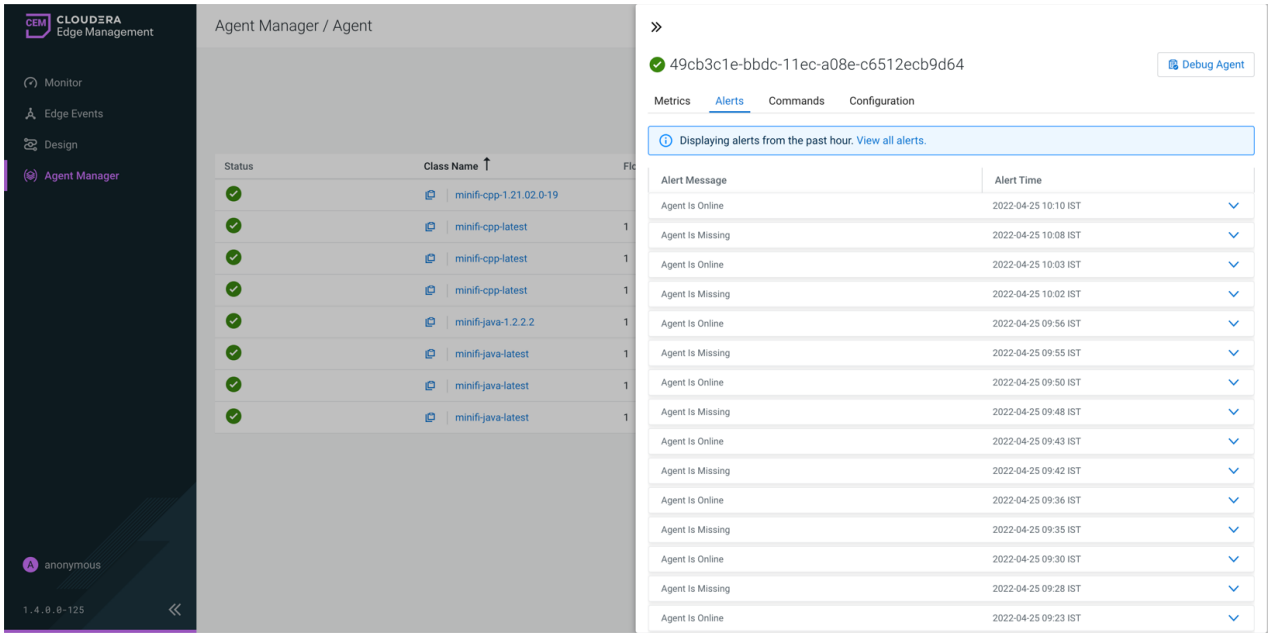
Usage indicators help you monitor resource consumption on the agent.

Connection Queues

Displays all connections used in a given agent accompanied by its metrics.

Alerts tab

It shows alerts related to the agent from the past hours. For example, you can monitor if there was no heartbeat from the agent for a specific time, or when it became available again. To view all alerts, click View all alerts.

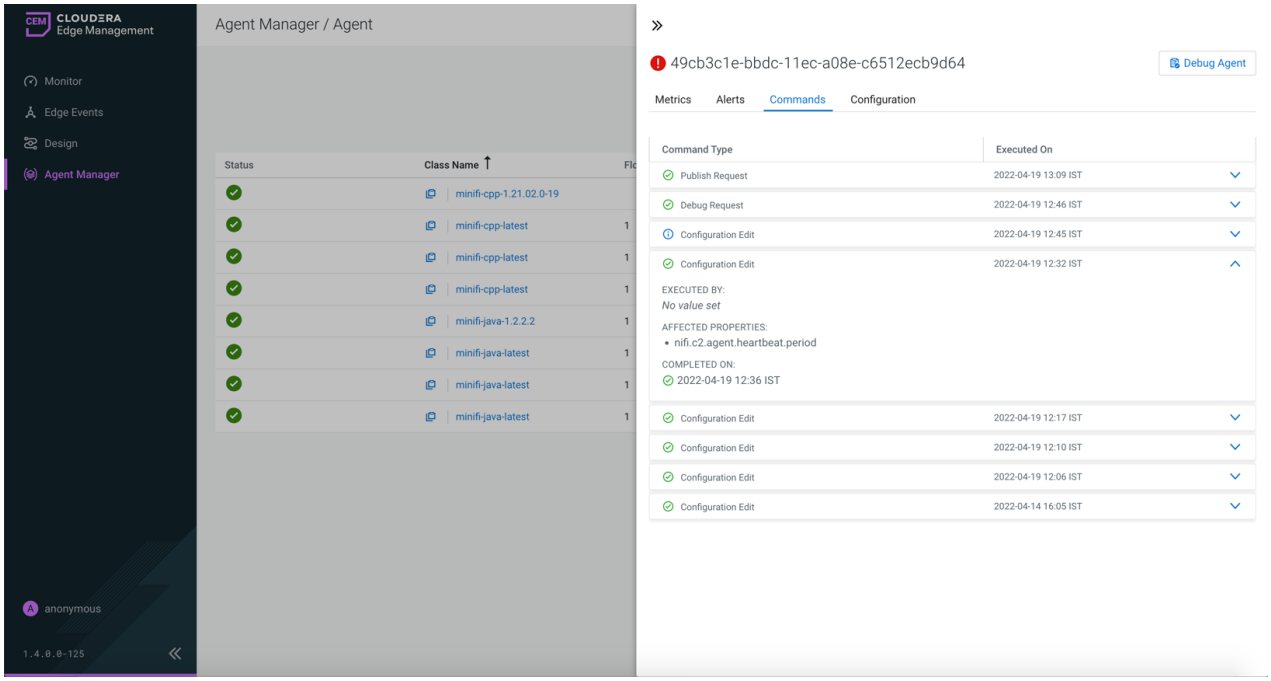


Commands tab

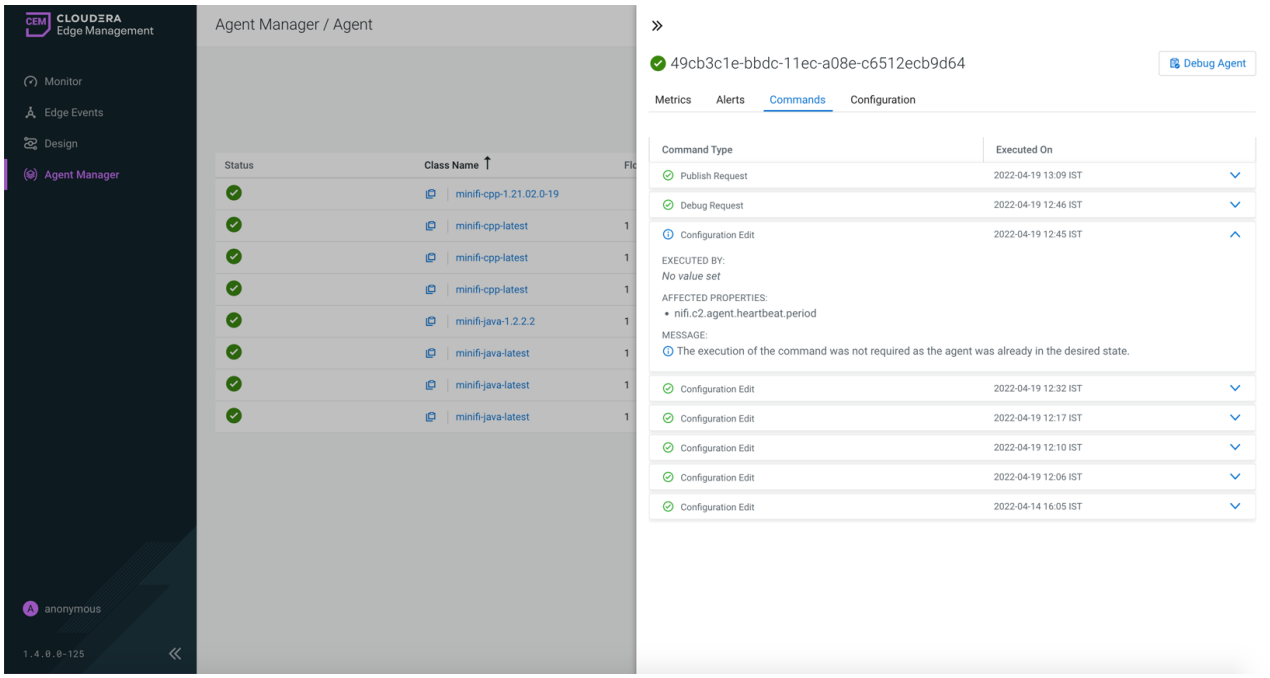
It lists commands executed on the agent. You can check the last 20 commands sent to the agent along with their statuses.

You can customize the number of displayed items using the efm.agentManager.commands.displayLimit property in the efm.properties file.

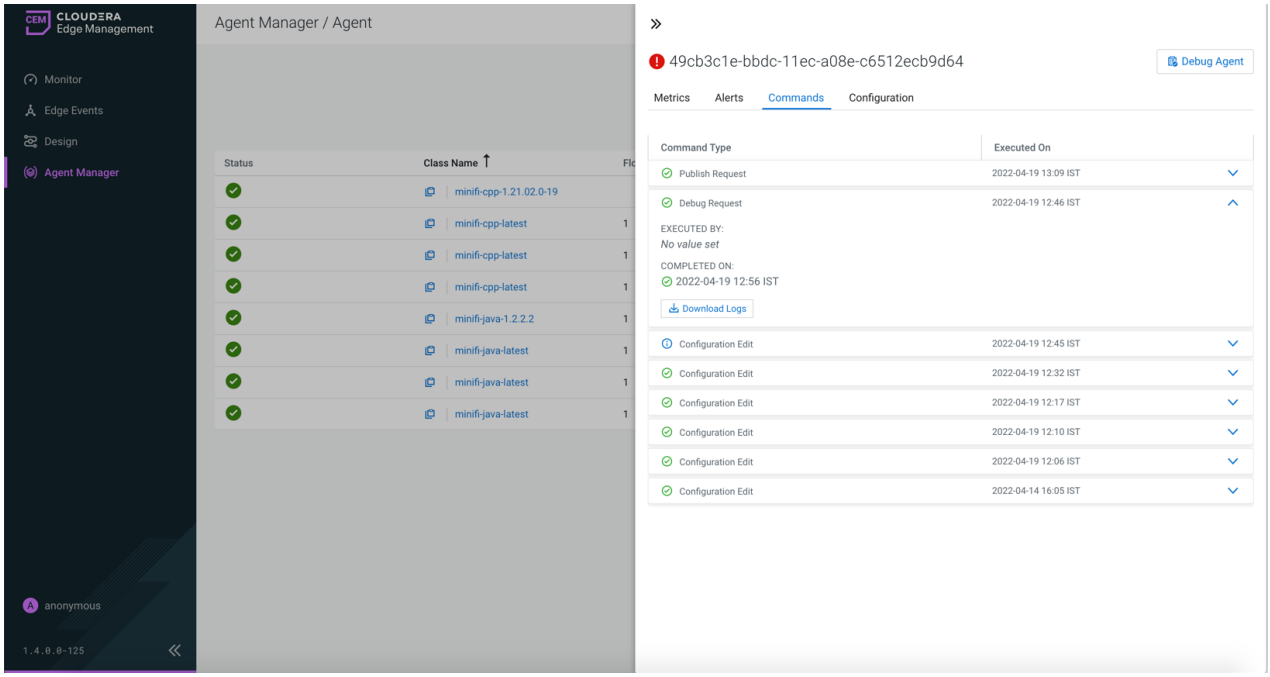
The details view may vary depending on the type of the command. For example, for **Configuration Edit**, you can check the affected property name, as shown in the following image:



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

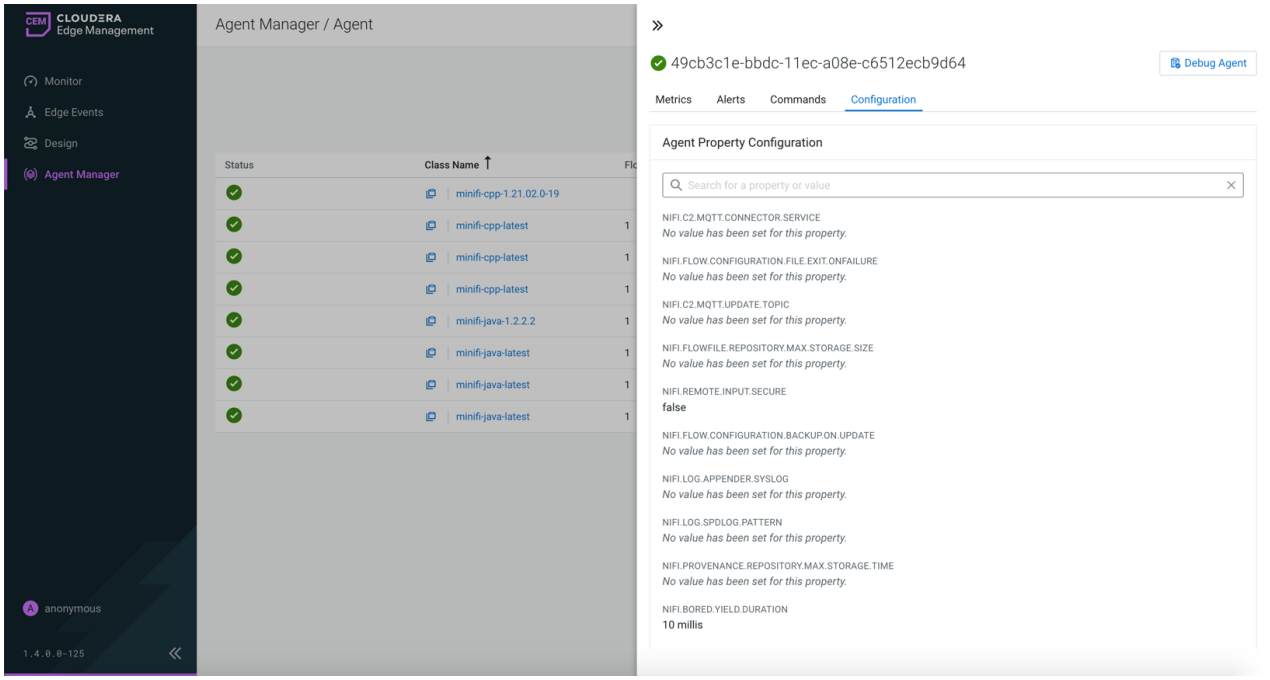


For debug requests, you are prompted with a Download Logs button to download the debug bundle retrieved from the agent, as shown in the following image:



Configuration tab

It displays the agent configuration. You can review the property configuration of an agent, provided the selected agent supports this functionality. Properties displayed can be filtered by both name and value.



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

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

Related Information

Debugging agent in Cloudera Edge Management
Monitoring deployments in Cloudera Edge Management

Starting/Stopping a flow

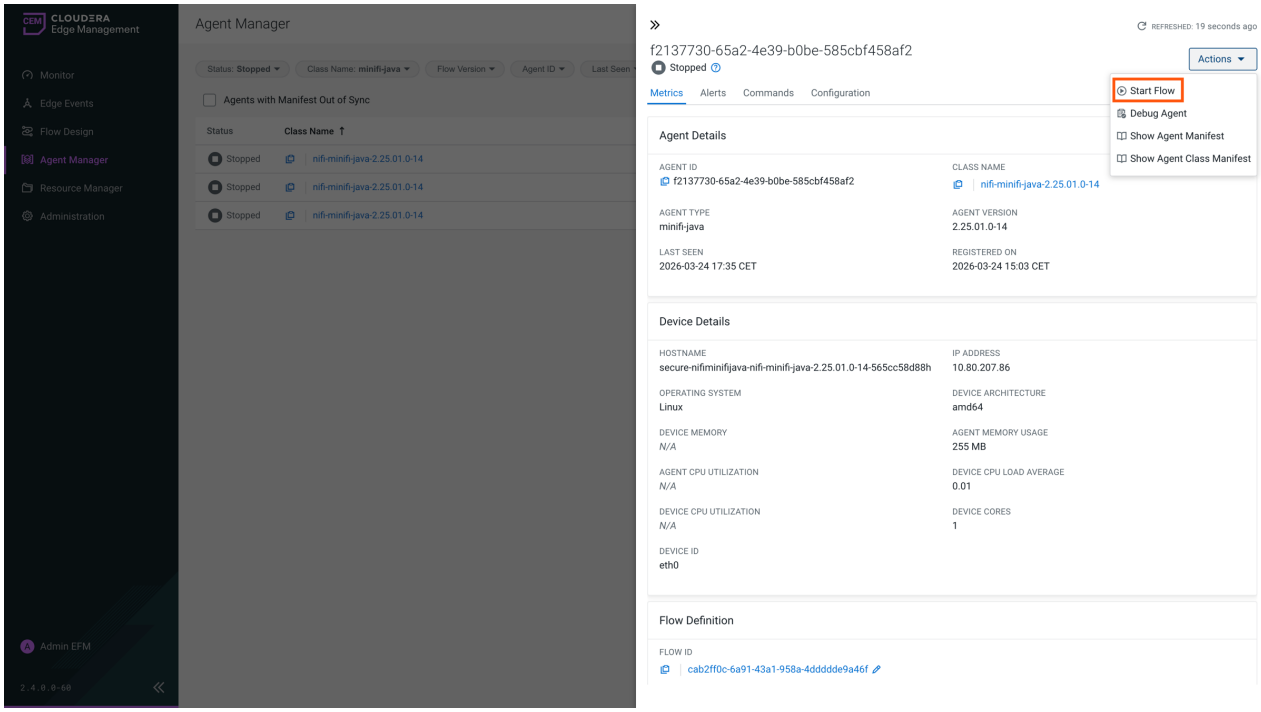
You can start and stop running a flow after deployment. This way you have greater control over when your flows are actively processing data.

1. From the Agent Manager # View Agent Details at the end of an agent row to open the agent details pane.
2. Click Actions Start Flow .

You can:

- Temporarily pause data collection for maintenance or troubleshooting.
- Quickly stop flows in response to operational issues.

Note: The start and stop actions are only available if the flows are published.

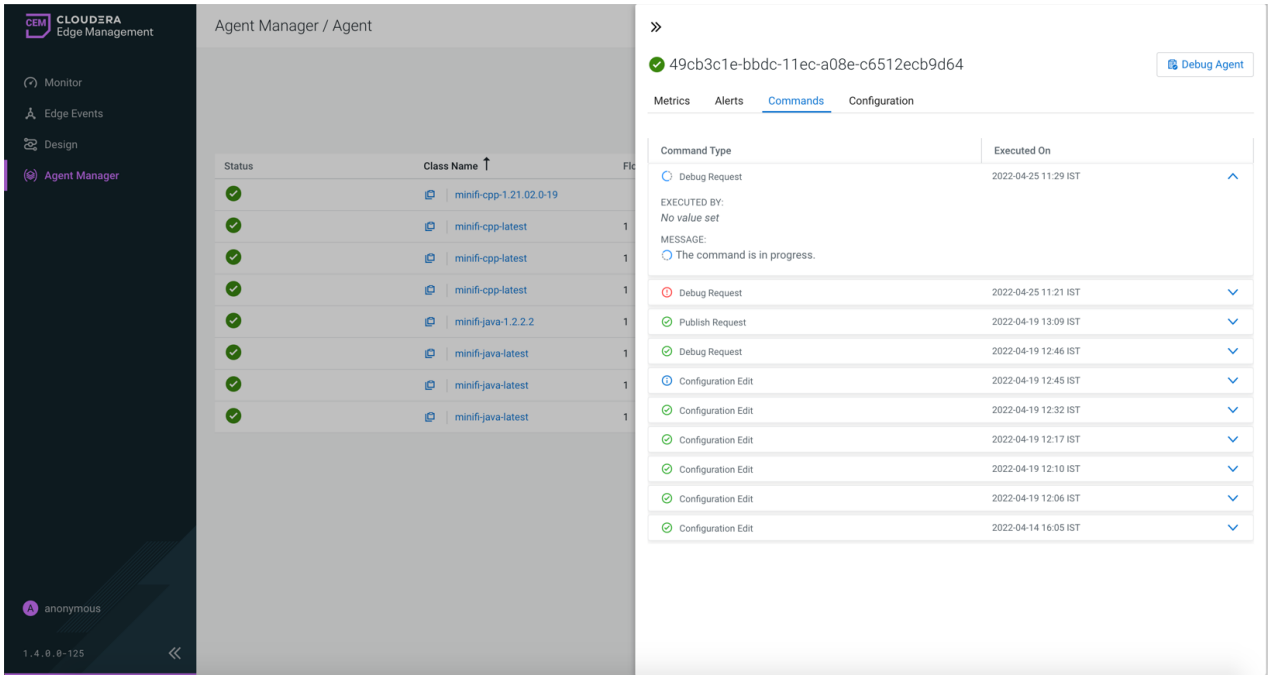


When a dataflow is started, the menu item changes to Stop Flow, allowing you to stop the running flow.

Debugging agent in Cloudera Edge Management

The debug command functionality allows you to collect debug information from agents using the C2 protocol.

Click Debug Agent to retrieve a debug bundle from the agent. 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 server.

When the files are available in the 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 C++ 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

Viewing the agent manifest

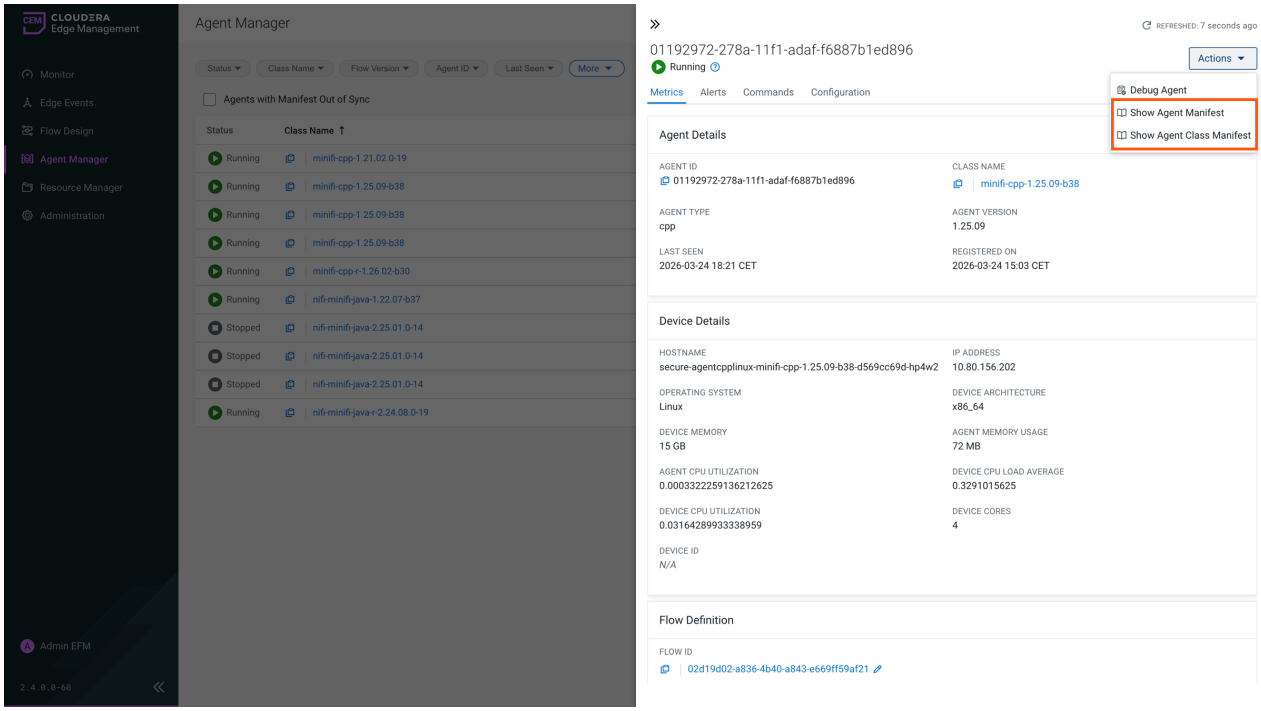
You can view the agent manifest to inspect the configuration and capabilities of an agent or an agent class, including details used during deploying or running a flow.

1. From the Agent Manager # View Agent Details at the end of an agent row to open the agent details pane.
2. Click Actions Show Agent Manifest or Show Agent Class Manifest.

The agent manifest contains the agent's configuration and capabilities.

The agent class manifest contains configuration and capability details for the associated agent class.

You can copy and download the manifest files.

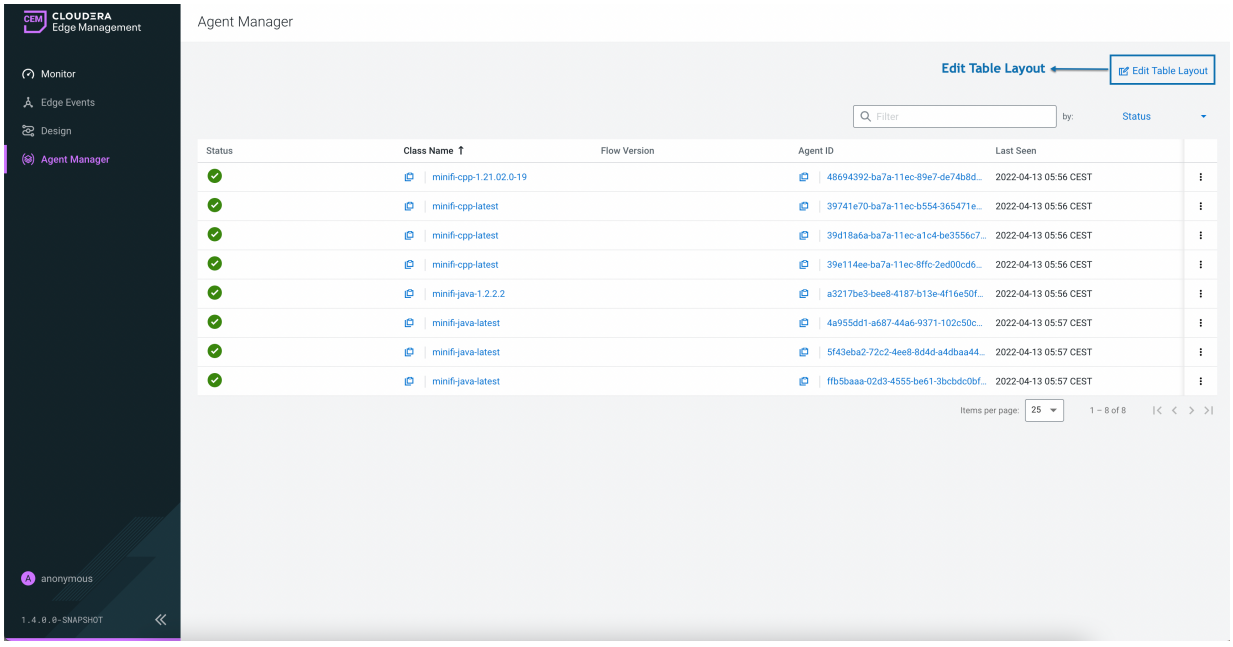


Customizing the agent list view in the Agent Manager

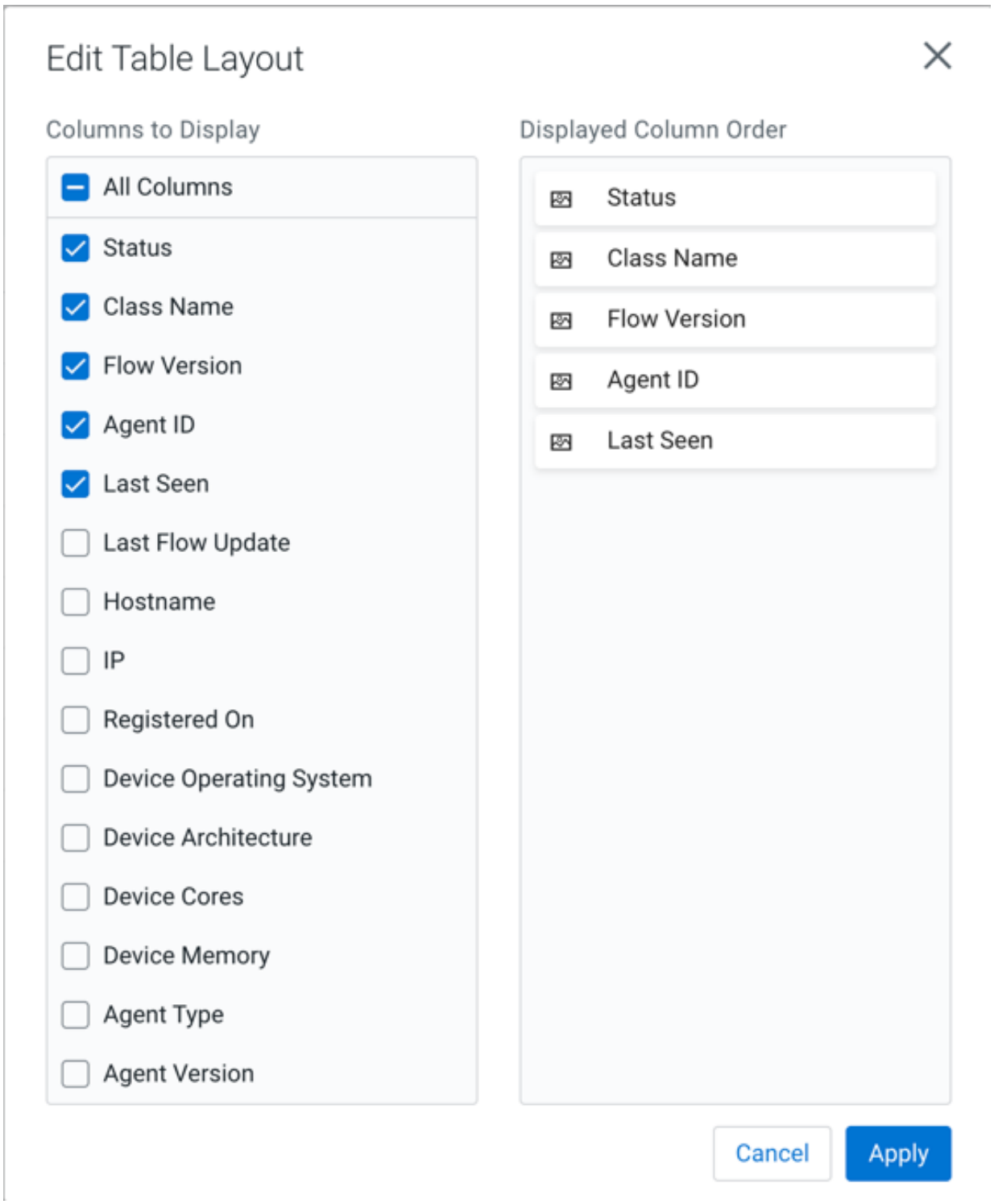
Learn how to customize the agent listing view to display the information most relevant to your specific requirements.

The table layout editor allows you to select which columns are displayed and change the order of columns.

1. To open the editor, go to the Agent Manager page and click Edit Table Layout in the top-right corner.



The Edit Table Layout dialog appears:



2. Customize the displayed columns.

   In the Columns to Display section, you can select the columns you want to show and clear the selection for the ones you want to hide.
3. In the Displayed Column Order section, drag and drop columns to change their order.
4. Click Apply to update the table layout.

The selected layout is reflected in the page URL. You can bookmark or share this URL to reuse the same table configuration.

Working with dataflows in Cloudera Edge Management

Dataflows in Cloudera Edge Management define how data is collected, processed, and transferred across edge environments. Using a visual flow designer, you can create pipelines by connecting processors that ingest, transform, and route data between systems. Dataflows provide a structured and versioned way to design, deploy, and manage data movement at the edge.

Building a dataflow in Cloudera Edge Management

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

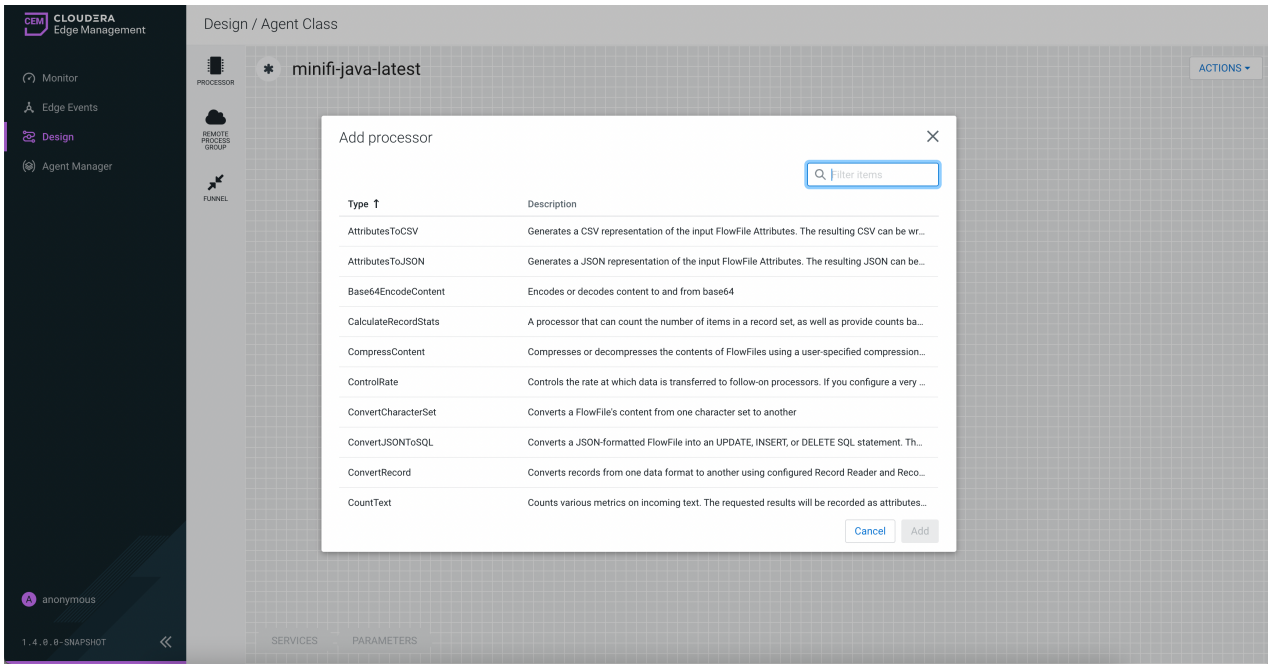
For additional information about flow creation and related concepts, check out the video on the Cloudera Edge Management YouTube playlist:https://www.youtube.com/embed/XCeJsJt5itc

Adding components to the canvas in Cloudera Edge Management

Learn how to add each of the components available in the Components Toolbar in the Edge Flow Manager UI. You can add processors, remote process groups, and funnels.

Processor

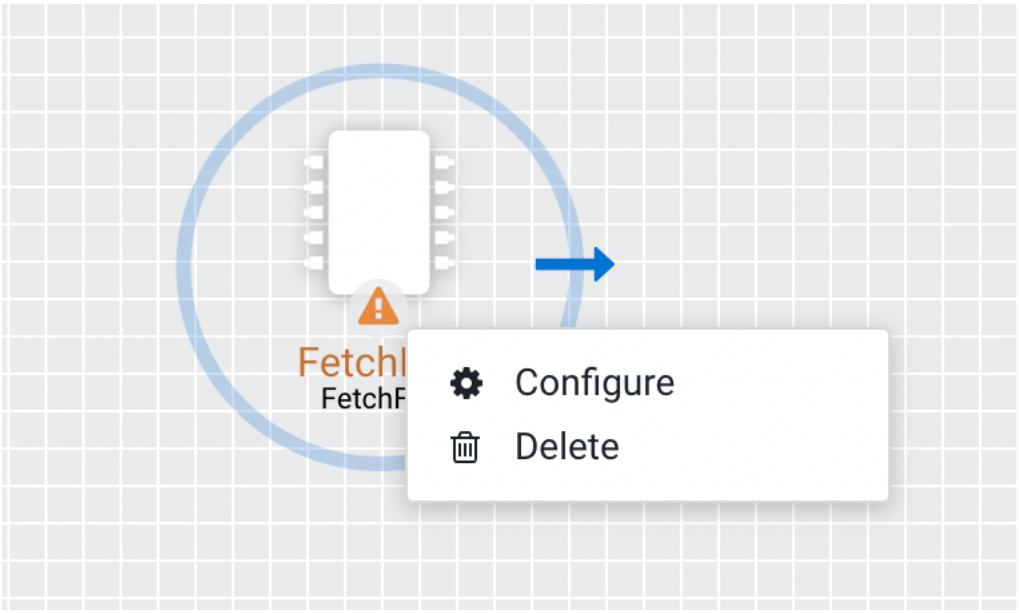
The processor is the most commonly used component, as it is responsible for data ingress, egress, routing, and manipulating. There are many different types of processors. When you drag a processor onto the canvas, the Add Processor dialog appears, as shown in the following image, which allows you to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

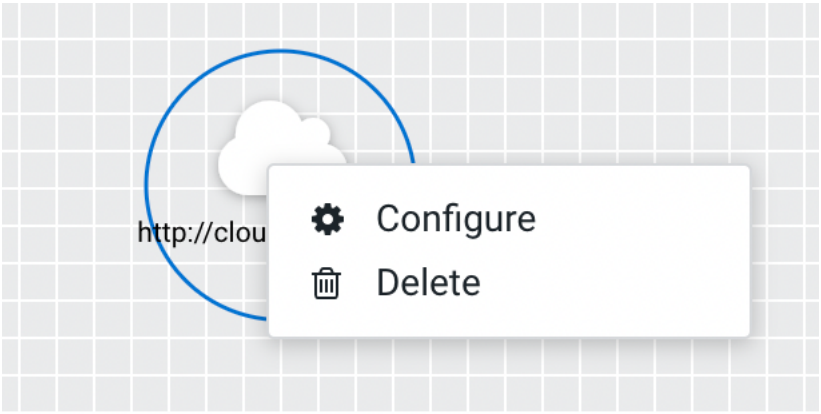
After you drag a processor onto the canvas, you can configure properties of the processor, parameterize processor property values, or delete the processor. To configure properties, double-click on the processor, or right-click on the processor and select Configure from the context menu. To delete a processor, 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.

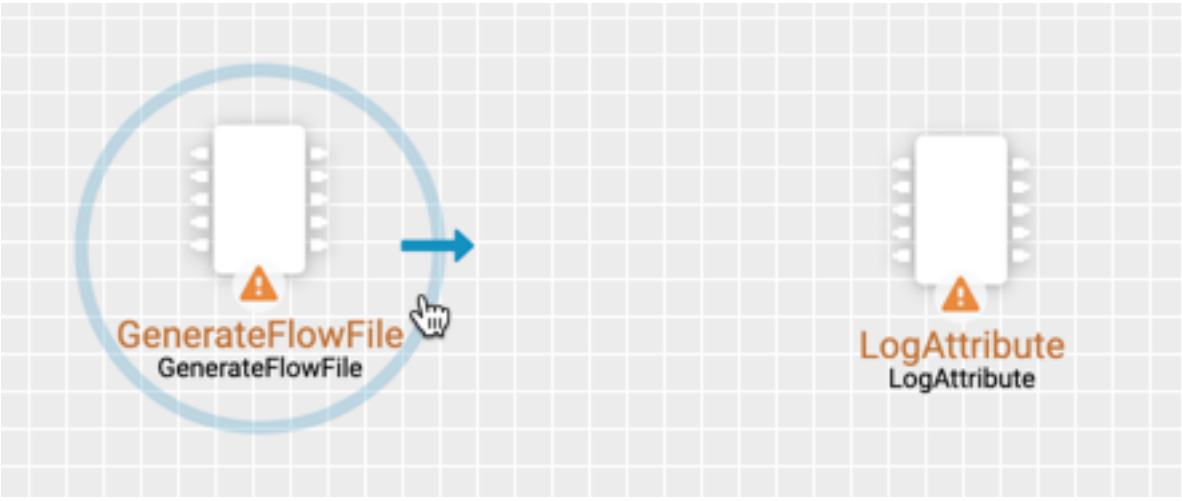
Connecting components in Cloudera Edge Management

After you add processors and other components to the canvas of the Edge Flow Manager 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

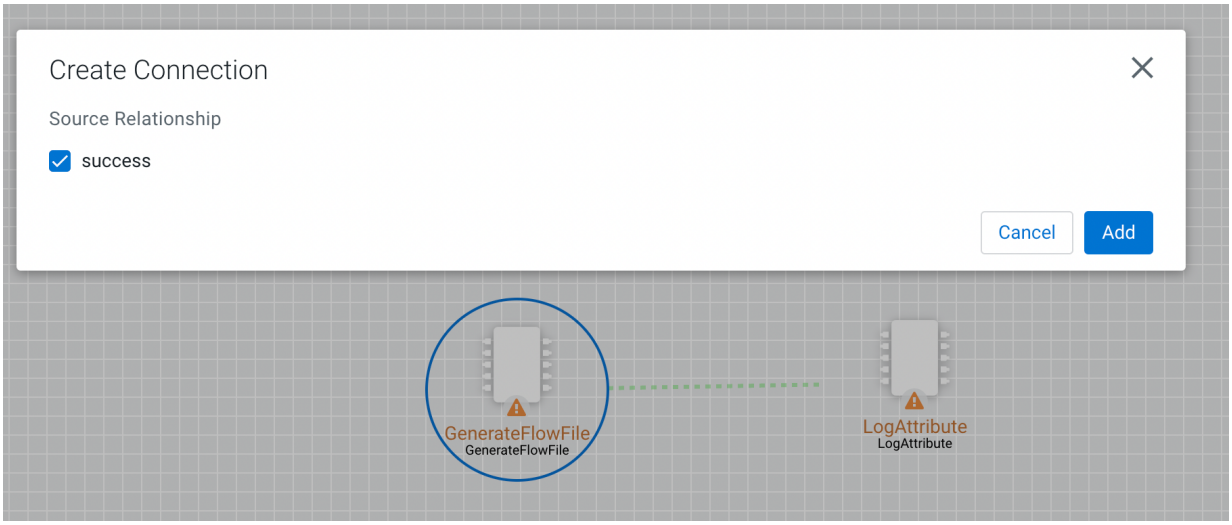
1. Hover the mouse over a component.

   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 processor in Cloudera Edge Management

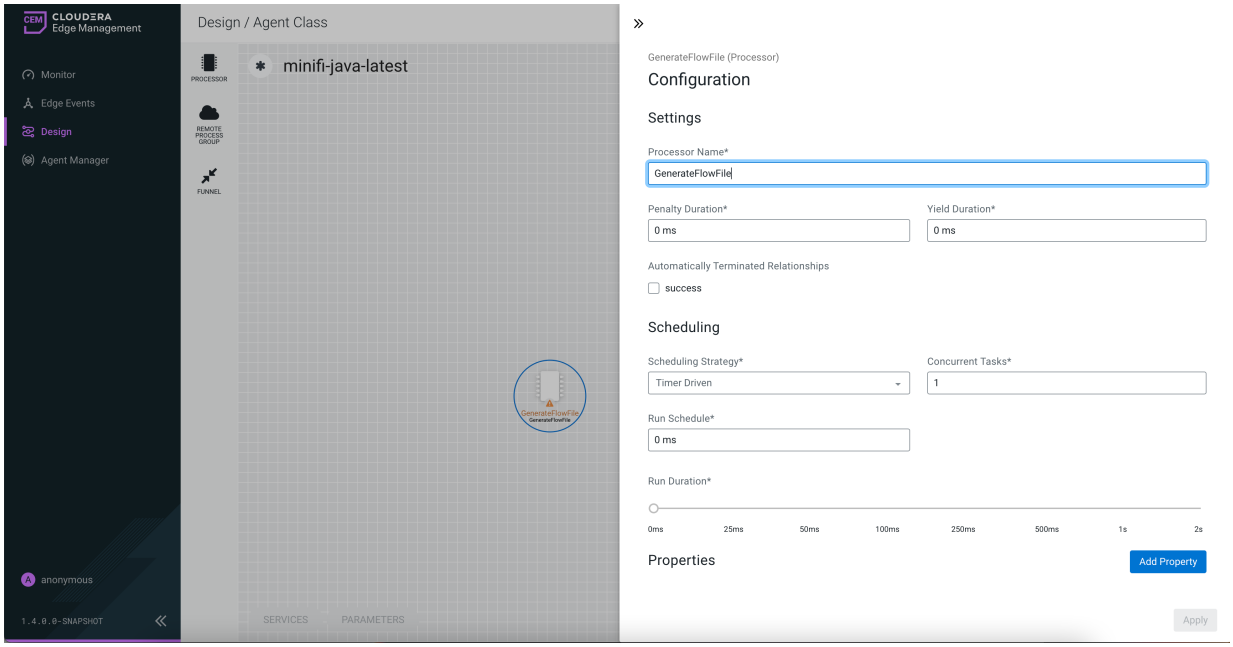
Learn how to configure a processor using the Edge Flow Manager UI in Cloudera Edge Management.

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.
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.

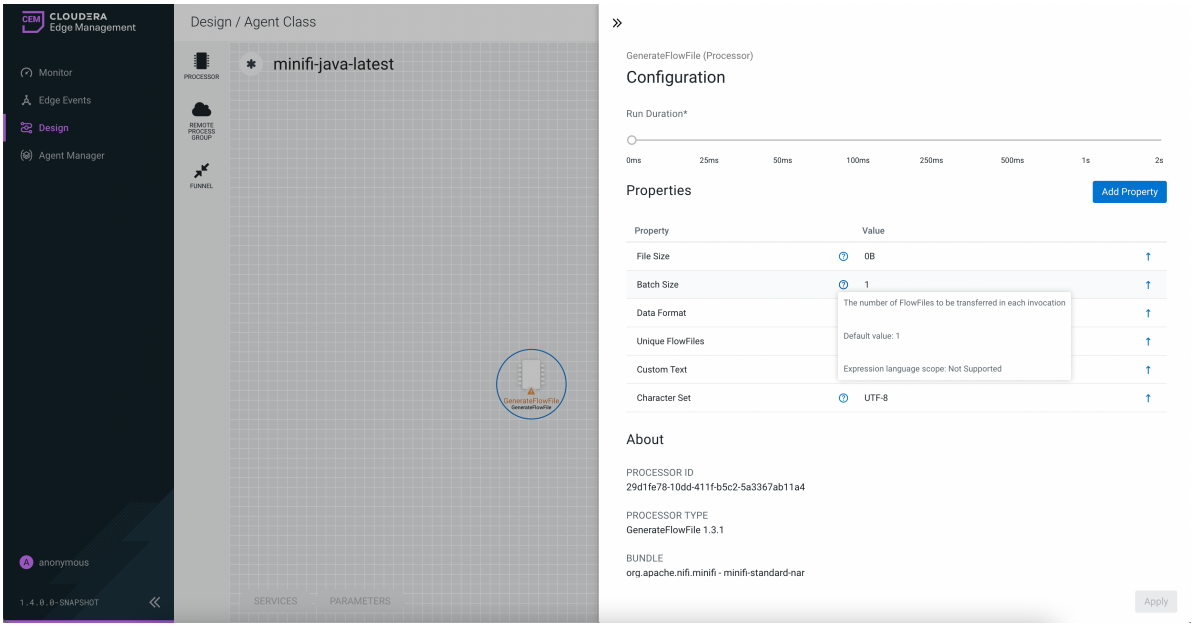
- Scheduling. The Scheduling section contains the following configuration items:

Properties	Description
Scheduling Strategy	There are two options for scheduling components: • 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.

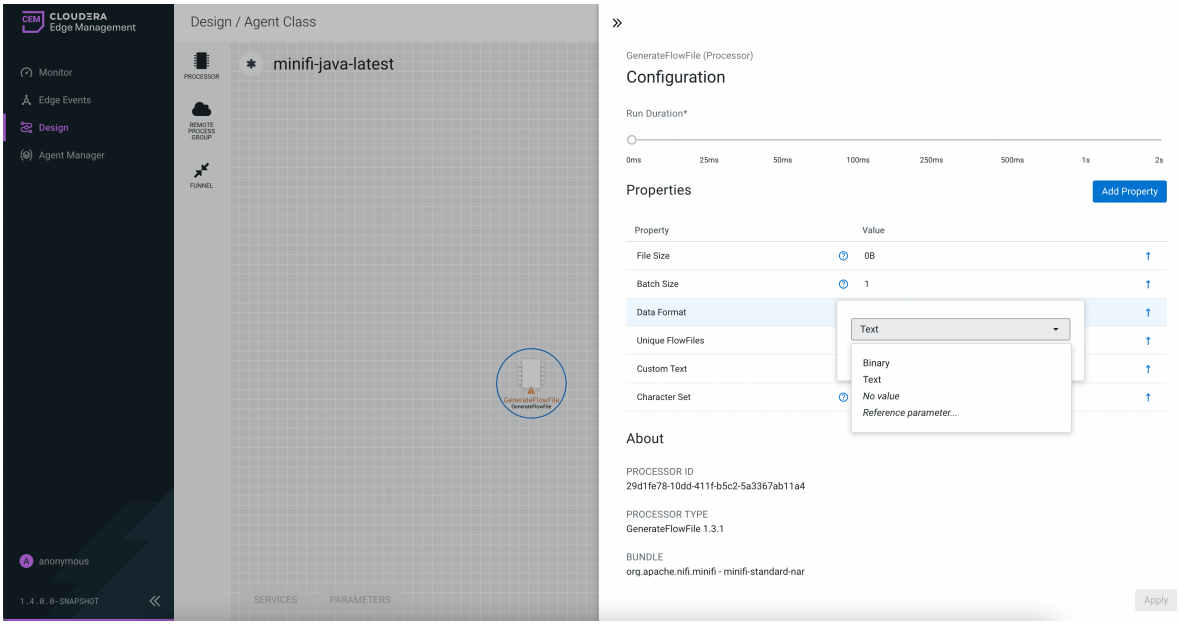
Properties	Description
Run Schedule	This dictates how often the processor must be scheduled to run. The valid values for this field depend on the selected scheduling strategy (see above). When you select the Event Driven scheduling strategy, this field is not available. When you select the Timer Driven 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.

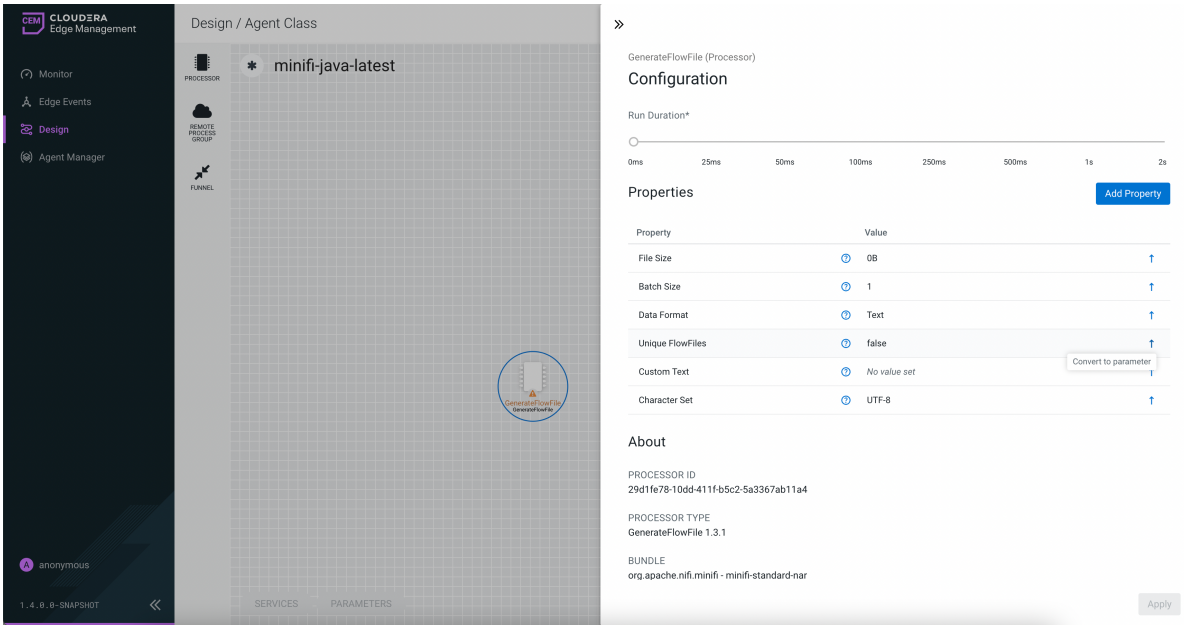
  A GenerateFlowFile processor, by default, has four properties including Batch Size, Data Format, File Size, and Unique FlowFiles. Next to the name of each property, there appears a small question-mark symbol ( ) indicating that additional information is available. Hovering over this symbol with the mouse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



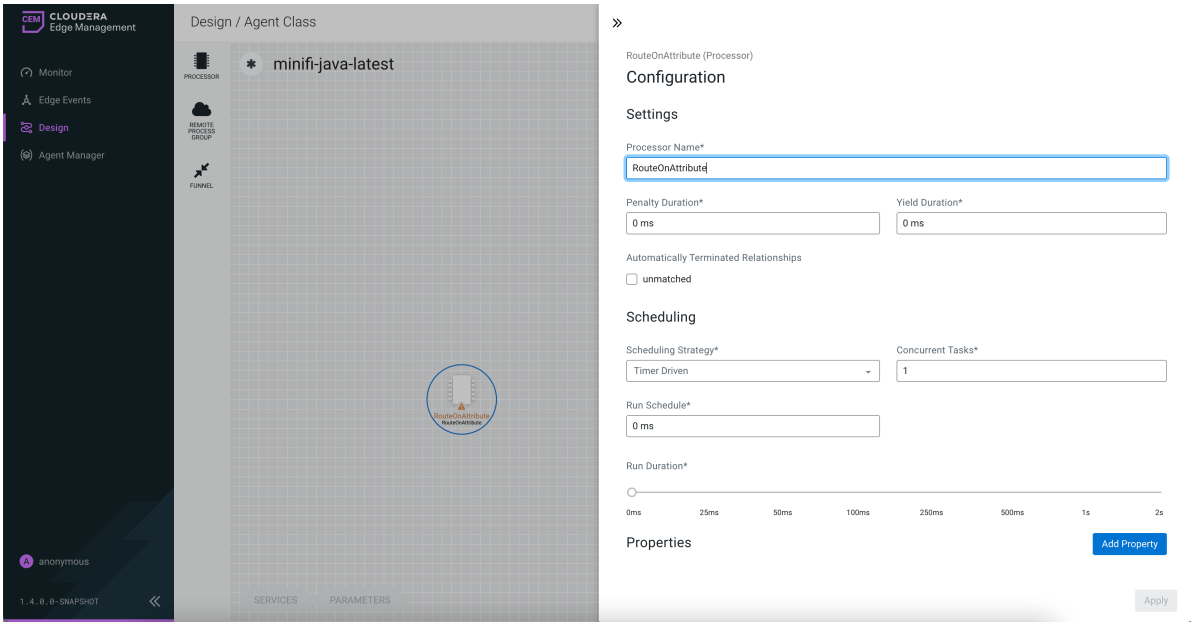
Clicking on the value for the property allows you to change the value. Depending on the values that are allowed for the property, you are either provided a drop-down from which to choose a value, or a text area to type a value. Here is an example of GenerateFlowFile processor with the drop-down for the 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:

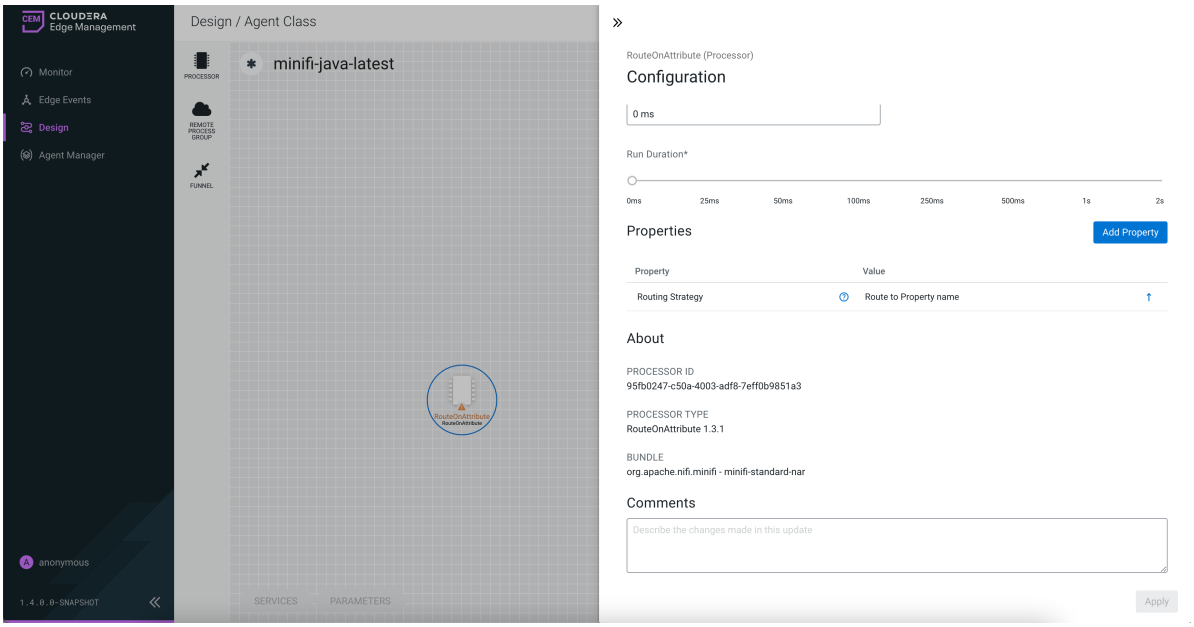


For some processors, there appears an Add Property button, beside the Properties section, for adding a user-defined property. When you click this button, a dialog opens, which allows you to enter the name and value of a new property. Not all processors allow user-defined properties. The RouteOnAttribute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

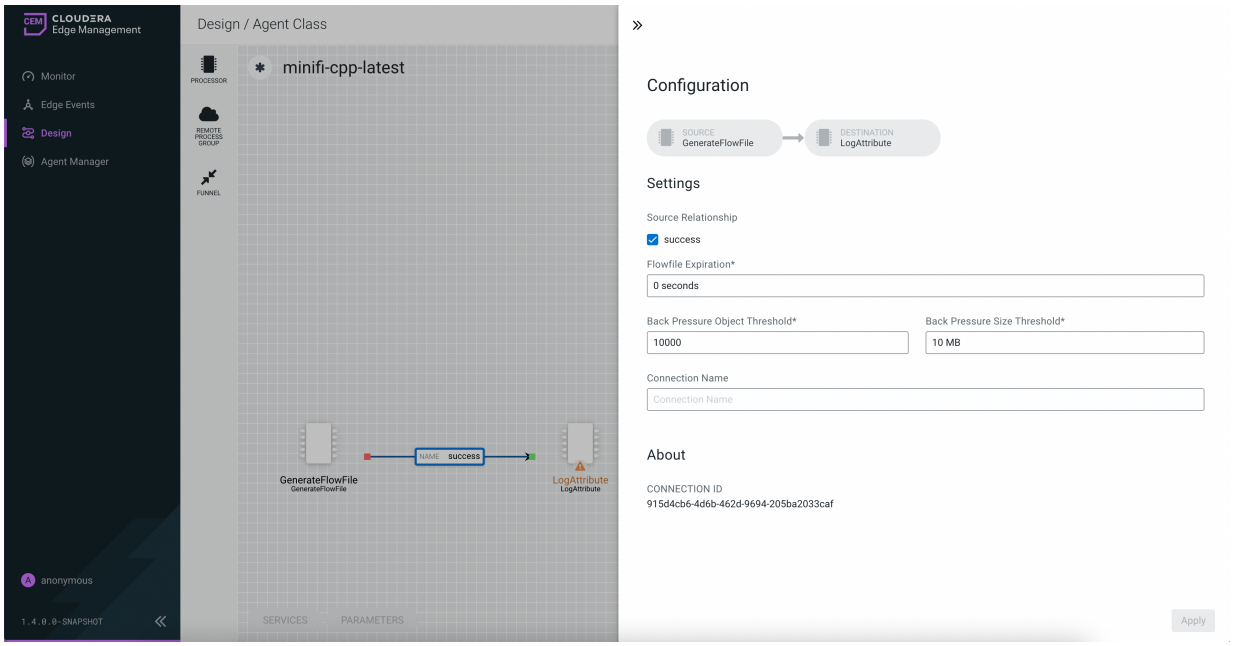
Configuring a connection in Cloudera Edge Management

After you create a connection, you can change the configuration properties of the connection or move the connection using the Edge Flow Manager UI in Cloudera Edge Management.

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.

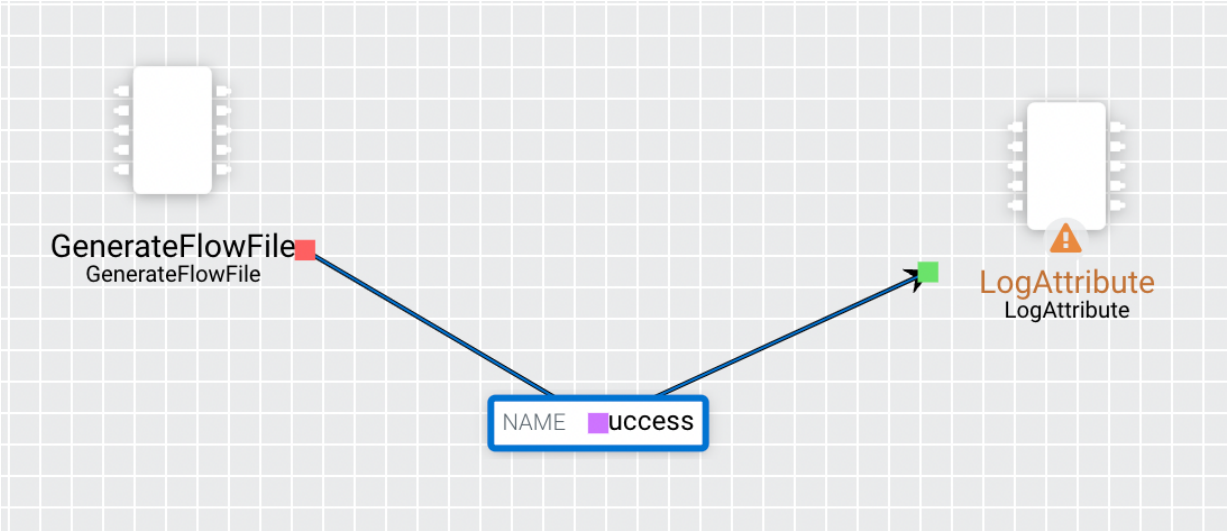
Bending connections in Cloudera Edge Management

Learn how to bend a connection using the Edge Flow Manager UI in Cloudera Edge Management.

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.

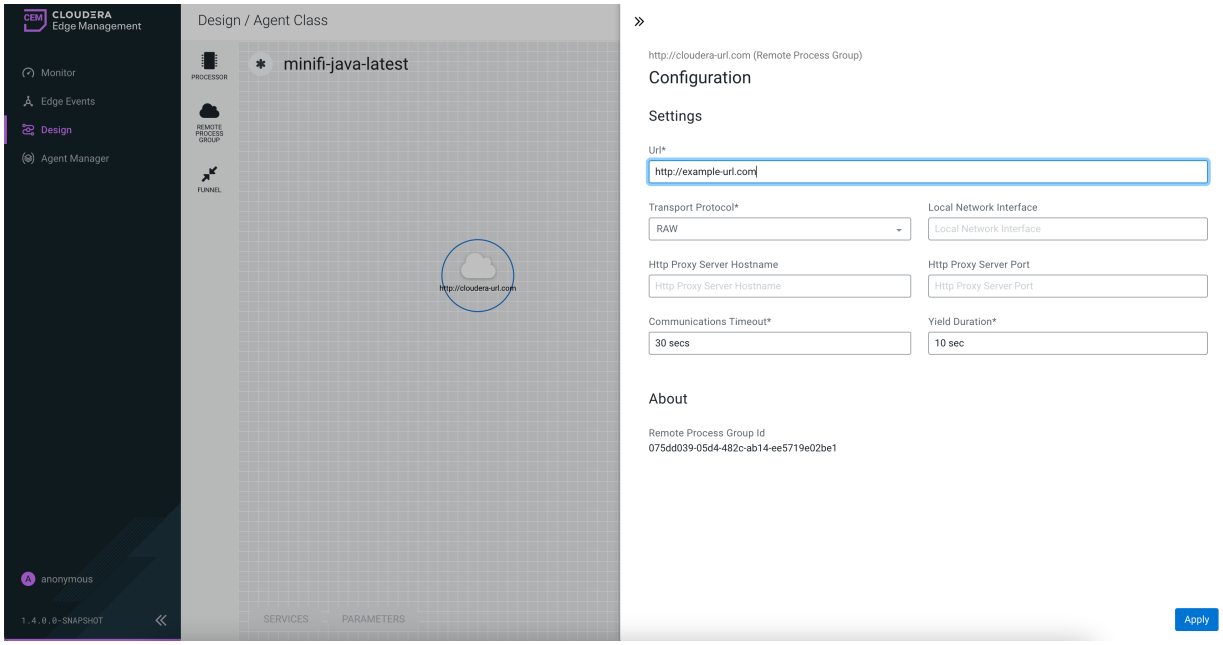
Configuring a remote process group in Cloudera Edge Management

Learn how to configure a remote process group using the Edge Flow Manager UI in Cloudera Edge Management.

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: • 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.
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.

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

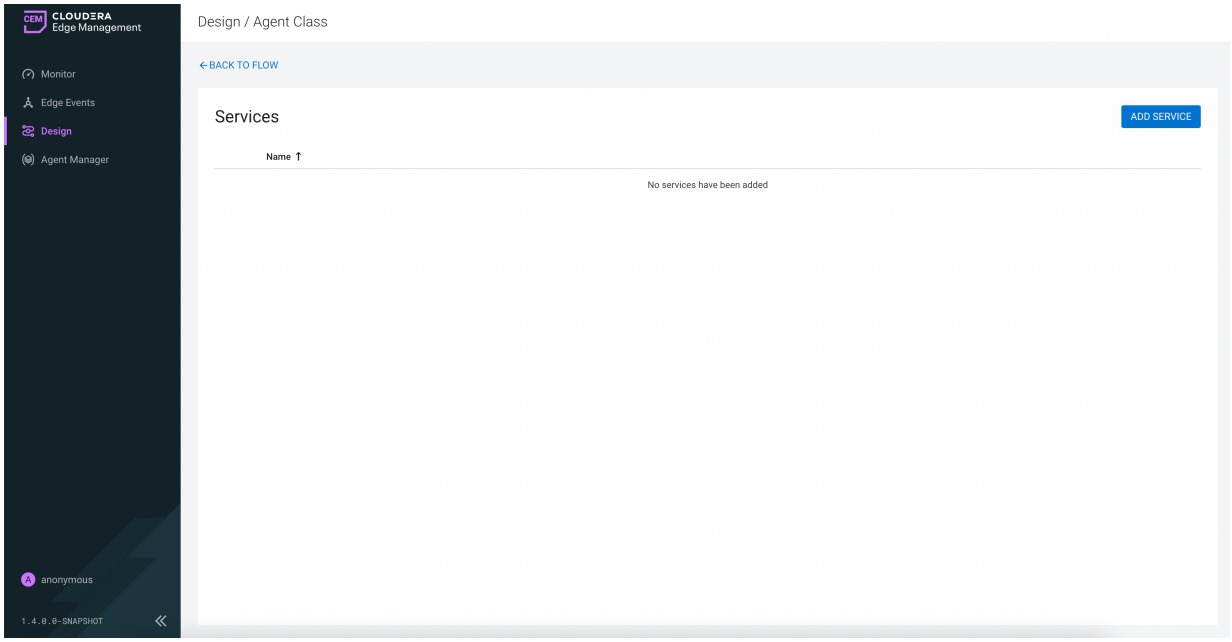
Adding services in Cloudera Edge Management

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 Edge Flow Manager UI in Cloudera Edge Management.

Procedure

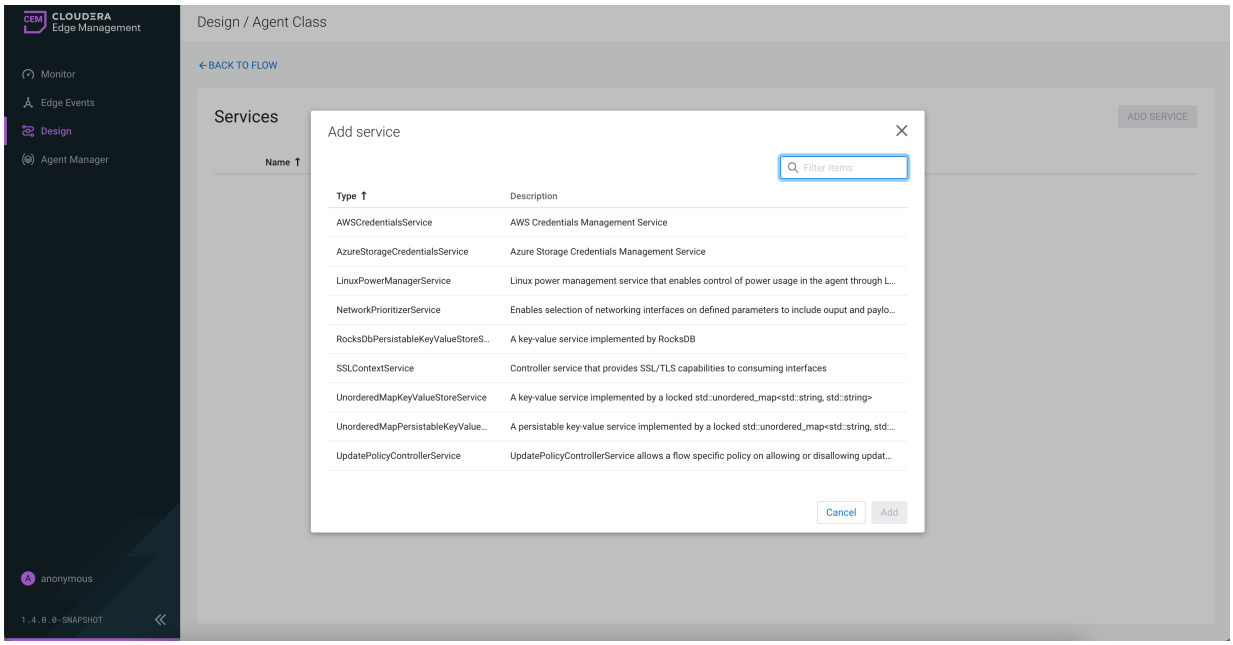
1. To add a service, click the SERVICES button at the bottom-left corner of the canvas, or simply right-click on the canvas and select Services.

   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:

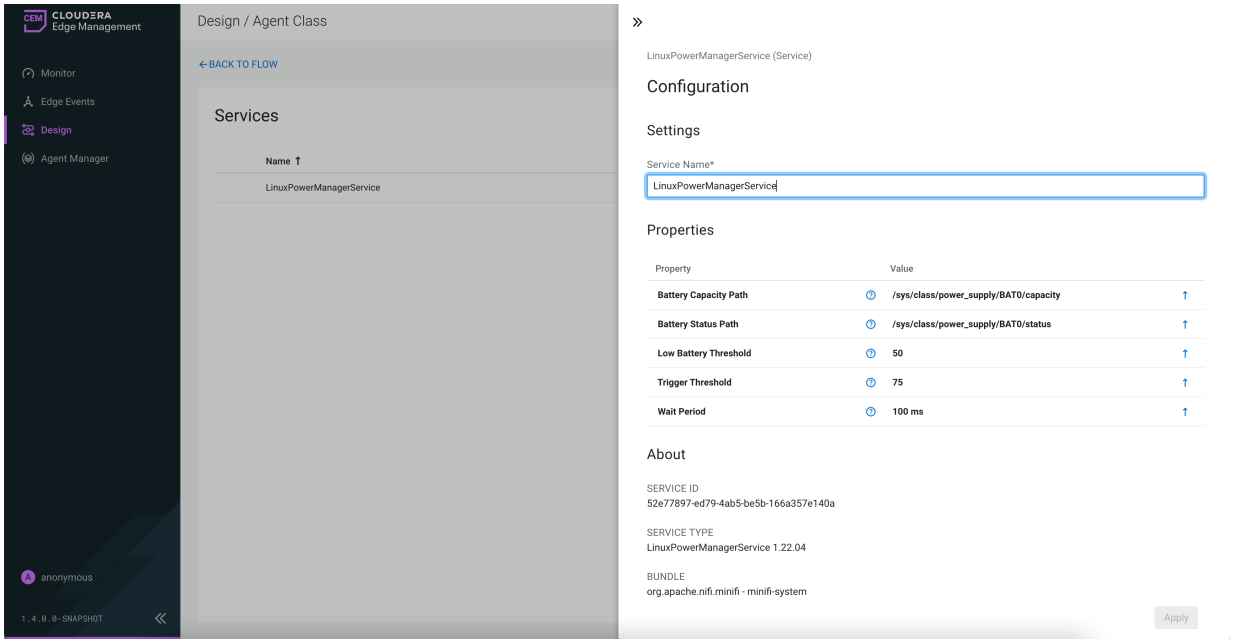


3. Select the service you want to add and click Add, or simply double-click on the name of the service to add it.

   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.

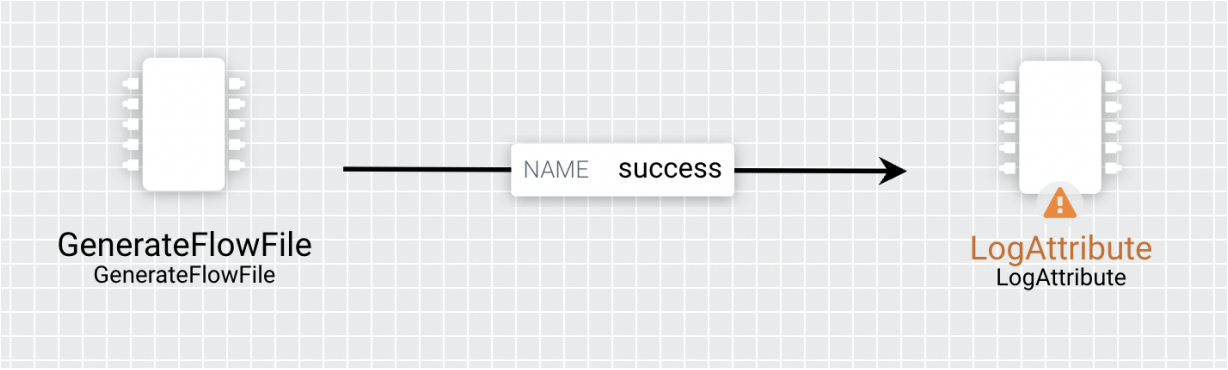
Example dataflow in Cloudera Edge Management

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 in action.

Procedure

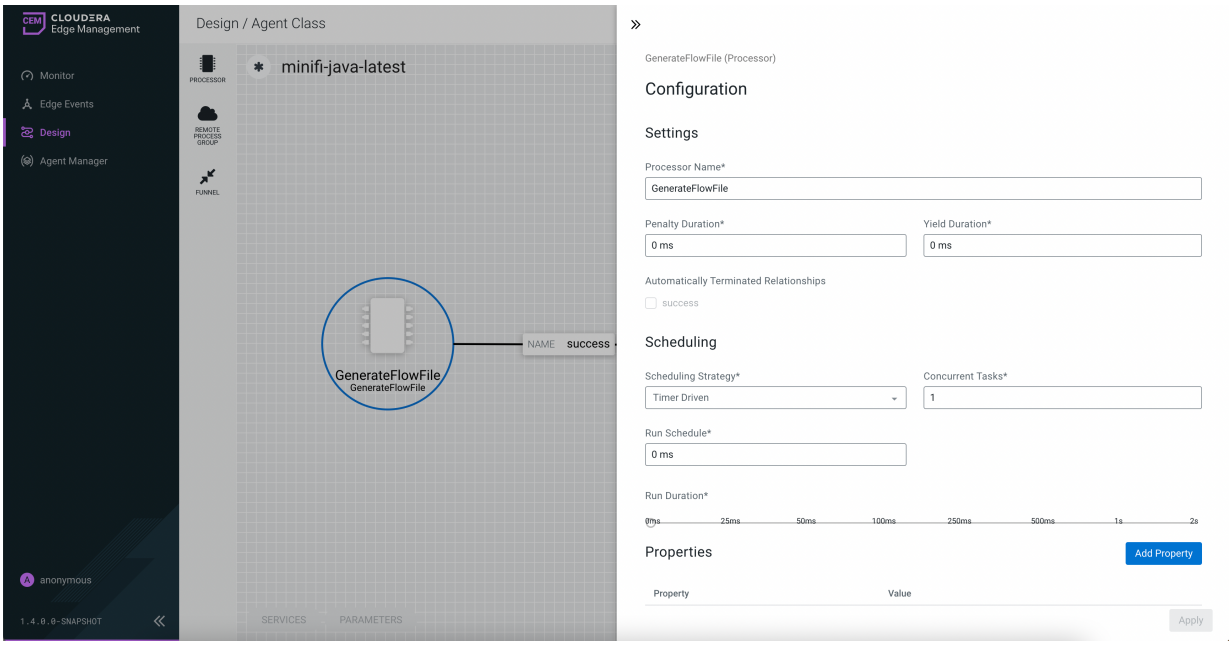
1. Drag the GenerateFlowFile and LogAttribute processors to the canvas and connect them (using the guidelines provided above).

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



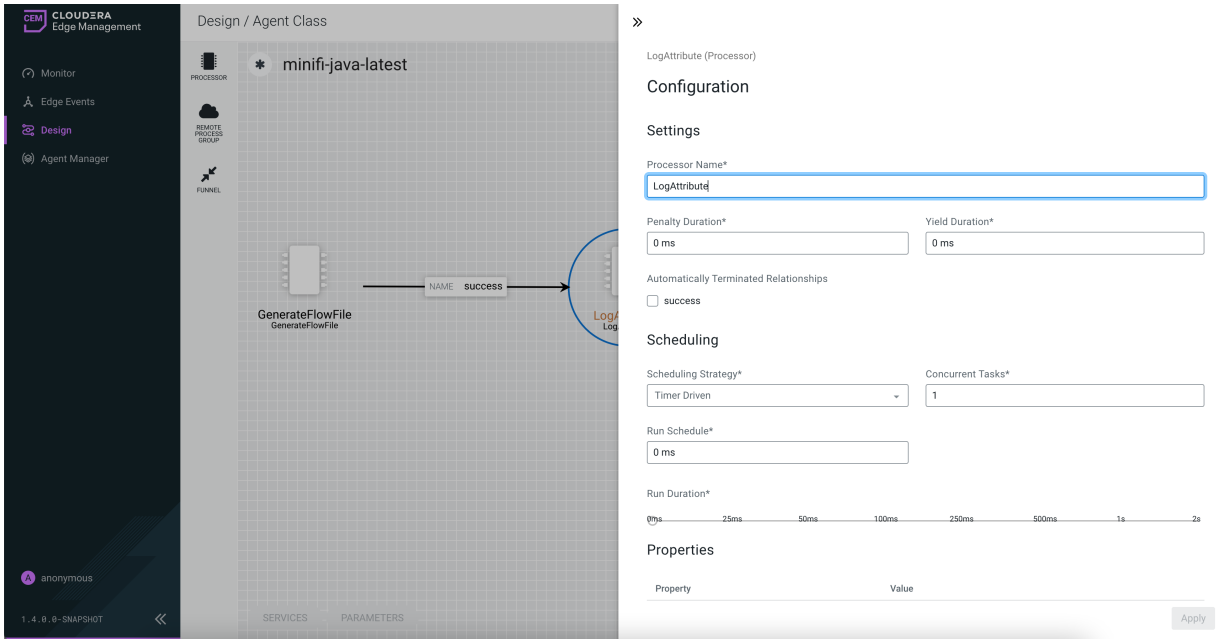
2. Configure the GenerateFlowFile processor and click Apply to apply the changes.
   - In the Scheduling section, set Run Schedule to: 5 sec. Note that the GenerateFlowFile processor can create many FlowFiles very quickly. Therefore, setting the Run Schedule is important, so that the flow does not overwhelm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

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



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.

Managing dataflows in Cloudera Edge Management

Publishing a dataflow in Cloudera Edge Management

Learn how to publish a dataflow using the Edge Flow Manager UI in Cloudera Edge Management.

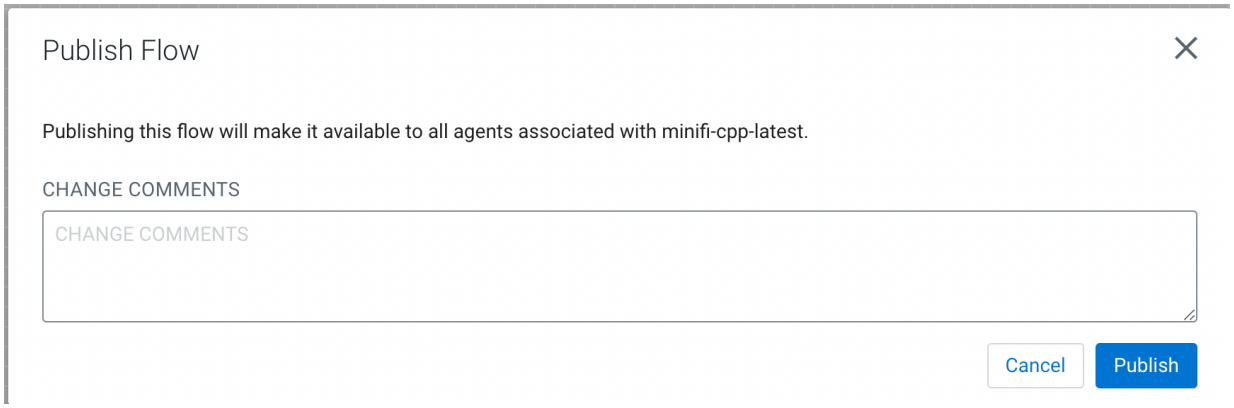
About this task

Note: Publishing is an asynchronous process and agents update their flow as they periodically heartbeat to the Cloudera Edge Management 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:



2. Enter comments if desired, and click Publish.

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

Reverting a dataflow in Cloudera Edge Management

Learn how to revert a dataflow using the Edge Flow Manager UI in Cloudera Edge Management.

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:

2. Select Revert to complete the process.

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

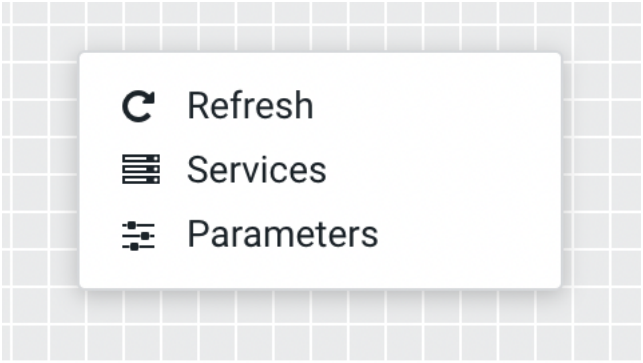
Refreshing a dataflow in Cloudera Edge Management

Learn how to refresh a dataflow using the Edge Flow Manager UI in Cloudera Edge Management.

Procedure

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

The following image shows the Refresh option:



Managing flow versions in Cloudera Edge Management

Learn how to undo and redo changes in your draft dataflows, view and load published flow versions, and switch between versioned flows using the Cloudera Edge Management UI in Cloudera Edge Management.

Undo/redo draft changes

You can undo changes in the Cloudera Edge Management Flow Designer made after the last publish event, allowing you to revert accidental edits. Similarly, you can redo changes that were previously reverted.

Procedure

1. Locate the Undo/Redo icons at the bottom right corner of the Flow Designer.
2. Click  to revert changes made since the last flow publish event, or click  to reapply changes that were previously undone.

   Note: These functions do not affect the Flow Parameters functionality.

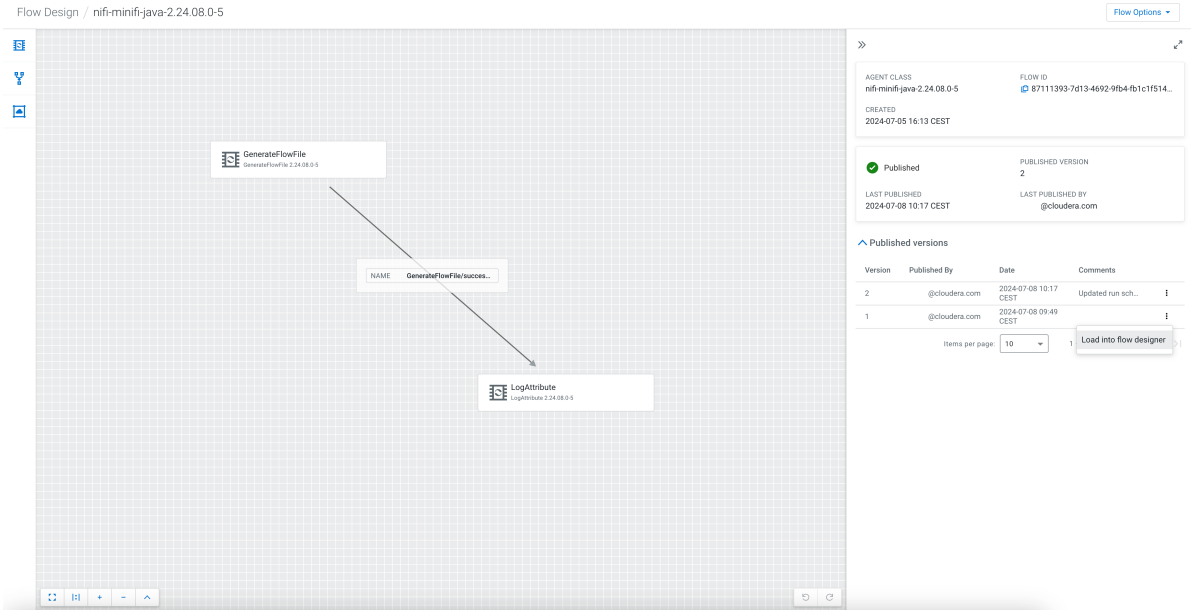
Switch between published flow versions

You can view and load previously published flow versions in the Cloudera Edge Management Flow Designer.

Procedure

1. View published flow versions.
   a) Open the right-side navigation pane when no component is selected.
   b) Review the list of previously published versions.

   Each entry includes basic information like the publish date, publisher, and any comments added while publishing the flow.



2. Load a previous flow version.
   a) In the right-side navigation pane, find the version you want to load.
   b) Click the Load into flow designer option on the selected line.

   This action replaces the current flow and parameters in the designer with the selected version.

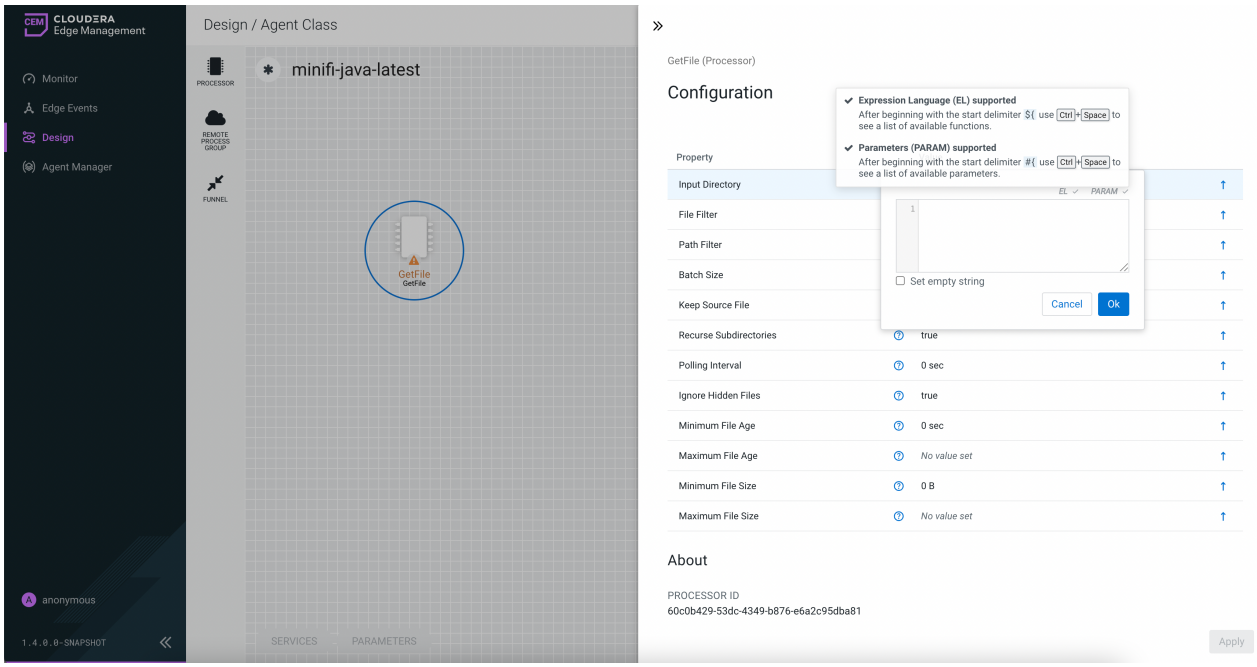
   Note: Loading a previous flow version into the designer does not automatically publish it. If you want to deploy the selected version to the agents, you need to do it manually using Flow Options Publish .

Working with parameters in Cloudera Edge Management

Parameters allow you to parameterize the values of processors and service properties in a flow including sensitive properties. This helps you manage configurations more efficiently and avoid duplicating values across components.

You can create and manage parameters directly in the Edge Flow Manager UI.

When configuring a property, you can check whether it supports parameters by clicking in the value field. Eligibility indicators show whether parameters are supported( ) or not ( )



Note: Properties that reference services cannot use parameters.

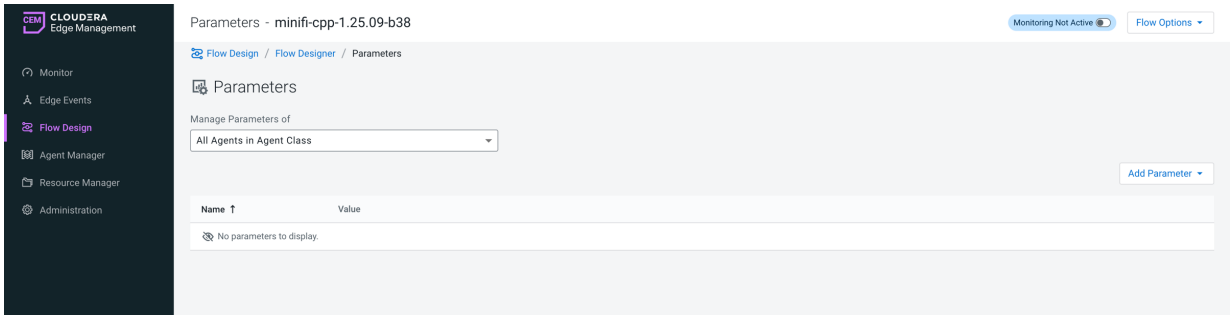
Adding parameters in Cloudera Edge Management

Learn how to add parameters using the Edge Flow Manager UI in Cloudera Edge Management.

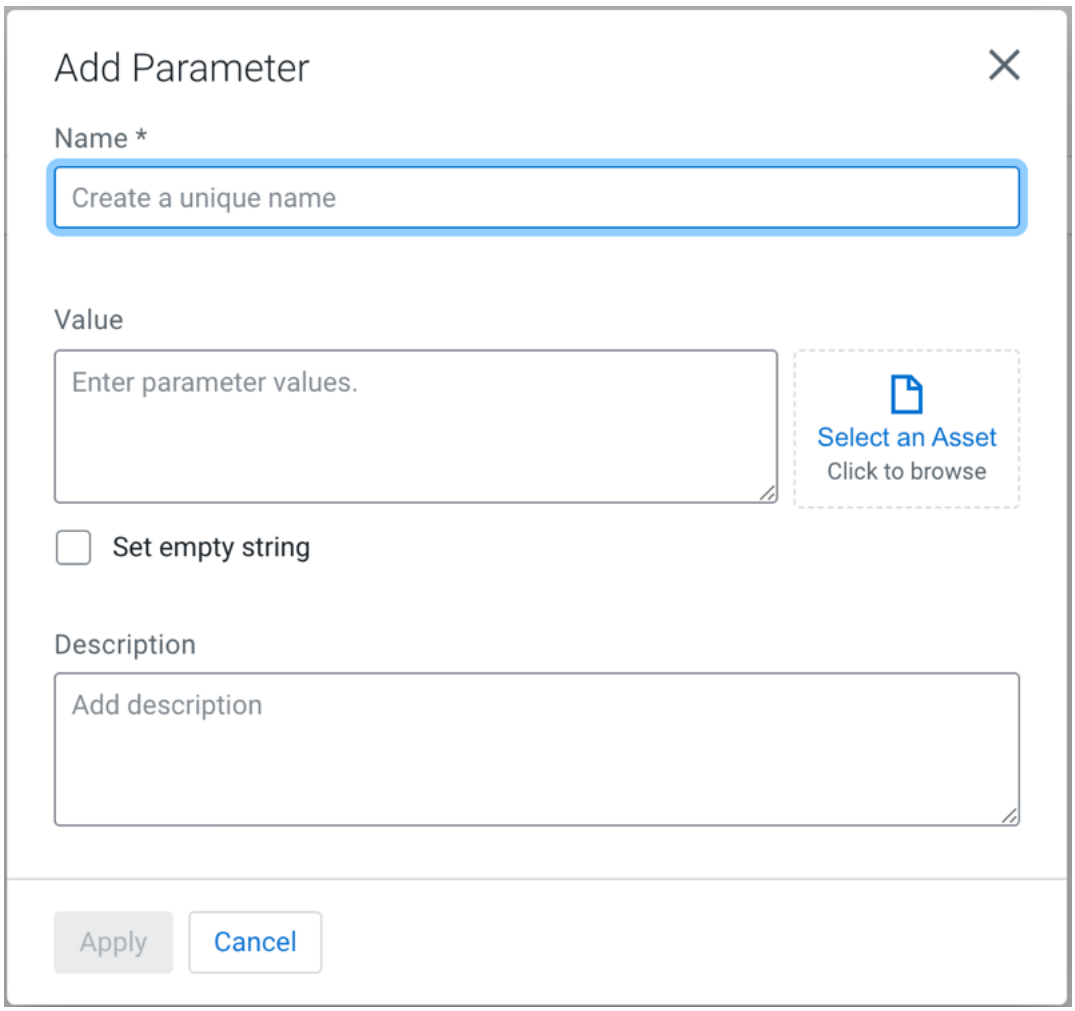
Procedure

1. In the Flow Designer, click Flow Options Parameters in the top-right corner.

   The **Parameters** page opens.



2. To add a new parameter, click ADD PARAMETER.



3. Configure the following properties.

Property	Description
Name	Enter a name for the parameter. Only alpha-numeric characters (a-z, A-Z, 0-9), hyphens (-), underscores (_), periods (.), and spaces are allowed.
Value	Enter the value 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 the parameter value to an empty string.
Sensitive Value	Specify whether the parameter value is sensitive. If enabled, the value of the parameter is not shown in the UI after applying. This setting cannot be changed after the parameter is created.
Description	(Optional) Enter a description of the parameter.

4. Click Add to add the parameter.
5. Click Apply to save the changes.

Referencing assets in flow parameters

When creating a parameter, you can define a static value or use an asset from the Resource Manager. If you combine asset management with parameterization by creating an asset-based parameter, you can decouple flow logic from environment-specific file paths while ensuring correct runtime resolution on agents.

About this task

Instead of referencing files by their physical location on the agent, you can define a parameter that points to an assigned asset. At runtime, Edge Flow Manager resolves this reference to the correct file path on each agent.

By using asset references you can reuse the same asset across multiple flows and agent classes and update the underlying asset without modifying the flow.

Before you begin

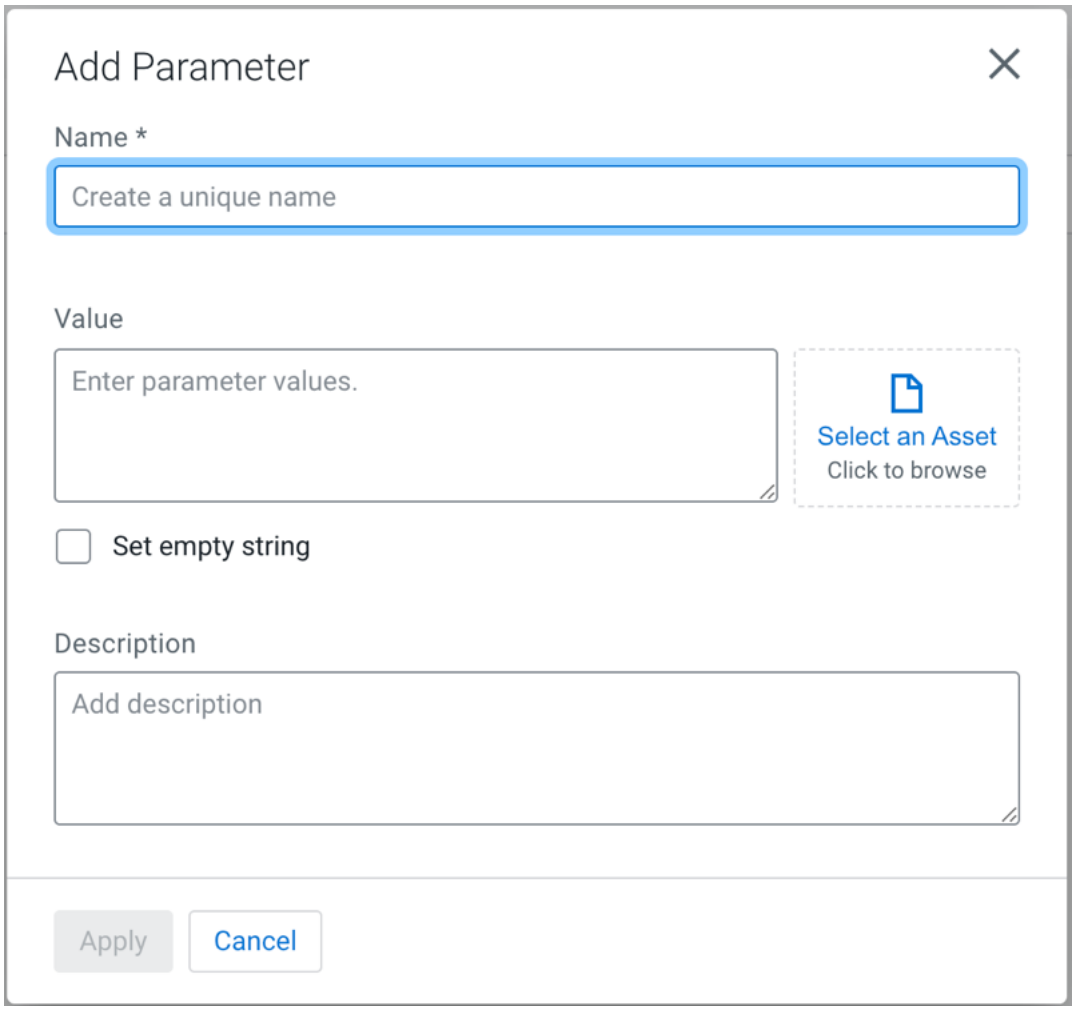
Before referencing an asset in a flow parameter, ensure that:

- The asset is uploaded in the Resource Manager.
- The asset is assigned to the target agent class.
- The asset is successfully synchronized to all agents in the class.

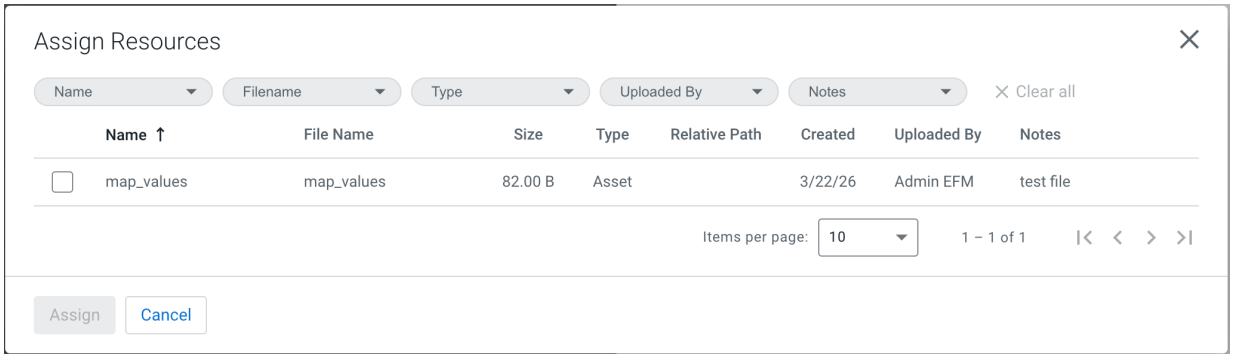
Procedure

1. Open your flow in the Flow Designer.
2. Go to Flow Options Parameters .
3. Click Add parameter.

   The Add Parameter dialog appears.



4. Click Select an Asset.



5. Choose an asset from the available list.
6. Click Assign.
7. Add a name for the parameter.

   Note: Only alpha-numeric characters (a-z, A-Z, 0-9), hyphens (-), underscores (_), periods (.), and spaces are allowed.

8. Add a description that explains what the parameter is for.
9. Click Apply to create the new parameter.
10. Click Apply to save your changes.

    Important:

    Each parameter can reference one asset.

Results

The parameter now represents the selected asset. You can use the parameter in a flow so the processor would use the asset indirectly through the parameter. When the flow runs on an agent:

- The agent detects that the parameter references an asset.
- The asset is already synchronized to the agent through the Resource Manager.
- The parameter is resolved to the local file path of the asset on the agent.
- The processor reads the file from that resolved path.

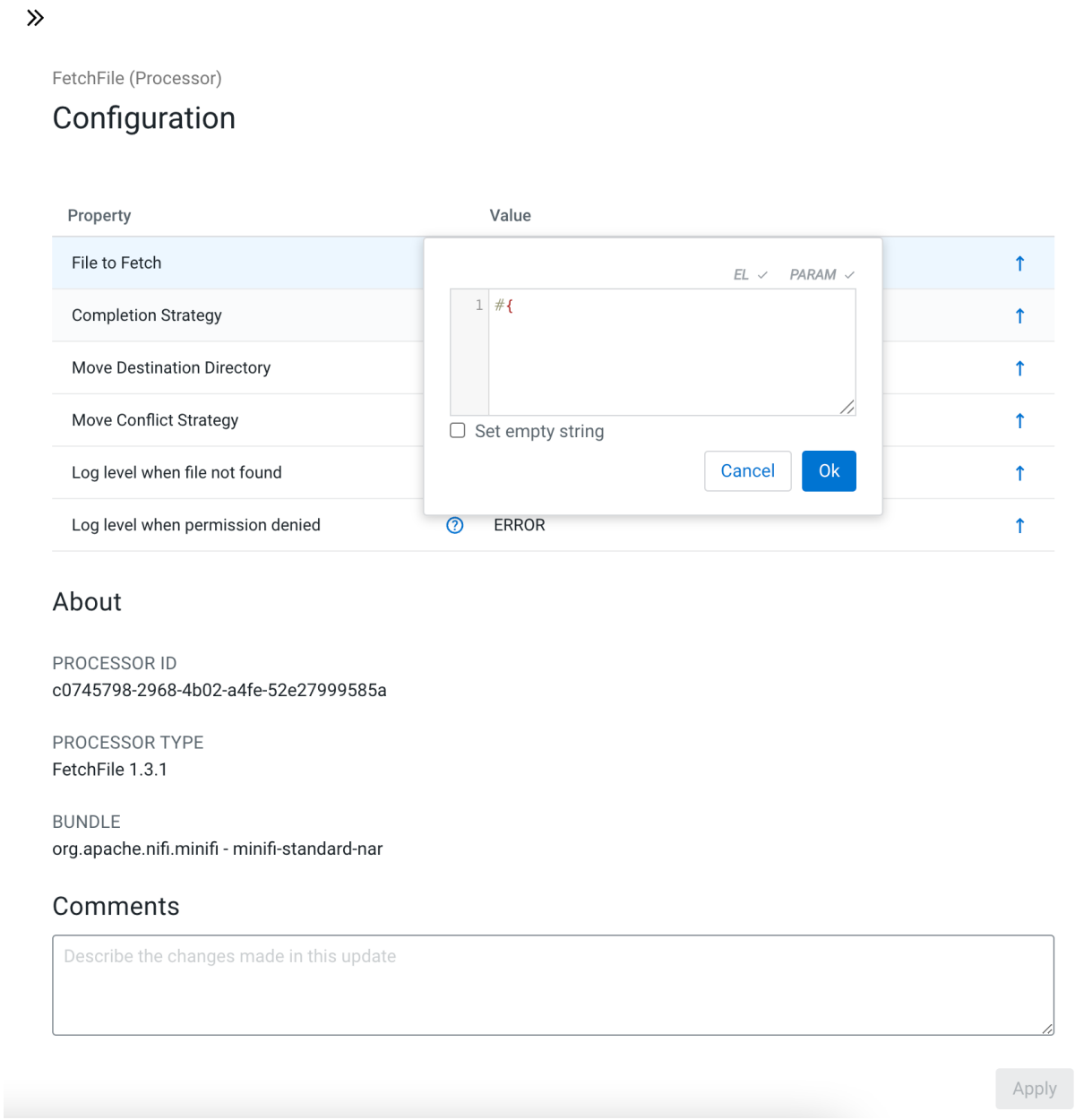
Using parameters in Cloudera Edge Management

Learn how to reference and use existing parameters while configuring flow components.

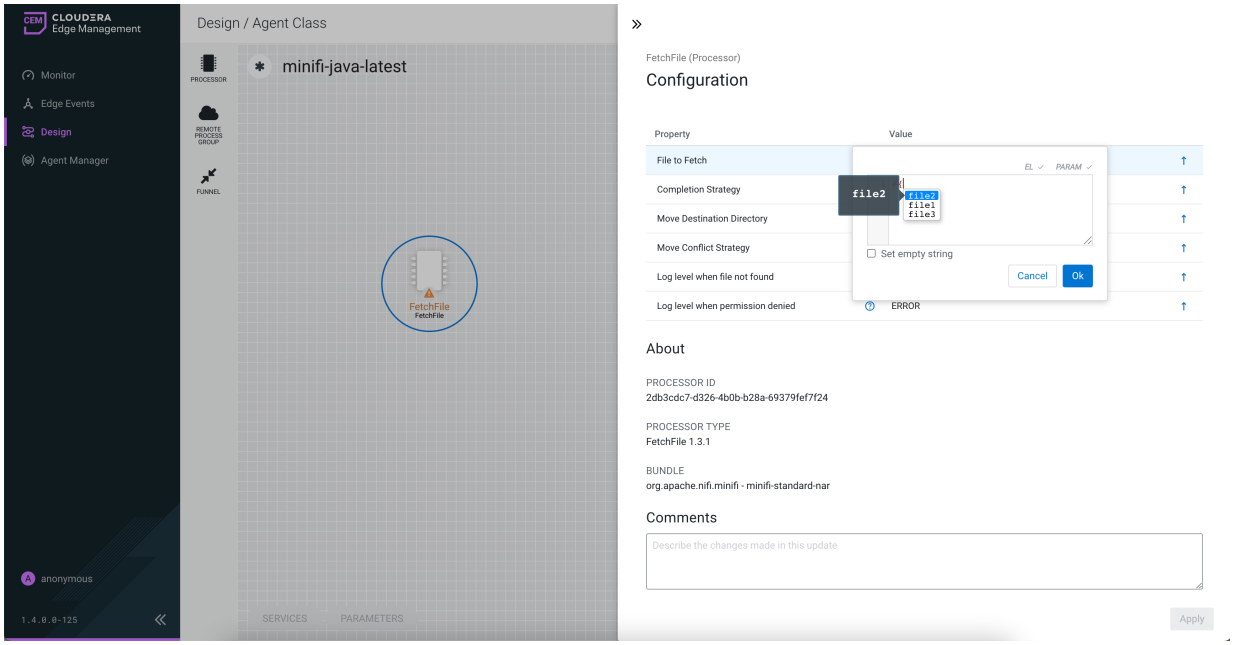
Referencing existing parameters

Existing parameters can be referenced for a processor or service property value during configuration.

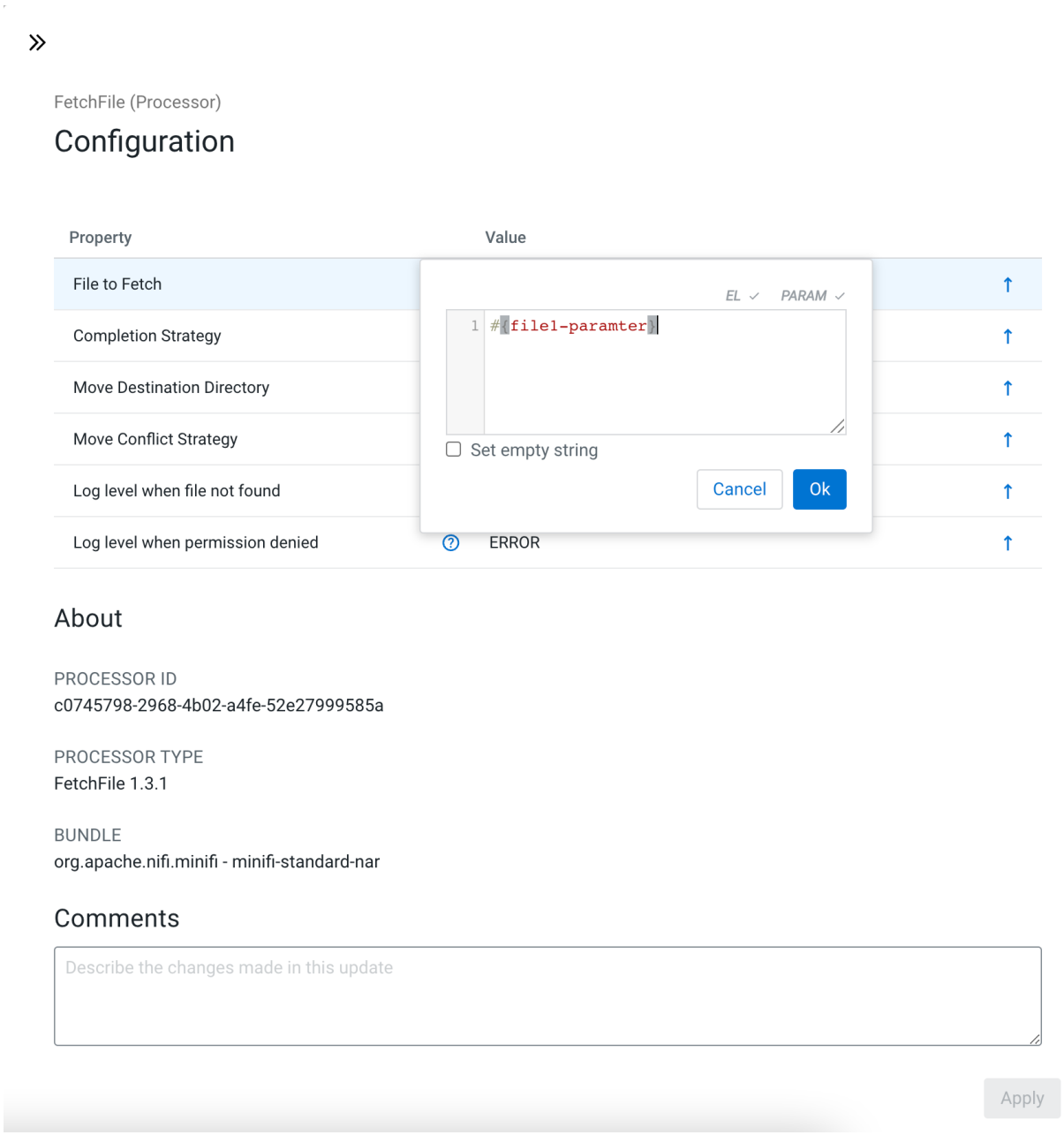
1. To reference an existing parameter, select the property value field and clear the default value if one exists.
2. Enter the start delimiter #{.



3. Enter the parameter name, or press Control+Space to view the list of available parameters.

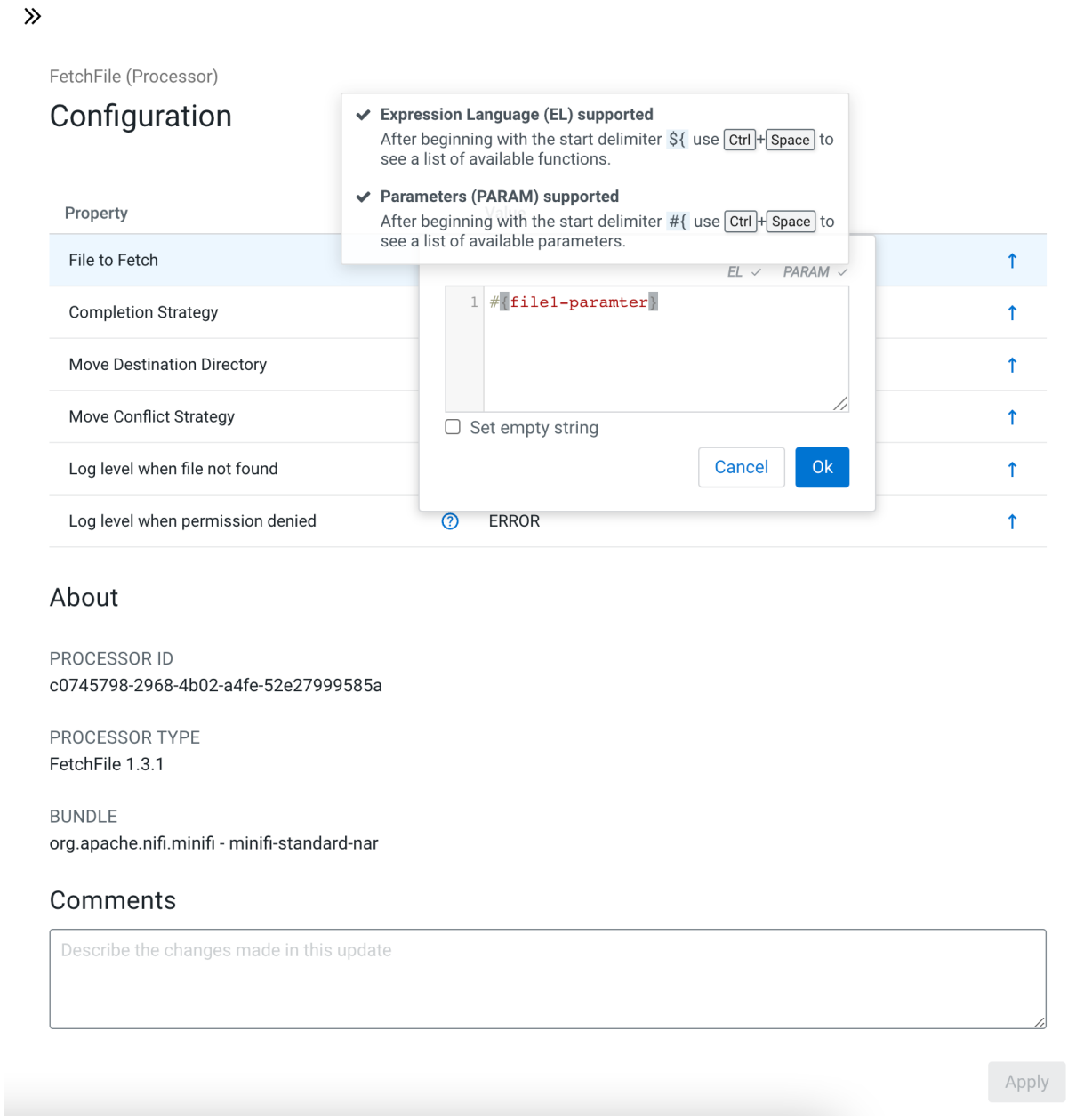


4. Complete the reference with a closing curly brace } and select Ok.



5. Click Apply to save the changes.

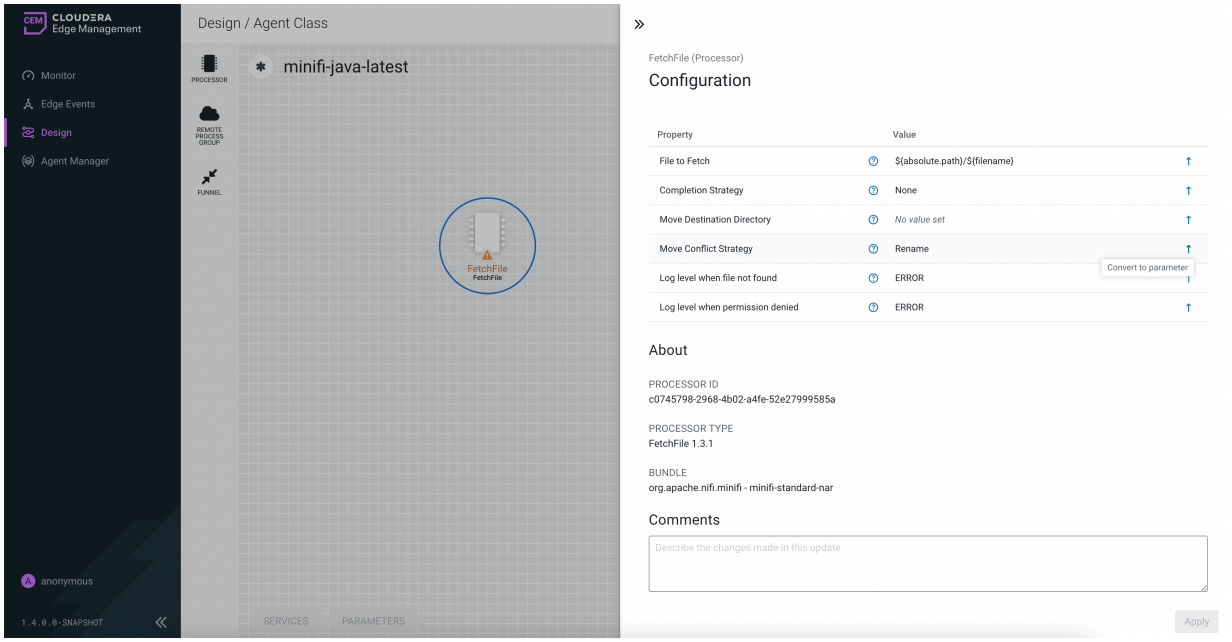
   You can hover over the Expression Language and Parameters indicators for help text on this process.



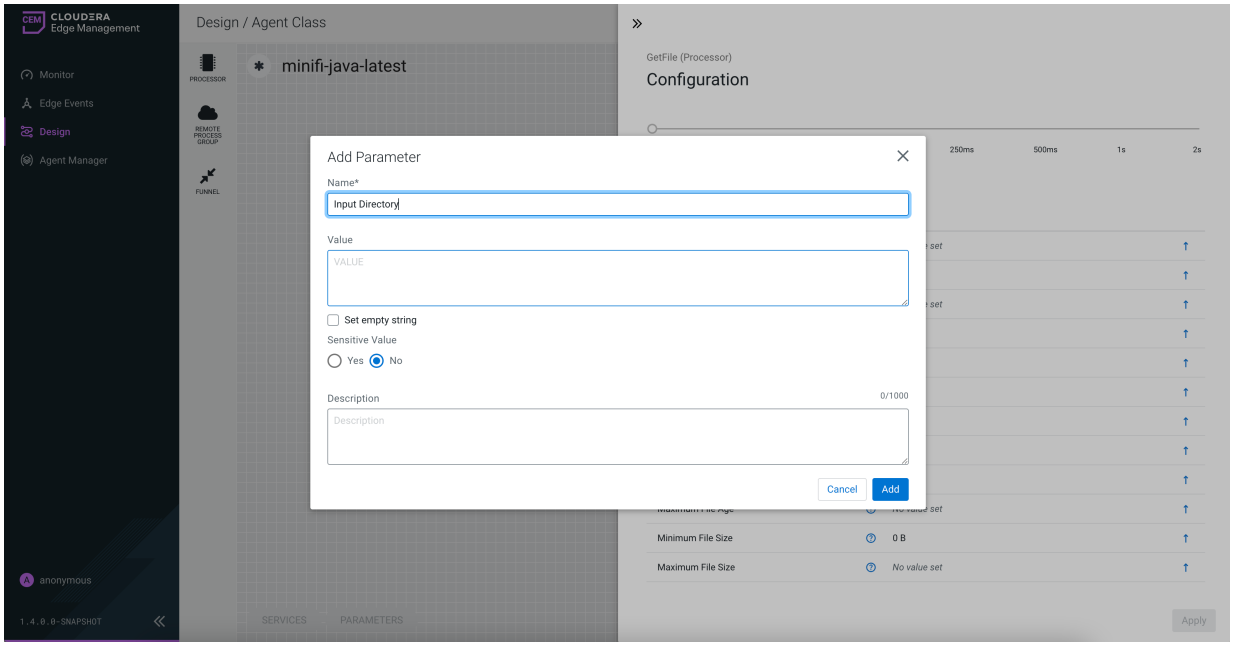
Converting property values to parameters

You can create parameters during the configuration of processors or services. Instead of entering a property value, you can convert the property value to a parameter.

1. Click  (the Convert to parameter icon) for the chosen property.

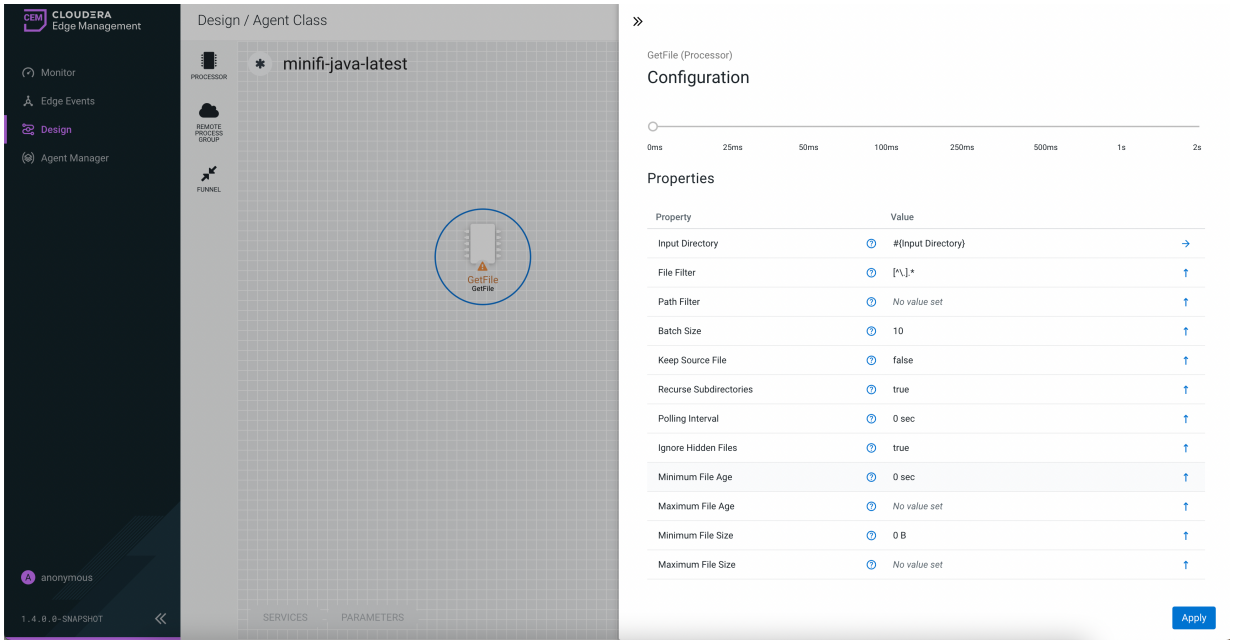


The Create Parameter modal window appears.



2. Configure the parameter properties.
3. Click Add to create the parameter.

   The property will automatically reference the new parameter using the correct syntax.

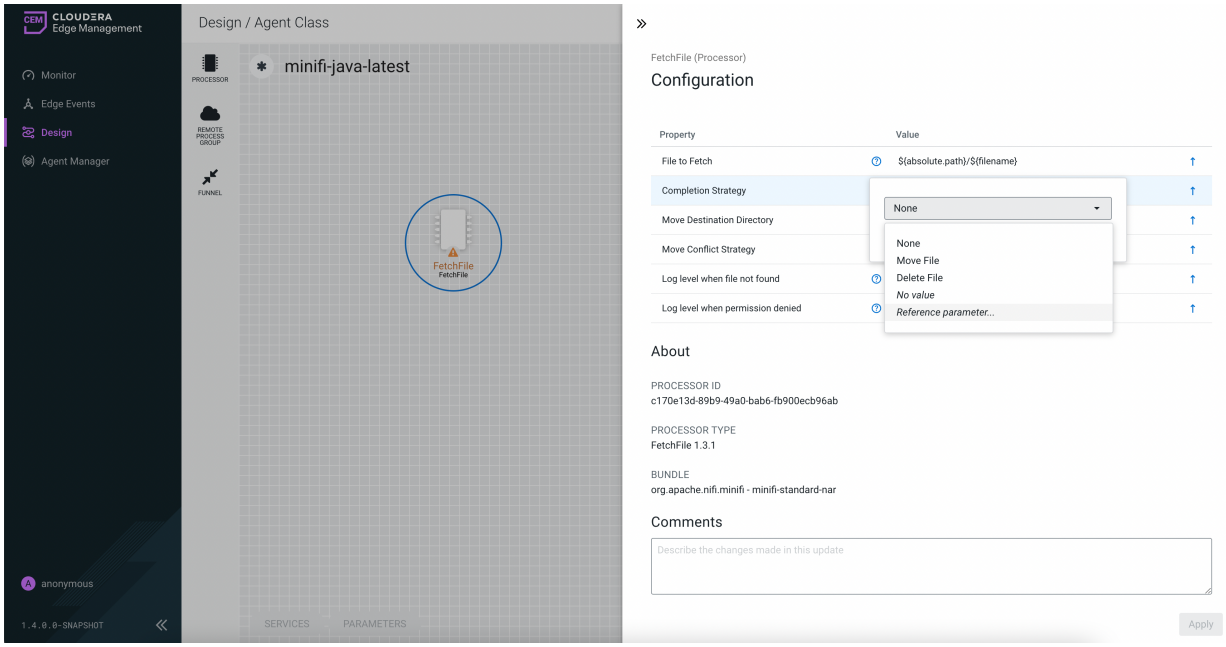


4. Click Apply to save the changes.

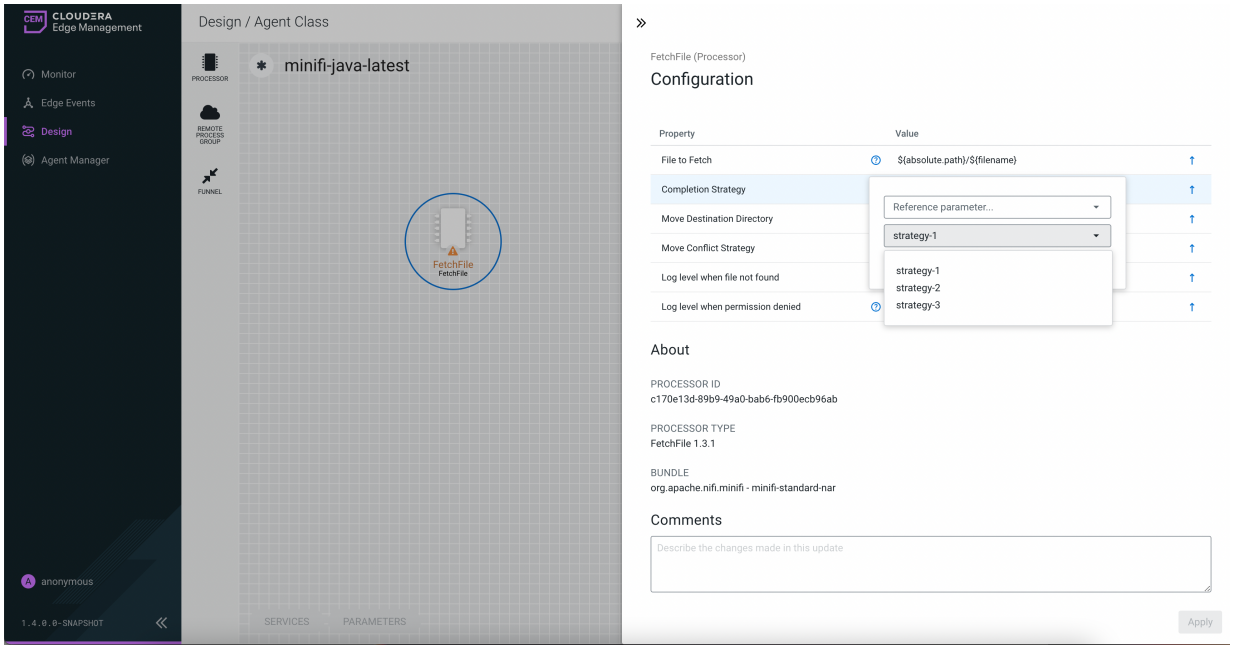
Converting selectable property values to parameters

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.

Using parameters in expressions

Parameters can be referenced using the #{} syntax, with options for escaping and combining parameters.

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}

If needed, escape the # using an additional # at the beginning.

Examples

If parameter abc has a value of xxx and parameter def has a value of yyy, 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.

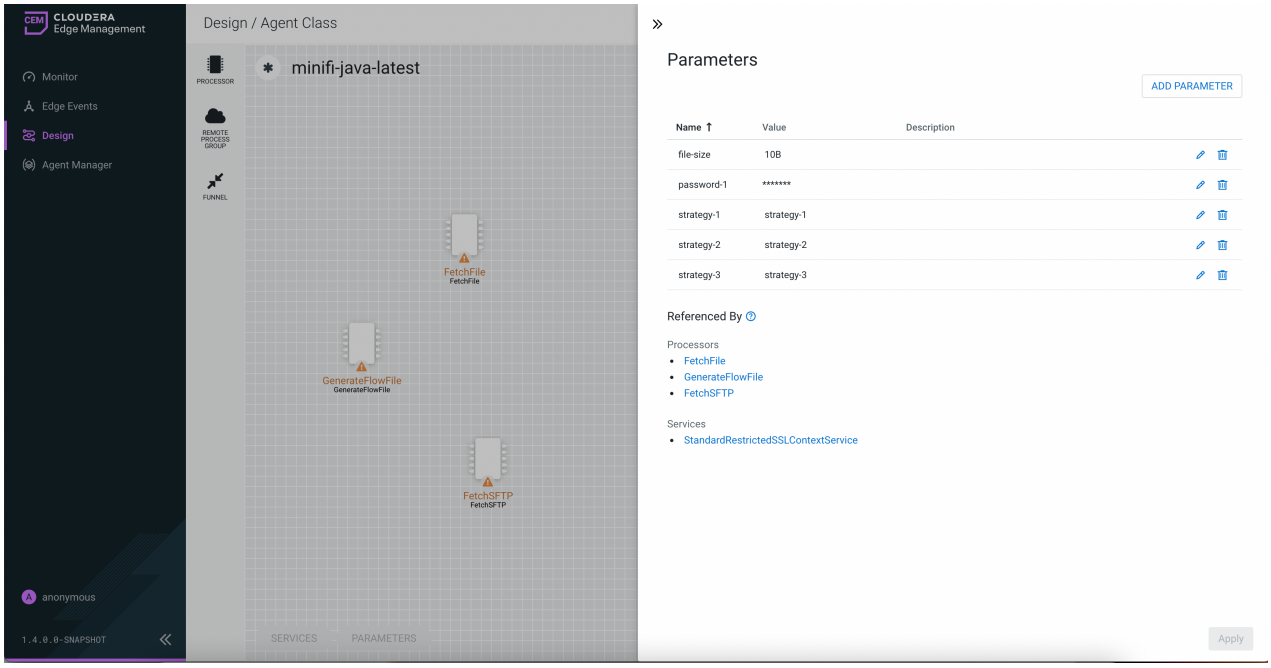
For example:

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

This replaces xxx with zzz for the abc parameter.

Referenced parameters

The Parameters window lists all parameters used in a flow, along with the components that reference them.



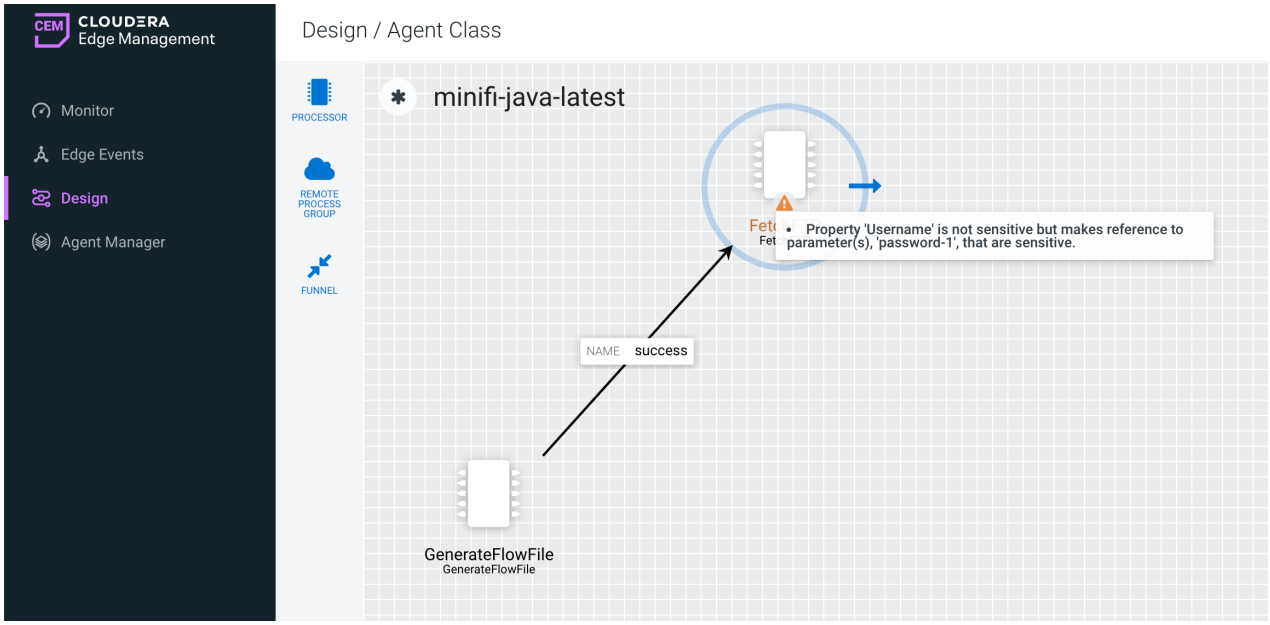
Select a specific parameter to view the processors and services that use it.

Using parameters with sensitive properties

Non-sensitive properties should only be referenced by non-sensitive parameters, and sensitive properties should only be referenced by sensitive parameters.

When publishing versioned flows:

- The value of a sensitive parameter is not sent to the flow registry, only the information that the property references the sensitive parameter.
- If a non-sensitive property references a sensitive parameter (or the other way round), the UI marks the component as invalid.



Processor-specific configurations

Some processors provide additional configuration options beyond the standard Configuration tab. These appear as extra, processor-specific tabs in the configuration dialog and offer specialized functionality for certain use cases.

Currently supported processors include:

- JoltTransformJSON – Provides a tab for design-time validation of Jolt transformations
- UpdateAttribute – Provides a tab for simplified rule-based configuration

Important: The processor-specific tab is currently available only for MiNiFi Java agents.

Testing JoltTransformJSON processor configuration

Use this feature to validate and refine Jolt transformations during flow design by testing them against sample JSON input before running the flow.

About this task

The Test Jolt Transform tab allows you to test how a Jolt specification transforms a JSON input.

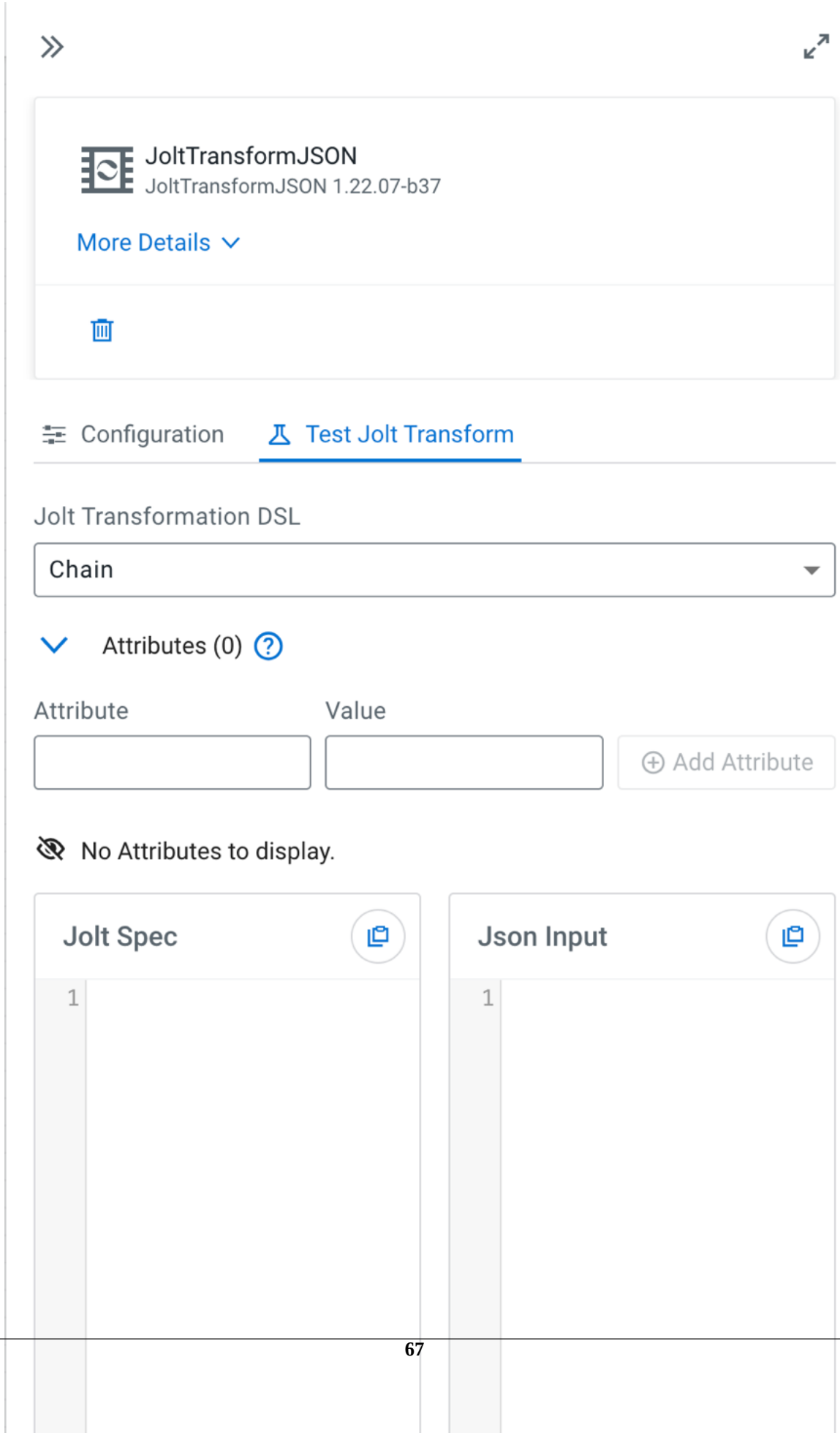
You can:

- Provide a sample JSON input
- Define or modify a Jolt specification
- Run the transformation
- Review the output immediately

This helps you verify and refine transformation logic before applying it in a running flow and design and debug flows more efficiently.

Procedure

1. Open the JoltTransformJSON processor configuration.
2. elect the Test Jolt Transform tab.


JoltTransformJSON
JoltTransformJSON 1.22.07-b37
More Details
Configuration
Test Jolt Transform
Jolt Transformation DSL
Chain
Attributes (0)
Attribute
Value
Add Attribute
No Attributes to display.
Jolt Spec
Json Input
1
1

3. Configure the transformation.

   Jolt Transformation DSL

   Select the transformation type.

   Jolt Spec

   Enter or modify the Jolt specification that defines the transformation.

   Note:

   The Jolt Transformation DSL and Jolt Spec values are automatically loaded from the processor configuration if already defined.

   Changes applied here overwrite the processor configuration.

4. Add attributes for testing.

   You can define attributes to simulate runtime values used in your Jolt specification.

   a) Expand the Attributes section.
   b) Enter an attribute name and the corresponding value.
   c) Click Add Attribute.
   d) Repeat these steps to add multiple attributes.

   Note:

   Attributes defined here are only used during testing and are not applied during flow execution.

5. Paste your sample JSON into the Json Input field.
6. Click Test Transform.

   This testing does not require starting or running the flow.

7. Review the result in the Output field.

   The test output reflects how the processor would transform the input during execution.

Configuring rules in the UpdateAttribute processor

The UpdateAttribute processor updates FlowFile attributes using the Attribute Expression Language and can also remove attributes based on regular expressions. In addition to standard configuration options, the processor allows you to apply conditional logic using rules.

About this task

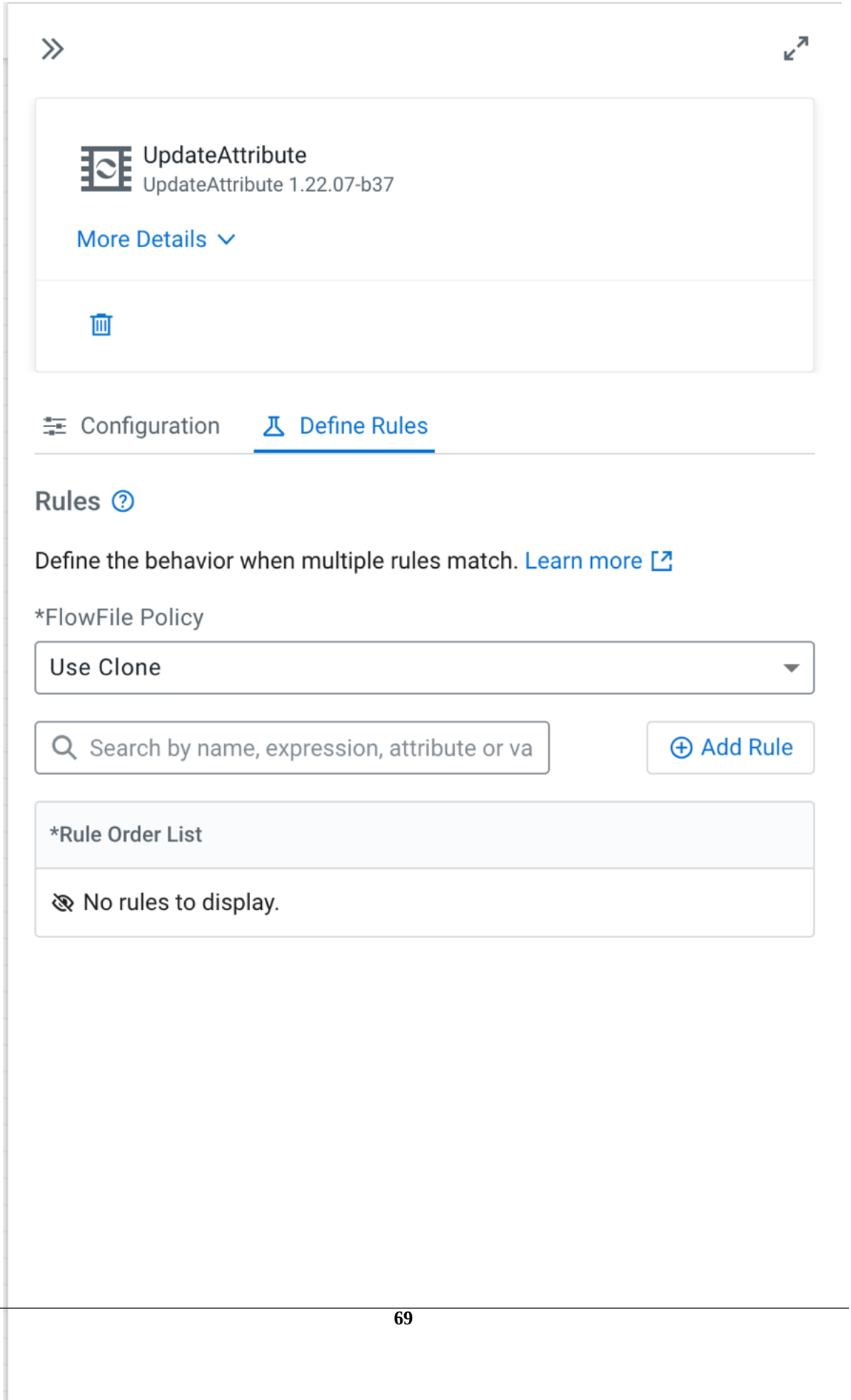
Use Define Rules tab to:

- Apply complex conditional logic without manual property configuration
- Manage multiple attribute updates in a structured way
- Improve readability and maintainability of your flow logic

Rules are a set of conditions and associated actions. Each rule follows an “if–then” structure, so if all conditions are met, then the defined actions are applied. Conditions define when the rule applies, and actions define what changes are made.

Procedure

1. Open the UpdateAttribute processor configuration.
2. Select the Define Rules tab.


UpdateAttribute
UpdateAttribute 1.22.07-b37
More Details
Configuration
Define Rules
Rules
Define the behavior when multiple rules match. Learn more
*FlowFile Policy
Use Clone
Search by name, expression, attribute or va
Add Rule
*Rule Order List
No rules to display.

3. Select a FlowFile policy.

   Use Clone

   Applies matching rules to a copy of the FlowFile

   Use Original

   Applies matching rules to the original FlowFile in the specified rule order

   Note:

   ules are evaluated based on the selected FlowFile policy.

4. Click Add Rule.
5. Enter a rule name.
6. Add one or more conditions.
   a) Click Add Condition.
   b) Define the condition using expressions.

   Note:

   All conditions in a rule must evaluate to true for the rule to match.

7. Add one or more actions.
   a) Click Add Action.
   b) Specify:
      - Attribute name
      - Valu
8. Click Apply to save the rule.

   Note:

   Invalid configurations prevent saving changes (the Apply button is disabled).

   You can duplicate existing rules to reuse and modify them.

Example

Rule: CheckForLargeFiles

Conditions:

- ${filename:equals('fileOfInterest')}
- ${fileSize:toNumber():ge(1048576)}
- ${fileSize:toNumber():lt(1073741824)}

Action:

- filename = ${filename}.meg

This rule appends .meg to the filename if the:

- filename is fileOfInterest, and
- file size is between 1 MB and 1 GB.

Example

Rule: CheckForGiantFiles

Conditions:

- ${filename:equals('fileOfInterest')}
- ${fileSize:toNumber():gt(1073741824)}

Action:

- filename = ${filename}.gig

This rule appends .gig to the filename if the:

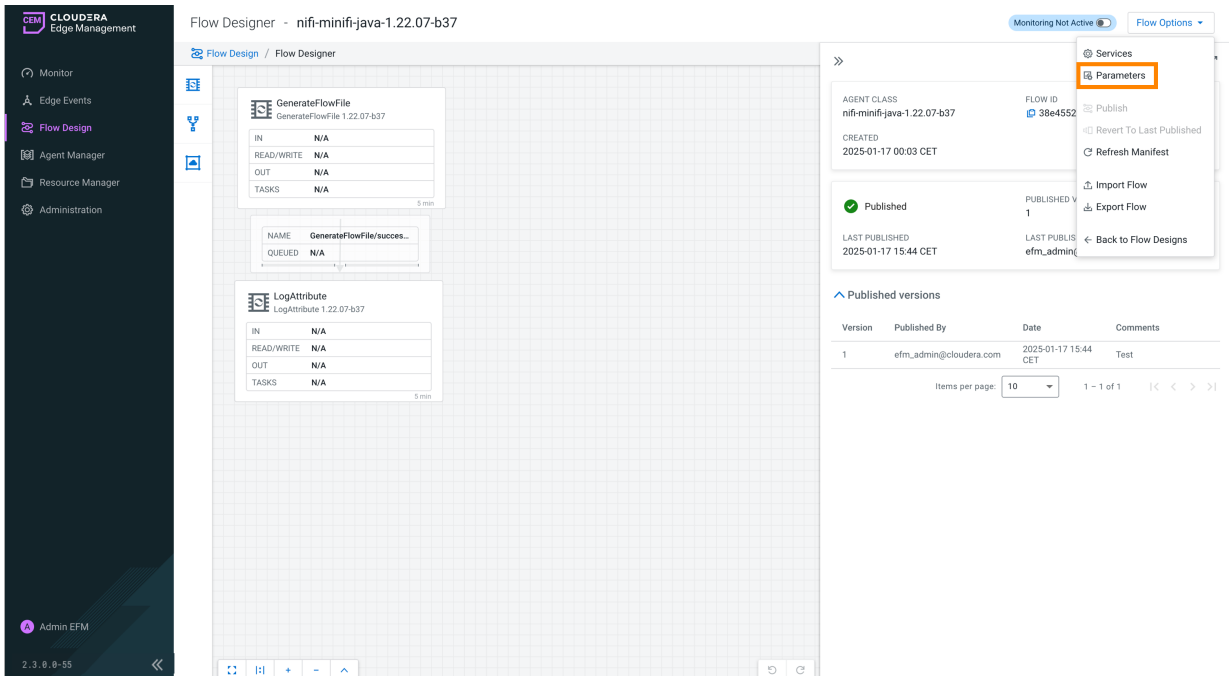
- filename is fileOfInterest, and
- file size is greater than 1 GB.

Managing agent parameters using the Cloudera Edge Management UI

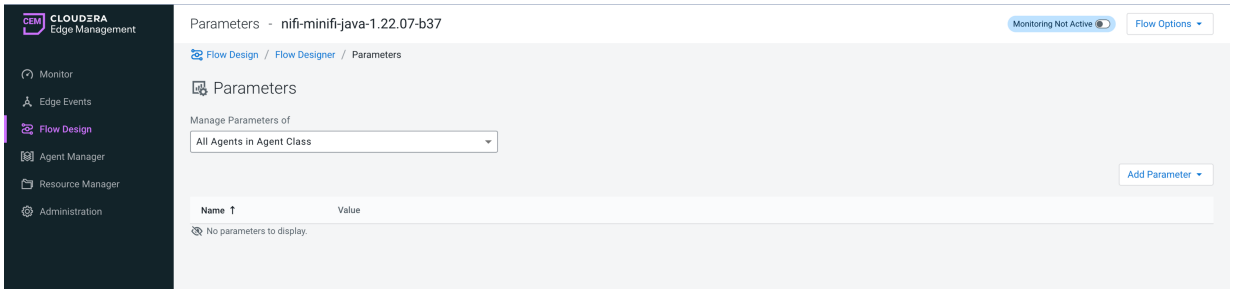
Learn how to configure parameters for individual agents and override flow parameters on the agent-level, enabling more granular configuration within an agent class.

Procedure

1. Navigate to the Flow Designer interface for a flow in Cloudera Edge Management.
2. Click Flow Options Parameters from the drop-down menu in the upper-right corner to access the parameter management interface.



3. Use the Manage Parameters of dropdown menu to filter parameters by scope.
   - You can view parameters for all agents in an agent class.
   - You can filter and display parameters for a specific agent.



4. Review the list of existing parameters displayed, which includes:
   - Name: the parameter's name.
   - Value: the current assigned value to the parameter.

5. To modify a parameter value:
   a) Click the arrow icon at the end of the parameter row to open the parameter editor on the right-hand side.
   b) Update the Value field as needed.
   c) Set the parameter to an empty string by selecting the Set empty string checkbox.
   d) Close the parameter editor panel.

   You can identify the changes in the parameter list by the badges displayed.

   New

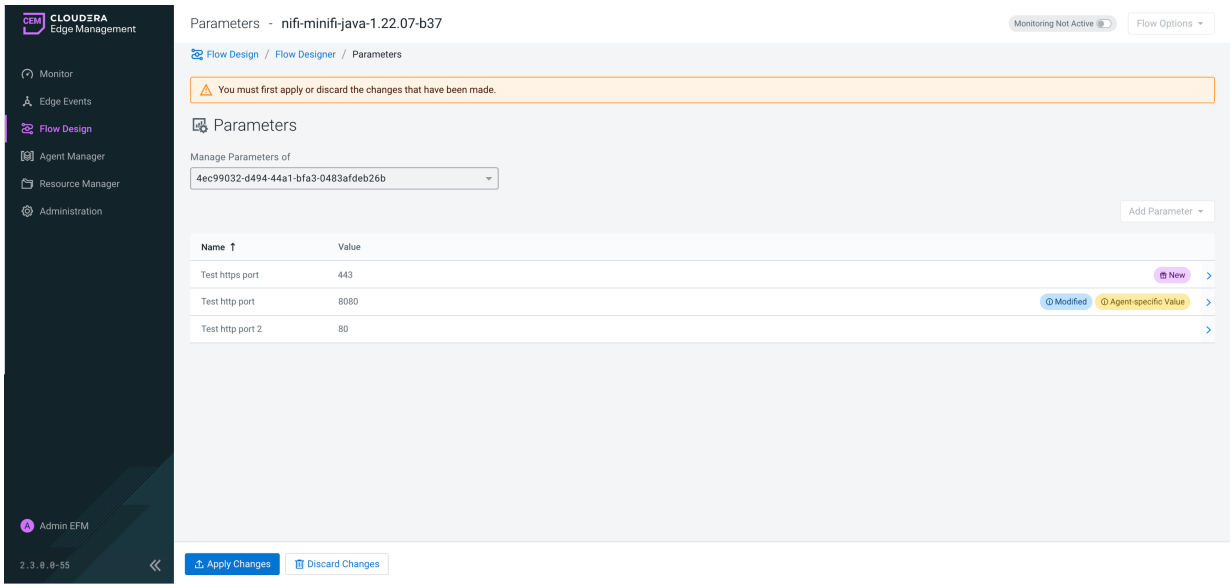
   Indicates a newly added parameter.

   Modified

   Indicates the parameter value has been changed.

   Agent-specific Value

   Highlights that the parameter has a custom value unique to the selected agent.



Note: You can restore the original class-level parameter value by clicking Restore Class Value in the parameter editor panel.

6. Click Apply Changes to save your updates or Discard Changes to revert.

Managing agent parameters using the REST API in Cloudera Edge Management

Learn how to configure programmatically using the Edge Flow Manager's REST API at both agent and agent class levels. This method ensures consistency across agents and provides flexibility for per-agent configurations.

Parameter contexts define the scope and overriding behavior of parameters, ensuring that values are applied at the correct level: agent, agent class, or flow. Parameters never exist on their own, they only exist in a parameter context. Any collection of replacement key-value pairs is known as a parameter context.

Edge Flow Manager allows setting parameter contexts at the following levels:

Levels	Description
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 Edge Flow Manager 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 Edge Flow Manager 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 Edge Flow Manager, it 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 Edge Flow Manager 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 *Edge Flow Manager REST API*.
- Parameter Mappings section in *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

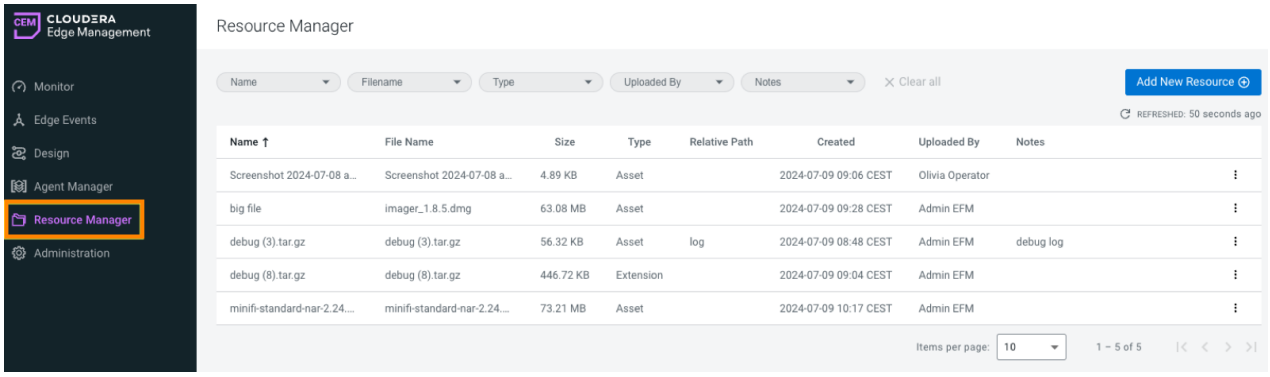
REST API Reference
Using agent parameters

Managing resources

Resource Manager in the Edge Flow Manager UI extends the Asset Push functionality. It allows you to manage existing assets and extensions, upload new assets and extensions, assign them to selected agent classes, and deploy new MiNiFi extensions or ML models at runtime.

Unlike Asset Push, Resource Manager works for both new and offline agents, ensuring they receive the new files. It functions in secure and unsecure environments as well. For secure environments, establish a secure connection between Edge Flow Manager and the agents. For more information about setting up a secure connection, see *TLS configuration for Cloudera Edge Management*.

All resources are listed on the **Resource Manager** page. When you upload a new resource , it is automatically synchronized between the Edge Flow Manager nodes. Only users with an Operator role can upload and delete resources, or assign them to agent classes.

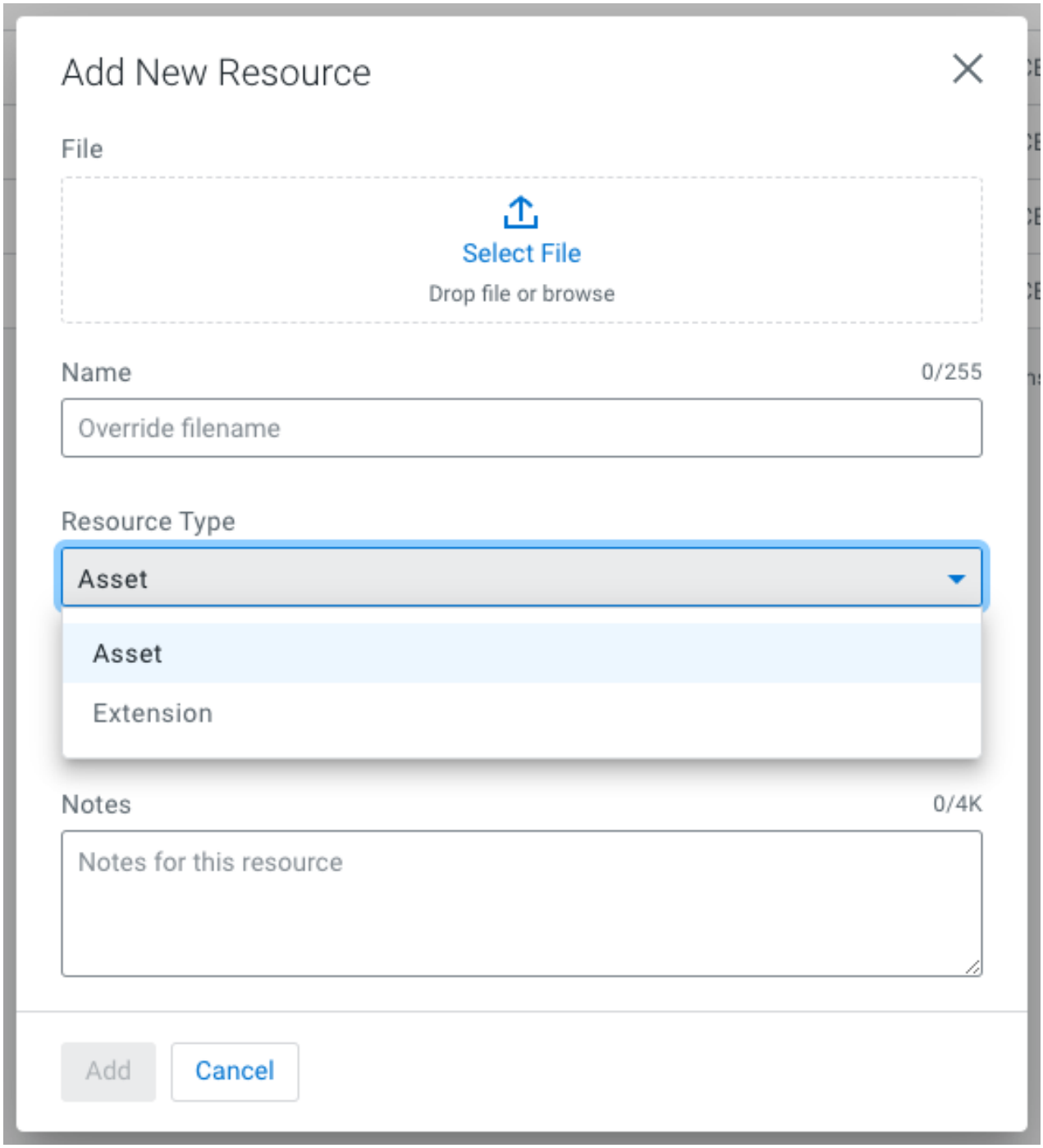


You can edit the name of an existing resource, and download or delete it using the pop-up menu at the end of each resource row.

Click Add New Resource to upload new resources. In the **Add New Resource** pop-up, you can upload a single file or a batch of files at once. If you upload one file at a time, you can change its name. For batch uploads, you can only edit file names one by one from the **Resource Manager** page.

Specify the agents' resource location using the Relative path on the agent field. This is relative to the path set with the efm.resourcemanager.repositoryPath property.

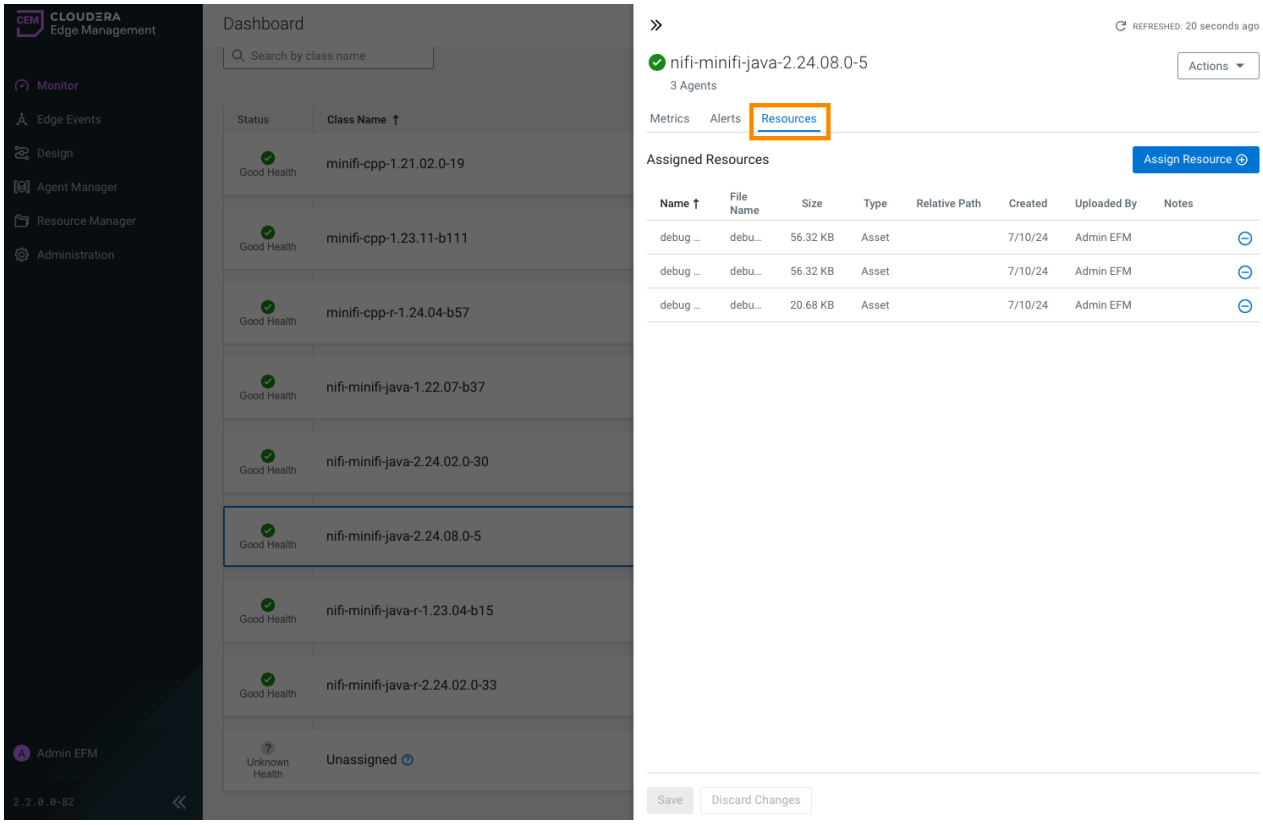
You can upload assets or extensions. Agents will start to download the new resources after their next heartbeat. If a download fails, the agent will retry the download in the next heartbeat iteration. On the agent, assets are downloaded to the asset directory, and extensions are downloaded to the extensions directory within the agent's base path.



Resources tab

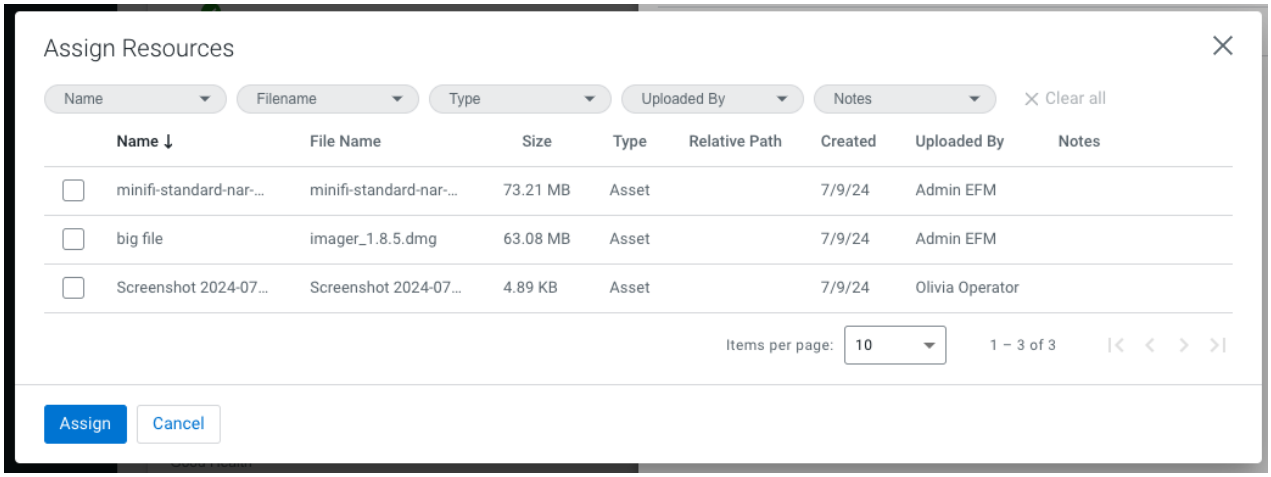
The **Resources** tab is available on the agent class details page, if the class contains agents that are compatible with resource assignment.

On this tab, you can see all assigned resources and assign new ones.

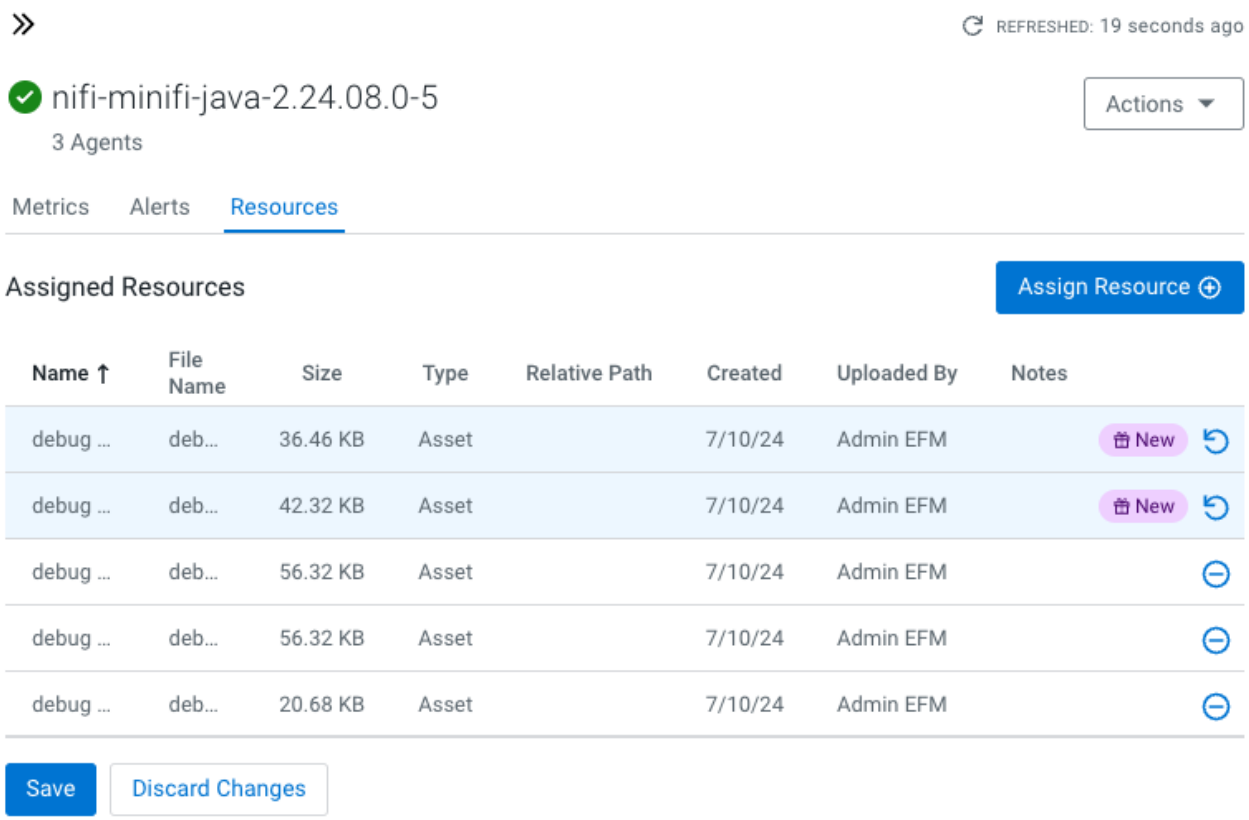


Click Assign Resource to see and select available resources for assignment.

Note: Clicking the Assign button does not immediately synchronize the newly selected resources to the agents.



To start downloading the selected resources to the agents, you have to save the new assignments on the agent class details page. You can review and modify the new assignment before saving.



Resource management properties

You can find here the key resource management properties, which help configure synchronization, caching, and operational parameters, ensuring smooth and efficient management of resources across nodes and agents.

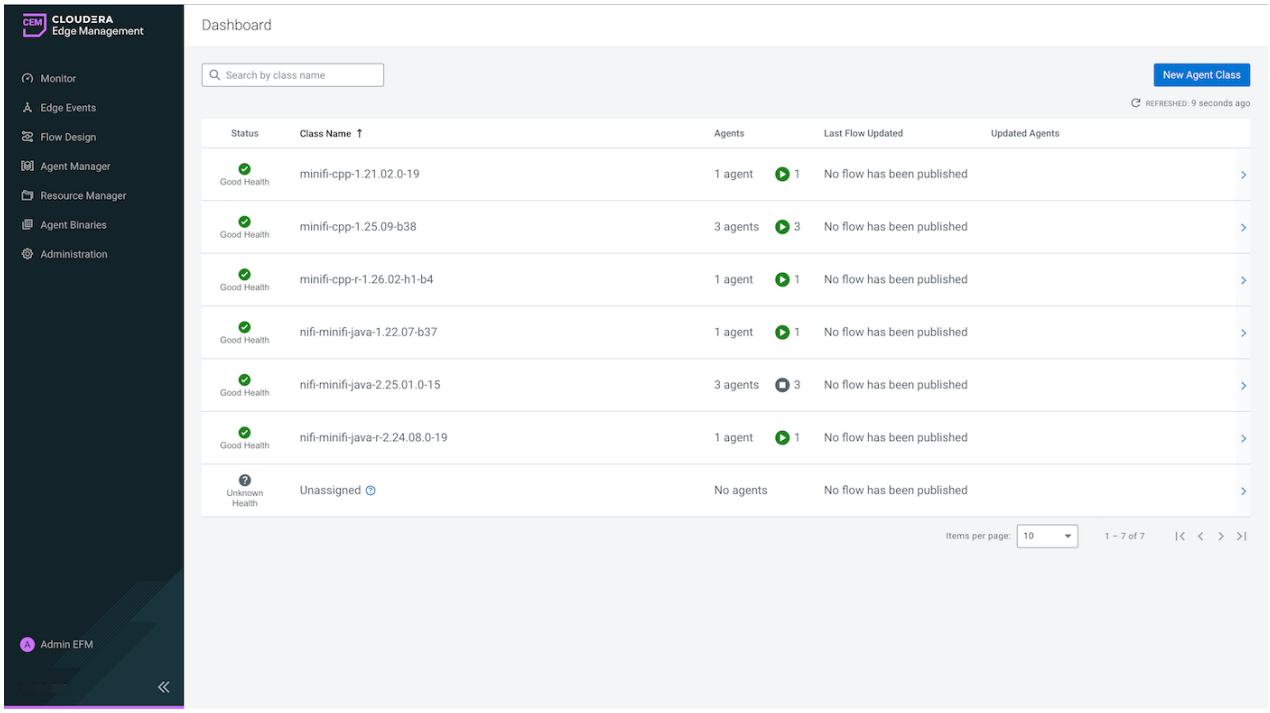
Property name	Dafault value	Description
efm.resourcemanager.resourceManagerPort	9010	Port for the resource synchronization endpoint
efm.resourcemanager.nodeAliveTtl	5 sec	Time-to-live after which a node is considered unavailable
efm.resourcemanager.resourceSynchronizationTriggerInterval	5 sec	Interval at which resource synchronization between nodes is triggered
efm.resourcemanager.nodeResourceCacheUpdateTriggerInterval	2 sec	Interval at which the resource list cache of the nodes is updated
efm.resourcemanager.repositoryPath	resources	Path to repository where Edge Flow Manager stores uploaded resources
efm.resourcemanager.blockSyncResourceReTriggerDuration	10 minutes	Duration for which Edge Flow Manager blocks re-triggering the SYNC RESOURCE operation
efm.operation.monitoring.rollingOperationsSize.sync.resource	10	Maximum number of resources sent to agents at once

Managing deployments in Cloudera Edge Management

Learn about the Dashboard page in Cloudera Edge Management, which provides a centralized view for monitoring and managing C2 server and agent deployments. It is the default landing page in the Edge Flow Manager UI.

The Dashboard combines monitoring and operational capabilities in a single view. It allows you to track agent health, flow updates, and command status in real time and access agent class details.

To access the Dashboard, click Monitor in the left navigation pane.



Agent class list

The Dashboard provides the following information for each agent class:

Status

Indicates the overall health of agents in an agent class, based on heartbeat activity:

- Good Health: Recent heartbeats are received from all agents in the agent class.
- Concerning Health: Some agents have not sent a heartbeat within the time period defined by the efm.monitor.maxHeartbeatInterval.
- Bad Health: No agents have sent a heartbeat in the given class for the time period defined in theefm.monitor.maxHeartbeatInterval.
- Unknown Health: The agents in the class may not yet be registered, or the heartbeat intervals cannot be determined.

An indicator next to the status icon shows that there are alerts associated with the agent class.

Possible alert messages include:

- A recent heartbeat has not been received for *[A NUMBER OF]* agents. - You can click View Recent Alerts to learn more about the issue.
- Errors occurred during previous flow publishing. - This lists the most frequent errors since the last publish. The number in brackets indicates how many times the error occurred. You can click View Errors to learn more about the issue.
- Agent Class contains agents with expiring certification within 30 days. - This notifies you that the certification of an agent will expire soon and you should renew it. You can click View Recent Alerts to learn more about the issue. Once the renewal is addressed, the alert disappears.
- Agent property sync is in progress
- Resource sync is in progress

Configuration inconsistency alerts indicate that agents in a class are temporarily not aligned. This can occur during updates or when a change has not yet propagated to all agents. These alerts highlight configuration drift, allowing you to investigate and resolve it before they cause

inconsistent behavior across agents. To investigate,, switch to the **Agent Manager** view, which displays an indicator for agents whose manifest is out of sync.

Examples of configuration inconsistencies:

- Manifest out of sync
- Agent certification expiration
- Operation in progress

Class Name

Displays the name of the agent class.

Agents

Displays the total number of agents assigned to the agent class, including both online and missing agents. Status icons next to the agent count indicate how many agents are running or stopped.

Last Flow Updated

Indicates when the flow was last updated for the agent class.

Updated Agents

Shows the status of the last batched command (for example flow publishing or property updates). Batched commands cannot fully complete for all agents if there are missing agents in the class.

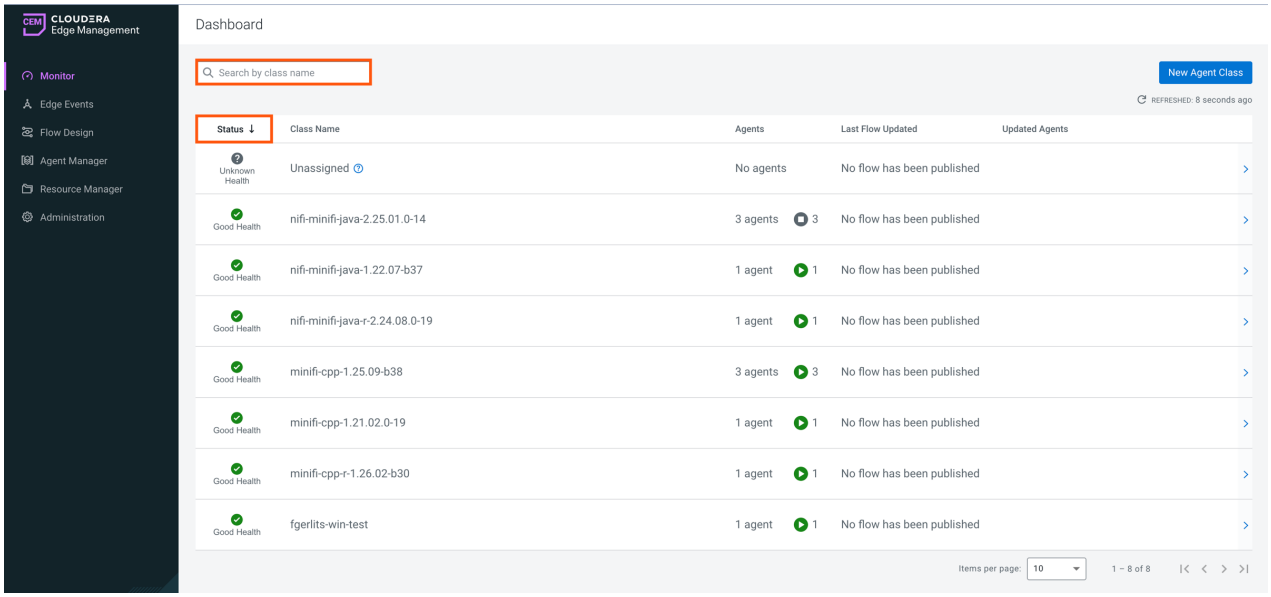
Arrow

Click the arrow at the end of a row to open the agent class details pane and review metrics, alerts, resources, and configurations.

Sorting and searching agent classes

You can sort data by most of the columns in ascending or descending order by clicking the column header.

To search for an agent class by class name, add your search string in the search field.



Items per page

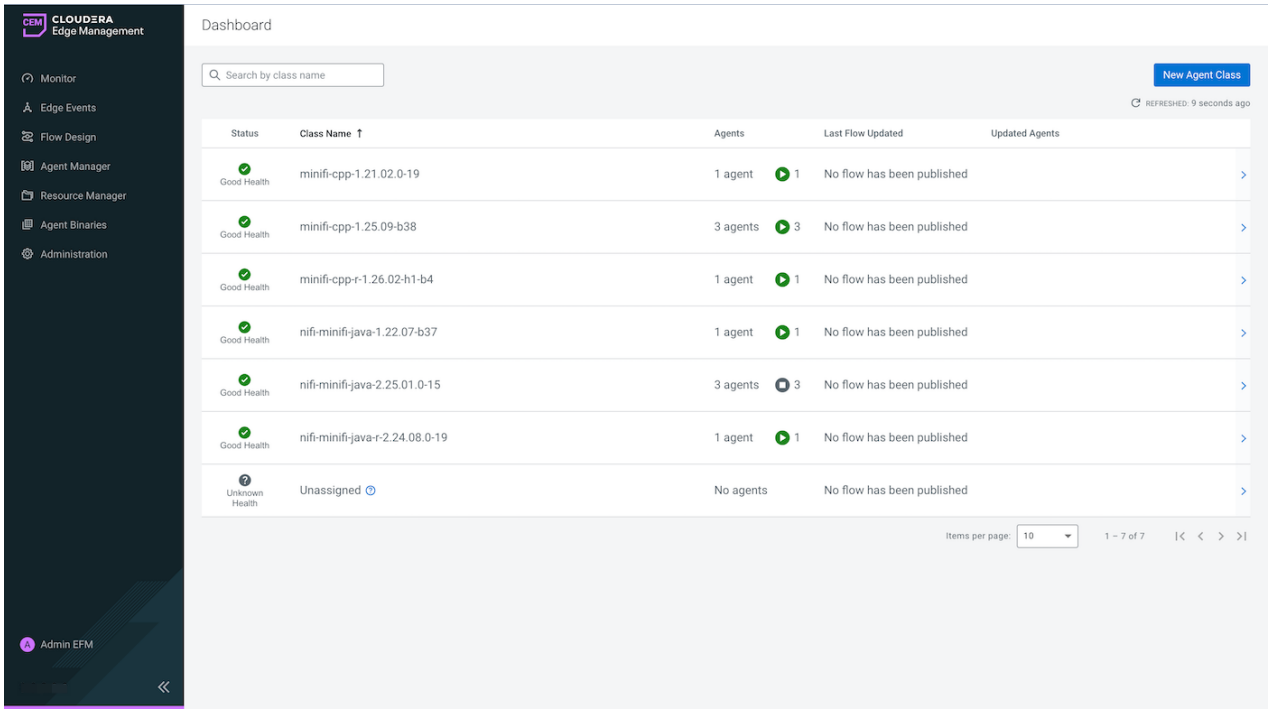
Use the Items per page dropdown at the bottom of the table to control how many agent classes are displayed on a page.

If multiple pages are available, use the pagination controls to navigate between pages (first, previous, next, or last).

Viewing agent class details in Cloudera Edge Management

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

From the Dashboard, click > View Agent Details at the end of an agent class row to open the agent class details pane.



Metrics tab

It displays the following information:

Flow Definition

Displays details about the class including flow ID and published flow version.

Dashboard

Provides a link to the Grafana dashboard, if enabled. For more information, see *Monitoring Metrics with Grafana*.

Repositories (Class Average)

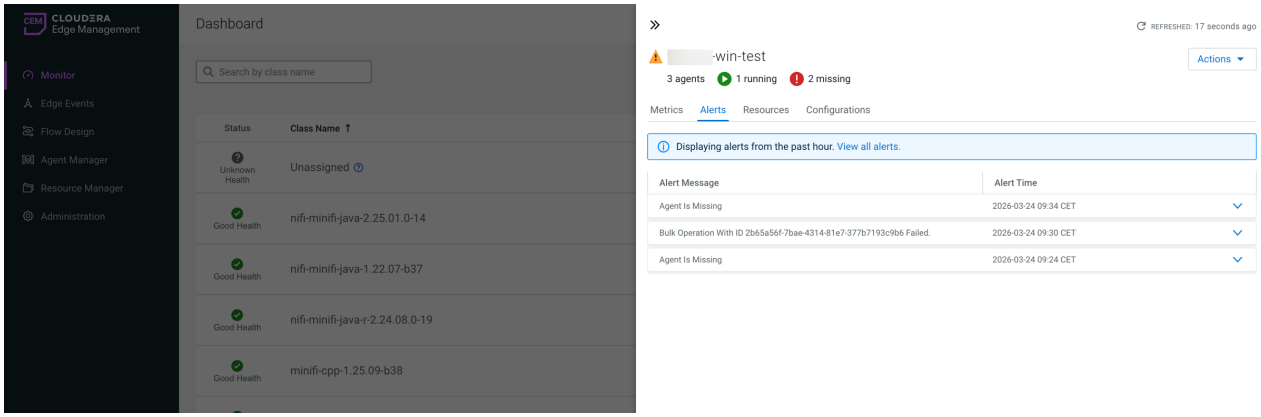
Shows the average usage details of the repositories across all agents in the agent class.

Connection Queues (Class Average)

Shows an aggregate view of all connections across agents in the class.

Alerts tab

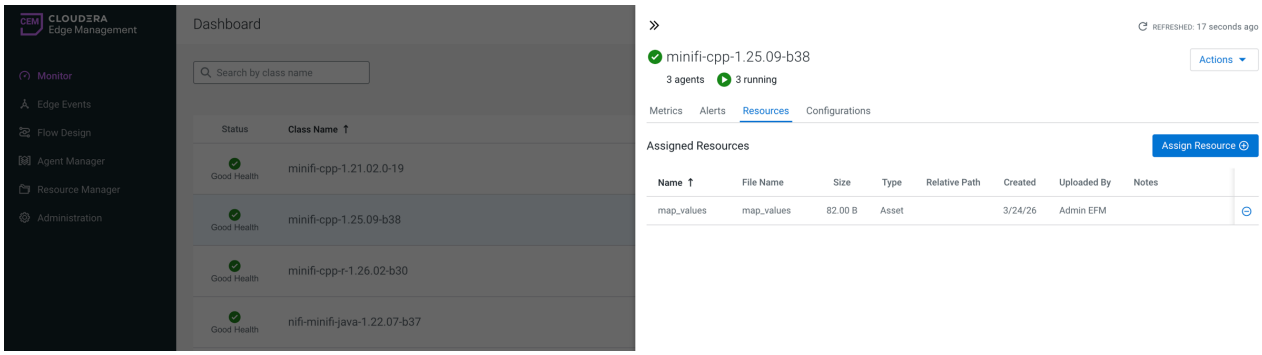
It displays information on alerts and their timestamps.



1. From the Dashboard, click the arrow icon at the end of an agent class row to open the agent class details pane.
2. Switch to the Alerts tab to view alerts and their timestamps.
3. Click ⌄ at the end of the alert row to display more information about the alert.
4. Click View all alerts to navigate to the Edge Events page, which is filtered by alert event type.

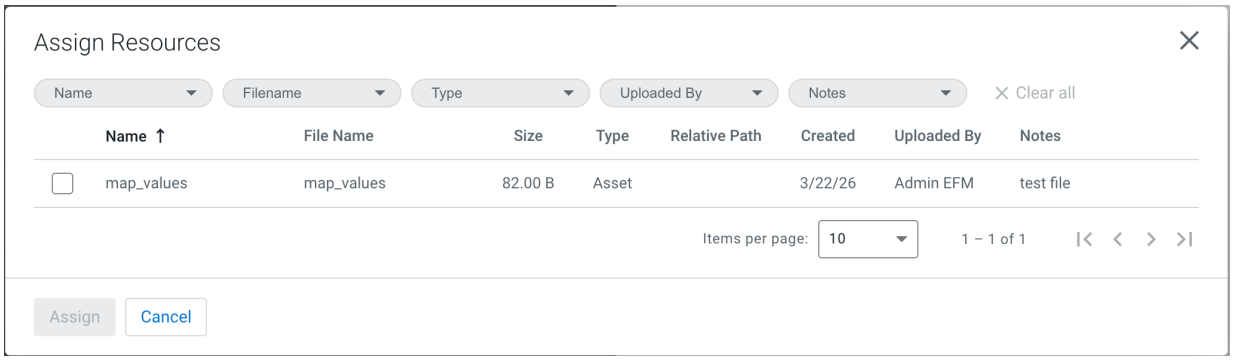
Resources tab

The table lists all resources assigned to the agent class, including details such as file name, type, size, and relative path on the agent. If no resources are assigned, the table displays No resource assigned.



You can also use this tab to assign new resources to the agent class.

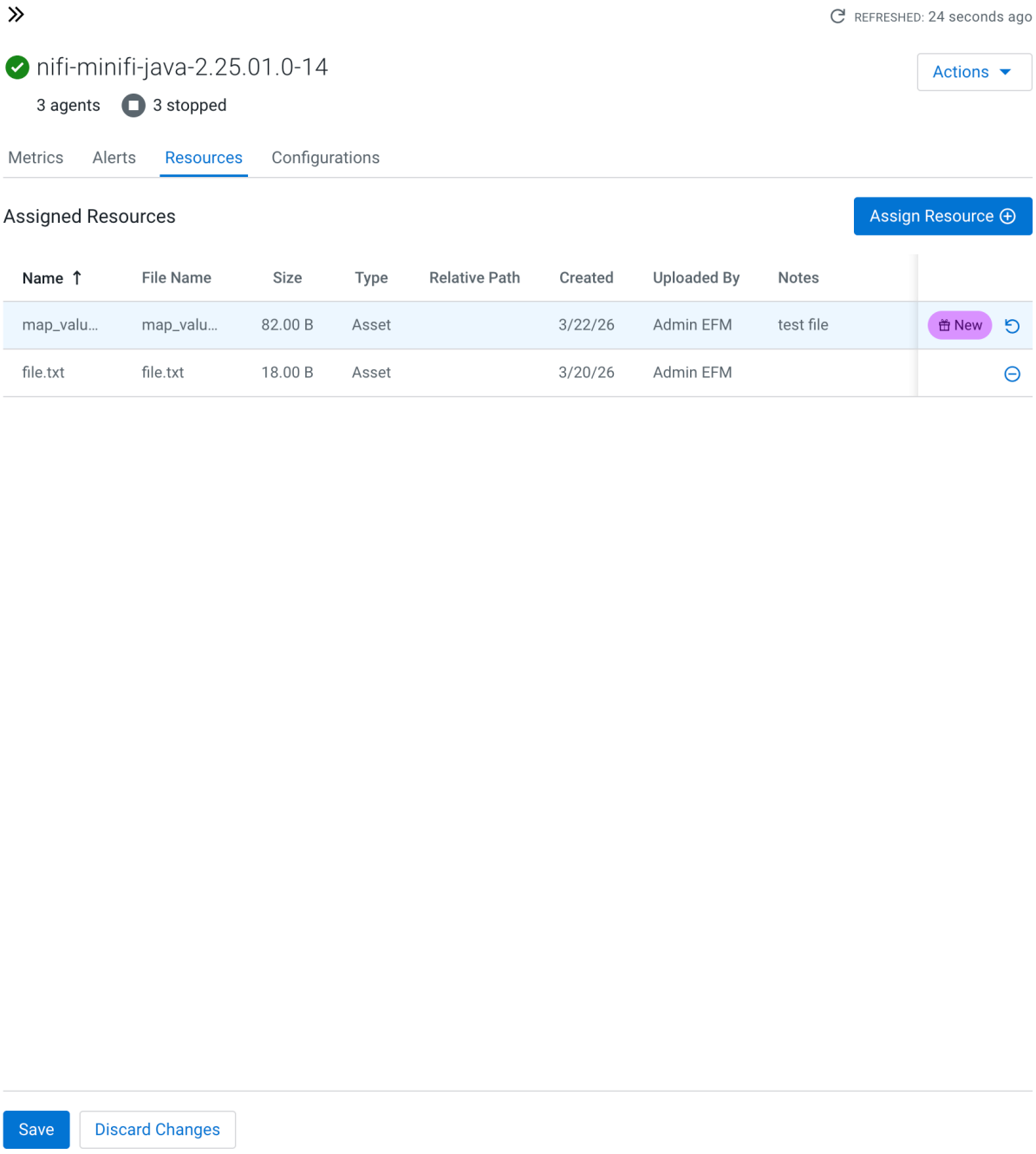
1. Click Assign Resource.
2. Select a resource from the list.



3. Confirm the assignment by clicking Assign.

Note: Clicking the Assign button does not immediately synchronize the newly selected resources to the agents.

4.



5. Save the new assignment(s) on the agent class details page.

   You can review and modify the new assignment before saving.

Assigned resources are synchronized to all agents in the class and become available for use in flows and processor configurations.

Configurations tab

It allows you to modify properties through the Edge Flow Manager for all agents within an agent class.

To update a property:

1. Find the property in the list you want to update.
2. Click the Edit icon in the property row.

3. Enter a new value.
4. Click Apply to save the changes.

   Click Cancel to discard unsaved modifications.

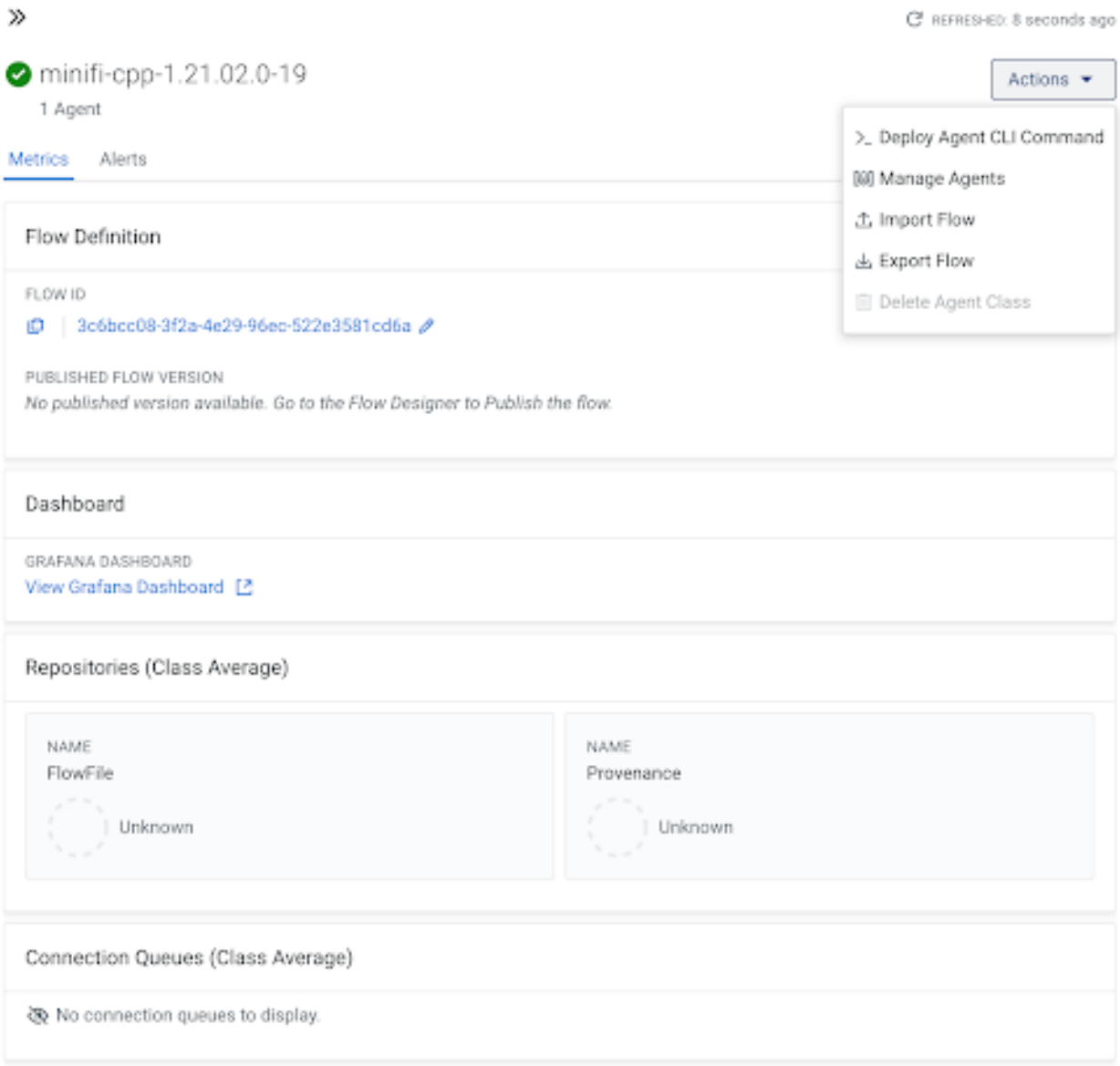
Any configuration changes made here are applied to all agents in the class.

Agent class actions

You can perform several management actions on agent classes, including working with flows, managing agents, editing configurations, and accessing agent details.

From the Dashboard, click the arrow icon at the end of an agent class row to open the agent class details pane.

The Actions menu in the top-right corner provides various commands and tools for managing agent classes.



Starting/Stopping a flow in an agent class

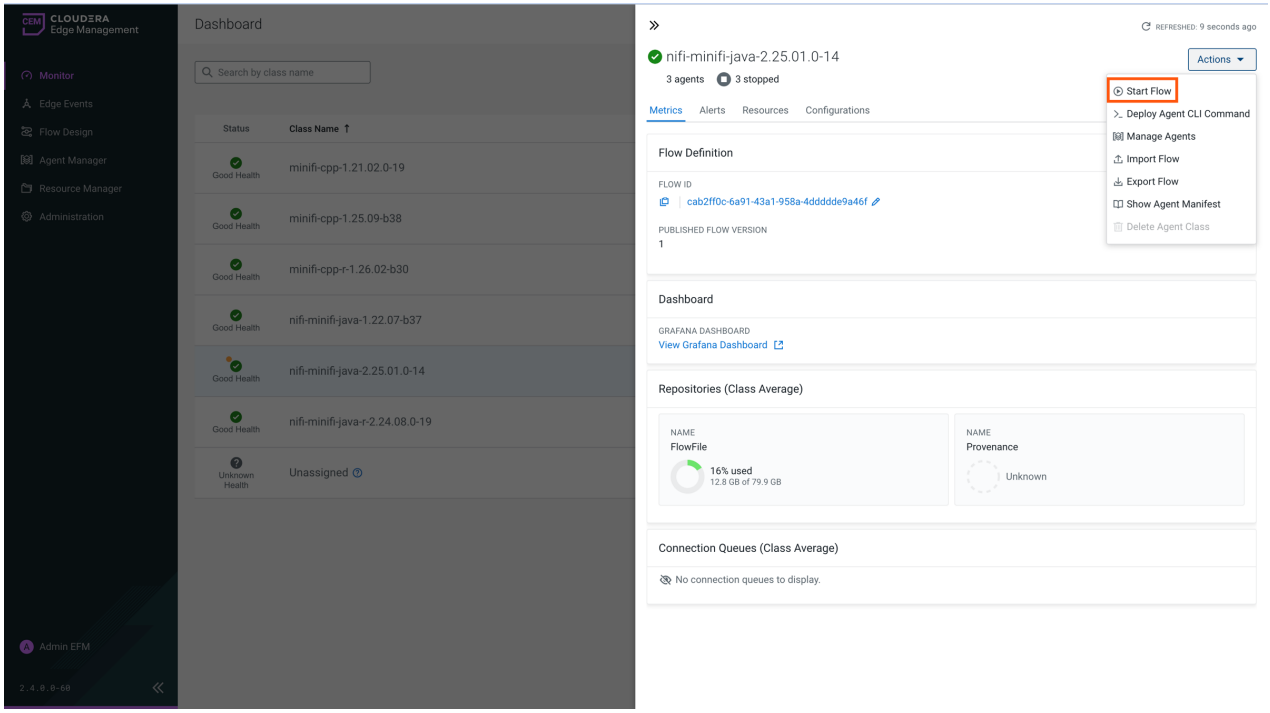
You can start and stop running a flow after deployment. With this, you have greater control over when your flows are actively processing data.

1. From the Dashboard, click the arrow icon at the end of an agent class row to open the agent class details pane.
2. Click Actions Start Flow .

You can:

- Temporarily pause data collection for maintenance or troubleshooting.
- Deploy flows to agents but only start them when ready for staged rollouts.
- Quickly stop flows in response to operational issues.

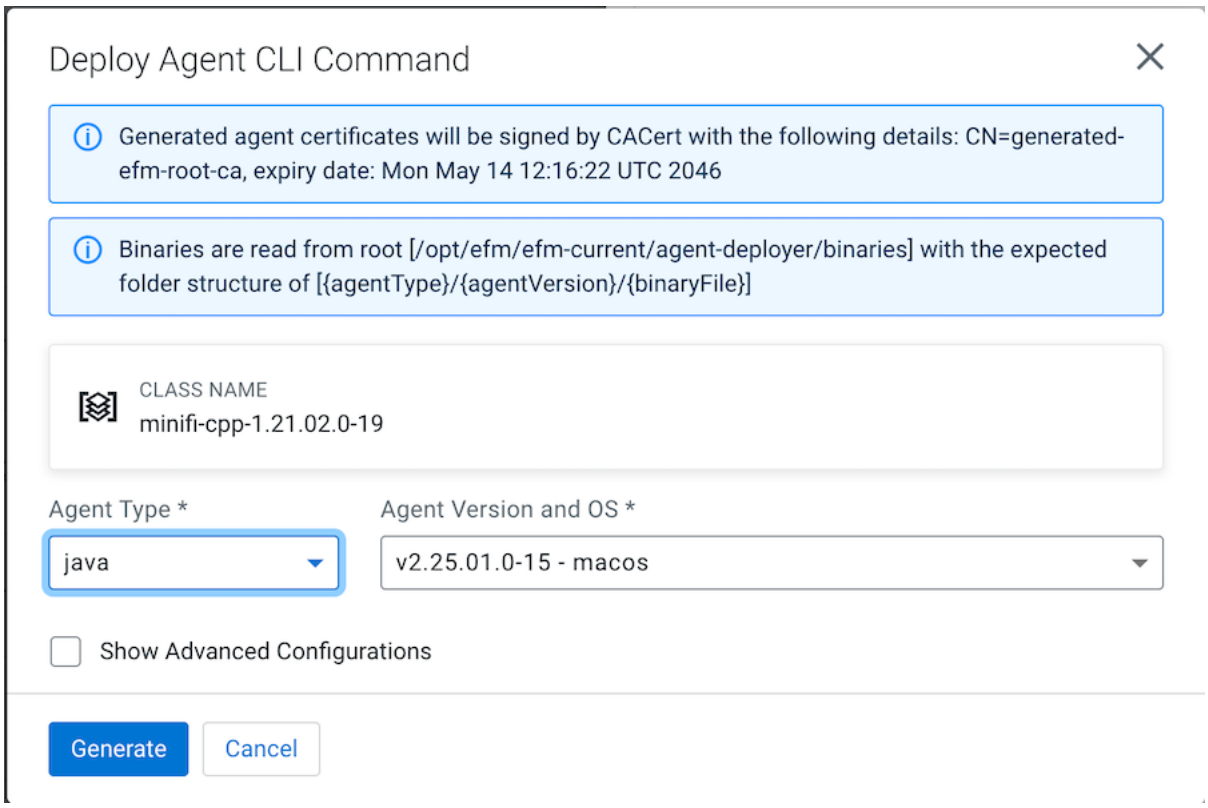
Note: The start and stop actions are only available if the flows are published.



When all dataflows are started the menu item changes to Stop Flow, allowing you to stop all running flows at once.

Deploying agent CLI command

This option allows you to deploy specific agent commands through the command-line interface (CLI). It provides flexibility for advanced operations or troubleshooting tasks that are not available through the UI.



Managing agents of an agent class

This option allows you to view and manage the agents assigned to an agent class.

To view the list of agents assigned to the selected agent class, click Actions Manage Agents .

For more details about how to manage agents, see *Managing agents in Cloudera Edge Management*.

Importing a flow into a new agent class

This option allows you to import a pre-existing flow into the agent class.

Exporting a flow from a new agent class

This option allows you to export the current dataflow associated with the agent class. Exported flows can be shared or stored for backup purposes.

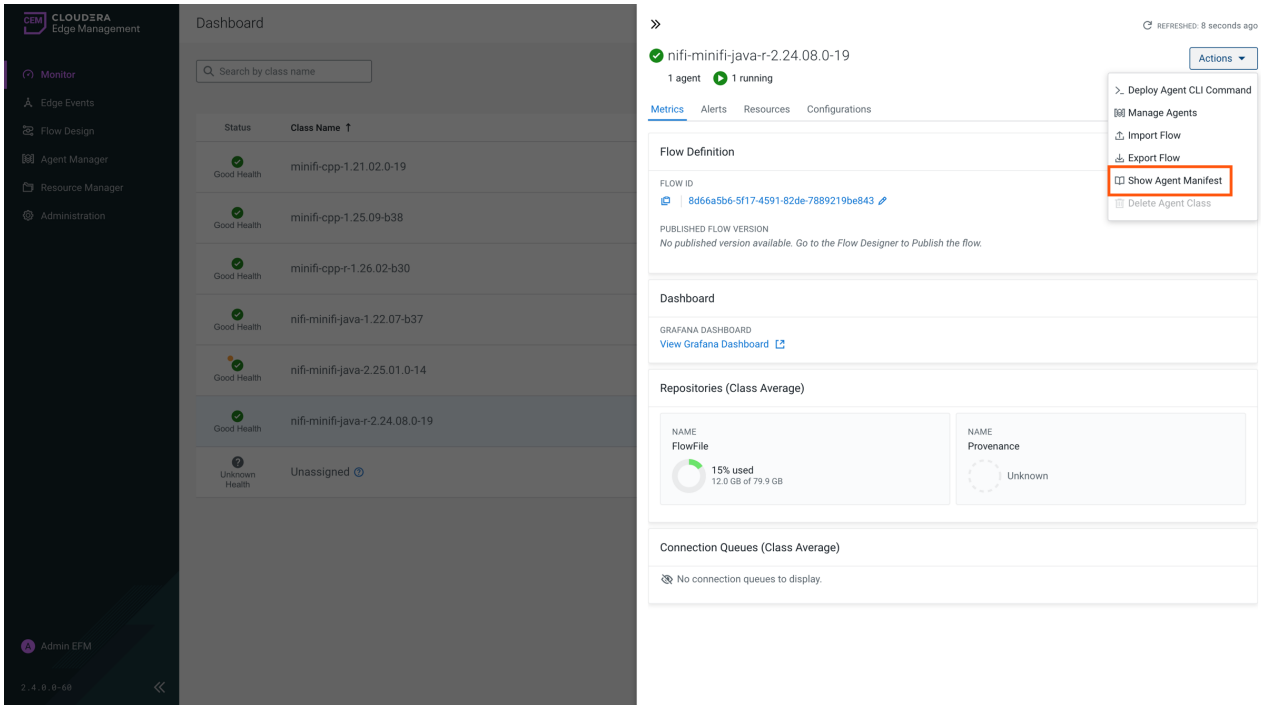
Viewing the agent manifest

This option allows you to view the agent manifest to inspect the configuration and capabilities of an agent, including details used during deploying or running a flow.

1. From the Dashboard, click the arrow icon at the end of an agent class row to open the agent class details pane.
2. Click Actions Show Agent Manifest .

The agent manifest contains the agent's configuration and capabilities.

You can copy and download the manifest files.



Deleting an agent class

This option allows you to permanently remove an agent class and its associated flow from Edge Flow Manager.

When you delete an agent class, it also removes the flow assigned to that class, but events associated with the deleted class will remain available in the **Edge Events** view. Edge Flow Manager does not provide any rollback option. You have to export the flow before deleting it, if you want to restore it in the future.

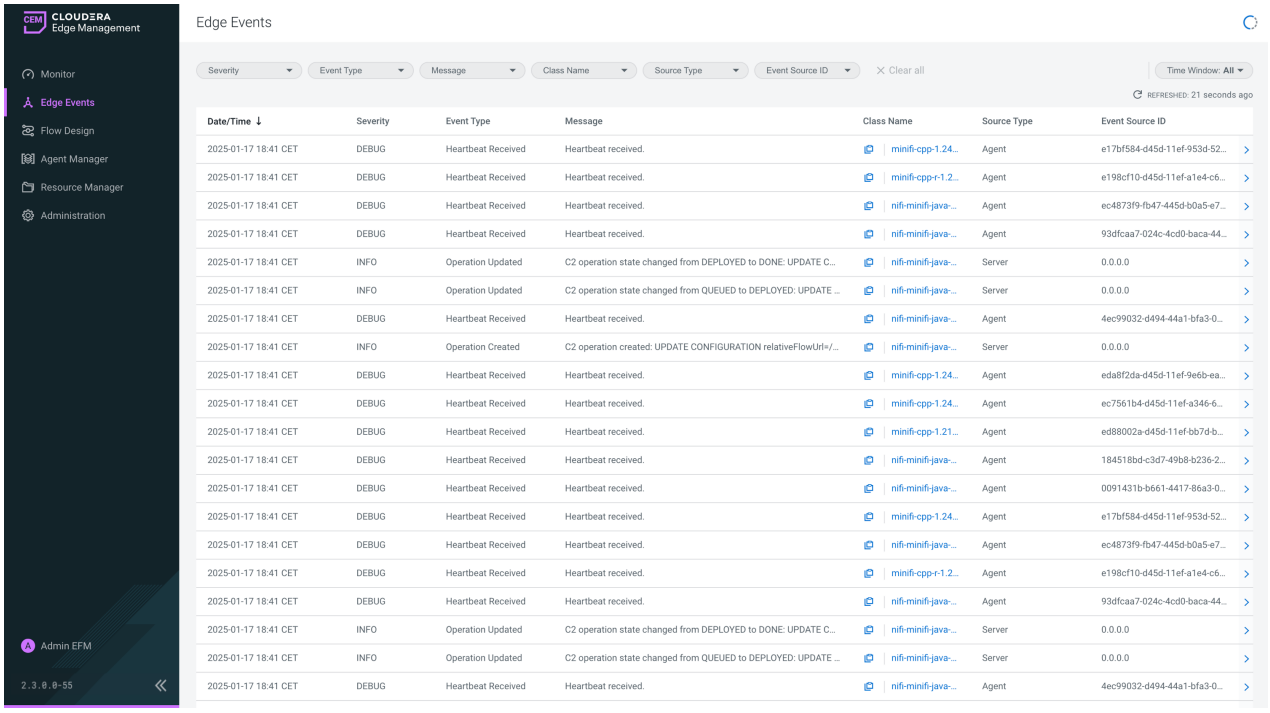
Note: You can only delete an agent class if no online agent is assigned to it.

Monitoring events in Cloudera Edge Management

Learn about the options available on the **Edge Events** screen that enable you to monitor C2 server and agent events.

The **Edge Events** interface allows you to monitor events effectively by providing detailed information on event type, severity, and source. You can gain deeper insights to ensure optimal system performance and address issues proactively. With options for sorting, filtering, and time-based views, you can analyze and manage events across your deployments. You can also access metrics and alert details from this page.

To access the **Edge Events** screen, click Edge Events in the left navigation pane.



The **Edge Events** interface provides the following details for events:

- Date/Time: The timestamp when the event occurred.
- Severity: It indicates the event's level of importance (for example: DEBUG, ERROR, INFO, ALERT).
- Event Type: The category of the event (for example: Heartbeat Received).
- Message: A short description of the event.
- Class Name: The agent class associated with the event. Clicking the value in this field opens the Metrics tab from the Details pane of a deployment on the Dashboard, allowing you to track detailed metrics and alerts for the class.
- Source Type: It indicates whether the event originates from an agentor another source.
- Event Source ID: A unique identifier for the event source.

The number of rows displayed on a page can be configured (20, 50, or 100) using the Rows per page drop-down option in the bottom-right corner.

For detailed information about a specific event, click the arrow at the end of the event row. A pane will appear, displaying the event's detailed information. To view updates, click Show now next to the Updates are available message in the upper-right corner of the screen.

Sorting and filtering options

Sorting

Click a column name to sort events in ascending or descending order.

Filtering

Use the filter fields at the top of the screen to narrow down the list of events. You can also filter the events by Severity, Event Type, Message, Class Name, Source Type, Event Source ID, and Time Window.

The Severity and Class Name filters provide drop-down menus with predefined options, allowing you to select from available values using checkboxes. This ensures consistency and accuracy for fields with a fixed set of valid options. Similarly, Time Window offers predefined values to refine the event list based on a preferred time range. You can choose from the following options:

- All
- Last Hour

- Last 4 Hours
- Last 24 Hours
- Last 7 Days
- Since Last Publish

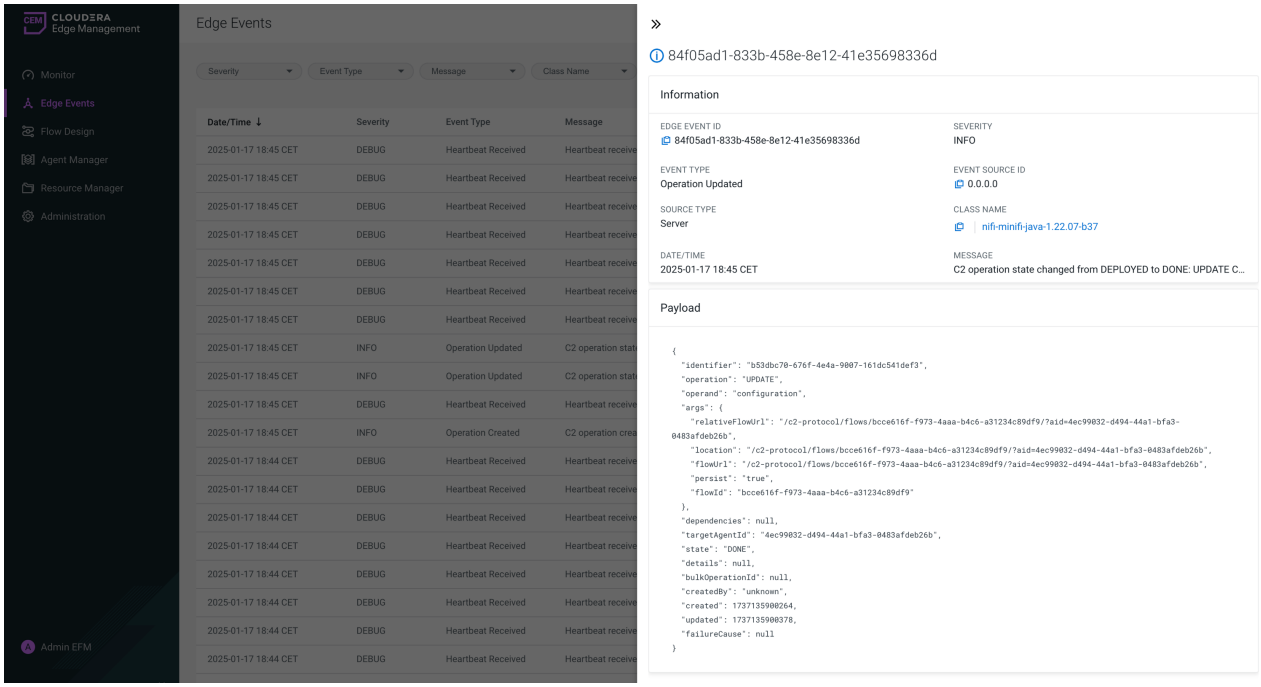
Important: To view events using Since Last Publish, you must select a single agent class and the flow has to be published.

The other filters, like Event Type or Message, include a free-text field. They allow you to enter custom values filtering data that can vary widely and cannot be predefined. Once you have entered or selected a value, press Enter on your keyboard to apply the filter and update the event list.

You can apply filters to multiple columns simultaneously to refine your data further. After applying filters, you can share the URL with others so they can view the same filtered event list.

Event details

You can view the details of an event by clicking the arrow at the end of the event row.



Related Information

Monitoring deployments in Cloudera Edge Management
Managing agents in Cloudera Edge Management

Monitoring metrics with Grafana in Cloudera Edge Management

in Cloudera Edge Management 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 in Cloudera Edge Management.

You need to perform the following tasks before you start visualizing in Cloudera Edge Management metrics with Prometheus and Grafana.

Enabling Prometheus metrics in in Cloudera Edge Management

Ensure that the following metrics exporting property is enabled in in Cloudera Edge Management 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 in Cloudera Edge Management. For instructions about how to install Prometheus, see the Prometheus website.
2. Configure your prometheus.yml file to scrape the in Cloudera Edge Management instance. For example:

   # Global config

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

   # The following is a scrape configuration for in Cloudera Edge Management. 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.
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 the Grafana website.
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 the Prometheus data source documentation.

3. Download the in Cloudera Edge Management Grafana dashboard templates.

   You can download the Grafana dashboard templates from Cloudera GitHub.

4. Import each JSON dashboard through the Grafana UI. Follow the Import dashboards instructions provided in the Grafana documentation.

   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.

   in Cloudera Edge Management dashboards should now be available in the Grafana UI.

Navigating to Grafana dashboard in in Cloudera Edge Management

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:https://www.youtube.com/embed/grE8pRwS0IM

NiFi expression language

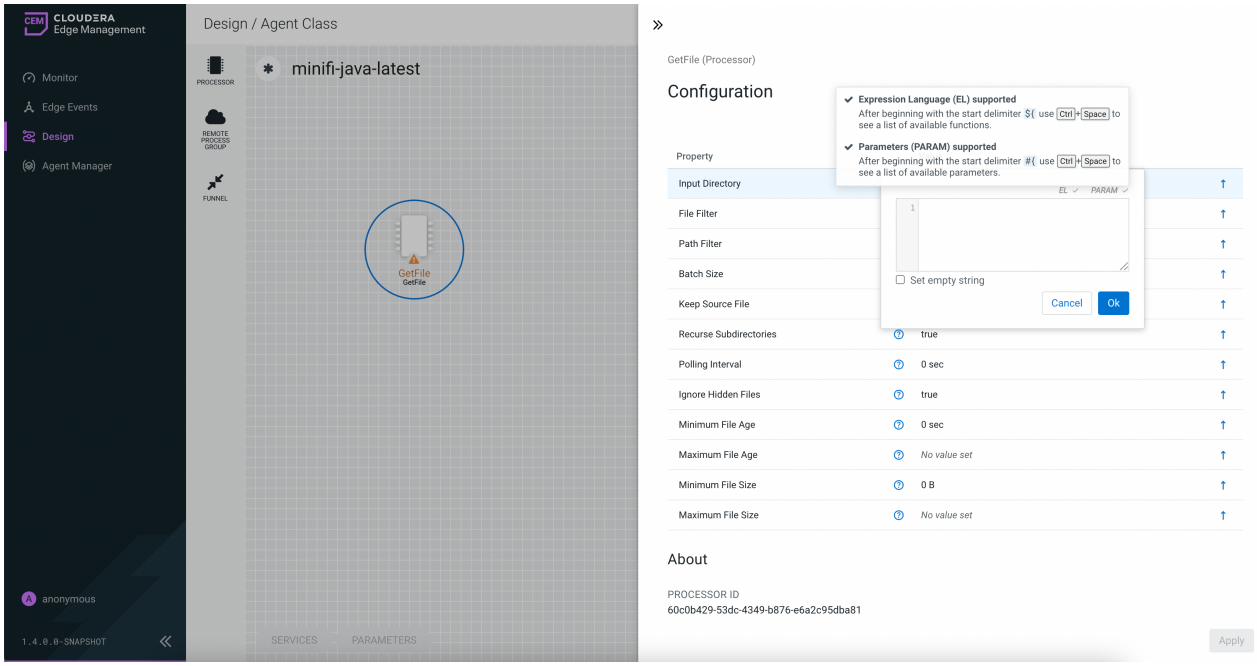
The NiFi Expression Language provides the ability to reference the attributes of flow files, compare them to other values, and manipulate their values.

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.



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.