

## Getting Started

Date published: 2020-06-22

Date modified: 2020-09-15

# CLOUdera

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

# Contents

<b>What is Cloudera Edge Management?</b> .....	<b>4</b>
MiNiFi.....	4
Edge Flow Manager.....	4
<b>Terminology</b> .....	<b>4</b>
<b>Downloading and Installing CEM</b> .....	<b>5</b>
<b>Accessing the CEM UI</b> .....	<b>6</b>
<b>Understanding the CEM UI</b> .....	<b>7</b>
<b>Creating your First Dataflow</b> .....	<b>8</b>
<b>Monitoring</b> .....	<b>13</b>

## What is Cloudera Edge Management?

Cloudera Edge Management (CEM) is a solution that enables you to manage, control, and monitor agents that are deployed on the edge devices in IoT implementations. You can use these agents to collect real-time data originating from the devices to create and push actionable intelligence and insights to the place of data origin.

CEM consists of two components:

- **MiNiFi** - A lightweight edge agent that implements the core features of Apache NiFi, focusing on data collection and processing at the edge.
- **Edge Flow Manager (EFM)** - An agent management hub that supports a graphical flow-based programming model to develop, deploy, and monitor edge flows on thousands of MiNiFi agents.

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

### MiNiFi

MiNiFi is an edge agent and can be deployed into thousands of edge devices to collect data. It is a light-weight version of NiFi and acts as a runtime at the edge to execute data flows. It comes in two flavors: Java and C++.

- **Java agent**

The Java agent is able to run most of the available processors of NiFi, but is a larger binary distribution and consumes greater system resources. If you need maximum flexibility to make routing and processing decisions at your point of origin of data, the Java agent is a good fit.

- **C++ agent**

The C++ agent is a smaller binary, consumes low system memory but it is able to run a limited subset of NiFi processors. If your primary concern is gathering and pushing data to downstream consumers and minimizing system impact, the C++ agent is a good fit. The smaller size enables it to be embedded within various types of devices as small as Raspberry Pi or other sensors used within IoT implementations.

### Edge Flow Manager

Edge Flow Manager (EFM) is a management hub that supports a GUI-based tool to manage, control, and monitor MiNiFi agents deployed in the field. EFM helps reduce the time and cost of developing IoT applications by enabling you to visually build flows for collecting and processing data without writing any code.

EFM allows you to develop, deploy, run, and monitor edge flow applications and machine learning models at the edge. EFM offers an easy-to-use NiFi-like user interface that allows users to leverage many of the NiFi processors to design data flows that can be pushed out to the edge. These data flows can instruct the edge agent to collect specific data points from the edge device as well as process it at the edge and stream it into the enterprise. These flows can also be changed from the same user interface and can be deployed to the edge to any specific class of devices. This allows the user to change the behavior of a specific set of agents in the field based on specific criteria.

## Terminology

Learn the terminologies used in CEM.

### Agent

Apache MiNiFi Java or C++ agent. MiNiFi implements the core features of Apache NiFi, focusing on data collection and processing the data at the edge.

### Agent Manifest Resolution Strategy

The logic used by the EFM server to assign an Agent Manifest to an Agent Class for the purpose of controlling which components are available in the Flow Designer for that Agent Class.

### Bucket

A container in NiFi Registry that stores and organizes flows.

### C2 Protocol

The MiNiFi C2 (Command and Control) protocol is an open standard defined as part of the Apache NiFi project and licensed using the Apache Software License version 2 (ALv2). It provides the ability for MiNiFi agents to communicate status to a central server and for that server to control many agents using encoded operations.

In CEM, the Edge Flow Manager acts as a C2 Server for MiNiFi agents, which is achieved using its implementation of the open C2 Protocol standard.

### Agent Class

An agent class allows you to configure and monitor a group of MiNiFi agents.

### Connection

You create an automated dataflow by dragging components from the CEM Components toolbar to the canvas and then connect the components together by using connections. Each connection consists of one or more relationships. For each connection that is drawn, you can determine which relationships should be used for the connection. This allows data to be routed in different ways based on its processing outcome.

### Content Repository

Content repository is a repository where the actual content bytes of a given flowfile live.

### Dataflow

Dataflow is an automated and managed flow of information between systems.

### Edge

Edge is the device that you want to manage, control, and monitor through CEM. To do so, you install the MiNiFi agent at the edge device to collect data and then pushes intelligence back to the same edge device.

### Flowfile Repository

Flowfile repository is a repository where CEM keeps track of the state of what it knows about a given flowfile that is presently active in the flow.

### Heartbeat

MiNiFi agents communicate their health and status to EFM through heartbeats. By default, agents heartbeat every second, but this interval can be configured on each agent to an interval appropriate for each use case. The agent heartbeat includes device information, agent status, and metrics for the currently running dataflow. The EFM server tracks agents through their heartbeats, and can respond to heartbeats with operations, such as dataflow configuration updates.

Optionally, agent health tracking can be enabled through a maximum heartbeat interval in the `efm.properties` file. If this interval threshold is exceeded without receiving a heartbeat from an agent, the agent will be flagged as offline/missing and an alert will be shown in the EFM web UI.

### Provenance Repository

Provenance repository is a repository where data from all provenance events is stored.

## Downloading and Installing CEM

You download and install the CEM software to start working with it.

## Download the software

The download location of CEM software bits is <https://www.cloudera.com/downloads.html>.

For more information about how to obtain the CEM software bits, see *Obtaining the CEM Software*.

## Install CEM

You should install Java 8 on the machine on which you will install the EFM Server, NiFi Registry, and each machine onto which you will install a MiNiFi Java agent.

To install EFM and use it, you need to perform the following steps:

1. Install database.
2. Install NiFi Registry.
3. Download the EFM server binaries.
4. Install EFM server.
5. Configure EFM server.
6. Open network ports.
7. Start EFM server.

For details on how to install EFM server, see *Installing the EFM Server*.

To install MiNiFi agents and use it, you need to perform the following steps:

1. Install your MiNiFi agents.
2. Configure your MiNiFi agents.
3. Start MiNiFi agents.

For details on how to install, configure, and start MiNiFi agents, see *Install the MiNiFi Agents*.

For detailed system requirements, and supported processors for MiNiFi Java and C++ agents, see *Support Matrix*.

## Related Information

[Install EFM server](#)

[Install the MiNiFi agents](#)

[System requirements](#)

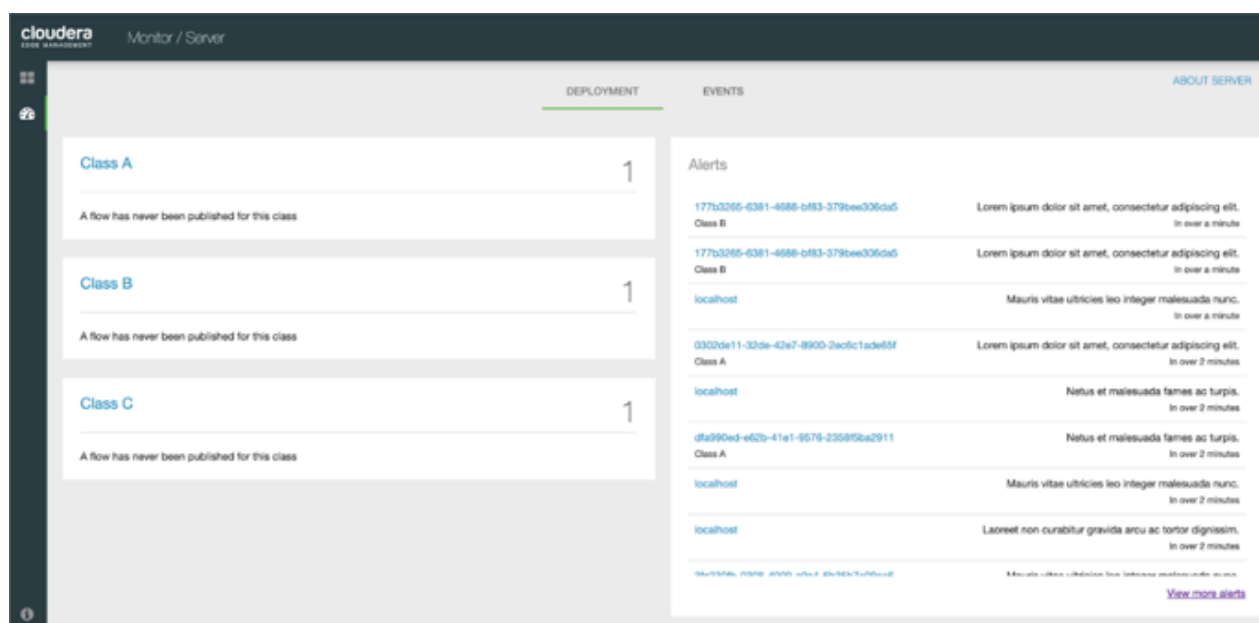
[Obtaining the CEM software](#)

# Accessing the CEM UI

After you start the EFM server, you can bring up the CEM User Interface (UI) in order to build, edit, and publish dataflows. You can also monitor the EFM server and agent events by accessing the CEM UI.

To get started, open a web browser and navigate to `http://<hostname>:10080/efm/ui`. The port can be changed by editing the `efm.properties` file in the EFM conf directory, but the default port is 10080.

The UI opens and displays the Deployment Monitor screen, as shown in the following image:



The first column, from the left, displays the classes available in the system. The classes allow you to create a single configuration template for multiple agents.

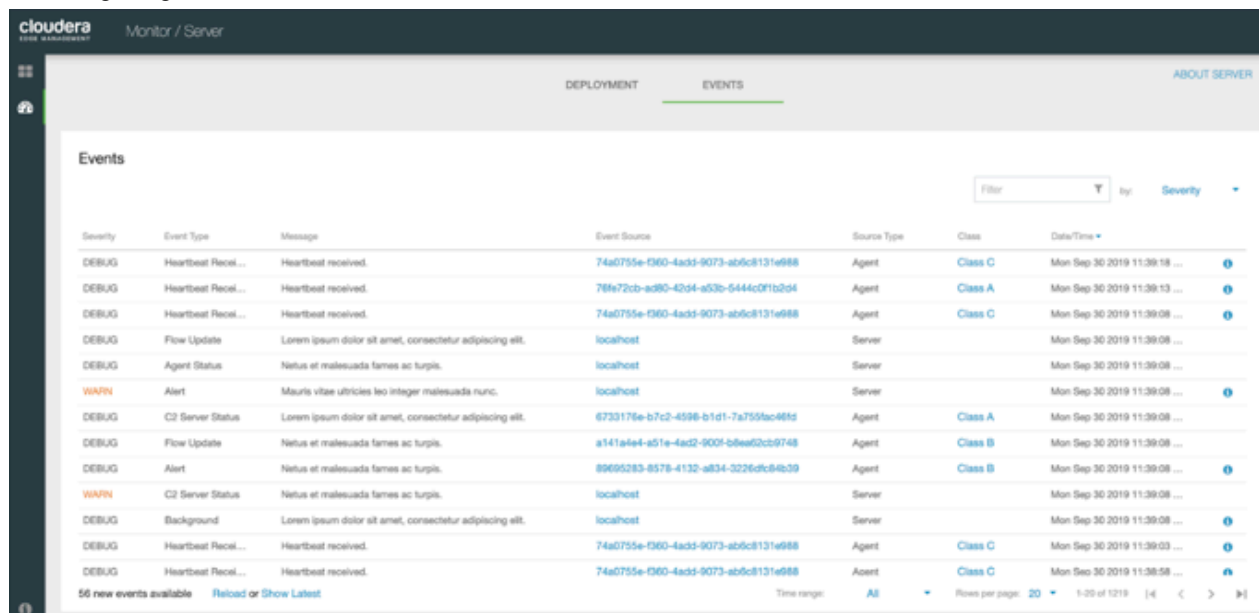
The Alerts column displays the alerts available in the system. To view more alerts, click the View more alerts link.

To fetch details about the C2 server, click the About Server link. The About This Server dialog displays the server URL, registry URL, registry bucket ID, and registry bucket name.

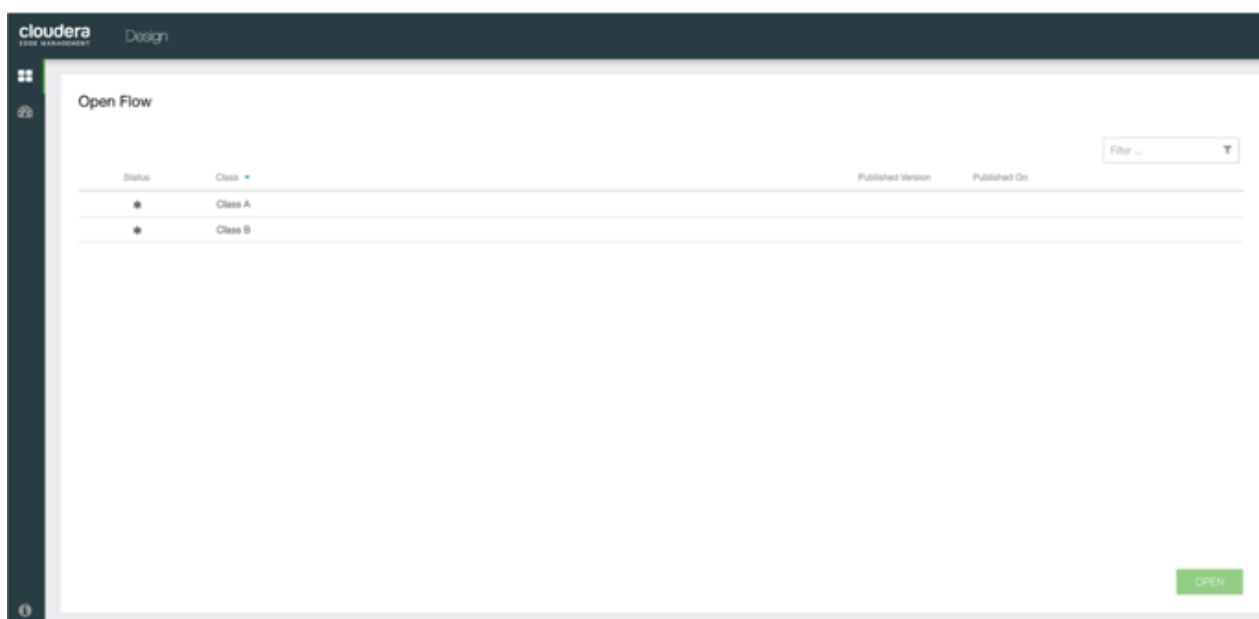
## Understanding the CEM UI

Understanding the CEM UI helps you to monitor events, design dataflows, and build dataflows.

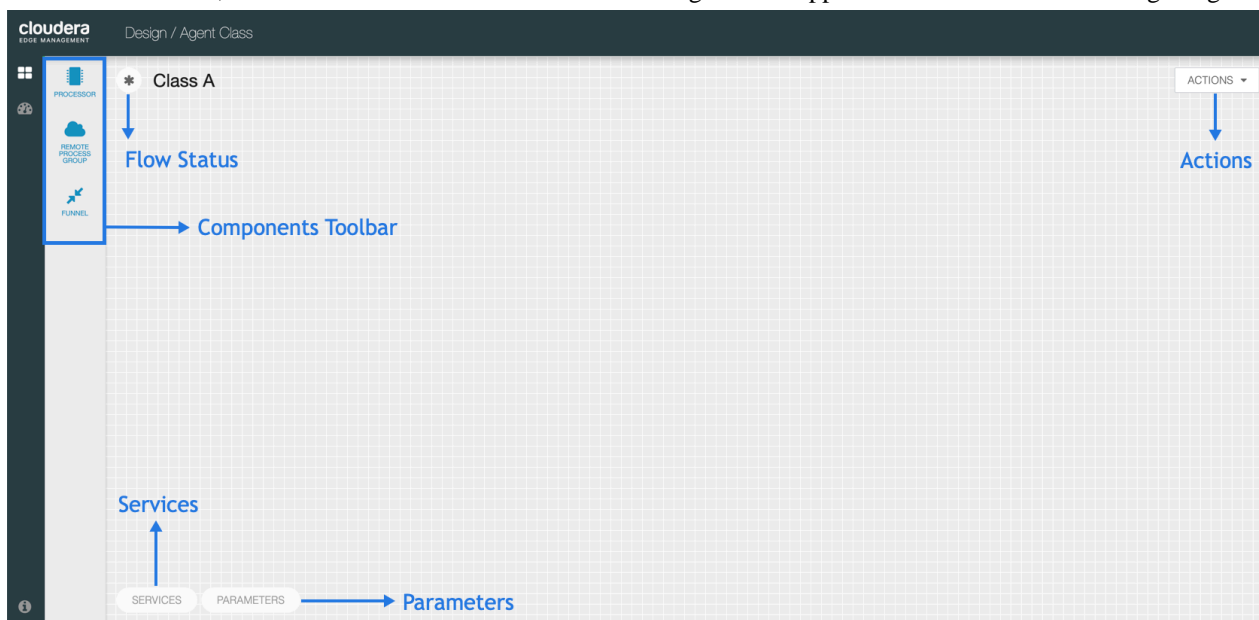
To monitor events, navigate to the Events page by clicking Events. The Events Monitor page appears, as shown in the following image:



To design a dataflow, navigate to the Flow Designer page by clicking the Design icon. The Open Flow screen initially appears, as shown in the following image, and displays information about the classes in the system:



To build a dataflow, double-click on the desired class. The Design screen appears as shown in the following image:



## Creating your First Dataflow

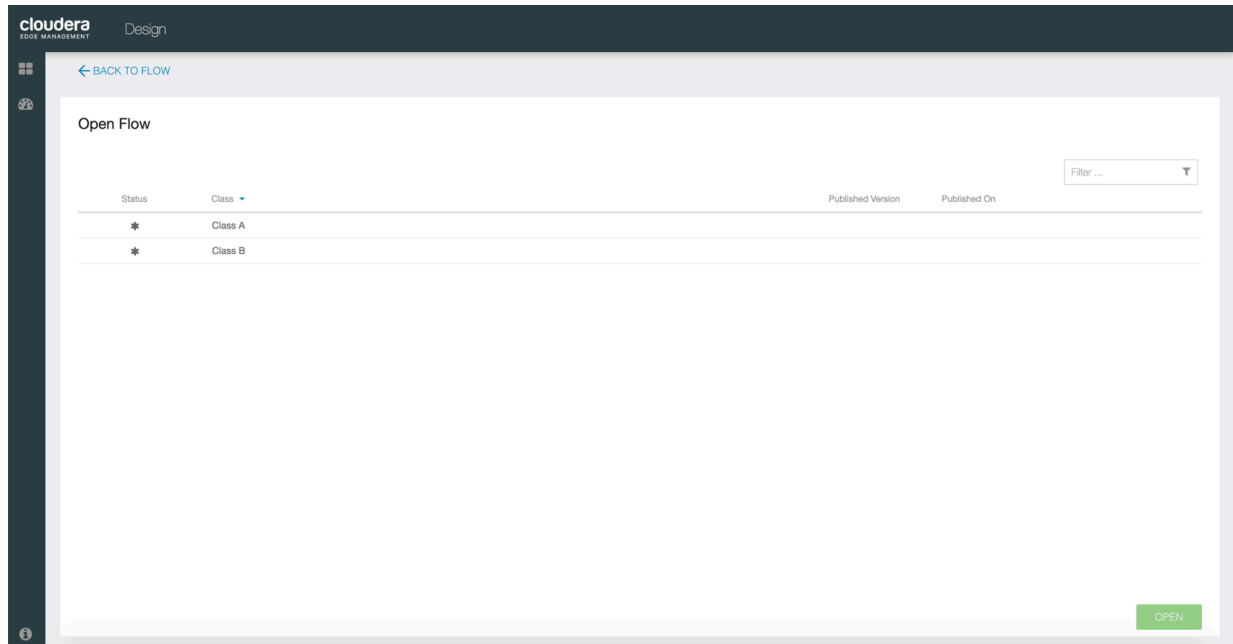
You can create an automated dataflow using the CEM UI. Simply drag components from the toolbar to the canvas, configure the components to meet specific needs, and connect the components together.

Perform the following steps to create a simple and your first dataflow:



1. Click the Design icon (🏠) to navigate to the Flow Designer.

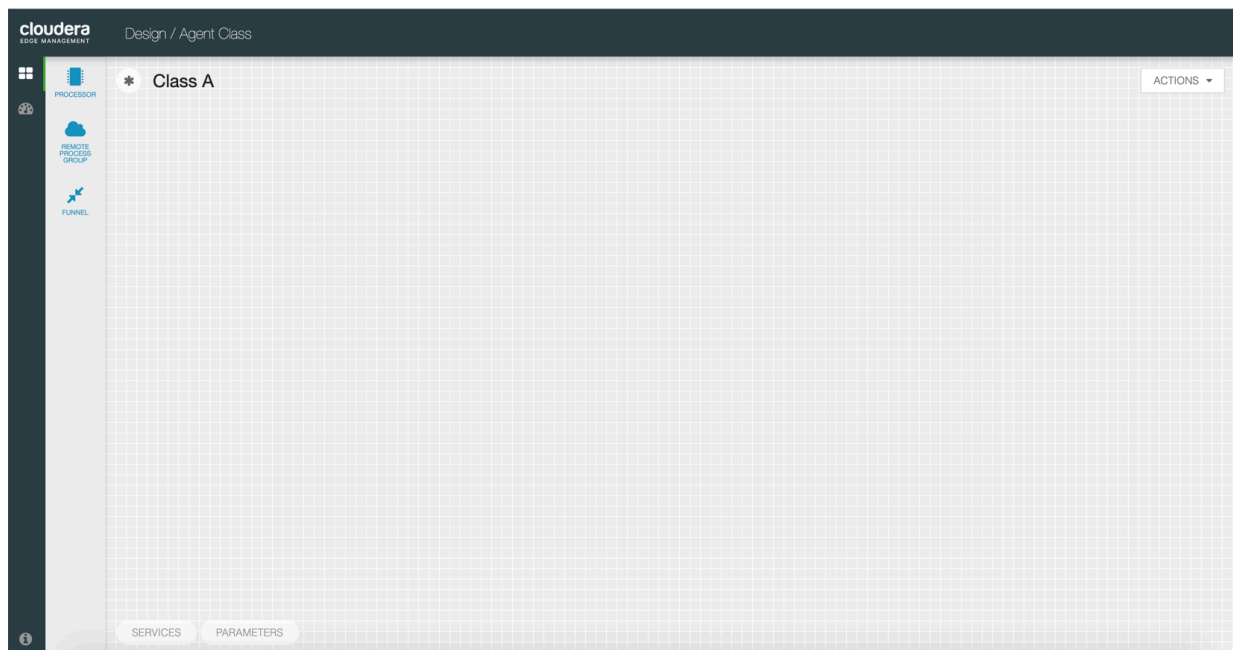
The Open Flow screen appears, as shown in the following image:



2. Select the class you want to create the dataflow in and click OPEN.

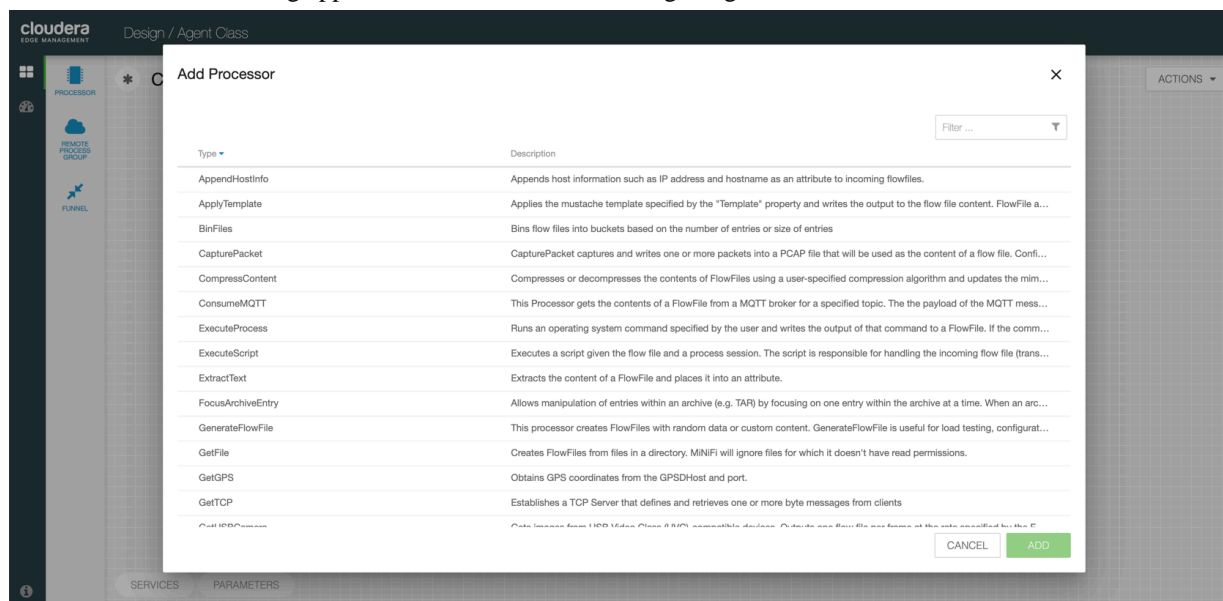
For example, select Class A.

The Design screen appears, as shown in the following image:



3. Drag the PROCESSOR option onto the canvas.

The Add Processor dialog appears, as shown in the following image:



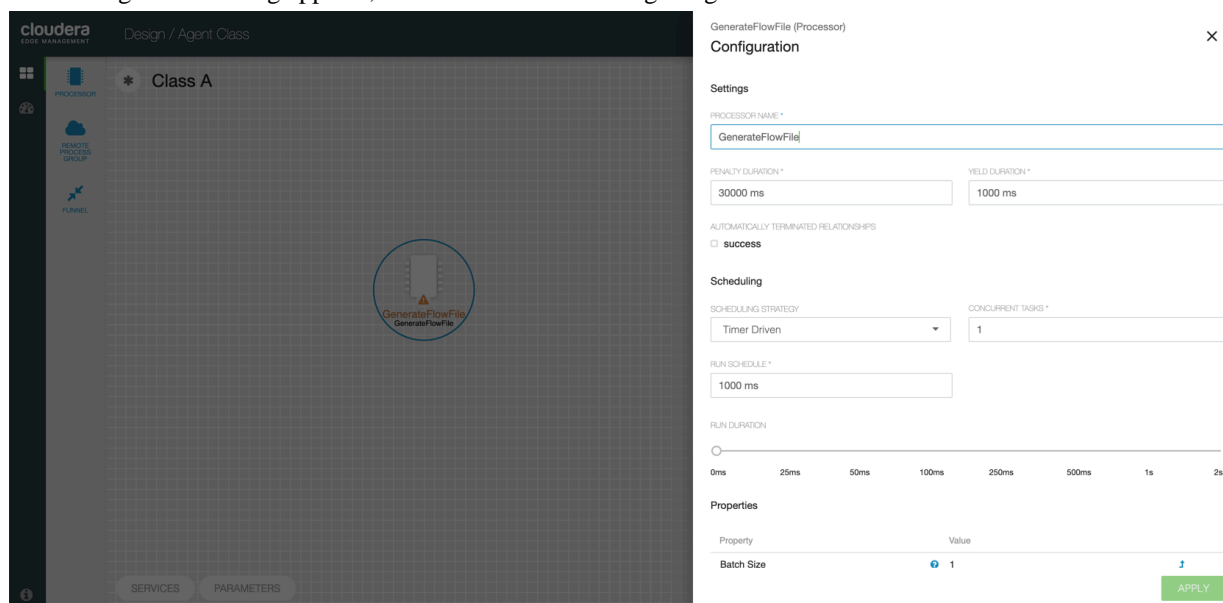
4. Select a processor and click the ADD button to add the selected processor to the canvas at the location that it was dropped.

Alternatively, you can double-click on a processor type to add it to the canvas.

For example, add GenerateFlowFile processor.

5. Double-click on the processor, or right-click on the processor and select Configure from the context menu to configure the processor properties.

The Configuration dialog appears, as shown in the following image:

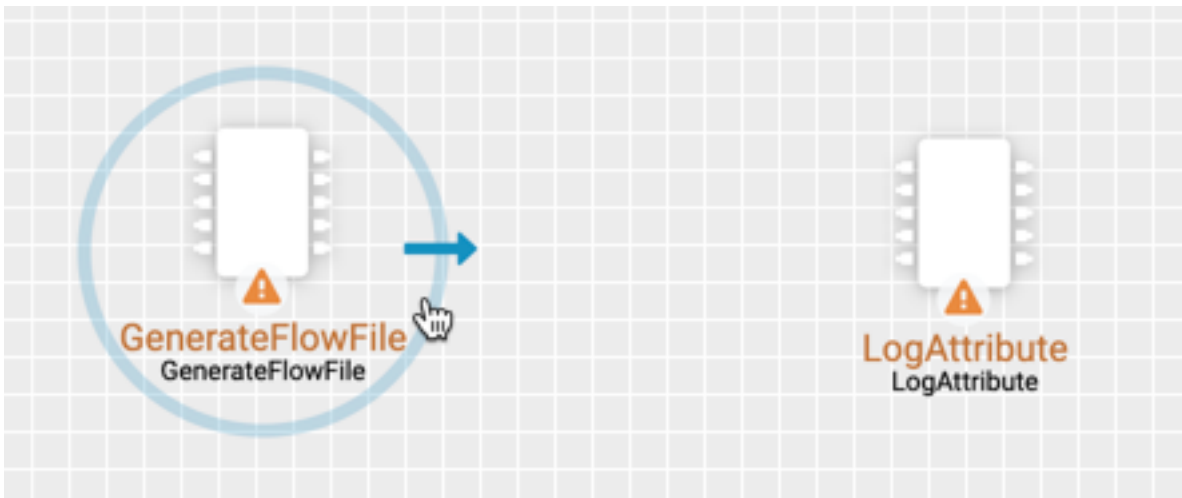


6. Configure the processor properties, and click the APPLY button to apply the changes.
7. Follow steps 3-6 to add another processor and configure the processor properties.

For example, add LogAttribute processor.

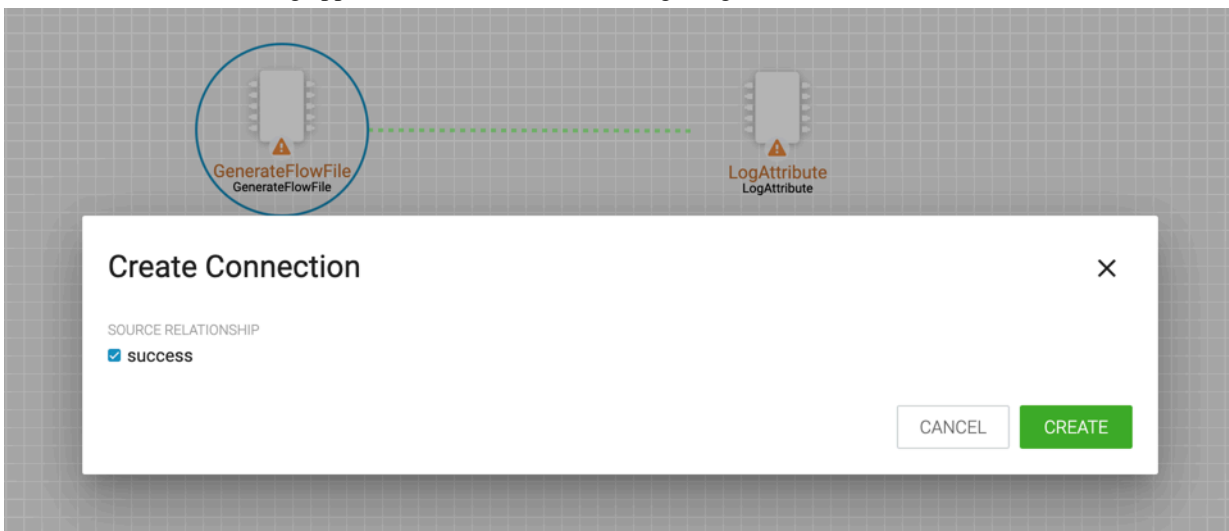
8. To connect the processors, hover the mouse over the GenerateFlowFile processor.

An arrow appears as shown in the following image:



9. Drag the arrow from the GenerateFlowFile processor to the LogAttribute processor until the LogAttribute processor is highlighted, and then release the mouse.

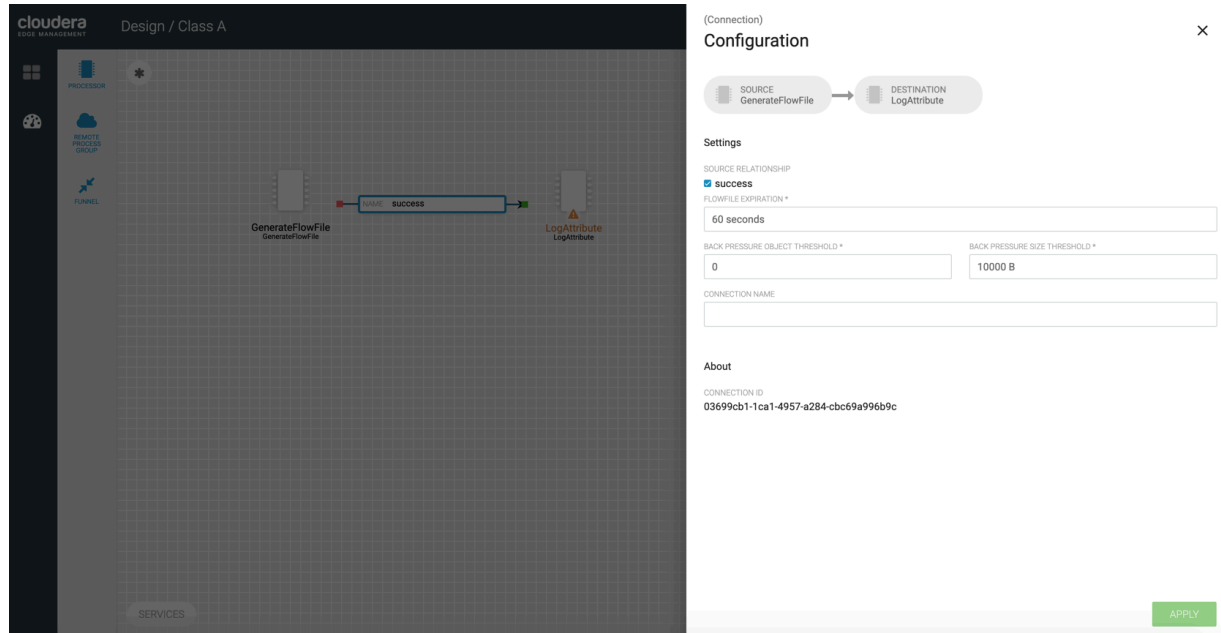
A Create Connection dialog appears as shown in the following image:



10. Select CREATE to create the connection.

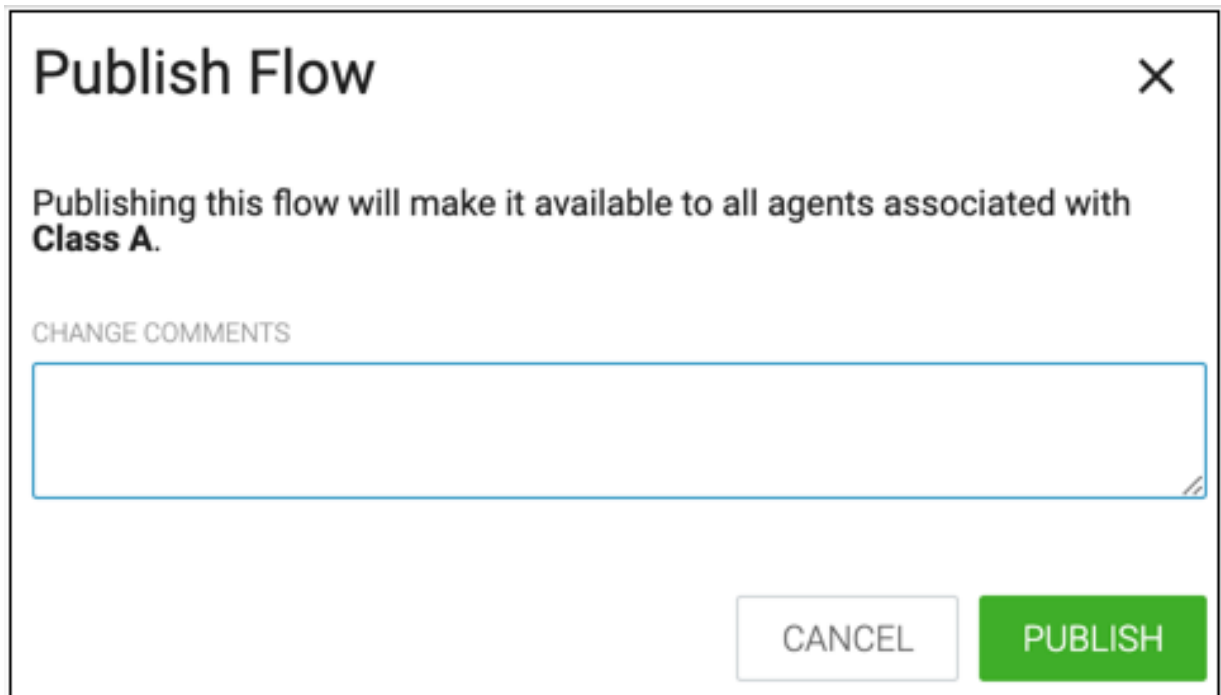
11. To change the configuration of a connection, right-click on the connection and select the Configure option, or double-click on the connection.

The Configuration dialog opens as shown in the following image:



12. Configure the connection properties and click the **APPLY** button to apply the changes.
13. 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:



14. Enter comments if desired, and click **PUBLISH**.

For more information on the above actions, see *Building a DataFlow*.

### Related Information

[Building a DataFlow](#)

# Monitoring

You can use CEM to monitor deployments and events.

While working with CEM, you can monitor the following things:

- Deployments

You need to navigate to the Deployments Monitor screen to monitor CEM deployments. After you navigate to the Deployments Monitor screen, you can monitor the classes, alerts, and server details.

For more information, see *Monitoring Deployments* section.

- Events

You need to navigate to the Events Monitor screen to monitor CEM events. After you navigate to the Events Monitor screen, CEM provides the following details for severity, event type, message, event source, source type, class, and date/time. You can also sort and filter the events according to your needs. You can use the reload and show latest links in the UI to view new event details. You can also fetch details of each event available in the system.

For more information, see *Monitoring Events* section.

## Related Information

[Monitoring Deployments](#)

[Monitoring Events](#)