Cloudera Edge Management 2.3.1

Deploying an Edge Flow Manager Cluster

Date published: 2019-04-15
Date modified: 2025-02-28

CLOUDERA

Legal Notice

Contents

Deploying an Edge Flow Manager cluster

Learn how to deploy an Edge Flow Manager cluster and the requirements to do so including supported number of nodes, supported databases, load balancer, and MiNiFi agents.

The Edge Flow Manager server component of Cloudera Edge Management supports clustered deployments for horizontal scalability and high availability. An EFM cluster deployment requires the following:

- Three or more Edge Flow Manager nodes

  Two-node clusters are not supported.
- A shared and external Edge Flow Manager database (MySQL, MariaDB, or PostgreSQL)
- An Edge Flow Manager load balancer for Web UI users

  You can use any load balancer of your choice, provided it can proxy HTTP(s) traffic and can be configured as described in this document.
- MiNiFi agents that forward client requests to one of the Edge Flow Manager cluster nodes

In addition to communicating with the external, centralized, and shared Edge Flow Manager database, Edge Flow Manager cluster members also communicate with each other to establish ephemeral state, such as distributed locks and caches, within the cluster. Due to this, each Edge Flow Manager node must have network connectivity to every other cluster member.

Edge Flow Manager cluster database setup

Before you first run or launch the Edge Flow Manager cluster, you must learn how to install and configure MySQL, MariaDB, or PostgreSQL.

For more information, see *Installaing Databases*.

You can use an existing external database of a single and standalone Edge Flow Manager instance for the cluster deployment. None of the existing data will be lost.

However, you cannot use an H2 database of a standalone Edge Flow Manager instance and you must start over with an external database. When moving from H2 to MySQL, MariaDB, or PostgreSQL, prior to stopping the standalone instance, export the existing flows from the Edge Flow Manager Flow Designer and import into the cluster to avoid data loss. For more information on exporting and importing flows, see the REST API Reference.

Related Information
Installing databases for Edge Flow Manager

Edge Flow Manager cluster configuration

Learn how to configure every Edge Flow Manager cluster member to make it part of the same cluster.

You can find the cluster configuration in the efm.properties file located in the conf directory of the Edge Flow Manager installation. The following is an example cluster configuration:

```
# Web Server Properties
#  address: the hostname or ip address of the interface to bind to; to bind
to all, use 0.0.0.0
efm.server.address=10.0.0.1
efm.server.port=10090
efm.server.servlet.contextPath=/efm
efm.cluster.enabled=true
# Database Properties
efm.db.url=jdbc:mysql://database.example.come:3306/efm
efm.db.driverClass=com.mysql.cj.jdbc.Driver
```

```
efm.db.username=efm
efm.db.password=efmPassword
efm.db.maxConnections=50
efm.db.sqlDebug=false
```

> **Note:** Except for the node-specific configurations, such as efm.server.address and efm.cluster.address, which can be a hostname or IP address, all other configurations must match for all nodes in the cluster. It is especially important that the nodes use the same database URL (efm.db.url), and the same list of cluster members (efm.infinispan.jgroups[tcpping.initial_hosts]). If these values do not match on all cluster nodes, the cluster fails on startup.

Cluster node discovery

Cluster node discovery allows you to configure multiple node discovery strategies to establish an Edge Flow Manager cluster, tailored to different deployment environments.

You have to define the efm.infinispan.environment property with the name of the environment where Edge Flow Manager is deployed. The default value is tcp, and the supported values are tcp, tcp-fixed, and kubernetes.

Here are the specifics for each environment setting:

tcp

TCP discovery uses IP multicast to discover the initial membership, eliminating the need for manual configuration on each node. This is the default setting.

Property required: efm.infinispan.environment=tcp

tcp-fixed

You have to provide a static list of host:port combinattions for all nodes.

Properties required:

- efm.infinispan.environment=tcp-fixed
- efm.infinispan.jgroups[tcpping.initial_hosts]

    Example value: localhost[7801],localhost[7802],localhost[7803]

    > **Note:** Port numbers have to match the efm.infinispan.jgroups[bind.port.jgroups.tcp .port] startup settings where Infinispan is running. The default port is 7800.

kubernetes

This strategy is designed to automatically discover pods based on namespace and label filtering.

Properties required:

- efm.infinispan.environment=kubernetes
- efm.infinispan.jgroups[kubeping.namespace]

    Example value: cem-namespace
- efm.infinispan.jgroups[kubeping.labels]

    Example value: app=efm

> **Note:**
> Setting up labels and namespaces in Kubernetes is the responsibility of the user.

Edge Flow Manager cluster high availability

Learn why you are able to operate in an Edge Flow Manager cluster with high availability.

In a clustered mode, Edge Flow Manager is highly available.

Even if individual nodes are stopped the cluster can continue operation. Nodes can stop or fail due to a variety of reasons, such as routine maintenance, loss of network connectivity, or underlying physical server host failure. By default, an Edge Flow Manager cluster can continue operating with up to two nodes down. If the failed nodes are restored, the cluster reforms. Note that any load that was served by a failed node is shifted to other cluster nodes. Hence capacity may fall if all nodes in the cluster were optimally utilized with regards to resources including network, CPU, memory, and disk.