Cloudera Edge Management 1.5.1

Cloudera Edge Management Upgrade

Date published: 2019-04-15
Date modified: 2023-04-18

https://docs.cloudera.com/

Legal Notice

Contents

Overview of Edge Flow Manager upgrade

Learn how to upgrade the Edge Flow Manager (EFM) server.

Note: Prior to CEM 1.1.0 release, the MySQL database driver was distributed with software. It will no longer be distributed with software. To use it, you need to download and configure the MySQL database driver. Download the MySQL database driver from https://dev.mysql.com/downloads/connector/j/. For more details, see *Installing and Configuring MySQL*.

Related Information

Installing and configuring MySQL

Upgrading EFM

To upgrade to the latest version of Edge Flow Manager (EFM), you must download the tar file, uncompress the tar file, configure properties, stop the old server, and start the new server.

Procedure

1. Download the tar file of the latest release.
2. Unzip the tar file.
3. Configure the following:
   a) Copy over any custom configuration property values from the previous install to the new install.
   b) Copy over the database driver from the previous install to the new install. Either take the driver you downloaded earlier (as described in *Installing databases for EFM*) or you can download it again.
   c) In order to not lose any data, confirm that the database properties that start with efm.db.* match identically to the previous install so that the new version of the server connects to the old database. If you are using H2 database, do not forget to sync the content of the database folder between the old and the new installation.
   d) Set the efm.encryption.password property.

      This is a master password used for encrypting sensitive data saved to the EFM server. You can set it through the efm.properties file, a command line argument, or an OS environment variable.

      By default, the EFM application uses AES encryption. The encryption key used is deterministically derived from an encryption password that the admin user must provide to the application at runtime. The property

that is read for the encryption password is efm.encryption.password. You can set the value for this property in following ways:

- As a command line argument: ./bin/efm.sh --efm.encryption.password=myEfmPassword
- As a Java System Property: -Defm.encryption.password=myEfmPassword
- As an OS environment variable: export EFM_ENCRYPTION_PASSWORD=myEfmPassword
- As a key/value pair in the efm.properties file: efm.encryption.password=myEfmPassword

Note: The master encryption password must be at least 12 characters long. It must be the same for all EFM instances.

The derived encryption key length is determined by your Java Runtime Environment encryption strength profiles.

- Unlimited Strength Encryption active: AES 256-bit key
- Unlimited Strength Encryption inactive: AES 128-bit key

It is strongly recommended to enable Unlimited Strength Encryption in your Java Runtime Environment.

e) Before you upgrade to version 1.3.0, decide what port you should use to access EFM, as the default EFM port is changed to 10090 (was 10080). This accounts for the changes in recent versions of browsers, including Chrome and Firefox, that are blocking HTTP, HTTPS, and FTP access to TCP port 10080 to prevent the ports from being abused in NAT Slipstreaming 2.0 attacks.

f) Set the efm.security.user.auth.enabled property, in the efm.properties file, to true if you are using a secured EFM with user authentication.

g) If user authentication is enabled, users no longer have access to all parts of EFM and need to be granted access policies to specific agent classes. You can set an initial admin identity for the user that grants access to other users using the efm.security.user.auth.adminIdentities property in the efm.properties file. For more information, see *Securing EFM*.

h) As SSO, OIDC and SAML are now available, you can migrate to those user authentication mechanisms if you were previously using Knox or mTLS with client certificates for user authentication.

4. Optional. Configure EFM to run as a service using, for example, init.d or systemd depending on your Linux distribution.
5. Stop the old server.
6. Start the new server.

- Use the following command to run as a background process:

```
/path/to/efm-<version>/bin/efm.sh start
```

- You can install EFM as an OS service and start it by using the OS service commands. For example, use the following command if EFM is installed as an OS service:

```
service efm start
```

Related Information

Securing Cloudera Edge Management

Installing databases for EFM

Installing EFM as an operating system service

The Edge Flow Manager (EFM) executable supports installation as a service on most Linux distributions. This is an optional installation step that is not required if you prefer to start the EFM server from the efm.sh executable included in the EFM bin directory.

You can start the application as a service by using either init.d or systemd.

Install EFM as an init.d service

To install EFM as an init.d service, symlink bin/efm.sh to init.d.

```
$ sudo ln -s /path/to/efm/bin/efm.sh /etc/init.d/efm
```

Once installed, you can start and stop the service as you would other OS services. For example:

```
$ service efm start
```

To configure EFM to start automatically on system boot, use update-rc.d. See man update-rc.d for information on using this utility.

Note: The EFM application runs as the user who owns the efm.sh launch script. It is recommended to never run as root. The recommended best practice is to create a specific user for running efm. Then use chown to make that user the owner of efm.sh. For example:

```
$ chown efm:efm /path/to/efm/bin/efm.sh
```

It is also recommended to use Unix or Linux filesystem permissions in order to secure the EFM installation. The rule of setting minimal access permissions applies. All files in the EFM installation should only be accessible to the EFM run-as user. Configuration files should be made read-only (for example, chmod 40 0 <file>). Executable files, such as those in the bin directory, should be made read and executable only (for example, chmod 500 <file>). Directories in the EFM install location should be readable and writable to the EFM user (for example, chmod 600 <dir>).

Install EFM as a systemd service

Most modern Linux distributions now use systemd as the successor to init.d (System V). In many cases you can continue to use init.d, but it is also possible to launch EFM using systemd as a service configuration.

To install EFM as a systemd service, create a file named efm.service in the /etc/systemd/system directory. For example:

```
[Unit]
Description=efm
After=syslog.target

[Service]
User=efm
ExecStart=/path/to/efm/bin/efm.sh
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

Note: When using systemd, the run-as user, the PID file, and the console log file are managed by systemd and therefore must be configured by using appropriate fields in the service script. Consult the service unit configuration man page for more details.

To configure EFM to start automatically on system boot, use systemctl. See man systemctl for information on using this utility.