

Upgrade 1.2.1

Cloudera Edge Management Upgrade

Date published: 2019-10-25

Date modified: 2020-09-15

CLouDERA

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

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Overview of Cloudera Edge Management Upgrade.....	4
Upgrading EFM.....	4
Installing EFM as an operating system service.....	5
Upgrading MiNiFi Agents.....	6

Overview of Cloudera Edge Management Upgrade

To upgrade CEM, you must upgrade the EFM server. But upgrading the MiNiFi agents are optional.



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 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) 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.
 - c) 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.

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
```

Installing EFM as an operating system service

The 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 400 <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.

Upgrading MiNiFi Agents

Upgrade your MiNiFi Java or MiNiFi C++ agents.

Upgrade MiNiFi Java agents

1. Stop the previous MiNiFi instance.
2. Unpack the new assembly.
3. Copy the following files from the conf directory of the previous instance to the conf directory of the new instance:
 - agent-identifier
 - device-identifier (if exists, it is only generated for some platforms)
 - config.yml
 - bootstrap.conf
4. Migrate the following directories to the root of the new instance:
 - content_repository
 - flowfile_repository
 - provenance_repository
 - state
5. Start the new MiNiFi instance.

Upgrade MiNiFi C++ agents

1. Stop the previous MiNiFi instance.
2. Unpack the new assembly.
3. Copy the following files from the conf directory of the previous instance to the conf directory of the new instance:
 - minifi-log.properties
 - minifi-uid.properties
 - config.yml
 - minifi.properties
4. Migrate the following directories to the root of the new instance:
 - content_repository
 - flowfile_repository
 - provenance_repository
5. Migrate the following file(s) to the root of the new instance from the root of the previous instance:
 - .device_id
6. Start the new MiNiFi instance.