

Installation 1

# Cloudera Edge Management Installation

Date of Publish: 2019-04-15



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

# Contents

<b>Before You Begin.....</b>	<b>3</b>
System Requirements.....	3
Install the JDK.....	3
<b>Installing the EFM Server.....</b>	<b>4</b>
Installing Databases.....	4
Installing MySQL.....	4
Configuring the EFM Server Metadata Store in MySQL.....	5
Install Postgres.....	5
Configure Postgres to Allow Remote Connections.....	6
Configure the EFM Server Metadata Store in Postgres.....	6
Install NiFi Registry.....	6
Obtaining the EFM Server Binaries.....	7
Install EFM Server.....	7
Configure the EFM Server.....	8
Open Network Ports.....	8
Start the EFM Server.....	8
<b>Install the MiNiFi Agents.....</b>	<b>9</b>
Install your MiNiFi Agent.....	9
Configure your MiNiFi Agents.....	9
Start MiNiFi Agents.....	10
<b>Configure Security.....</b>	<b>10</b>
Configure EFM Server TLS.....	10
Configure MiNiFi Agent TLS.....	10

## Before You Begin

### System Requirements

Before you begin your installation, carefully review the system requirements to understand operating system, database, browser, and JDK support.

#### Operating System Support

Operating System	Version
RHEL/CentOS	7.0, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6
Debian	9
Ubuntu	14.04, 16.04, 18.0
Windows	8, 10, Server 2012, Server 2012, R2 Server 2016

#### JDK Support

JDK	Version
OpenJDK	JDK8
OracleJDK	JDK8

#### Supported Databases

Database	Version
Postgres	9.6
MySQL	5.7, 5.6, 5.5

#### Browser Support

Browser	Version
Chrome	70.0, 69.0
Firefox	63.0, 62.0

### Install the JDK

You should install JDK 8 on the machine on which you will install the EFM Server, and each machine onto which you will install a MiNiFi Agent.

#### Procedure

1. Download JDK from the appropriate website.
2. Run the installation command appropriate for your operating system:

For RHEL/CentOS:

```
yum install java-1.8.0-openjdk-devel
```

For Debian and Ubuntu:

```
apt-get install openjdk-8-jdk
```

### Related Information

[Open JDK Download](#)

[Oracle JDK Download](#)

## Installing the EFM Server

### Installing Databases

Installing CEM requires a relational datastore for metadata. You can use either MySQL or Postgres. These topics describe how to install MySQL and Postgres, and how to create a database for the CEM metastore.



#### Note:

You should install either Postgres or MySQL; both are not necessary.

### Installing MySQL

You can install MySQL 5.5 or 5.6.

#### About this task

If you have already installed a Postgres database, you may skip these steps.

#### Procedure

1. Log in to the machine on which you want to install the MySQL metastore to use for the EFM Server.
2. Install MySQL and the MySQL community server, and start the MySQL service:

```
yum localinstall \
https://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm

yum install mysql-community-server

systemctl start mysqld.service
```

3. Obtain the randomly generated MySQL root password.

```
grep 'A temporary password is generated for root@localhost' \
/var/log/mysqld.log |tail -1
```

4. Reset the MySQL root password. Enter the following command. You are prompted for the password you obtained in the previous step. MySQL then asks you to change the password.

```
/usr/bin/mysql_secure_installation
```

## Configuring the EFM Server Metadata Store in MySQL

EFM Server requires a relational data store to store metadata. To use MySQL for this, install MySQL and create a database for the CEM metastore.

### Procedure

1. Launch the MySQL monitor:

```
mysql -u root -p
```

2. Create the database for the CEM metastore:

```
CREATE DATABASE EFM DEFAULT CHARACTER SET utf8;
```

3. Create the efm user account, replacing the final IDENTIFIED BY string with your password:

```
CREATE USER 'efm@%' IDENTIFIED BY '9oNio)exlndL';
```

4. Assign privileges to the efm account:

```
GRANT ALL PRIVILEGES ON *.* TO 'efm@%' WITH GRANT OPTION;
```

5. Commit the operation:

```
commit;
```

## Install Postgres

The EFM Server requires a relational database to store metadata. You may use Postgres 9.6 for this.

### About this task

If you have already installed a MySQL database, you may skip these steps.

### Procedure

1. Install Red Hat Package Manager (RPM) according to the requirements of your operating system:

```
yum install https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm
```

2. Install Postgres version 9.6:

```
yum install postgresql96-server postgresql96-contrib postgresql96
```

3. Initialize the database:

For example, if you are using CentOS 7, use the following syntax:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

4. Start Postgres.

For example, if you are using CentOS 7, use the following syntax:

```
systemctl enable postgresql-9.6.service  
systemctl start postgresql-9.6.service
```

5. Verify that you can log in:

```
sudo su postgres
```

```
psql
```

## Configure Postgres to Allow Remote Connections

It is critical that you configure Postgres to allow remote connections before you deploy CEM. If you do not perform these steps in advance of installing the EFM Server, the installation fails.

### Procedure

1. Open `/var/lib/pgsql/9.6/data/pg_hba.conf` and update to the following

```
# "local" is for Unix domain socket connections only
local all all trust

# IPv4 local connections:
host all all 0.0.0.0/0 trust

# IPv6 local connections:
host all all ::/0 trust
```

2. Open `/var/lib/pgsql/9.6/data/postgresql.conf` and update to the following:

```
listen_addresses = '*'
```

3. Restart Postgres:

```
systemctl stop postgresql-9.6.service
systemctl start postgresql-9.6.service
```

## Configure the EFM Server Metadata Store in Postgres

The EFM Server requires a relational data store to store metadata. To use Postgres for this, create a database for the CEM metastore.

### Procedure

1. Log in to Postgres:

```
sudo su postgres
psql
```

2. Create a database called `efm` with the password `efm`:

```
create database efm;
CREATE USER efm WITH PASSWORD 'efm';
GRANT ALL PRIVILEGES ON DATABASE "efm" to efm;
```

## Install NiFi Registry

Using CEM requires a NiFi Registry instance to store your dataflows. Installing NiFi Registry requires that you download the software and unpack it.

### Procedure

1. Download the NiFi Registry `.zip` or `.targz` files from the download location available in the *CEM Release Notes*.

2. Extract the NiFi Registry software to the desired directory.
3. Start NiFi Registry:

```
bin/nifi-registry.sh run
```

## Obtaining the EFM Server Binaries

You may download the EFM Server from the location provided to you in your Cloudera support information.

### Procedure

1. Log into your host as root.
2. Download the EFM Server binaries to a directory on your host:

For CentOS/RHEL 7:

```
wget efm-server-repo-location
```

For Ubuntu:

```
wget efm-server-repo-location  
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD  
apt-get update
```

For Debian:

```
wget efm-server-repo-location  
apt-key adv --recv-keys --keyserver keyserver.debian.com B9733A7A07513CAD  
apt-get update
```

3. Confirm that your download is successful.

For CentOS/RHEL 7:

```
yum repolist
```

For Ubuntu/Debian:

```
apt-cache showpkg
```

## Install EFM Server

### Procedure

Install the EFM Server.

For RHEL/CentOS 7:

```
yum install efm
```

For Ubuntu / Debian:

```
apt-get install efm
```

## Configure the EFM Server

Once the EFM Server is installed, you can configure it by editing the `efm.properties` file. At minimum, you should edit the EFM Server address, set up the connection to NiFi Registry, and configure the connection to your database.

### Procedure

1. Open the `efm.properties` file located in `$EFM_HOME/conf/efm.properties`.
2. Configure the EFM Server address. Change `efm.server.address=localhost` to `efm.server.address={YOUR_DESIRED_IP OR 0.0.0.0}`.
3. Configure your connection by editing the following two properties:  
 Change `efm.nifi.registry.enabled=false` to `efm.nifi.registry.enabled=true`.  
 Change `efm.nifi.registry.url=http://localhost:18080` to `efm.nifi.registry.url=http://{YOUR_REGISTRY_IP}:18080`.
4. Configure one of the following properties to identify the NiFi Registry bucket you want EFM to use. Do not set both.  
`efm.nifi.registry.bucketId=`  
`efm.nifi.registry.bucketName=`
5. Edit the following database properties according to what your database requires:
  - `efm.db.url=jdbc:h2:./database/efm;AUTOCOMMIT=OFF;DB_CLOSE_ON_EXIT=FALSE;LOCK_MODE=3`
  - `efm.db.driverClass=org.h2.Driver`
  - `efm.db.username=efm`
  - `efm.db.password=`
  - `efm.db.maxConnections=5`
  - `efm.db.sqlDebug=false`

## Open Network Ports

You should ensure that the following ports are available for the EFM Server and its components.

Component	Port number
EFM Server UI	10080
NiFi Registry	18080
Constraint application protocol (CoAP)	8989

## Start the EFM Server

Once you have installed and configured the EFM Server you can start it. In general, you should start the EFM Server before you install and configure your MiNiFi Agents.

### Procedure

1. From the EFM Server home directory, run:

```
bin/efm.sh start
```



2. Access the UI by browsing to the following location:

```
http://{EFM_IP}:10080/efm/ui
```

## Install the MiNiFi Agents

### Install your MiNiFi Agent

To install your MiNiFi agent, you need to download the software for the Agent you want to install and extract it.

#### Procedure

1. Download the tar or zip files for the MiNiFi Java or C++ Agent.

```
wget {java.tar.gz}
```

```
wget {cpp.tar.gz}
```

2. To install the MiNiFi Agent, extract the file to your desired home directory.

### Configure your MiNiFi Agents

Once you have installed the MiNiFi agent, update the configuration files.

#### About this task

If you are configuring a MiNiFi Java agent the configuration file is `conf/bootstrap.conf`. If you are configuring a MiNiFi C++ agent, the file is `conf/minifi.properties`.

#### Procedure

1. From the MiNiFi home directory, open the appropriate configuration file.
2. Configure the Agent Class so that you can logically group MiNiFi instances according to their functionality. Specify the agent class:

```
nifi.c2.agent.class={AGENT_CLASS}
```

3. Configure the Agent ID. If you do not specify an Agent ID, MiNiFi generates a unique ID per agent instance.

```
nifi.c2.agent.identifier={AGENT_ID}
```

4. Configure your EFM Server endpoint:

```
nifi.c2.rest.url=http://{EFM_SERVER_IP}:10080/efm/api/c2-protocol/  
heartbeat  
nifi.c2.rest.url.ack=http://{EFM_SERVER_IP}:10080/efm/api/c2-protocol/  
acknowledge
```

5. Configure your heartbeat interval:

```
nifi.c2.agent.heartbeat.period={HEARTBEAT_INTERVAL}
```

6. If you are installing a MiNiFi C++ Agent, you may also configure metrics. Metrics are not yet available for the MiNiFi Java Agent.

```
c2.agent.heartbeat.reporter.classes=RESTRceiver
```

## Start MiNiFi Agents

Once MiNiFi is installed and configured, you can start it.

### Procedure

1. From a terminal window, navigate to the MiNiFi installation directory.
2. To start MiNiFi in the foreground, enter:

```
bin/minifi.sh run
```

3. To start MiNiFi in the background, enter:

```
bin/minifi.sh start
```

## Configure Security

### Configure EFM Server TLS

You can configure the server to authenticate users based on a client certificate provided for TLS mutual authentication. The server's TLS settings, including what certificates it will trust, are configured using the `efm.security.tls.*` prefixed properties in the `efm.properties` file.

### Procedure

1. Open the `efm.properties`.

The properties pertaining to TLS configuration are the following:

```
efm.server.ssl.enabled=false
efm.server.ssl.keyStore=/path/to/keystore.jks
efm.server.ssl.keyStoreType=jks
efm.server.ssl.keyStorePassword=
efm.server.ssl.keyPassword=
efm.server.ssl.trustStore=/path/to/truststore.jks
efm.server.ssl.trustStoreType=jks
efm.server.ssl.trustStorePassword=
efm.server.ssl.clientAuth=WANT
```

2. Change `efm.security.user.certificate.enabled=false` to `efm.security.user.certificate.enabled=true`.
3. Change `efm.server.ssl.clientAuth=WANT` to `efm.server.ssl.clientAuth=NEED`.

### Configure MiNiFi Agent TLS

You can configure MiNiFi Agent TLS by updating the configuration files.

### About this task

If you are configuring a MiNiFi Java agent the configuration file is `conf/bootstrap.conf`. If you are configuring a MiNiFi C++ agent, the file is `conf/minifi.properties`.

### Procedure

1. Open your MiNiFi Agent configuration file in a text editor.
2. Edit the security properties. For example:

```
Security Properties:
  keystore: /tmp/ssl/localhost-ks.jks
  keystore type: JKS
  keystore password: localtest
  key password: localtest
  truststore: /tmp/ssl/localhost-ts.jks
  truststore type: JKS
  truststore password: localtest
  ssl protocol: TLS
Sensitive Props:
  key:
    algorithm: PBEWITHMD5AND256BITAES-CBC-OPENSSL
    provider: BC
```