

Linux Installation Guide

Kinetic - Edge & Fog Processing Module (EFM) 1.2.0

Revised: November 30, 2017

Contents

Introduction	3
Features and Functions.....	3
The Edge and Fog Processing Module Components	4
Hardware Requirements.....	5
EFM Components Protocols and Ports	5
Licensing installation and requests.....	6
Required Third Party Libraries for EFM Installation on Linux.....	6
Secure Mode Operation.....	6
Basic component installation scenarios	7
General Concepts:.....	8
Defining an EFM Administrator User per node	8
Defining a non-root Linux account for installation and operation.....	8
Setting Environment Variables to override default values prior to installation	9
Installation of the EFM Components	10
Install help or usage:.....	11
Installing the Smart Licensing Tool:	13
Installing the EFM Message Broker:	13
Installing the System Administrator:	14
Installing the System Monitor:	14
Installing the ParStream Historian Database:.....	15

Installing all of the EFM Linux components:.....	17
Installing the C-Broker:	19
Installing DGLux5:	19
Upgrading to EFM version 1.2.0 from an existing installation	21
Configuring the SSL certificate to allow for secure inbound connections.....	23
Starting and stopping the EFM Message Broker	23
Starting and stopping the EFM ParStream Historian Database.....	24
Starting and Stopping EFM System Administrator, EFM System Monitor and EFM DataFlow Editor.....	25
Connecting to the EFM System Components	25
Troubleshooting.....	25
Linux Firewall issues.....	25
Proxy Server challenges and the EFM Message Broker.....	26
EFM Message Broker Server Configuration via the server.json file	27
Obtaining documentation and submitting a service request.....	31

Introduction

The Cisco Edge & Fog Processing Module (EFM) allows you to create a reliable data communications messaging system on top of your data networking infrastructure that provides data delivery and allows you to rapidly deploy applications where needed that can be at the edge, fog or in the datacenter. The Edge and Fog Processing Module is an open platform that allows for the addition of micro services or applications by anyone, allowing for unlimited capability and growth by adding software components that optimize the results of the application, system or outcome.

The EFM addresses the complexity of building an enterprise ready scalable data messaging system for one or many applications to reside upon. The EFM comes with a series of tools to manage the system, the EFM system administrator and the EFM system monitor.

Features and Functions

The system's key capabilities include:

- A framework for edge and fog processing. High performance.
- Reusable micro services for collecting data from, and providing control over, devices and machines, as well as processing the data prior to delivery to its destination.
- Different options for reliable transport of data through the system, encompassing both batch and real-time streaming options.
- Flexible mechanisms for integration with IT systems, reporting, and analytics.
- An architectural framework to extend fog processing to multiple tiers: east west (fog to fog) and north south (hierarchical processing leveraging network topology).
- Easy-to-use GUI tools to simplify development, deployment, and operation for all aspects of the system.
- A pervasive control paradigm and flow of information back to micro services, devices and machines for management, control, optimization and specific actions.
- A completely open and polyglot system, where third parties can provide devices, processing storage, software modules, analytics, applications, or any combination thereof.

This is the technology that makes IoT approachable, and leads to much faster industry adoption of the vision of IoT.

The Edge and Fog Processing Module Components

EFM Message Broker	<p>Provides reliable and flexible data delivery between any devices and micro services. The sources can be devices like sensors or other micro services and consumers can be micro services or user applications.</p> <p>The EFM Message Broker is a small footprint component working with other brokers to form a message bus.</p>
EFM Data Flow Editor	Defines message paths between devices and micro services.
EFM Data Flow Engine	Executes message paths between devices and micro services. It is recommended to be installed adjacent to the EFM Message Broker in order to perform data transformation and input sources that not in the canonical data format of the system.
EFM System Administrator	Configures and manages the message broker and micro services.
EFM System Monitor	A standalone tool for operators to obtain real-time functional status of a deployed solution.
Cisco ParStream (Historian Database)	Purpose-built database for scale to handle the massive volumes and high velocity of IoT data as well as analytics at the Edge.
EFM Tools Runtime Engine	A standalone runtime tool for visualizing dashboards and driving EFM System Administrator, EFM Data Flow Engine and EFM System Monitor.
Links	<p>DQL Link – DSA Query Language</p> <p>System Link – System Information</p> <p>ParStream Link – ParStream Historian Database</p>
Smart License Agent Tool for Nodes	The Smart License Agent client that allows system users to manage license registration for Node Product IDs
Smart License Agent Tool for Devices	The Smart License Agent client that allows system users to manage license registration for Device Product IDs

Hardware Requirements

EFM Message Broker EFM Data Flow Engine DQL Link System Link ParStream Link	Red Hat Linux 7.2, CentOS 7 or Ubuntu 16.04, 1GB RAM, Windows 2016 Server, 10 GB HD* - Recommended on the same system/VM
EFM Data Flow Editor	Automatically installs with EFM Message Broker and EFM Tools Runtime Engine. Access via a web browser
EFM System Administrator	Project installs on the same system as the EFM Message Broker and EFM Tools Runtime Engine. Accessed via a web browser
EFM System Monitor	Project installs on the same system as the EFM Message Broker and EFM Tools Runtime Engine. Accessed via a web browser
Cisco ParStream (Historian Database)	Red Hat Linux 7.2, CentOS 7 or Ubuntu 16.04, 6 CPU cores with 2GB RAM per core, 500 GB HD
EFM Tools Runtime Engine	Installs with EFM Message Broker
Smart License Agent Tool for Nodes and Smart License Agent Tool for Devices	Redhat Linux 7.2, CentOS 7 or Ubuntu 16.04, with 1GB RAM, 10 GB HD.

EFM Components Protocols and Ports

The protocols and ports used by the EFM Broker and the EFM Historian Database. The port values are configurable during and after installation.

TCP Port No.	Description
8080	Default http value for inbound connections to the Message Broker and Web access
8443	Default https value for inbound connections to the Message Broker and Web access
EFM Administrator Defined	Connection to ParStream Historian Server (when installed) from a local or remote ParStream DLink. This includes the registrationPort and server(s) port in the server INI file.

Licensing installation and requests

This product uses the Smart License Agent Tool (for Nodes and Devices) to manage the corresponding smart licenses. After installation, refer to the Kinetic - Edge and Fog Processing Module 1.0 Smart License Agent User Guide.

Required Third Party Libraries for EFM Installation on Linux

The following third party libraries are needed for some of the EFM components. The exact versions listed were tested to work with the EFM.

For the ParStream Historian Database:

CentOS 7, RHEL 7, Debian

Package	Version
libicu	libicu-50.1.2-15.el7.x86_64
gpgme	gpgme-devel-1.3.2-5.el7.x86_64, gpgme-1.3.2-5.el7.x86_64
openssl	openssl-libs-1.0.1e-60.el7_3.1.x86_64, openssl-devel-1.0.1e-60.el7_3.1.x86_64, openssl-1.0.1e-60.el7_3.1.x86_64

Ubuntu 16.04

Package	Version
libssl1.0.0	1.0.2g-1ubuntu4.6
libicu55	55.1-7ubuntu0.1

For the ParStream DSA Link and the Cisco Smart License Agent: Java 8 JRE or JDK

Secure Mode Operation

The can operate in Secure Mode to enhance the security features available for the EFM message broker and web server. Secure mode provides these following enhancements:

- HTTPS Strict Transport Security (HSTS), automatically redirecting inbound http connections to https for message broker and web traffic
- System dslink can not execute "system command"
- login page won't allow browser to remember password
- prevents the pages from being embedded in iframes

- prevents the command action that allows shell execution by the System Link Secure Mode is configured by indicating Y(es) during the installation of the message broker or placing the hidden file “.secureMode” in the EFM_server directory.

Using Secure Mode HSTS only affects inbound connections, outbound http and https connections are still supported.

Basic component installation scenarios

The EFM has many components, allowing for a diverse manner of architecting a solution. While there is no single deployment architecture, we will explain the basic deployment scenarios.

The first and simplest installation is a complete install on a single host. This scenarios allows for development and testing, but is not typical for a production system.

The EFM architecture can be divided into five main building blocks and are typically on different hosts. We can separate them as follows:

- EFM Smart License Agent(s) - this is the only node in the system that is required to connect to Cisco.com, either directly or via the Smart License Satellite. The Smart License Agent allows for license activation, revocation and renewals. Without license activation or periodic communication over the Internet to Cisco.com, the EFM is out of license compliance.
- EFM Message Broker with core DLinks (DQL, System, ParStream and Dataflow engine). The message broker is deployed on all nodes and is responsible for communications between all components across the system.
- A System Administrator node, typically one per system. This is the administrative console that allows for configuration and operation of the EFM System components. A message broker is installed on this node in addition to the EFM System Administrator project.
- A System Monitor node, also one per system. The EFM System Monitor allows operators to view the connectivity and operations of the message brokers and DLinks deployed throughout the system. The System Monitor is used as an operations console.¹

¹ The operation of the System Monitor acts as a subscription to the System link in every broker for telemetry. If certain parts of the EFM system are bandwidth restricted, the use monitor rules should be reduced.

- A Historian node, is deployed through a system to persist telemetry into a database. This is an add-on to a message broker. The ParStream DsLink is used to communicate between the message broker and the ParStream historian database.

The System Administrator and System Monitor use the message brokers for communications to all the nodes and system dslinks. Message broker to broker communications needs to be setup first before other tasks can be performed.

General Concepts:

Defining an EFM Administrator User per node

It is important to note that we do not define default username and passwords for EFM. The first user that is defined at install becomes the administrator of that node. After the install, using the System Administrator, the additional users that may be added. At least one user requires administrative privileges for that node.

Defining a non-root Linux account for installation and operation

As a Linux security best practice, it is recommend creating a non-root account for installing and operating the EFM. We suggest creating account “EFM” for this purpose, and the following examples throughout the documentation will reference this name. The *adduser* command creates a new user “EFM” and a new group “EFM”.

To add a user, follow the instructions below for each supported operating system:

Redhat 7.2/Centos 7.2

Adding the “EFM” user account with a password and creating a home directory

```
$ sudo adduser EFM -m
$ sudo passwd EFM
```

Ubuntu 16.04

```
$ sudo adduser EFM
Enter new UNIX password: <password>
Retype new UNIX password: <password>
Enter the new value, or press ENTER for the default
  Full Name []: EFM
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
```


For all operating systems:

As the host administrator (sudo), create the EFM root installation directory. Unless otherwise defined, this will be /opt/cisco/kinetic. Also change owner and group to “EFM”.

```
$ sudo mkdir -p /opt/cisco/kinetic
$ sudo chown EFM /opt/cisco/kinetic
$ sudo chgrp EFM /opt/cisco/kinetic
```

Logging in as user “EFM” from the current user

```
$ su EFM
```

Place the EFM software image in the EFM home directory.

The software should be downloaded from CCO at www.cisco.com under Support and Downloads.

Unzip the image

```
$ unzip EFM-1-2-0.zip
```

Change into unzipped folder

```
$ cd EFM-1-2-0
```

Setting Environment Variables to override default values prior to installation

Setting environment variables allows to override default value used during the installation script. A system installation will function properly by installing with default values, but there may be a need to change the root installation directory or broker socket ports because of deployment requirements.

In addition, by being able to define values, the installation can be scripted for multiple installations on many nodes.

For example, a sample script for installation:

```
$ export EFM_ROOT=/opt/cisco/kinetic
$ ./efm-linux broker
```

Environment Variable	Description	Default
EFM_ROOT	should be set to the <i>absolute</i> path of the install root (i.e. folders EFM_server, parstream, and dart-sdk will be created below that path)	/opt/cisco/kinetic
EFM_GUI_LOGIN	set to the <i>name</i> of the EFM GUI Admin User	EFMAdmin
EFM_GUI_PHRASE	set to the <i>pass phrase</i> of the EFM GUI Admin User	Not set
EFM_BROKER_PRIV_KEY_PEM	overwrites the default value key.pem for certKeyName in server.json (i.e. with letsencrypt would suggest: /etc/letsencrypt/live/fully.qualified.domain.name/privkey.pem)	
EFM_BROKER_FULL_CHAIN_PEM	overwrites the default value server.pem for certName in server.json (i.e. with letsencrypt would suggest: /etc/letsencrypt/live/fully.qualified.domain.name/fullchain.pem)	
EFM_BROKER_SECURE_PORT	overwrite the default value for httpsPort in server.json	8443
EFM_BROKER_CLEARTEXT_PORT	overwrite the default value for port in server.json (disable this insecure access port by setting value 0 in production systems!)	8080
EFM_BROKER_IS_ALWAYS_OFFLINE	will overwrite the default value to set isAlwaysOffline true in server.json if set (isAlwaysOffline indicates that a server is expected to never have a full internet connection)	false
EFM_BROKER_WORKERS	may be set to positive integers in [1, 128] and should match the number of logical cpu cores as maximum and only if machine is dedicated to broker use and does not e.g. run local links or ParStream DB (if set and valid will overwrite the CPU core count derived one as value of the workers key in servers.json)	Maximum of: 1 and CPU Cores / 3
EFM_INSTALL_LOGS	set to the <i>absolute</i> path of a folder for install logs (default: case logs subfolder of EFM-1.0 e.g. \$HOME/EFM-1-2-0/logs if EFM-1-2-0.zip unpacked in \$HOME folder) (e.g. needed in case the unpacked install components are stored on a read-only medium)	
EFM_DEBUG	the installer is run with more explicit warn and fail early flags	

Installation of the EFM Components

The EFM installer `efm-linux` is a tool that is designed to allow for interactive or non-interactive installation of the EFM components. The defaults allow for non-root users to operate the system.

All the examples below rely on default values and invocations refer to a fictitious \$HOME being /home/userid and further assume, that the package EFM-1-2-0.zip has been unpacked inside that folder.

Install help or usage:

Executing `./efm-linux` displays a summary help screen.

```
Edge & Fog Processing Module - Installer and checksum tool v1.2.0

Synopsis: efm-linux [env|help|install|report|upgrade|verify|version]
[admin|broker|cbroker|dart|dglux|license|monitor|parstream]

Note: Call with help for extended version including sample usage(s) or with env for environment info.

Copyright:
  (c) 2017 Cisco and/or its affiliates. All rights reserved.
```

Executing `./efm-linux help` displays an extended help screen.

```
Edge & Fog Processing Module - Installer and checksum tool v1.2.0

Synopsis: efm-linux [env|help|install|report|upgrade|verify|version]
[admin|broker|cbroker|dart|dglux|license|monitor|parstream]

Note: Call with help for extended version including sample usage(s) or with env for environment info.

Copyright:
  (c) 2017 Cisco and/or its affiliates. All rights reserved.
```

```
[EFM@efm-rh73-148 EFM-1-2-0]$ ./efm-linux help
INFO: Interactive mode enabled (reason default)
Edge & Fog Processing Module - Installer and checksum tool v1.2.0

Synopsis: efm-linux [env|help|install|report|upgrade|verify|version]
[admin|broker|cbroker|dart|dglux|license|monitor|parstream]
```

Sample usage(s):

```
efm-linux install          # -> install all components (dart, broker, admin, monitor, license,
and parstream)
efm-linux upgrade        # -> upgrades all platform components (broker, admin, monitor, and
parstream)

efm-linux broker         # -> installs broker, data flow editor and dsa run-time
efm-linux dart           # -> installs dart run-time
efm-linux admin          # -> installs admin

efm-linux install license # -> installs smart license agent
efm-linux parstream      # -> installs parstream historian database

efm-linux verify         # -> report current settings and SHA512 checksums of components

efm-linux verify dart    # -> calculate SHA512 checksum of dart run-time

efm-linux version        # -> report version banner of this tool
```



```
efm-linux [help]          # -> this help screen / usage info.
```

Notes:

Installing admin and monitor will also install Dart VM and Broker run-time.

Environment Variables:

EFM_ROOT should be set to the *absolute* path of the install root (default: /opt/cisco/kinetic)
(i.e. folders efm_server, parstream, and dart-sdk will be created below that path)
Current value is: '/opt/cisco/kinetic' (without the enclosing single quotes)

EFM_GUI_LOGIN may be set to the *name* of the EFM GUI Admin User (default: efmAdmin)
Current value is: <UNSET>

EFM_GUI_PHRASE may be set to the *pass phrase* of the EFM GUI Admin User (default: not set)
(This variable will be ignored for now in interactive install sessions)
Current value is: <UNSET>

EFM_BROKER_PRIV_KEY_PEM overwrites the default value key.pem for certKeyName in server.json
(i.e. with letsencrypt would suggest:
/etc/letsencrypt/live/fully.qualified.domain.name/privkey.pem)
Current value is: <UNSET>

EFM_BROKER_FULL_CHAIN_PEM overwrites the default value cert.pem for certName in server.json
(i.e. with letsencrypt would suggest:
/etc/letsencrypt/live/fully.qualified.domain.name/fullchain.pem)
Current value is: <UNSET>

EFM_BROKER_SECURE_PORT will overwrite the default value of 8443 for httpsPort in server.json
Current value is: <UNSET>

EFM_BROKER_CLEARTEXT_PORT will overwrite the default value of 8080 for port in server.json
(disable this insecure access port by setting value 0 in production systems!)
Current value is: <UNSET>

EFM_BROKER_IS_ALWAYS_OFFLINE will overwrite the default value to set isAlwaysOffline true in
server.json if set
(isAlwaysOffline indicates that a server is expected to never have a full internet connection)
Current value is: <UNSET>

EFM_BROKER_WORKERS may be set to positive integers in [1, 128] and should match the number of
logical cpu cores
as maximum and only if machine is dedicated to broker use and does not e.g. run local links or
ParStream DB
(if set and valid will overwrite the CPU core count derived one as value of the workers key in
servers.json)
Current value is: <UNSET>

EFM_INSTALL_LOGS may be set to the *absolute* path of a folder for install logs (default:
/home/efm/EFM-1-2-0)
(e.g. needed in case the unpacked install components are stored on a read-only medium)
Current value is: <UNSET>

EFM_UNATTENDED may be set to fast enable unattended operation solely controlled by efm.json file
content.

Setting this to anything else than an empty value is equivalent to setting INTERACTIVE to false
in eff.json

EFM_DEBUG: For execution in debug mode, please set EFM_DEBUG environment variable to nonempty value

Copyright:

(c) 2017 Cisco and/or its affiliates. All rights reserved.

Executing `./efm-linux version` displays the version.

```
Edge & Fog Processing Module - Installer and checksum tool v1.2.0
```

Executing `./efm-linux env` reports the EFMective environment variable values and their source.

```
INFO: Interactive mode enabled (reason default)
INFO: Effective environment variable status detected:
REPORT: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - -
REPORT: EFM_ROOT: '/opt/cisco/kinetic' (without the enclosing single quotes)
REPORT: EFM_GUI_LOGIN(from config): efmAdmin
REPORT: EFM_GUI_PHRASE(from config): <empty>; Format: PBKDF2
REPORT: EFM_BROKER_PRIV_KEY_PEM(from config): key.pem
REPORT: EFM_BROKER_FULL_CHAIN_PEM(from config): cert.pem
REPORT: EFM_BROKER_SECURE_PORT(from config): 8443
REPORT: EFM_BROKER_CLEARTEXT_PORT(from config): 8080
REPORT: EFM_BROKER_IS_ALWAYS_OFFLINE(from config): BOOLEAN_FALSE
REPORT: EFM_BROKER_WORKERS(from config): 1
REPORT: EFM_INSTALL_LOGS(from config):
REPORT: EFM_UNATTENDED(from env): BOOLEAN_FALSE
REPORT: EFM_DEBUG: <UNSET>
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 -
```

Installing the Smart Licensing Tool:

In this example no global variables are set, all defaults are used.

Typing `./efm-linux license`

```
INFO: Interactive mode enabled (reason default)
INFO: The component EFM Smart License Agent has been installed
```

Installing the EFM Message Broker:

In this example no global variables are set, all defaults are used.

This installs the EFM Message Broker and the DQL, System, ParStream and Dataflow engine DSLinks.

Executing `./efm-linux broker`

```
INFO: Interactive mode enabled (reason default)
INFO: The component EFM Smart License Agent has been installed
[EFM@efm-rh73-148 EFM-1-2-0]$ ./efm-linux broker
INFO: Interactive mode enabled (reason default)
INFO: Sub-task install Dart VM (required by Dart broker) ...
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 02:01:35 2017) on "linux_x64"
INFO: Sub-task install Dart VM (required by Dart broker) succeeded
INFO: EFM Message Broker server configuration template copied successfully from /home/efm/EFM-1-2-
0/Components/server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]: y
```



```
INFO: EFM Message Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFM_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFM_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFM_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFM_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFM_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFM_BROKER_WORKERS to 1
INFO: EFM Message Broker server configuration updated with path
INFO: Installation created secure by default setup, good.
INTERACTION: Keep blocking unsecured creation of upstream connections over HTTP using the EFM Message
Broker installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: EFM Message Broker user configuration template copied ...
INTERACTION: Perform EFM Message Broker custom user configuration now? [Y/n]: y
INFO: EFM Message Broker custom user configuration requested, continuing ...
INTERACTION: EFM GUI Admin login is (efmAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the EFM Message Broker user configuration and grant admin
rights ...
INFO: Creation of EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully
created."]
INFO: Grant admin rights for EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was
successfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - -
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044xe6cq4K994BkNLL0axkF6Va3YSKnlh6Jcp7l5c0iaAE=7IwNWFzvG6dV0W2QR84TlsG2YcoyEXKnTvqEx5XNdVI=
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the EFM Message Broker can execute and expose
its version? [Y/n]:
WARNING: Execution of /opt/cisco/kinetic/dart-sdk/bin/dart
/opt/cisco/kinetic/efm_server/bin/daemon.dart verify failed with exit status 1
INFO: The component EFM Message Broker has been installed
```

Installing the System Administrator:

In this example no global variables are set, all defaults are used.

The System Administrator is an option component that can be added to the broker for managing the system. It requires that the broker previously has been installed and configured to properly function and thus will trigger install of broker otherwise.

Executing ./efm-linux admin

```
INFO: Interactive mode enabled (reason default)
INFO: The component EFM System Administrator has been installed
```

Installing the System Monitor:

In this example no global variables are set, all defaults are used.

Typing ./efm-linux monitor

Since the System Monitor is envisioned to operate on an operations console, the installation takes the necessary steps of installing the message broker and system monitor project as a bundle. So in contrast to above admin install, the below sample starts with an empty target folder.

```
INFO: Interactive mode enabled (reason default)
INFO: Sub-task install EFM Message Broker (required by Monitor) ...
INFO: Sub-task install Dart VM (required by Dart broker) ...
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 02:01:35 2017) on "linux_x64"
INFO: Sub-task install Dart VM (required by Dart broker) succeeded
INFO: EFM Message Broker server configuration template copied successfully from /home/efm/EFM-1-2-
0/Components/server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]: y
INFO: EFM Message Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFM_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFM_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFM_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFM_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFM_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFM_BROKER_WORKERS to 1
INFO: EFM Message Broker server configuration updated with path
INFO: Installation created secure by default setup, good.
INTERACTION: Keep blocking unsecured creation of upstream connections over HTTP using the EFM Message
Broker installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: EFM Message Broker user configuration template copied ...
INTERACTION: Perform EFM Message Broker custom user configuration now? [Y/n]: y
INFO: EFM Message Broker custom user configuration requested, continuing ...
INTERACTION: EFM GUI Admin login is (efmAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the EFM Message Broker user configuration and grant admin
rights ...
INFO: Creation of EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully
created."]
INFO: Grant admin rights for EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was
successfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - -
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044JKSKyWMOXrwLl7FTY8LCTk8fIROVGZDwzaCgLhmbuPE=1sXnzzaLRksZeJgENV6TO4t5olVzZe7kUDbzgI6+Wcs=
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the EFM Message Broker can execute and expose
its version? [Y/n]:
WARNING: Execution of /opt/cisco/kinetic/dart-sdk/bin/dart
/opt/cisco/kinetic/efm_server/bin/daemon.dart verify failed with exit status 1
INFO: Sub-task install EFM Message Broker (required by Monitor) succeeded
INFO: The component EFM System Monitor has been installed
```

Installing the ParStream Historian Database:

Important note on ParStream Historian installation:

- Operational data, e.g. partitions, configuration files or journals should never be stored in the ParStream installation directory.

In this example no global variables are set, all defaults are used.

This installs the ParStream Historian Database.

Executing `./efm-linux parstream`

This will install the ParStream Historian Database to the folder specified in the environment variable `EFM_ROOT`². In the following example `EFM_ROOT` was set to `/home/efm/opt`³.

```
[EFM@efm-rh72-149 EFM-1-2-0]$ rm -rf /opt/cisco/kinetic/*./efm-linux parstream^C
[EFM@efm-rh72-149 EFM-1-2-0]$ ./efm-linux parstream
INFO: EFM_ROOT = /opt/cisco/kinetic taken neither from file, nor from environment variable, but instead
from platform default!
INFO: Interactive mode enabled (reason default)
INFO: Check of platform support - passed. (ok)
WARNING: WARNING: Only 1 cpu cores detected for this machine
WARNING: WARNING: Detected soft limit of open files 1024 smaller than suggested minimal 131072 for EFM
ParStream Historian Database
INTERACTION: Installed EFM ParStream Historian Database. Check that the parstream-server can execute
and expose its version? [Y/n]:
INFO: Check that the parstream-server can execute and expose its version requested
INFO: For future execution of the EFM ParStream Historian Database possible environment settings are:
INFO: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - -
- - - -
INFO: LD_LIBRARY_PATH=/opt/cisco/kinetic/parstream/lib
INFO: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- >8 -
INFO: For future execution of the EFM ParStream Historian Database server command is
/opt/cisco/kinetic/parstream/bin/parstream-server
INFO: Executing check that the parstream-server can execute and expose its version
DEBUG: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - -
- - - -
DEBUG: [2017-11-29T02:51:38]:unknown-000000:PROT-77001: ***** Starting Cisco ParStream Server 5.1.0
20171106T163434Z Release
DEBUG: [2017-11-29T02:51:38]:unknown-000000:PROT-77083: **** cmdlineargs: --version
DEBUG: [2017-11-29T02:51:38]:unknown-000000:PROT-77086: **** PID: 12670 (host: efm-rh72-149)
DEBUG: /opt/cisco/kinetic/parstream/bin/parstream-server version: 5.1.0 built: 20171106T163434Z
(34acfa088f7671db01b25b9276c1de0c033c076f)
DEBUG: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- >8 -
INFO: The component EFM ParStream Historian Database has been installed
```

It is recommended to set the environment `PARSTREAM_HOME`, `LD_LIBRARY_PATH` and `PATH` as suggested by the install script:

² Please refer to section “*Setting Environment Variables to override default values prior to installation*” for more information about environment variables.

³ We use the user `EFM` in this example.


```
export PARSTREAM_HOME=/home/efm/opt/parstream
export LD_LIBRARY_PATH=$PARSTREAM_HOME/lib:$LD_LIBRARY_PATH
export PATH=$PATH:$PARSTREAM_HOME/bin
```

After installing ParStream, the installer will ask you to verify the installation. If everything was installed successfully, it will print the ParStream version and exit.

In the ParStream installation folder (In this example found in `/home/efm/opt/parstream`) an *examples* directory exists. These examples provide you with basic ParStream configuration files. In each example's directory a *conf* directory exists. In this directory you'll find the `parstream.ini` file. This file provides the configuration parameters the ParStream Historian Database will read on startup.

In order to execute a specific example navigate to the example's directory. Here you'll find the shell script `run_cluster.sh`. When invoking this shell script the respective example will be executed.

For more details regarding the configuration of ParStream and further details, please refer to the ParStream documentation accompanying this document. The additional documentation files accompany this document .

Installing all of the EFM Linux components:

In this example no global variables are set, all defaults are used.

For a complete installation of all the components on a host system, you can use this command option. We do not envision production systems will be designed with all the components on a single system, there are circumstances for development testing and learning environments that desire a quicker installation with one command.

This command installs the EFM Smart Licensing Tool, EFM Message Broker, EFM System Administrator, EFM System Monitor, EFM DataFlow Editor and the EFM ParStream Historian Database.

Executing `./efm-linux install`

```
INFO: EFM_ROOT = /opt/cisco/kinetic taken neither from file, nor from environment variable, but instead
from platform default!
INFO: Interactive mode enabled (reason default)
INTERACTION: Installed platform dart run-time. Check that the Dart VM can execute and expose its
version? [Y/n]:
INFO: Dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 02:01:35 2017) on "linux_x64"
```



```
INFO: The component Dart VM has been installed
INFO: EFM Message Broker server configuration template copied successfully from /home/efm/EFM-1-2-0/Components/server.json
INTERACTION: Reconfigure the EFM Message Broker server configuration? [y/n]: y
INFO: EFM Message Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFM_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFM_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFM_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFM_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFM_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFM_BROKER_WORKERS to 1
INFO: EFM Message Broker server configuration updated with path
INFO: Installation created secure by default setup, good.
INTERACTION: Keep blocking unsecured creation of upstream connections over HTTP using the EFM Message Broker installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: EFM Message Broker user configuration template copied ...
INTERACTION: Perform EFM Message Broker custom user configuration now? [Y/n]:
INFO: EFM Message Broker custom user configuration requested, continuing ...
INTERACTION: EFM GUI Admin login is (efmAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the EFM Message Broker user configuration and grant admin rights ...
INFO: Creation of EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully created."]
INFO: Grant admin rights for EFM GUI Admin User efmAdmin succeeded with ["User 'efmAdmin' was successfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - -
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044e3FmKKhOoJcNmqcG/TKqkR5zxPukivjf7NcKuhyJoFoQ=WMR7xs3riFCOcLlZ9nsdabRvbbhlcp+K/xlu00ga7Z4=
REPORT: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the EFM Message Broker can execute and expose its version? [Y/n]:
WARNING: Execution of /opt/cisco/kinetic/dart-sdk/bin/dart
/opt/cisco/kinetic/efm_server/bin/daemon.dart verify failed with exit status 1
INFO: The component EFM Message Broker has been installed
INFO: The component EFM System Administrator has been installed
INFO: The component EFM System Monitor has been installed
INFO: The component EFM Smart License Agent has been installed
INFO: Check of platform support - passed. (ok)
WARNING: WARNING: Only 1 cpu cores detected for this machine
WARNING: WARNING: Detected soft limit of open files 1024 smaller than suggested minimal 131072 for EFM ParStream Historian Database
INTERACTION: Installed EFM ParStream Historian Database. Check that the parstream-server can execute and expose its version? [Y/n]:
INFO: Check that the parstream-server can execute and expose its version requested
INFO: For future execution of the EFM ParStream Historian Database possible environment settings are:
INFO: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< -
- - - -
INFO: LD_LIBRARY_PATH=/opt/cisco/kinetic/parstream/lib
INFO: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 -
INFO: For future execution of the EFM ParStream Historian Database server command is
/opt/cisco/kinetic/parstream/bin/parstream-server
INFO: Executing check that the parstream-server can execute and expose its version
DEBUG: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< -
- - - -
DEBUG: [2017-11-29T02:55:27]:unknown-000000:PROT-77001: ***** Starting Cisco ParStream Server 5.1.0
20171106T163434Z Release
DEBUG: [2017-11-29T02:55:27]:unknown-000000:PROT-77083: **** cmdlineargs: --version
```


This installs the DGLux5 server only and the dataflow dslink. See the document *EFM and DGLux5 Installation Guide* for details on installing and running the EFM message broker and the DGLux5 on the same host.

If the EFM broker has already been installed with the DART or the DART VM has not been installed as a standalone, then it must be installed before the DGLux5 can be run. The DART VM is used both the EFM message broker and DGLux5.

Typing `.\efm-windows dart`

```
INFO: EFM_ROOT = /opt/cisco/kinetic taken neither from file, nor from environment variable, but instead
from platform default!
INFO: Interactive mode enabled (reason default)
INTERACTION: Installed platform dart run-time. Check that the VM can execute and expose its version?
[Y/n]:
INFO: dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 02:01:35 2017) on "linux_x64"
INFO: The component Dart VM has been installed
```

Upgrading to EFM version 1.2.0 from an existing installation

If an existing version 1.0.0, 1.0.1 or 1.1.0 is installed and running on a Linux server it is possible to upgrade to the version 1.2.0 without a fresh install. It is highly recommended performing a backup of the host prior to proceeding with the upgrade.

Assuming that the EFF version 1.0.x or 1.1.0 is already installed and running. Here are the prerequisites to upgrading:

- Download the EFM 1.2.0 package
- Stop the running message broker (see below)
- Stop the running ParStream Historian Database (see below)

Please take note of the following name changes that might impact the upgrade process:

- If no explicit EFF_ROOT was set for an installation with EFF 1.0.x, or 1.1.0, all components were installed below `/opt/cisco/iotdc`
- The environment variable name for the install root folder has been changed from EFF_ROOT to EFM_ROOT
- If you now upgrade to EFM 1.2, the new default path `/opt/cisco/kinetic` is empty. So, you have two alternatives:
 - either set EFM_ROOT to `/opt/cisco/iotdc` to upgrade the setup in the prior location
 - or move `/opt/cisco/iotdc` to `/opt/cisco/kinetic` (and adapt config entries in `eff_server/server.json` where applicable)
- be aware, that the `efm-tool` will rename `eff_` subfolders during the upgrade process. An existing `eff_server` will be renamed to `efm_server`, `eff_license` will become `efm_license`.

Important notes on ParStream Historian installation:

- Operational data, e.g. partitions, configuration files or journals should never be stored in the ParStream installation directory. When upgrading an existing EFM installation, we may decide to replace the ParStream installation folder by the new version. When upgrading from a previous version of EFF (1.0.0, 1.0.1 or 1.1.0) to EFM 1.2.0, the installer will create a backup folder containing all files and directories in the existing ParStream installation directory.
- After upgrading to a new version of ParStream, please make sure that `PARSTREAM_HOME` and `LD_LIBRARYPATH` are set appropriately. Refer to the ParStream manual, accompanying this document, for more information.
- [Specific usage] If you are using BLOB columns in your existing installation, please update all ParStream DSA Links before updating the ParStream database server.

Perform the following steps to upgrade:

Prepare upgrade like a fresh install and unpack the EFM-1-2-0.zip accordingly and change current working directory to EFM-1-2-0 folder all as described above in the install section.

```
$> unzip -q EFM-1-2-0.zip
$> cd EFM-1-2-0
```

Set environment variable to the same destination as the current version 1.0.x installation, for example:

```
export EFM_ROOT=/opt/cisco/kinetic
```

Stop the broker if running:

```
EFM_ROOT/dart-sdk/bin/dart EFM_ROOT/efm_server/bin/daemon.dart stop
```

Finally upgrade your EFM installation to 1.2.0 with the following command and follow the instructions

```
./efm-linux upgrade
```

After a successful upgrade, two steps are required:

1. Update the static URL path to the “.well-known” resource inside efm_server/server.json config file, so that it matches the upgraded location (just change eff_server to efm_server inside path) and then
2. Restart the message broker (see below).

Configuring the SSL certificate to allow for secure inbound connections

A necessary step for the message broker and web server to allow incoming secure connections is to properly install the SSL certificate files and define a non-null (or not empty value) value for the certificate password in the server.json file.

The EFM system ships with a self-signed SSL certificate and certificate private key. ⁴ It is optional to install user certificates. The placement of the certificate files are in the \$EFM_ROOT/efm_server/certs folder.

This is accomplished during the broker installation or editing the \$EFM_ROOT/efm_server/server.json file and adding a non-null value to the “certPassword” field. Without this value, the SSL functionality will not operate.

Starting and stopping the EFM Message Broker

This step starts the EFM Message Broker, EFM Data Flow Engine and Editor and the DSLinks that might were installed.

Note: Assuming the default installation path of /opt/cisco/kinetic/efm_server in the following examples.

Starting and Stopping EFM Message Broker:

To start the EFM Message Broker, EFM Data Flow Engine and Editor execute the following:

```
/opt/cisco/kinetic/dart-sdk/bin/dart /opt/cisco/kinetic/efm_server/bin/daemon.dart start
```

To stop the EFM Message Broker, EFM Data Flow Engine and Editor execute the following:

```
/opt/cisco/kinetic/dart-sdk/bin/dart /opt/cisco/kinetic/efm_server/bin/daemon.dart stop
```

If the message broker ports are in the range of 1-1024, the Linux operating system will require sudo or root priveledges to start the application.

```
$sudo /opt/cisco/kinetic/dart-sdk/bin/dart /opt/cisco/kinetic/efm_server/bin/daemon.dart start
```

```
$sudo /opt/cisco/kinetic/dart-sdk/bin/dart /opt/cisco/kinetic/efm_server/bin/daemon.dart stop
```

⁴ Note that this self-signed certificate will present a browser security exception. To eliminate the security warning, a valid certificate should be installed.

Starting and stopping the EFM ParStream Historian Database

To start the ParStream Historian Database without the `run_cluster.sh` script, navigate into one of the example's directories. If you want to follow this example please navigate to the `noauthentication` directory (Found here `/home/efm/opt/parstream/examples/noauthentication`, if installed as user `efm`).

In each example directory a `conf` folder is located. Within the `conf` folder the `parstream.ini` file exists. The configuration of the ParStream Historian Database cluster is also configured by the `parstream.ini` file. Each cluster node has its own section in the file. The `noauthentication` example configures a one node cluster.

For this purpose the section `server.first` is specified in the `parstream.ini` file. In order to start the cluster node execute the following command in the `noauthentication` directory⁵:

```
$PARSTREAM_HOME/bin/parstream-server first
```

This will start the ParStream Historian Database with the global parameters found in the `parstream.ini` file. The node specific parameters are read from the node's ini file section (In this case `server.first`).

After the cluster's initialization phase it will start to listen on port 9042 for incoming client connections.

In order to shutdown the ParStream Historian Database execute the following command⁶:

```
$PARSTREAM_HOME/bin/pnc -p 9042
```

This will establish a connection the ParStream Historian Database server listening on port 9042. Issue the following command when command prompt `Cisco ParStream=>` appears:

```
Cisco ParStream=> ALTER SYSTEM CLUSTER SHUTDOWN;
```

This will shut the cluster down. For more information about configuring, starting and stopping ParStream Historian Database clusters refer to the ParStream accompanying the package.

⁵ If the command fails, make sure that all environment variables are set as described.

⁶ Make sure that the ParStream Historian Database server is still running and listening on port 9042.

Starting and Stopping EFM System Administrator, EFM System Monitor and EFM DataFlow Editor

These components are available when the EFM Message Broker is running. No other steps are necessary to start these tools.

Connecting to the EFM System Components

All the EFM tools require the administrator user and login for access.

Application	Insecure Port (if supported) ⁷	Secure Port
EFM System Administrator	http://[Server IP Address]/efm-admin	https://[Server IP Address]/efm-admin
EFM System Monitor	http://[Server IP Address]/efm-monitor	https://[Server IP Address]/efm-monitor
EFM Data Flow Editor	http://[Server IP Address]/dataflow.html	https://[Server IP Address]/dataflow.html

Troubleshooting

Linux Firewall issues

Redhat and CentOS initially is configured by default with the firewall service turned on and blocks all incoming connections. It is necessary to read the operating system guide to turn off or allow only the known service ports for the EFM connections. But the command below can be used for troubleshooting or staging. The proper configuration needs to be defined by the host administrator.⁸

To turn off the firewall on Redhat/CentOS/ the following command can be executed:

```
$ sudo service firewalld stop
```

⁷ Note if the server configuration is using Safe Mode, incoming connections will automatically redirect to the secure port if configured. If no secure port is defined or certificates and passwords are not properly configured, the secure port will not accept connections and therefore in Safe Mode the access to the system is unavailable.

⁸ The firewall service will restart on next reboot unless the administrator modifies the default configuration

Proxy Server challenges and the EFM Message Broker

In some environments it might be necessary to define a proxy server to access the Internet due to security restrictions. The EFM message broker uses a localhost communication to connect to the DSLinks on the same host and usually any proxy server configuration inhibits some of this functionality from functioning properly.

We have observed in the System Administrator that some DSLinks connect to the message broker, while others do not if there is a proxy server configured.

In order to successfully connect to all the DSLinks it stopping the message broker be necessary, remove the proxy settings and start again the message broker. For example:

- Stop the message broker with `/opt/cisco/kinetic/efm_server/bin/daemon.sh stop`
- Remove the proxy server settings in the environment or system configuration
- Start the message broker with `/opt/cisco/kinetic/efm_server/bin/daemon.sh start`

EFM Message Broker Server Configuration via the server.json file

Example server.json configuration file located in the \$EFM_ROOT/efm_server folder.

```
{
  "allowAllLinks": true,
  "allowBrowserCaching": false,
  "allowPasswordChanges": true,
  "alternativeBrokerUrl": null,
  "authType": "file",
  "broadcast": false,
  "brokerName": "broker-",
  "certKeyName": "key.pem",
  "certName": "cert.pem",
  "certPassword": "",
  "corsProxyRules": "",
  "dartRuntimeManagerVmFlags": [],
  "debug": false,
  "defaultPermission": null,
  "disableFileSecurity": false,
  "disabledLinks": [],
  "distributionUrl": "NO",
  "downstreamName": "downstream",
  "enableCertificateGeneration": true,
  "enableGit": false,
  "enableIPv6": false,
  "enableSingleSignOnServer": false,
  "enableUptimeChecker": true,
  "formatDg5": false,
  "generatedCertificateSubject": "/C=US/ST=California/L=Oakland/O=DGLogik Inc./OU=Customers/CN=*",
  "guestLoginRedirectPath": "/assets/",
  "hooks": {},
  "host": "0.0.0.0",
  "httpPathClassification": {},
  "httpsPort": 8443,
  "isAlwaysOffline": false,
  "javaRuntimeManagerVmFlags": [],
  "keepCustomAssets": true,
  "linkConfig": {},
  "linkManagerEnvironment": {},
  "linkRepositoryUrl": "https://dsa.s3.amazonaws.com/links/links.json",
  "logRotationInterval": 0,
  "loggers": [],
  "loginRedirectPath": "/",
  "observe": false,
  "passwordHasherIterations": 1000,
  "passwordHasherKeyLength": 32,
  "port": 8080,
  "proxies": {},
  "quarantine": false,
  "runBrokerInMain": true,
  "runPortChecks": true,
  "serverLogLevel": "INFO",
  "serverVmFlags": [],
  "ssoProviderUrl": null,
  "static": {
    "/.well-known": "/opt/cisco/kinetic/efm_server/.well-known"
  },
  "storageDriver": "simple",
  "timeHttpRequests": false,
  "twoFactorAuth": "none",
  "updateInterval": 200,
  "upstream": {},
  "uptimeCheckUrl": null,
}
```

```

"useDartRuntimeManager": false,
"useJavaRuntimeManager": false,
"useRuntimeManager": false,
"userTimeout": 525600,
"workers": 1
}

```

In below table the default values are those, that are assumed by the server, if the key is not present in the server.json

Option	Description	Default Value	Comments
debug	Enable/Disable Debugging Mode	false	For production site, this should always be false, debug:true may result in memory leak and bugs. port
port	HTTP Port to listen on. If this is less than or equal to 0, then the server does not listen on any port for HTTP.	8080	At least one of port or httpsPort must have a valid port number assigned.
httpsPort	HTTPS port to listen on. If this is less than or equal to 0, and/or certName or certPassword is not provided, then the server does not listen on any port for HTTPS. Ensure that if you install a custom certificate, you fill in the certName, certKeyName and certPassword fields.	8443	At least one of port or httpsPort must have a valid port number assigned. certName
certName	SSL certificate file name. Leave blank to disable HTTPS		
certPassword	SSL certificate password. Set to null to disable HTTPS		
certKeyName	SSL private key file name. Leave blank to disable HTTPS		
disableFileSecurity	When this value is true, then any user can access any file. When this is false, file permissions are checked.	false	
broadcast	When this value is true, the server's broker is broadcast to the local network for discovery by other machines. When this value is false, the broadcast service is not enabled.	true	
workers	Number of Server Workers. For low end devices, this should stay at 1. For large machines, this can be set up to a	For single-core machines, this is 1, for other devices,	

	maximum of 128. It is recommended that you do not exceed the number of logical processors on your machine.	this is 2.	
static	Configures a static directory mapping. This is used to serve files and directories on the server. Example: { "/static": "/srv/http/static" }	{"/.well-known": "/path/to/dsa/dglux-server/.well-known"}	
defaultPermission	Default permission setting for the root node. When this value is null, permissions are disabled, and everything has the config permission.		
allowAllLinks	When the value is true, all incoming DSLink connections will be accepted to /downstream. When the value is false, an incoming DSLink without proper authentication will be rejected unless quarantine is enabled.	true	
quarantine	** This setting has no EFMect when allowAllLinks is true ** When the value is true, a new incoming DSLink without a token will be put in /sys/quarantine. A quarantined DSLink can only work as a responder. Use the /sys/quarantine/authorize to move a quarantined DSLink to /downstream.	false	
isAlwaysOffline	Indicates that a server is expected to never have a full internet connection.	false	
useDartRuntimeManager	When the value is true, the Dart Runtime Manager is used for Dart DSLinks. The Dart runtime manager reduces resource consumption by merging Dart DSLinks into a single process.	false	
useJavaRuntimeManager	When the value is true, the Java Runtime Manager is used for Java DSLinks. The Java runtime manager reduces resource consumption by merging Java DSLinks into a single process.	false	

guestLoginRedirectPath	Determines the URI that a user is redirected to when login is complete.	/	
authType	Determines the authentication provider to use.	file	
twoFactorAuth	<p>Determines the two factor authentication provider to use.</p> <p>Supported Two-Factor Authentication Providers</p> <ul style="list-style-type: none"> • none: Don't enable two factor authentication. • duo: Duo Two-Factor Authentication 	none	
enableIPv6	Toggles support for IPv6 connections	false	
keepCustomAssets	When the value is true, custom assets in www/assets are kept upon updating EFM Server.	false	
formatDg5	When this value is true, EFM client will save dg5 in a formatted and json with key sorted, makes it easy to track changes.	false	

Obtaining documentation and submitting a service request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.