



Linux Installation Guide

Kinetic - Edge & Fog Processing Module (EFM) 1.5.0

Revised: December 10, 2018

Table of Contents

Linux Installation Guide	1
Kinetic - Edge & Fog Processing Module (EFM) 1.5.0	1
Introduction	3
Features and functions	3
The Edge and Fog Processing Module components	4
Hardware requirements	5
EFM components' protocols and ports	5
Additional ports post installation	6
Licensing installation and requests	6
Required third party libraries for EFM installation on Linux	6
Secure Mode operation	8
Basic component installation scenarios	8
General concepts	10
Defining an EFM Administrator User per node	10
Defining a non-root Linux account for installation and operation.....	10
Setting environment variables to override default values prior to installation	12
Configuring SELinux on CentOS 7 to work with EFM Asset Manager	13
Installation of the EFM components	14
Installing help or usage	14
Installing the Smart Licensing Tool.....	17
Installing the EFM Message Broker	17
Installing the System Administrator	18



Installing the System Monitor	18
Installing the Asset Manager and its dependencies.....	20
Installing the ParStream Historian Database	22
Installing the C-Broker.....	23
Installing DGLux5.....	24
Installing all of the EFM Linux components.....	25
Upgrading to EFM version 1.5.0 from an existing installation	29
Configuring the SSL certificate to allow for secure inbound connections	31
Starting and stopping the EFM Message Broker.....	31
Starting and stopping the EFM ParStream Historian Database	31
Starting and stopping EFM System Administrator, EFM System Monitor, and EFM DataFlow Editor	32
Starting and stopping the EFM Asset Manager	32
Configuring the EFM Asset Manager	33
Connecting to the EFM System components.....	33
Troubleshooting	34
Linux Firewall issues	34
Proxy Server challenges and the EFM Message Broker	34
Configuring the EFM Message Broker server via the server.json file	35
Obtaining documentation and submitting a service request.....	48

Introduction

The Cisco Edge and Fog Processing Module (EFM) allows you to create a reliable data communications messaging system on top of your data networking infrastructure. This system provides data delivery and allows you to rapidly deploy applications, where needed, that can be at the edge or fog or in the data center. The EFM 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 upon which one or many applications can reside. 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 possible, and leads to faster industry adoption of the IoT vision.

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. We recommend installing it adjacent to the EFM Message Broker for performing data transformation and inputting sources that are 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	<ul style="list-style-type: none"> • DQL Link - DSA Query Language • System Link - System Information • ParStream Link - ParStream Historian Database • JDBC Link - Java Database Connectivity • Alarming Link • Asset Link - Asset Manager • Sensor Simulation - demonstration sensor simulator
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

EFM Asset Manager	<p>Detects and manages Assets throughout the EFM Messaging system</p> <p><i>Note: The Asset Manager is in Preview Release.</i></p>
-------------------	--

Hardware requirements

<p>EFM Message Broker EFM Data Flow Engine DQL Link System Link ParStream Link</p>	<p>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</p>
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 Asset Manager	<p>4GB RAM,</p> <p>10 GB HD - Recommended on the same system/VM</p>

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
443	Default https value for Asset Manager
EFM Administrator Defined	Connection to ParStream Historian Server (when installed) from a local or remote ParStream DsLink. This includes the registrationPort and server(s) port in the server INI file.

Additional ports post installation

The EFM DsLinks are microservices. They may expose additional protocol ports that are listening for incoming connections. It is necessary to verify these ports in the DsLink documentation and configure the host to allow the incoming connections as desired. For example, the MQTT DsLink provides an optional server that can listen on port TCP 8443 to MQTT clients. If the MQTT Server DsLink functionality is desired, the host must allow for the proper firewall access for this to receive the connections.

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 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
libcuc	libcuc-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
libc5	55.1-7ubuntu0.1

For the ParStream DSA Link and the Cisco Smart License Agent, use Java 8 JRE or JDK.

For improved memory handling, Cisco ParStream 5.2.0 now uses tcmalloc. If not already installed in your environment, you can install the depended library with the following commands:

- On Ubuntu16.04 and Debian8: apt-get install libtcmalloc-minimal4
- On CentOS7: yum install gperftools-libs

For the EFM Asset Manager, use nginx, Java 8 JRE or JDK.

Secure Mode operation

The EFM 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), which automatically redirects inbound http connections to https for message broker and web traffic
- System dslink cannot 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 no single deployment architecture exists, we will explain the basic deployment scenarios.

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

The EFM architecture can be divided into six 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 DSLinks (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. The EFM System Administrator 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 DSLinks deployed throughout the system. The System Monitor is used as an operations console.¹
- 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 Asset Manager is a microservices component of the of Cisco Kinetic EFM that detects and manages assetsthroughout the EFM message system. The Device DSLinks expose the assets that are required to have been specified using the Cisco EFM Device Object Model Standard structureto allow for auto-discovery. The resulting list of approved assets not only appears in the Asset Manager graphical interface, but also creates a node structure in the EFM data path that can be used as input for other applications.

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

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.

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 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 recommended to create 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 --force-badname
```

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 /opt/cisco/kinetic -p
$ 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-1-5-0.zip
```

Change into unzipped folder:

```
$ cd EFM-1-5-0
```

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 the root installation directory or broker socket ports may need to be changed 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/privatekey.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 *absolute* path of a folder for install logs (default: case logs subfolder of EFM-1.5 e.g. \$HOME/EFM-1-5-0/logs if EFM-1-5-0.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	

Configuring SELinux on CentOS 7 to work with EFM Asset Manager

In most cases, CentOS 7 comes with an enforced Security Enhanced Linux Kernel (SELinux) that blocks any inbound and outbound connections of the EFM Asset Manager on port 8443. Before installing the EFM components, it is recommended to either configure SELinux properly or to set the SELinux mode to “permissive.”

Retrieve current status of SELinux:

```
$ sudo sestatus
```

Set SELinux mode to permissive:

```
$ sudo setenforce permissive
```

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`. It is further assumed that the package `EFM-1-5-0.zip` has been unpacked inside that folder.

Installing help or usage

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

```
Edge & Fog Processing Module - Installer and checksum tool v1.5.0

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 v1.5.0

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-1-5-0]$ ./efm-linux help
INFO: Interactive mode enabled (reason default)
Edge & Fog Processing Module - Installer and checksum tool v1.5.0

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, parstream and asset manager)
efm-linux upgrade         # -> upgrades all platform components (broker, admin,
monitor, parstream and asset-manager)
```

```
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 asset-manager   # -> installs asset-manager

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-1-5-0)
(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 eff.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 v1.5.0

Executing `./efm-linux env` reports the EFM effective 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< - - - - 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: EFM_ASSET_MANAGER_GUI_LOGIN(from config):
REPORT: EFM_ASSET_MANAGER_GUI_PASSWORD(from config): ***** (replaced by 16 x
'*)
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-1-5-0]$ ./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-1-5-0/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< - - - -
REPORT: Username is: efmAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044xe6cq4K994BkNLL0axkF6Va3YSKnLh6Jcp7l5c0iaAE=7IwNWFZvG6dV0W2QR84TlsG2YcoyeXKnTvqEx5XNdV
I=
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 optional component that can be added to the Broker for managing the system. It requires that the Broker has been installed previously and configured to properly function and thus will otherwise trigger installation of the Broker.

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 process takes the necessary steps of installing the Message Broker and System Monitor project as a bundle. So, in contrast to above admin install, the following 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-1-5-0/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:
044JKSKyWMOXrwl17FTY8LCTk8fIRoVGZDwzaCgLhmbuPE=1sXnzzaLRksZeJgENV6TO4t5o1VzZe7kUdbzgI6+Wc
s=
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 Asset Manager and its dependencies

The Asset Manager is a component of the of Kinetic - EFM that detects and manages assets throughout the EFM message system. The Device DSLinks expose the assets that are required to have been specified using the Cisco EFM Device Object Model Standard structure to allow for auto-discovery.

Executing ./efm-linux asset-manager

```
efm-user@efmuser-PC:~/Downloads/EFM-1-5-0$ ./efm-linux asset-manager
INFO: Interactive mode enabled (reason default)
INFO: Installing Asset Manager ...
INFO: Check of platform support - passed. (ok)
INFO: Java found at /usr/bin/java
INFO: Java seems to be version 8 (ok)
INFO: Nginx found at /usr/sbin/nginx
INTERACTION: Asset Manager already exists. Overwrite all but application and user
configuration files? [Y/n]: Y
INFO: Asset Manager overwrite requested, evaluating rules ...
INFO: Unpacking /home/efm-user/Downloads/EFM-1-5-0/Components/asset-link.zip
INFO: Unpacking /home/efm-user/Downloads/EFM-1-5-0/Components/simulation-link.zip
INFO: Unpacking /home/efm-user/Downloads/EFM-1-5-0/Components/asset-manager.zip
INFO: Skipping existing configuration file /home/efm-
user/efm/asset_manager/config/users.json
INFO: Skipping existing configuration file /home/efm-
user/efm/asset_manager/config/application-conf.json
Please execute a 'Rescan' and 'Start Link' on the broker to start the Asset Link. Done?
[Y/n]: Y

*****
*****
Don't forget to execute '/home/efm-
user/efm/asset_manager/installation/install_dependencies.sh' with root permissions to
finish the Asset Manager installation
*****
*****

INFO: The component Asset Manager has been installed
```

After the Asset Manager has been installed, the following script needs to be run to install the dependencies.

Executing(as root user) `sudo /home/efm-user/efm/asset_manager/installation/install_dependencies.sh`

INFO: Done.

After installing the dependencies script, nginx needs to be restarted so configuration changes can be applied. Execute the following command before starting the Asset Manager:

```
sudo service nginx restart
```

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-1-5-0]$ rm -rf /opt/cisco/kinetic/*./efm-linux parstream
[EFM@efm-rh72-149 EFM-1-5-0]$ ./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< - - - -
INFO: LD_LIBRARY_PATH=/opt/cisco/kinetic/parstream/lib
INFO: - - - - >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< - - - -
```

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.



```
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 -
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:

```
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, this is found in `/home/efm/opt/parstream`), an *examples* directory exists. These examples provide you with basic ParStream configuration files. A *conf* directory exists in each example's directory where 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.

Installing the C-Broker

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

Typing `./efm-linux cbroker`

```
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)
INTERACTION: Installed EFM C-Broker & Life Cycle Manager. Check that the broker can
execute and expose its version? [Y/n]:
INFO: Check that the broker can execute and expose its version requested
INFO: For future execution of the EFM C-Broker & Life Cycle Manager possible environment
settings are:
INFO: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - - 8< - - - -
INFO: LD_LIBRARY_PATH=/opt/cisco/kinetic/efm_cbroker/lib
```

```

INFO: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- >8 - - - - >8 -
INFO: For future execution of the EFM C-Broker & Life Cycle Manager:
/opt/cisco/kinetic/efm_cbroker/bin/broker
INFO: Executing check that the broker can execute and expose its version
DEBUG: - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8< - - - - 8<
- - - - 8< - - - -
DEBUG: IOT-DSA c-sdk version: 1.2.0
DEBUG: - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - - >8 - - - -
- - >8 - - - - >8 -
INFO: The component EFM C-Broker & Life Cycle Manager has been installed

```

It is recommended to set the environment `LD_LIBRARY_PATH` as suggested by the install script (and the `PATH` for ease of execution):

```

export LD_LIBRARY_PATH=/opt/cisco/kinetic/efm_cbroker/lib:$LD_LIBRARY_PATH
export PATH=$PATH:/opt/cisco/kinetic/efm_cbroker/bin

```

Installing DGLux5

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

Typing `./efm-linux dglux`

```

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: AcuityBrands DGLux Server server configuration template copied successfully from
/home/efm/EFM-1-5-0/Components/server_dglux_certs.json
INFO: The component AcuityBrands DGLux Server has been installed

```

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-linux 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

```


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, but 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, EFM ParStream Historian Database, and the EFM Asset Manager.

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-1-3-0/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 Adminlogin 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:
0445nwoHcQpKm89jDNOSgWmz1qhjBnPBpKvmV5icpk+0P0=LQIj9+HzjImaD6or44Vu7CYXlwtgWuic2OeS4oPXDs=
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]:
INFO: EFM Message Broker check succeeded with ['DSA Version: 1.1.2', 'DGLux5 Build:
r7907', 'DGLux Server Build: 1217', 'Verifying Environment...', 'Verifying
Configuration...']
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)
INFO: Check of cpu core count - passed. (ok)
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< - - - - -
INFO: LD_LIBRARY_PATH=/opt/cisco/kinetic/parstream/lib
INFO: - - - - - >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< - - - - -
DEBUG: [2018-02-27T05:49:13]:unknown-000000:PROT-77001: ***** Starting Cisco ParStream
Server 5.1.0 20171106T163434Z Release
DEBUG: [2018-02-27T05:49:13]:unknown-000000:PROT-77083: **** cmdlineargs: --version
DEBUG: [2018-02-27T05:49:13]:unknown-000000:PROT-77086: **** PID: 3922 (host:
centos7)
DEBUG: /opt/cisco/kinetic/parstream/bin/parstream-server version: 5.1.0 built:
20171106T163434Z (34acfa088f7671db01b25b9276c1de0c033c076f)
DEBUG: - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - -
- - >8 - - - - - >8 -
INFO: The component EFM ParStream Historian Database has been installed
INFO: Installing Asset Manager ...
INFO: Check of platform support - passed. (ok)
INFO: Java found at /usr/bin/java
INFO: Java seems to be version 8 (ok)
INFO: Nginx found at /usr/sbin/nginx
INFO: Unpacking /home/notroot/EFM-1-5-0/Components/asset-link.zip
```


Upgrading to EFM version 1.5.0 from an existing installation

If an existing version 1.0.0, 1.0.1, 1.10 or 1.2.0 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 to perform 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 pre-requisites to upgrading:

- Download the EFM 1.5.0 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 affect 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 1.5, 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, 1.2.0 or 1.1.0) to EFM 1.5.0, 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 that accompanies 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:

1. Prepare upgrade like a fresh install and unpack the EFM-1-5-0.zip accordingly and change the current working directory to EFM-1-5-0 folder all as described above in the Installation section.

```
$> unzip -q EFM-1-5-0.zip
$> cd EFM-1-5-0
```

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

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

3. Stop the broker if running:

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

4. Finally upgrade your EFM installation to 1.5.0 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.

Starting and stopping the EFM Message Broker

This step starts the EFM Message Broker, EFM Data Flow Engine and Editor, and the DSLinks that 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 privileges 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
```

Starting and stopping the EFM ParStream Historian Database

To start the ParStream Historian Database without the `run_cluster.sh` script, navigate to 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.

⁴ Note that this self-signed certificate will present a browser security exception. To eliminate the security³¹ warning, a valid certificate should be installed.

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 shut down 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.

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.

Starting and stopping the EFM Asset Manager

This step starts the Asset Manager that is already installed.

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

- To start the EFM Asset Manager, execute the following as non root user: `.bin/start.sh`
- To stop the EFM Asset Manager, execute the following as non root user: `.bin/stop.sh`

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.

The logs can be found under `logs/` of the default installation path of asset manager.

Configuring the EFM Asset Manager

The EFM Asset Manager configuration files are located in the `$(EFM_ROOT)/asset_manager/config` folder.

- To modify rediscovery period (in minutes) of assets, modify the `restartIntervalMinutes` value in the `$(EFM_ROOT)/asset_manager/config/application-conf.json` file. For changes to take effect, the EFM Asset Manager needs to be stopped and started.
- To modify the HTTPS port, modify the `listen` value in the `$(EFM_ROOT)/asset_manager/config/ssl.conf` file. For changes to take effect, the `nginx` needs to be stopped and started.

Connecting to the EFM System components

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

Application	Insecure Port (if supported) ⁷	Secure Port
EFM System Administrator	<code>http://[Server IP Address:Port]/efm-admin</code>	<code>https://[Server IP Address:Port]/efm-admin</code>
EFM System Monitor	<code>http://[Server IP Address:Port]/efm-monitor</code>	<code>https://[Server IP Address:Port]/efm-monitor</code>
EFM Data Flow Editor	<code>http://[Server IP Address:Port]/dataflow.html</code>	<code>https://[Server IP Address:Port]/dataflow.html</code>
EFM Asset Manager	Not Supported	<code>https://[Server IP Address]</code>

Defaults, unless changed during installation or configuration, are:

Application	Insecure Port (if supported)	Secure Port
EFM System Administrator	<code>http://[Server IP Address:8080]/efm-admin</code>	<code>https://[Server IP Address:8443]/efm-admin</code>
EFM System Monitor	<code>http://[Server IP Address:8080]/efm-monitor</code>	<code>https://[Server IP Address:8443]/efm-monitor</code>
EFM Data Flow Editor	<code>http://[Server IP Address:8080]/dataflow.html</code>	<code>https://[Server IP Address:8443]/dataflow.html</code>
EFM Asset Manager		<code>https://[Server IP Address:443]</code>

⁷ Note that 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. 33

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 consult the Operating System Guide to turn off or allow only the known service ports for the EFM connections. The proper configuration needs to be defined by the host administrator.⁸

For the firewall to allow for incoming connections on the 443 and 8443 and on Redhat/CentOS, the following commands can be executed:

```
$ sudo firewall-cmd --add-port=443/tcp --permanent
$ sudo firewall-cmd --add-port=8443/tcp --permanent
```

If using unencrypted connections to the broker on the default 8080 port, this will need to be added:

```
$ sudo firewall-cmd --add-port=8080/tcp --permanent
```

You must restart the firewall to implement the changes.

```
$ sudo firewall-cmd --reload
```

Note that if any incoming connections for DSLinks to the ParStream database, etc. exist, those specific ports should be configured to allow incoming connections.

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:

⁸ Unless properly configured and made permanent, on RedHat and CentOS, the firewall service will restart in the default configuration.

- 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`

An alternative to removing the use of the proxy server is to define an exclusion list that includes the localhost. In this manner at least the localhost will not be forwarded to the proxy server and communications between the Message Broker and the DSLinks that are on the local host form a connection.⁹

Configuring the EFM Message Broker server via the server.json file

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

```
{
  "debug": false,
  "host": "0.0.0.0",
  "port": 8080,
  "httpsPort": 8443,
  "certName": "cert.pem",
  "certKeyName": "key.pem",
  "certPassword": "",
  "enableHSTS": false,
  "enableCSRFProtection": false,
  "strictFileUpload": {
    "enabled": false,
    "useClamAV": false,
    "extensions": [
      "dg5",
      "dgi",
      "crt",
      "key",
      "woff",
      "ttf",
      "gif",
      "svg",
      "png",
      "jpg",
      "xml",
      "json",
      "sql",
      "csv"
    ]
  },
  "disableFileSecurity": false,
```

⁹ See <http://xmodulo.com/how-to-configure-http-proxy-exceptions.html> for examples on Linux.

```
"isAlwaysOffline": false,
"broadcast": false,
"workers": 1,
"updateInterval": 200,
"static": {
  "/.well-known": "/home/efm-user/efm/efm_server/.well-known"
},
"linkConfig": {},
"disabledLinks": [],
"uptimeCheckUrl": null,
"upstream": {},
"quarantine": false,
"allowAllLinks": true,
"defaultPermission": [
  [
    ":config",
    "config"
  ],
  [
    ":write",
    "write"
  ],
  [
    ":read",
    "read"
  ],
  [
    ":user",
    "read"
  ],
  [
    ":trustedLink",
    "config"
  ],
  [
    "default",
    "config"
  ]
],
"useRuntimeManager": false,
"useDartRuntimeManager": false,
"useJavaRuntimeManager": false,
"passwordHasherIterations": 1000,
"passwordHasherKeyLength": 32,
"loginRedirectPath": "/",
"guestLoginRedirectPath": "/assets/",
"authType": "file",
"twoFactorAuth": "none",
"runPortChecks": true,
"storageDriver": "simple",
"downstreamName": "downstream",
"enableUptimeChecker": true,
"loggers": [],
"proxies": {},
```

```

"hooks": {},
"distributionUrl": "NO",
"linkRepositoryUrl": "https://dsa.s3.amazonaws.com/links/links.json",
"serverVmFlags": [],
"userTimeout": 525600,
"allowBrowserCaching": false,
"serverLogLevel": "INFO",
"enableLogCompression": true,
"logRotationInterval": 0,
"enableIPv6": false,
"dartRuntimeManagerVmFlags": [],
"javaRuntimeManagerVmFlags": [],
"allowPasswordChanges": true,
"keepCustomAssets": true,
"linkManagerEnvironment": {},
"timeHttpRequests": false,
"generatedCertificateSubject": "/C=US/ST=California/L=Oakland/O=DGLogik
Inc./OU=Customers/CN=*",
"enableCertificateGeneration": true,
"alternativeBrokerUrl": null,
"httpPathClassification": {},
"corsProxyRules": "",
"enableGit": false,
"enableSingleSignOnServer": false,
"maxQueueSize": 256,
"ssoProviderUrl": null,
"formatDg5": false,
"allowedCorsRegexString": null,
"loginAuditFileName": "audit.log",
"loginAudit": false,
"blockOutsideGuests": false,
}

```

In the following table, the default values are listed 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	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.



Linux Installation Guide - Cisco Kinetic EFM, Release 1.5.0

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	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		
enableHSTS	When this value is true, the HTTP server will always redirect to the HTTPS server, and the HTTPS server will have HSTS enabled to route requests automatically to the HTTPS server.		
enableCSRFProtection	When this value is true, the HTTP server will add specific headers and cookies to help mitigate Cross-Site Request Forgery attacks.	false	
disableFileSecurity	When this value is true, then any user can access any file. When this is false, file permissions are checked.	false	

<p>strictFileUpload</p>	<p>strictFileUpload is a configuration map that contains 3 fields. When this option is enabled, it will affect various file upload capabilities. Notably, it prevents guest users from being able to upload a file; It will limit uploads to only explicitly permitted file extensions; and possibly scan uploaded files for viruses.</p> <p>The configuration options are:</p> <p>enabled</p> <p>When set to true, strictFileUpload is enabled. If false, it will disable strictFileUpload checks.</p> <p>useClamAV</p> <p>When set to true, the server will attempt to find <u>Clam Antivirus</u> on the system and if located, it will try to utilize this to scan any file uploads the server receives from a user. If this value is false, or the ClamAV was not found on the system, antivirus scans will be skipped, but other strictFileUpload conditions still apply if enabled.</p> <p>extensions</p> <p>An allow list of permissible file extensions (omitting the leading .) When strictFileUpload is enabled, the filename must end in one of these extensions or the upload will be rejected.</p>	<pre>{ "enabled": false, "useClamAV": false, "extensions": ["dg5", "dgi", "crt", "key", "woff", "ttf", "gif", "svg", "png", "jpg", "xml", "json", "sql", "csv"] }</pre>	
<p>disableFileSecurity</p>	<p>When this value is true, then any user can access any file. When this is false, file permissions are checked.</p>	<p>false</p>	
<p>broadcast</p>	<p>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.</p>	<p>true</p>	
<p>isAlwaysOffline</p>	<p>Indicates that a server is expected to never have a full internet connection. This will prevent the server from trying to download the list of DSLinks available in the remote repository.</p>	<p>false</p>	

workers	<p>Number of Server Workers. For low end devices, this should stay at 1. For large machines, this can be set up to a maximum of 128. It is recommended that you do not exceed the number of logical processors on your machine.</p>	<p>For single-core machines, this is 1, for other devices, this is 2.</p>	
updateInterval	<p>Only affects the responder. When this setting specified, a responder must not send stream updates to server more often than the minimum interval in milliseconds, value subscriptions in the responder should be cached.</p> <p>If a value subscription update is already cached, it must update the cache with the new value to prevent useless updates or updating an incorrect value.</p> <p>This value only affects the time between two updates of the same stream.</p> <p>If the responder does not respect the interval, the requestor might close the connection due to flooding.</p>	200	
static	<p>Configures a static directory mapping. This is used to serve files and directories on the server.</p> <p>Example:</p> <pre>{ "/static": "/srv/http/static" }</pre>	<pre>{".well-known": "/path/to/dsa/dglux-server/.well-known" }</pre>	
linkConfig	<p>Each DSLink may optionally specify its own configuration parameters to use. These configuration parameters can be see under the <code>/sys/links/<linkName>/configs</code> node. If you modify one of those parameters, the value is updated in the server configuration, as opposed to directly modifying the DSLink's configuration file. This value will vary depending on the DSLinks installed, their given names and the configuration parameters they may provide. It should be modified from the DSA node tree rather than by hand.</p>		

<p>uptimeCheckURL</p>	<p>The server has a built-in checker to verify it is still running, and restart it if it goes offline or drops connections. By default the checker will attempt to connect to localhost. However if the server is bound to a different interface in the host parameter, you will need to specify the correct URL for the server. It should end in /ping. This only applies when enableUptimeChecker is enabled.</p> <p>Example:</p> <pre>"https://169.254.100.100/ping"</pre>		
<p>defaultPermission</p>	<p>Default permission setting for the root node. When this value is null, permissions are disabled, and everything has the config permission.</p>	<pre>[[":config", "config"], [":write", "write"], [":read", "read"], [":user", "read"], [":trustedLink", "config"], ["default", "none"]]</pre>	

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	<p>** This setting has no effect when allowAllLinks is true **</p> <p>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.</p>	false	
isAlwaysOffline	Indicates that a server is expected to never have a full internet connection.	false	
useRuntimeManager	<p>This value enables both the useDartRuntimeManager and useJavaRuntimeManager on the server if applicable.</p> <p>WARNING: Setting this value has no effect on a Windows based server</p>		
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	
passwordHasherIteration	When using file based authType, passwords are encrypted locally using PBKDF2. This value determines the number of iterations of the PBKDF2 algorithm used to encode the password.	10000	
passwordHasherKeyLength	When using file based authType, this value determines the number of bytes that the encoded password should store.	32 bytes	

loginRedirectPath	Determines the URI that a user is redirected to when login is complete.	/	
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	
runPortChecks	When set to true, this option will verify that the configured ports for the server (HTTP and HTTPS) are valid and available for use prior to actually starting the server.	true	
storageDriver	This option is available for future expansion for how data is persisted at various QOS levels. Currently only simple is supported.	simple	
downstreamName	This value is the name of the downstream connections node. Previously releases used a downstream name of <code>conns</code> . However it is recommended to leave this as the default value, as other Requester DSLinks may make assumptions of the correct path.	downstream	
enableUptimeChecker	The server also comes with a checker which will periodically check to verify that the server is still up and running and responsive. Setting this value to false will disable the uptime checker.	true	
loggers	The server contains a number of specialized loggers, particularly for debugging, which may be added here to retrieve verbose logging information. Some examples include "File Service" and "Execute". These would normally be advised to be enabled at the request of support.		
proxies	This value is a Map of path (key) and URI (value) pairs. Requests to the path will be forwarded to the URI	{}	

hooks	This value is designed to execute a specific command line program at various server states. Currently the only supported state is <code>ready</code> which executes when the server has finished loading. The Map contains keys of state (eg <code>ready</code>) and value of a list of command line programs to execute.	{}	
distributionURL	This value is the url used to check for updates of the EFM server. This value can be managed in the <code>/sys/config</code> nodes (generally should not change from default).	NO	
linkRepository	This value is the url used to check for updates for any of the DSA Links installed via repository. This value can be managed in the <code>/sys/config</code> nodes (generally should not change from default).		
serverVmFlags	This value is a list of flags to add to the server when being started. They only apply to the EFM server and not any managed links.		
userTimeout	Number of minutes of user inactivity (nothing being loaded from the server) after which session times out. This is a general setting, cannot be set per user.	525600	
allowBrowserCaching	When enabled, this value will add Cache-Control headers for 300 seconds on static files such as <code>.dg5</code> , images, etc.	false	
serverLogLevel	Sets the log level verbosity. Levels are: NONE; SEVERE; WARNING; INFO; FINE; FINEST; ALL; DEBUG. Each level will report its level and all prior to it. (Example: INFO will log all INFO, WARNING and SEVERE messages).	INFO	
enableLogCompression	If this value is true, then when log files reach approximately 5MB in size, they will be rotated and compressed. Log files will be renamed to <code><logfile>.<millisecondTimeStamp>.gz</code>	true	

logRotationInterval	This value is the number of seconds to wait before rotating log files. When this option is enabled (anything greater than 0) enableLogCompression will not be used. After the specified period, a log file will be renamed to <filename>.<number> the higher the number the older the log file. Any files greater than 2 will be removed.	0	
enableIPv6	Toggles support for IPv6 connections	false	
dartRuntimeManagerVmFlags	When the useRuntimeManager or useDartRuntimeManager options are enabled and the platform supports the use of a runtime manager, then the flags provided here are passed to the Dart VM prior to starting the DSLink manager.	[]	
javaRuntimeManagerVmFlags	When the useRuntimeManager or useJavaRuntimeManager options are enabled and the platform supports the use of a runtime manager, then the flags provided here are passed to the Java VM prior to starting the DSLink manager.	[]	
allowPasswordChanges	When true, this value will enable passwords to be updated via the /change_password URL (after the user has logged in). This is only work if supported by the current (authType)[#authtype]	true	
keepCustomAssets	When the value is true, custom assets in www/assets are kept upon updating EFM Server.	false	
linkManagerEnvironment	This value is a map of environment variables to set when the DSLink manager is started.	{}	
timeHttpRequests	If enabled, this value will cause all HTTP requests to the EFM Server to be timed and the log will be updated with the request and elapsed duration.	false	
generatedCertificateSubject	If the option enableCertificateGeneration is enabled, this is the subject used when generating the self-signed certificate.	/C=US/ST=California/L=Oakland/O=Acuity Brands Inc./OU=Customers/CN=*	

enableCertificateGeneration	When this option is set to true, the server will attempt to generate self-signed SSL certificates prior to launching the server. This will set the appropriate certName, certKeyName. If these values are not empty, then certificate generation will be skipped.	true	
alternativeBrokerUrl	If you wish for all DSLink connections to be forwarded to a separate broker rather than the default broker, you would specify the URI of the alternative broker here. This was primarily used for legacy installations.	null	
httpPathClassification	This value is a map of paths which may match a specific classification string. The key is the classification and the value is a list of paths which match that classification. If enabled and a requested path to the server matches a path in that classification, then that request will be treated as that type of classification request even if not matching the original hardcoded path. Currently the only supported classification is <code>session</code> .	{}	
corsProxyRules	The EFM server may also be used to proxy requests to external servers. To limit the locations which the proxy can access, a list of addresses, separated by new lines, may be added to the string.	" "	
enableGit	This value will enable git version control over your project directory. When enabled, modifications to files in the project will be committed to a git repository at the same file path, and can be used in project management to see a history of changes and even revert changes.	false	
enableSingleSignOnServer	In an environment where there are multiple instances of the EFM Server installed on the network, it is possible to allow all instances to refer to one primary server for authentication. When this option is enabled, this server will act as a primary server and allow other EFM Server instances to query this server for a user session on this server and if found, share it with the other instance. This requires the <code>ssoProviderUrl</code> be supplied to the other EFM Server instances.	false	

maxQueueSize	This value is the maximum number of items stored in the queue to be sent, if the queue reaches a volume greater the behaviour will vary depending on the QOS settings (merged, dropped etc).	256	
ssoProviderUrl	When this value is supplied, it must be the URI of another EFM server instance. This server will request an existing session from the supplied server and if found grant access via that session. If no session is found, the user will be prompted to log into that server and will be redirected to this instance once successfully authenticated.	null	
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	
allowedCorsRegexString	If you wish to allow, but restrict, the access of external sites to interface with your EFM server, you can set a Regular Expression string here which much match for the external server requests to be completed.	null	
loginAuditFileName	This value only applies when <i>loginAudit</i> is enabled. This will be the filename, within the <code>/logs</code> directory, in which the login audits are recorded.	audit.log	
loginAudit	When enabled, this option will log to the <i>loginAuditFileName</i> any user logins, it will record the DateTime, username, and the IP address from which the request originated. It will also log any time a user's IP address changes during an active session.	false	
blockOutsideGuests	Enable this value if you wish to require a valid user login to view all projects.	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.