

# Linux Installation Guide

---

*Cisco IoT Data Connect - Edge and Fog Fabric (EFF) 1.0.1*

**Revised:** August 25, 2017

## Contents

Introduction .....	3
Features and Functions .....	3
The Edge and Fog Fabric Components .....	4
Hardware Requirements .....	5
EFF Components Protocols and Ports .....	5
Licensing installation and requests .....	6
Required Third Party Libraries for EFF Installation on Linux .....	6
Secure Mode Operation .....	6
Basic component installation scenarios .....	7
General Concepts: .....	8
Defining an EFF 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 EFF Components .....	10
Install help or usage: .....	11
Installing the Smart Licensing Tool: .....	12
Installing the EFF Message Broker: .....	12
Installing the System Administrator: .....	13
Installing the System Monitor: .....	14
Installing the ParStream Historian Database: .....	15

Installing all of the EFF Linux components: .....	17
Upgrading to EFF version 1.0.1 from an existing version 1.0 installation .....	20
Configuring SSL certificate for secure connections .....	21
Starting and stopping the EFF Message Broker .....	21
Starting and stopping the EFF ParStream Historian Database .....	21
Starting and Stopping EFF System Administrator, EFF System Monitor and EFF DataFlow Editor .....	23
Connecting to the EFF System Components .....	23
Troubleshooting .....	23
Linux Firewall issues .....	23
Proxy Server challenges and the EFF Message Broker .....	24
EFF Message Broker Server Configuration via the server.json file .....	25
Obtaining documentation and submitting a service request .....	29

## Introduction

The Cisco Edge and Fog Fabric (EFF) allows you 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 the can be at the edge, fog or in the datacenter. The Edge and Fog Fabric 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 EFF addresses the complexity of building an enterprise ready scalable data messaging system for one or many applications to reside upon. The EFF comes with a series of tools to management the system, the EFF system administrator and the EFF 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 Fabric Components

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

## Hardware Requirements

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

## EFF Components Protocols and Ports

The protocols and ports used by the EFF Broker and the EFF 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
EFF Administrator Defined	Connection to ParStream Historian Server (when installed) from a local or remote ParStream DQLink. 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 IoT Data Connect - Edge Fog and Fabric 1.0 Smart License Agent User Guide.

## Required Third Party Libraries for EFF Installation on Linux

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

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 EFF 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 eff\_server directory.

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

## Basic component installation scenarios

The EFF 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 scenario allows for development and testing, but is not typical for a production system.

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

- EFF 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 EFF is out of license compliance.
- EFF 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. This is the administrative console that allows for configuration and operation of the EFF System components. A message broker is installed on this node in addition to the EFF System Administrator project.
- A System Monitor node, also one per system. The EFF 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.<sup>1</sup>

---

<sup>1</sup> The operation of the System Monitor acts as a subscription to the System link in every broker for telemetry. If certain parts of the EFF 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 EFF Administrator User per node

It is important to note that we do not define default username and passwords for EFF. 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 EFF. We suggest creating account “eff” for this purpose, and the following examples throughout the documentation will reference this name. The *adduser* command creates a new user “eff” and a new group “eff”.

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

#### **Redhat 7.2/Centos 7.2**

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

```
$ sudo adduser eff -m -p <password>
```

#### **Ubuntu 16.04**

```
$ sudo adduser eff
```

```
Enter new UNIX password: <password>
```

```
Retype new UNIX password: <password>
```

```
Enter the new value, or press ENTER for the default
```

```
Full Name []: eff
```

```
Room Number []:
```

```
Work Phone []:
```

```
Home Phone []:
```

```
Other []:
```



**For all operating systems:**

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

```
$ sudo mkdir /opt/cisco; sudo chown eff /opt/cisco; sudo chgrp eff /opt/cisco
```

Logging in as user “eff” from the current user

```
$ su eff
```

Place the EFF software image in the eff home directory.

The software should be downloaded from CCO at [www.cisco.com](http://www.cisco.com) under Support and Downloads.

Unzip the image

```
$ unzip EFF_1.0.zip
```

Change into unzipped folder

```
$ cd EFF-1.0
```

## 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 EFF_ROOT=/opt/cisco
```

```
$ ./eff-linux broker
```

Environment Variable	Description	Default
EFF_ROOT	should be set to the *absolute* path of the install root (i.e. folders eff_server, parstream, and dart-sdk will be created below that path)	/opt/cisco/iotdc

EFF_GUI_LOGIN	set to the <i>*name*</i> of the EFF GUI Admin User	effAdmin
EFF_GUI_PHRASE	set to the <i>*pass phrase*</i> of the EFF GUI Admin User	Not set
EFF_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)	
EFF_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)	
EFF_BROKER_SECURE_PORT	overwrite the default value for httpsPort in server.json	8443
EFF_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
EFF_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
EFF_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
EFF_INSTALL_LOGS	set to the <i>*absolute*</i> path of a folder for install logs (default: case logs subfolder of EFF-1.0 e.g. \$HOME/EFF-1.0/logs if EFF-1.0.zip unpacked in \$HOME folder) (e.g. needed in case the unpacked install components are stored on a read-only medium)	
EFF_DEBUG	the script is run with more explicit warn and fail early flags	

## Installation of the EFF Components

The EFF installer `eff-linux` is a script that is designed to allow for interactive or non-interactive installation of the EFF 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 EFF-1.0.zip has been unpacked inside that folder.

## Install help or usage:

### Executing `./eff-linux` displays the help screen.

IoT DataConnect - Edge & Fog Fabric - Installer and checksum tool v1.0

Synopsis: `./eff-linux [install|report|verify]`  
`[broker|admin|license|monitor|standalone_monitor|parstream]`

Sample usage(s):

```
./eff-linux install          # -> install all components (broker, admin, monitor, license, and
parstream)

./eff-linux broker          # -> installs broker, data flow editor and run-time
./eff-linux admin           # -> installs admin
./eff-linux standalone_monitor # -> installs standalone_monitor (including its own broker)

./eff-linux license         # -> installs smart license agent
./eff-linux parstream       # -> installs parstream historian database

./eff-linux verify          # -> report current settings and SHA512 checksums of components

./eff-linux verify parstream # -> calculate SHA512 checksum of parstream historian database

./eff-linux version         # -> report version banner of this tool

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

Notes:

Installing the broker component is a precondition for admin and monitor installation.

Environment Variables:

`EFF_ROOT` should be set to the *absolute* path of the install root (default: `/opt/cisco/iotdc`)  
(i.e. folders `eff_server`, `parstream`, and `dart-sdk` will be created below that path)

`EFF_GUI_LOGIN` may be set to the *name* of the EFF GUI Admin User (default: `effAdmin`)

`EFF_GUI_PHRASE` may be set to the *pass phrase* of the EFF GUI Admin User (default: not set)  
(This variable will be ignored for now in interactive install sessions)

`EFF_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`)

`EFF_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`)

`EFF_BROKER_SECURE_PORT` will overwrite the default value of 8443 for `httpsPort` in `server.json`

`EFF_BROKER_CLEARTXT_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!)

`EFF_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)

`EFF_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`)

EFF\_INSTALL\_LOGS may be set to the *\*absolute\** path of a folder for install logs (default: /Users/sthagen/repo/eff/deploy/  
EFF-Install/logs) (e.g. needed in case the unpacked install components are stored on a read-only medium)

Copyright:

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

## Installing the Smart Licensing Tool:

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

### Typing `./eff-linux license`

```
[root@eff-rh72-145 EFF-1.0]# ./eff-linux license
[2017-04-11 16:48:07.824836Z]:<INFO>: Install logs will go into folder(/home/userid/EFF-1.0/logs)
[2017-04-11 16:48:07.946004Z]:<INFO>: Install source folder(/home/userid/EFF-1.0)
[2017-04-11 16:48:08.066312Z]:<INFO>: Install logs folder(/home/userid/EFF-1.0/logs)
[2017-04-11 16:48:08.178444Z]:<INFO>: Installer version(1.0.0)
[2017-04-11 16:48:08.278162Z]:<INFO>: Installing EFF Smart License Agent
[2017-04-11 16:48:08.404189Z]:<INFO>: Unpacking archive of EFF Smart License Agent (install trace
written to (/home/userid/EFF-1.0/logs/install_trace_license-20170411T184808.log) ...
Display all 63 lines of install log trace from (/home/userid/EFF-1.0/logs/install_trace_license-
20170411T184808.log. Continue? [y/N]
[2017-04-11 16:48:13.302061Z]:<INFO>: EFF Smart License Agent installed successfully.
Display README from /opt/cisco/iotdc/eff_license/readme.md. Continue? [y/N]
[2017-04-11 16:48:22.165830Z]:<INFO>: Success.
```

## Installing the EFF Message Broker:

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

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

### Executing `./eff-linux broker`

```
[root@eff-rh72-145 EFF-1.0]# ./eff-linux broker
[2017-03-27 06:29:44.997261Z]:<INFO>: Install logs will go into folder(/home/userid/EFF-1.0/logs)
Environment variable EFF_ROOT unset! Current default target set to '/opt/cisco/iotdc' (where everything
will be installed below) Continue? [y/N] y
[2017-03-27 06:29:58.661940Z]:<INFO>: Note: Selected to continue install into '/opt/cisco/iotdc'
This user has superuser rights! EFF should be installed as non root user ... Continue? [y/N] y
[2017-03-27 06:30:02.146610Z]:<WARNING>: Continuing as super user upon request!
[2017-03-27 06:30:02.158288Z]:<INFO>: Install source folder(/home/userid/EFF-1.0)
[2017-03-27 06:30:02.164737Z]:<INFO>: Install logs folder(/home/userid/EFF-1.0/logs)
[2017-03-27 06:30:02.171189Z]:<INFO>: Installer version(1.0.0)
[2017-03-27 06:30:02.177670Z]:<INFO>: EFF GUI Admin User(effAdmin) *default* as EFF_GUI_LOGIN not set
or empty
[2017-03-27 06:30:02.184095Z]:<INFO>: EFF_GUI_PHRASE not set or empty
[2017-03-27 06:30:02.190392Z]:<INFO>: Broker Cleartext Port(8080) i.e. enabled (consider disabling for
production setups)
[2017-03-27 06:30:02.199233Z]:<INFO>: Broker Cleartext Port(8443) i.e. enabled (strongly suggested for
production setups)
[2017-03-27 06:30:02.206206Z]:<INFO>: Broker Server Full Certificate Chain PEM Filepath(server.pem)
[2017-03-27 06:30:02.212605Z]:<INFO>: Broker Server Private Key PEM Filepath(key.pem)
[2017-03-27 06:30:02.219008Z]:<INFO>: Broker isAlwaysOffline = false (default, and suggested for setups
with internet access)
```



```
[2017-03-27 06:30:02.225267Z]:<INFO>: Detected CpuCoreCount(1) using CpuDetectionMethod(nproc)
[2017-03-27 06:30:02.231611Z]:<INFO>: Trying to set workers in broker config server.json to
CpuCoreCount(1) ...
[2017-03-27 06:30:02.238306Z]:<INFO>: Installing EFF Message Broker
[2017-03-27 06:30:02.246257Z]:<INFO>: Unpacking archive of EFF Message Broker (install trace written to
/home/userid/EFF-1.0/logs/install_trace_broker-20170327T023002.log) ...
Display all 7598 lines of install log trace from /home/userid/EFF-1.0/logs/install_trace_broker-
20170327T023002.log. Continue? [y/N]
[2017-03-27 06:30:17.388838Z]:<INFO>: Updating permissions on scripts of EFF Message Broker ...
[2017-03-27 06:30:18.230603Z]:<INFO>: EFF Message Broker installed successfully.
Change EFF GUI Admin User from effAdmin to something else. Continue? [y/N]
[2017-03-27 06:30:22.351833Z]:<INFO>: Keeping EFF GUI Admin User as effAdmin.
***** Set EFF GUI Admin User (effAdmin) pass phrase *****
Please enter the EFF GUI Runtime Engine effAdmin pass phrase:
Please verify the pass phrase:

[2017-03-27 06:30:30.253309Z]:<INFO>: Create EFF GUI Admin User (effAdmin) ...
Password change for user 'effAdmin' was successful.
ServerConfig(/opt/cisco/iotdc/eff_server/server.json) as is will instruct the server to listen on
8443(secure HTTPS) and 8080(insecure HTTP)
[2017-03-27 06:30:30.601483Z]:<INFO>: Detected CpuCoreCount(1) using CpuDetectionMethod(nproc)
[2017-03-27 06:30:30.607783Z]:<INFO>: Requested to set workers in broker config server.json to
EFF_BROKER_WORKERS(1) ...
Configuring default ServerConfig(/opt/cisco/iotdc/eff_server/server.json) (Backup at
/opt/cisco/iotdc/eff_server/server.json.bak) ...
[2017-03-27 06:30:30.634314Z]:<INFO>: Note: The empty string value to the key certPassword in
server.json matches the empty pass phrase
[2017-03-27 06:30:30.640481Z]:<INFO>: ... of the non-production private key delivered for test-only
setups to ease verification.
[2017-03-27 06:30:30.646798Z]:<INFO>: ... Please verify the consistency of the values given to the
server.json keys:
[2017-03-27 06:30:30.653115Z]:<INFO>: ... certKeyName (private key file path)
[2017-03-27 06:30:30.659471Z]:<INFO>: ... certName (certificate path), and
[2017-03-27 06:30:30.665860Z]:<INFO>: ... certPassword (pass phrase of private key or empty to
enforce prompt upon server start)
[2017-03-27 06:30:30.672181Z]:<INFO>: ... before going into production to allow the server to start
and expose only authentic and hardened systems.
[2017-03-27 06:30:30.678498Z]:<INFO>: Installation created secure by default setup, good.
Prohibit unsecured creation of upstream connections over HTTP using the EFF Dart installation. Keep?
[Y/n]
[2017-03-27 06:30:33.938599Z]:<INFO>: Setup kept secure as installed by default, good.
About to verify EFF Message Broker install. Continue? [Y/n]
DSA Version: 1.1.2
DGLux5 Build: r7487
DGLux Server Build: 1094
Verifying Environment...
Verifying Configuration...
[2017-03-27 06:30:35.517955Z]:<INFO>: Return code OK. See above for version info of EFF Message Broker.
[2017-03-27 06:30:35.524244Z]:<INFO>: Success.
```

### 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.

Executing `./eff-linux admin`



```
[root@eff-rh72-145 EFF-1.0]# ./eff-linux admin
[2017-03-27 06:33:45.377407Z]:<INFO>: Install logs will go into folder(/home/userid/EFF-1.0/logs)
Environment variable EFF_ROOT unset! Current default target set to '/opt/cisco/iotdc' (where everything
will be installed below) Continue? [y/N] y
[2017-03-27 06:33:51.042287Z]:<INFO>: Note: Selected to continue install into '/opt/cisco/iotdc'
This user has superuser rights! EFF should be installed as non root user ... Continue? [y/N] y
[2017-03-27 06:33:53.875457Z]:<WARNING>: Continuing as super user upon request!
[2017-03-27 06:33:53.887571Z]:<INFO>: Install source folder(/home/userid/EFF-1.0)
[2017-03-27 06:33:53.894208Z]:<INFO>: Install logs folder(/home/userid/EFF-1.0/logs)
[2017-03-27 06:33:53.900801Z]:<INFO>: Installer version(1.0.0)
[2017-03-27 06:33:53.907488Z]:<INFO>: Installing EFF System Administrator
[2017-03-27 06:33:53.915687Z]:<INFO>: Unpacking archive of EFF System Administrator (install trace
written to /home/userid/EFF-1.0/logs/install_trace_admin-20170327T023353.log) ...
Display all 73 lines of install log trace from /home/userid/EFF-1.0/logs/install_trace_admin-
20170327T023353.log. Continue? [y/N]
[2017-03-27 06:33:56.505690Z]:<INFO>: EFF System Administrator installed successfully. [2017-03-27
06:33:56.505690Z]:<INFO>: EFF System Administrator installed successfully.
```

### Installing the System Monitor:

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

Typing `./eff-linux standalone_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.

```
[root@eff-rh72-145 EFF-1.0]# ./eff-linux standalone_monitor
[2017-03-27 06:35:25.236790Z]:<INFO>: Install logs will go into folder(/home/userid/EFF-1.0/logs)
Environment variable EFF_ROOT unset! Current default target set to '/opt/cisco/iotdc' (where everything
will be installed below) Continue? [y/N] y
[2017-03-27 06:35:27.628709Z]:<INFO>: Note: Selected to continue install into '/opt/cisco/iotdc'
This user has superuser rights! EFF should be installed as non root user ... Continue? [y/N] y
[2017-03-27 06:35:35.917937Z]:<WARNING>: Continuing as super user upon request!
[2017-03-27 06:35:35.929872Z]:<INFO>: Install source folder(/home/userid/EFF-1.0)
[2017-03-27 06:35:35.936506Z]:<INFO>: Install logs folder(/home/userid/EFF-1.0/logs)
[2017-03-27 06:35:35.943134Z]:<INFO>: Installer version(1.0.0)
[2017-03-27 06:35:35.949719Z]:<INFO>: EFF GUI Admin User(effAdmin) *default* as EFF_GUI_LOGIN not set
or empty
[2017-03-27 06:35:35.956099Z]:<INFO>: EFF_GUI_PHRASE not set or empty
[2017-03-27 06:35:35.962472Z]:<INFO>: Broker Cleartext Port(8080) i.e. enabled (consider disabling for
production setups)
[2017-03-27 06:35:35.968800Z]:<INFO>: Broker Cleartext Port(8443) i.e. enabled (strongly suggested for
production setups)
[2017-03-27 06:35:35.975125Z]:<INFO>: Broker Server Full Certificate Chain PEM Filepath(server.pem)
[2017-03-27 06:35:35.981560Z]:<INFO>: Broker Server Private Key PEM Filepath(key.pem)
[2017-03-27 06:35:35.987851Z]:<INFO>: Broker isAlwaysOffline = false (default, and suggested for setups
with internet access)
[2017-03-27 06:35:35.994076Z]:<INFO>: Detected CpuCoreCount(1) using CpuDetectionMethod(nproc)
[2017-03-27 06:35:36.000378Z]:<INFO>: Trying to set workers in broker config server.json to
CpuCoreCount(1) ...
[2017-03-27 06:35:36.006877Z]:<INFO>: Installing EFF Message Broker
[2017-03-27 06:35:36.014402Z]:<INFO>: Unpacking archive of EFF Message Broker (install trace written to
/home/userid/EFF-1.0/logs/install_trace_standalone_monitor-20170327T023536.log) ...
Display all 7598 lines of install log trace from /home/userid/EFF-
1.0/logs/install_trace_standalone_monitor-20170327T023536.log. Continue? [y/N]
[2017-03-27 06:35:48.193307Z]:<INFO>: Updating permissions on scripts of EFF Message Broker ...
[2017-03-27 06:35:48.291443Z]:<INFO>: EFF Message Broker installed successfully.
Change EFF GUI Admin User from effAdmin to something else. Continue? [y/N]
[2017-03-27 06:35:54.240333Z]:<INFO>: Keeping EFF GUI Admin User as effAdmin.
```



```
***** Set EFF GUI Admin User (effAdmin) pass phrase *****
Please enter the EFF GUI Runtime Engine effAdmin pass phrase:
Please verify the pass phrase:

[2017-03-27 06:35:59.449431Z]:<INFO>: Create EFF GUI Admin User (effAdmin) ...
Password change for user 'effAdmin' was successful.
ServerConfig(/opt/cisco/iotdc/eff_server/server.json) as is will instruct the server to listen on
8443(secure HTTPS) and 8080(insecure HTTP)
[2017-03-27 06:35:59.807933Z]:<INFO>: Detected CpuCoreCount(1) using CpuDetectionMethod(nproc)
[2017-03-27 06:35:59.814511Z]:<INFO>: Requested to set workers in broker config server.json to
EFF_BROKER_WORKERS(1) ...
Configuring default ServerConfig(/opt/cisco/iotdc/eff_server/server.json) (Backup at
/opt/cisco/iotdc/eff_server/server.json.bak) ...
[2017-03-27 06:35:59.842672Z]:<INFO>: Note: The empty string value to the key certPassword in
server.json matches the empty pass phrase
[2017-03-27 06:35:59.848936Z]:<INFO>: ... of the non-production private key delivered for test-only
setups to ease verification.
[2017-03-27 06:35:59.855219Z]:<INFO>: ... Please verify the consistency of the values given to the
server.json keys:
[2017-03-27 06:35:59.861517Z]:<INFO>: ... ... certKeyName (private key file path)
[2017-03-27 06:35:59.867736Z]:<INFO>: ... ... certName (certificate path), and
[2017-03-27 06:35:59.873945Z]:<INFO>: ... ... certPassword (pass phrase of private key or empty to
enforce prompt upon server start)
[2017-03-27 06:35:59.880113Z]:<INFO>: ... before going into production to allow the server to start
and expose only authentic and hardened systems.
[2017-03-27 06:35:59.886355Z]:<INFO>: Installation created secure by default setup, good.
Prohibit unsecured creation of upstream connections over HTTP using the EFF Dart installation. Keep?
[Y/n]
[2017-03-27 06:36:03.148225Z]:<INFO>: Setup kept secure as installed by default, good.
About to verify EFF Message Broker install. Continue? [Y/n]
DSA Version: 1.1.2
DGLux5 Build: r7487
DGLux Server Build: 1094
Verifying Environment...
Verifying Configuration...
[2017-03-27 06:36:08.429884Z]:<INFO>: Return code OK. See above for version info of EFF Message Broker.
[2017-03-27 06:36:08.436363Z]:<INFO>: Success.
[2017-03-27 06:36:08.442940Z]:<INFO>: Removing non-monitor components ...
[2017-03-27 06:36:08.458091Z]:<INFO>: Successfully removed non-monitor components.
[2017-03-27 06:36:08.464845Z]:<INFO>: Installing EFF System Monitor
[2017-03-27 06:36:08.473034Z]:<INFO>: Unpacking archive of EFF System Monitor (install trace written to
/home/userid/EFF-1.0/logs/install_trace_standalone_monitor-20170327T023608.log) ...
Display all 44 lines of install log trace from /home/userid/EFF-
1.0/logs/install_trace_standalone_monitor-20170327T023608.log. Continue? [y/N]
[2017-03-27 06:36:11.418622Z]:<INFO>: EFF System Monitor installed successfully.
[2017-03-27 06:36:11.424939Z]:<INFO>: Change ownership of EFF Standalone System Monitor to created EFF
GUI Admin User (effAdmin) ...
Owner Changed.
[2017-03-27 06:36:11.660272Z]:<INFO>: EFF Standalone System Monitor installed successfully.
```

### Installing the ParStream Historian Database:

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

This installs the ParStream Historian Database.

Executing `./eff-linux parstream`

This will install the ParStream Historian Database to the folder specified in the environment variable `EFF_ROOT`<sup>2</sup>. In the following example `EFF_ROOT` was set to `/home/eff/opt`<sup>3</sup>.

```
[2017-06-23 09:21:57.338433Z]:<INFO>: Install logs will go into folder(/home/eff/repos/deploy/EFF-Install/logs)
[2017-06-23 09:21:57.353828Z]:<INFO>: Install source folder(/home/eff/repos/deploy/EFF-Install)
[2017-06-23 09:21:57.361605Z]:<INFO>: Install logs folder(/home/eff/repos/deploy/EFF-Install/logs)
[2017-06-23 09:21:57.370469Z]:<INFO>: Installer version(1.1.0)
[2017-06-23 09:21:57.374815Z]:<INFO>: ParStream version(4.4.3)
[2017-06-23 09:21:57.404274Z]:<INFO>: Installing EFF ParStream Historian Database (platform=UBUNTU-16)
...
[2017-06-23 09:21:57.845545Z]:<INFO>: Post-Install of local install mode for EFF ParStream Historian Database ...
[2017-06-23 09:21:57.851873Z]:<INFO>: EFF ParStream Historian Database installed successfully inside /home/eff/opt/cisco/iotdc
Suggested to configure PARSTREAM_HOME, LD_LIBRARY_PATH, and PATH as follows:
export PARSTREAM_HOME=/home/eff/opt/cisco/iotdc/parstream
export LD_LIBRARY_PATH=$PARSTREAM_HOME/lib:$LD_LIBRARY_PATH
export PATH=$PATH:$PARSTREAM_HOME/bin
About to verify ParStream install by calling parstream-server --version. Continue? [Y/n]
[2017-06-23 09:22:18.519668Z]:<INFO>: Setting up environment for Cisco ParStream Historian database.
[2017-06-23T11:22:18]:unknown-000000:PROT-77001: ***** Starting Cisco ParStream Server 4.4.3
20170317T142036Z Release
[2017-06-23T11:22:18]:unknown-000000:PROT-77083: **** cmdlineargs: --version
[2017-06-23T11:22:18]:unknown-000000:PROT-77086: **** PID: 17186 (host: eff-virtual-machine)
parstream-server version: 4.4.3 built: 20170317T142036Z (1ff15c03589d1b236f502a67fd857b513ea1c96b)
[2017-06-23 09:22:18.596716Z]:<INFO>: Return code OK. See above for version info of Cisco ParStream Historian database.
```

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

```
export PARSTREAM_HOME=/home/eff/opt/cisco/iotdc/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/eff/opt/cisco/iotdc/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.

---

<sup>2</sup> Please refer to section "*Setting Environment Variables to override default values prior to installation*" for more information about environment variables.

<sup>3</sup> We use the user *eff* in this example.



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 can be found in the "`EFF-1.0/Docs/ParStream 4.4.3`" folder.

### Installing all of the EFF 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 EFF Smart Licensing Tool, EFF Message Broker, EFF System Administrator, EFF System Monitor, EFF DataFlow Editor and the EFF ParStream Historian Database.

### Executing `./eff-linux install`

```
[root@eff-rh72-145 EFF-1.0]# ./eff-linux install
[2017-03-27 06:42:29.056127Z]:<INFO>: Install logs will go into folder(/home/userid/EFF-1.0/logs)
Environment variable EFF_ROOT unset! Current default target set to '/opt/cisco/iotdc' (where everything
will be installed below) Continue? [y/N] y
[2017-03-27 06:42:31.770697Z]:<INFO>: Note: Selected to continue install into '/opt/cisco/iotdc'
This user has superuser rights! EFF should be installed as non root user ... Continue? [y/N] y
[2017-03-27 06:42:33.418010Z]:<WARNING>: Continuing as super user upon request!
[2017-03-27 06:42:33.429465Z]:<INFO>: Install source folder(/home/userid/EFF-1.0)
[2017-03-27 06:42:33.435696Z]:<INFO>: Install logs folder(/home/userid/EFF-1.0/logs)
[2017-03-27 06:42:33.442015Z]:<INFO>: Installer version(1.0.0)
[2017-03-27 06:42:33.448347Z]:<INFO>: EFF GUI Admin User(effAdmin) *default* as EFF_GUI_LOGIN not set
or empty
[2017-03-27 06:42:33.454710Z]:<INFO>: EFF_GUI_PHRASE not set or empty
[2017-03-27 06:42:33.460923Z]:<INFO>: Broker Cleartext Port(8080) i.e. enabled (consider disabling for
production setups)
[2017-03-27 06:42:33.467061Z]:<INFO>: Broker Cleartext Port(8443) i.e. enabled (strongly suggested for
production setups)
[2017-03-27 06:42:33.473138Z]:<INFO>: Broker Server Full Certificate Chain PEM Filepath(server.pem)
[2017-03-27 06:42:33.479265Z]:<INFO>: Broker Server Private Key PEM Filepath(key.pem)
[2017-03-27 06:42:33.485349Z]:<INFO>: Broker isAlwaysOffline = false (default, and suggested for setups
with internet access)
[2017-03-27 06:42:33.491403Z]:<INFO>: Detected CpuCoreCount(1) using CpuDetectionMethod(nproc)
[2017-03-27 06:42:33.497518Z]:<INFO>: Trying to set workers in broker config server.json to
CpuCoreCount(1) ...
[2017-03-27 06:42:33.503849Z]:<INFO>: ParStream version(4.4.3)
[2017-03-27 06:42:33.510234Z]:<INFO>: Installing EFF Message Broker
[2017-03-27 06:42:33.517588Z]:<INFO>: Unpacking archive of EFF Message Broker (install trace written to
/home/userid/EFF-1.0/logs/install_trace_all-20170327T024233.log) ...
Display all 7598 lines of install log trace from /home/userid/EFF-1.0/logs/install_trace_all-
20170327T024233.log. Continue? [y/N]
```



```
[2017-03-27 06:42:47.546830Z]:<INFO>: Updating permissions on scripts of EFF Message Broker ...
[2017-03-27 06:42:50.340317Z]:<INFO>: EFF Message Broker installed successfully.
Change EFF GUI Admin User from effAdmin to something else. Continue? [y/N]
[2017-03-27 06:42:53.197747Z]:<INFO>: Keeping EFF GUI Admin User as effAdmin.
***** Set EFF GUI Admin User (effAdmin) pass phrase *****
Please enter the EFF GUI Runtime Engine effAdmin pass phrase:
Please verify the pass phrase:

[2017-03-27 06:42:57.510676Z]:<INFO>: Create EFF GUI Admin User (effAdmin) ...
Password change for user 'effAdmin' was successful.
ServerConfig(/opt/cisco/iotdc/eff_server/server.json) as is will instruct the server to listen on
8443(secure HTTPS) and 8080(insecure HTTP)
[2017-03-27 06:42:57.859530Z]:<INFO>: Detected CpuCoreCount(1) using CpuDetectionMethod(nproc)
[2017-03-27 06:42:57.865609Z]:<INFO>: Requested to set workers in broker config server.json to
EFF_BROKER_WORKERS(1) ...
Configuring default ServerConfig(/opt/cisco/iotdc/eff_server/server.json) (Backup at
/opt/cisco/iotdc/eff_server/server.json.bak) ...
[2017-03-27 06:42:57.891503Z]:<INFO>: Note: The empty string value to the key certPassword in
server.json matches the empty pass phrase
[2017-03-27 06:42:57.897435Z]:<INFO>: ... of the non-production private key delivered for test-only
setups to ease verification.
[2017-03-27 06:42:57.903359Z]:<INFO>: ... Please verify the consistency of the values given to the
server.json keys:
[2017-03-27 06:42:57.909436Z]:<INFO>: ... certKeyName (private key file path)
[2017-03-27 06:42:57.915520Z]:<INFO>: ... certName (certificate path), and
[2017-03-27 06:42:57.921427Z]:<INFO>: ... certPassword (pass phrase of private key or empty to
enforce prompt upon server start)
[2017-03-27 06:42:57.927386Z]:<INFO>: ... before going into production to allow the server to start
and expose only authentic and hardened systems.
[2017-03-27 06:42:57.933313Z]:<INFO>: Installation created secure by default setup, good.
Prohibit unsecured creation of upstream connections over HTTP using the EFF Dart installation. Keep?
[Y/n]
[2017-03-27 06:43:01.327041Z]:<INFO>: Setup kept secure as installed by default, good.
About to verify EFF Message Broker install. Continue? [Y/n]
DSA Version: 1.1.2
DGLux5 Build: r7487
DGLux Server Build: 1094
Verifying Environment...
Verifying Configuration...
[2017-03-27 06:43:04.533237Z]:<INFO>: Return code OK. See above for version info of EFF Message Broker.
[2017-03-27 06:43:04.539441Z]:<INFO>: Success.
[2017-03-27 06:43:04.545562Z]:<INFO>: Installing EFF System Administrator
[2017-03-27 06:43:04.552842Z]:<INFO>: Unpacking archive of EFF System Administrator (install trace
written to /home/userid/EFF-1.0/logs/install_trace_all-20170327T024304.log) ...
Display all 73 lines of install log trace from /home/userid/EFF-1.0/logs/install_trace_all-
20170327T024304.log. Continue? [y/N]
[2017-03-27 06:43:09.043588Z]:<INFO>: EFF System Administrator installed successfully.
[2017-03-27 06:43:09.049837Z]:<INFO>: Installing EFF System Monitor
[2017-03-27 06:43:09.057302Z]:<INFO>: Unpacking archive of EFF System Monitor (install trace written to
/home/userid/EFF-1.0/logs/install_trace_all-20170327T024309.log) ...
Display all 44 lines of install log trace from /home/userid/EFF-1.0/logs/install_trace_all-
20170327T024309.log. Continue? [y/N]
[2017-03-27 06:43:11.348301Z]:<INFO>: EFF System Monitor installed successfully.
[2017-03-27 06:43:11.374845Z]:<INFO>: Installing EFF ParStream Historian Database (platform=REDHAT-7)
...
Note: As super user you may install 'EFF ParStream Historian Database' either
1) sytem wide (super user only with automatic dependency resolution) or
2) local only (default)
Local only install of 'EFF ParStream Historian Database' with automatic dependency resolution.
Continue? [y/N] y
[2017-03-27 06:43:14.994345Z]:<INFO>: Non-default local plain install selected ...
Display all 141 lines of install log trace from /home/userid/EFF-1.0/logs/install_trace_EFF ParStream
Historian Database-20170327T024314.log. Continue? [y/N]
[2017-03-27 06:43:20.238743Z]:<INFO>: Post-Install of local install mode for EFF ParStream Historian
Database ...
```



```
[2017-03-27 06:43:20.246034Z]:<INFO>: EFF ParStream Historian Database installed successfully inside
/opt/cisco/iotdc
Suggested to configure PARSTREAM_HOME, LD_LIBRARY_PATH, and PATH as follows:
export PARSTREAM_HOME=/opt/cisco/iotdc/parstream
export LD_LIBRARY_PATH=$PARSTREAM_HOME/lib:$LD_LIBRARY_PATH
export PATH=$PATH:$PARSTREAM_HOME/bin
About to verify ParStream install by calling parstream-server --version. Continue? [Y/n]
[2017-03-27 06:43:23.032298Z]:<INFO>: Setting up environment for Cisco ParStream Historian database.
[2017-03-27T02:43:23]:unknown-000000:PROT-77001: ***** Starting Cisco ParStream Server 4.4.3
20170317T142015Z Release
[2017-03-27T02:43:23]:unknown-000000:PROT-77083: **** cmdlineargs: --version
[2017-03-27T02:43:23]:unknown-000000:PROT-77086: **** PID:          24403 (host: eff-rh72-145)
parstream-server version: 4.4.3 built: 20170317T142015Z (1ff15c03589dlb236f502a67fd857b513ealc96b)
[2017-03-27 06:43:23.093137Z]:<INFO>: Return code OK. See above for version info of Cisco ParStream
Historian database.
```

## Upgrading to EFF version 1.0.1 from an existing version 1.0 installation

If an existing version 1.0 is installed and running on a Linux server it is possible to upgrade to the version 1.0.1 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 is already installed and running. Here are the pre-requisites to upgrading:

- Download the upgrade package for Linux
- Stop the running message broker (see below)

The EFF ParStream Historian Database does not upgrade, so if installed it does not require stopping and starting for this process.

Perform the following steps to upgrade:

Unzip file with

```
unzip -q EFF-Upgrade-To-1-0-1.zip
```

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

```
export EFF_ROOT=/opt/cisco
```

Stop the broker if running:

```
$EFF_ROOT/eff_server/bin/daemon.sh stop
```

Finally upgrade your EFF installation to 1.0.1 with the following command and follow the instructions

```
EFF-Upgrade-To-1-0-1/eff-linux-osx upgrade
```

After a successful upgrade, restart the message broker (see below).

## Configuring SSL certificate for secure connections

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

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

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

## Starting and stopping the EFF Message Broker

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

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

### Starting and Stopping EFF Message Broker:

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

```
/opt/cisco/iotdc/eff_server/bin/daemon.sh start
```

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

```
/opt/cisco/iotdc/eff_server/bin/daemon.sh 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/iotdc/eff_server/bin/daemon.sh start
```

```
$sudo /opt/cisco/iotdc/eff_server/bin/daemon.sh stop
```

## Starting and stopping the EFF ParStream Historian Database

---

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

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/eff/opt/parstream/examples/noauthentication`, if installed as user `eff`).

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` directory<sup>5</sup>:

```
$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 command<sup>6</sup>:

```
$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 documentation found in the "`EFF-1.0/Docs/ParStream 4.4.3`" folder.

---

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

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

## Starting and Stopping EFF System Administrator, EFF System Monitor and EFF DataFlow Editor

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

### Connecting to the EFF System Components

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

Application	Insecure Port (if supported) <sup>7</sup>	Secure Port
EFF System Administrator	http://[Server IP Address]/eff-admin	https://[Server IP Address]/eff-admin
EFF System Monitor	http://[Server IP Address]/eff-monitor	https://[Server IP Address]/eff-monitor
EFF 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 EFF connections. But the command below can be used for troubleshooting or staging. The proper configuration needs to be defined by the host administrator.<sup>8</sup>

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

```
$ sudo service firewalld stop
```

---

<sup>7</sup> 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.

<sup>8</sup> The firewall service will restart on next reboot unless the administrator modifies the default configuration

## Proxy Server challenges and the EFF Message Broker

In some environments it might be necessary to define a proxy server to access the Internet due to security restrictions. The EFF 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/iotdc/eff_server/bin/daemon.sh stop`
- Remove the proxy server settings in the environment or system configuration
- Start the message broker with `/opt/cisco/iotdc/eff_server/bin/daemon.sh start`



## EFF Message Broker Server Configuration via the server.json file

Example server.json configuration file located in the \$EFF\_ROOT/eff\_server folder.

```
{
  "allowAllLinks": true,
  "allowBrowserCaching": false,
  "allowPasswordChanges": true,
  "alternativeBrokerUrl": null,
  "authType": "file",
  "broadcast": false,
  "brokerName": "broker-",
  "certKeyName": "key.pem",
  "certName": "server.pem",
  "certPassword": "",
  "corsProxyRules": "",
  "dartRuntimeManagerVmFlags": [],
  "debug": false,
  "defaultPermission": null,
  "disableFileSecurity": false,
  "disabledLinks": [],
  "distributionUrl": "https://dsa.s3.amazonaws.com/dists/dists.json",
  "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/iotdc/eff_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: <pre>{   "/static": "/srv/http/static" }</pre>	<pre>{"/.well-known": "/path/to/dsa/dglux- server/.well- known"}</pre>	
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	<b>** This setting has no effect when allowAllLinks is true **</b>  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	false	

	process.		
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"> <li>• none: Don't enable two factor authentication.</li> <li>• <a href="#">duo</a>: Duo Two-Factor Authentication</li> </ul>	none	
enableIPv6	Toggles support for IPv6 connections	false	
keepCustomAssets	When the value is true, custom assets in www/assets are kept upon updating EFF Server.	false	
formatDg5	When this value is true, eff 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](http://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.