Installation Guide for Windows

Cisco IoT Data Connect - Edge and Fog Fabric (EFF) 1.1.0

Revised: October 24, 2017

Contents

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

EFF Message Broker	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. The EFF Message Broker is a small footprint component working with other brokers to form a message bus.
EFF Data Flow Editor	Defines message paths between devices and micro services.
EFF Data Flow Engine	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.
EFF System Administrator	Configures and manages the message broker and micro services.
EFF 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.
EFF Tools Runtime Engine	A standalone runtime tool for visualizing dashboards and driving EFF System Administrator, EFF Data Flow Engine and EFF System Monitor.
Links	DQL Link – DSA Query Language System Link – System Information ParStream Link v3 – ParStream Historian Database
Smart License Agent Tool for Nodes	The Smart License Agent client that allows system users to manage license registration for Node Product IDs
Smart License Agent Tool for Devices	The Smart License Agent client that allows system users to manage license registration for Device Product IDs

Hardware Requirements

EFF Message Broker **EFF Data Flow Engine** **DQL Link** **System Link** **ParStream Link**	Red Hat Linux 7.2, CentOS 7 or Ubuntu 16.04, Windows 2016 Server 1GB RAM, 10 GB HD* - Recommended on the same system/VM
EFF Data Flow Editor	Automatically installs with EFF Message Broker and EFF Tools Runtime Engine. Access via a web browser
EFF System Administrator	Project installs on the same system as the EFF Message Broker and EFF Tools Runtime Engine. Accessed via a web browser
EFF System Monitor	Project installs on the same system as the EFF Message Broker and EFF 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
EFF Tools Runtime Engine	Installs with EFF 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.

EFF Components Protocols and Ports

The protocols and ports used by the EFF Broker and EFF Tools. The port values are configurable during and after installation.

EFF Broker and EFF Tools

Port No.	Exposure	Protocol	Description
8080	Public	HTTP	Default http or insecure port
8443	Public	HTTPS	Default https or secure port

Licensing installation and requests

This product uses the Smart License Agent Tool (for Nodes and Devices) to manage the corresponding licenses. After installation, refer to the IoT Data Connect - Edge Fog and Fabric 1.0 Smart License Agent User Guide.

Required Libraries for Installation on Windows

For the ParStream DSA Link: Java 8 JRE or JDK

Installation preparation

Place the EFF software image in the home directory.
The software should be downloaded from CCO at www.cisco.com under Support and Downloads.

Then **either** unpack the image by executing the unzip program directly (if on your system):

```
C:\Users\userid\> unzip -q EFF-1-1-0.zip
```

Or – **alternatively** - right click on the archive in the file explorer and select extract all (but do not create an additional folder as suggested per default, instead, we suggest to remove the EFF-1-1-0.zip from entry field)

Change into unzipped directory, e.g.:

```
C:\Users\userid\> cd %userprofile%\EFF-1-1-0
```

Executing the Installer

Execute the windows installer eff-windows as described in the following sections. Executing it without arguments yields a short usage info:

```
C:\Users\userid\EFF-1-1-0\> .\eff-windows
```

Installation of the EFF Components

The EFF installer `eff-windows` 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 C:\Users\userid and further assume, that the package EFF-1-1-0.zip has been unpacked inside that folder (as described above).

Install help or usage:

Executing `.\eff-windows` displays a summary help screen.

```
Edge and Fog Fabric - Installer and checksum tool v1.1.0

Synopsis: eff-windows [env|help|install|report|upgrade|verify|version] [admin|broker|dart|monitor]

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 `.\eff-windows help` displays an extended help screen.

```
INFO: EFF_ROOT = C:\Users\userid\iotdc taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
Edge and Fog Fabric - Installer and checksum tool v1.1.0

Synopsis: eff-windows [env|help|install|report|upgrade|verify|version] [admin|broker|dart|monitor]


Sample usage(s):

    eff-windows install            # -> install all platform components (dart, broker, admin, and
monitor)
    eff-windows upgrade            # -> upgrades all platform components (broker, admin, and monitor)

    eff-windows broker             # -> installs broker, data flow editor and dsa run-time
    eff-windows dart               # -> installs dart run-time
    eff-windows admin              # -> installs admin

    eff-windows verify             # -> report current settings and SHA512 checksums of components

    eff-windows verify dart        # -> calculate SHA512 checksum of dart run-time

    eff-windows version            # -> report version banner of this tool

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

Notes:

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

Environment Variables:
```

```
    EFF_ROOT should be set to the *absolute* path of the install root (default: $HOME/iotdc)
        (i.e. folders eff_server, parstream, and dart-sdk will be created below that path)
        Current value is: <UNSET>

    EFF_GUI_LOGIN may be set to the *name* of the EFF GUI Admin User (default: effAdmin)
        Current value is: <UNSET>

    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)
        Current value is: <UNSET>

    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)
        Current value is: <UNSET>

    EFF_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>

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

    EFF_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>

    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)
        Current value is: <UNSET>

    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)
        Current value is: <UNSET>

    EFF_INSTALL_LOGS may be set to the *absolute* path of a folder for install logs (default:
C:\Users\userid\EFF-1-1-0)
        (e.g. needed in case the unpacked install components are stored on a read-only medium)
        Current value is: <UNSET>

    EFF_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

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

Copyright:

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

Executing `.\eff-windows version` displays the version.

```
Edge and Fog Fabric - Installer and checksum tool v1.1.0
```

Executing `.\eff-windows env` reports the effective environment variable values and their source.

```
INFO: EFF_ROOT = C:\Users\userid\iotdc taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: Effective environment variable status detected:
REPORT: - 8< - - - - - - 8< - - - - - - 8< - - - - - - 8< - - - - - - 8< - - - - - - 8< - - - - - - 8<
- - - -
REPORT: EFF_ROOT(from platform default): C:\Users\userid\iotdc
REPORT: EFF_GUI_LOGIN(from config): effAdmin
REPORT: EFF_GUI_PHRASE(from config): <empty>; Format: PBKDF2
REPORT: EFF_BROKER_PRIV_KEY_PEM(from config): key.pem
REPORT: EFF_BROKER_FULL_CHAIN_PEM(from config): cert.pem
REPORT: EFF_BROKER_SECURE_PORT(from config): 8443
REPORT: EFF_BROKER_CLEARTEXT_PORT(from config): 8080
REPORT: EFF_BROKER_IS_ALWAYS_OFFLINE(from config): BOOLEAN_FALSE
REPORT: EFF_BROKER_WORKERS(from config): 1
REPORT: EFF_INSTALL_LOGS(from config):
REPORT: EFF_UNATTENDED(from env): BOOLEAN_FALSE
REPORT: EFF_DEBUG: <UNSET>
REPORT: - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - -
- - >8 -
```

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, and Dataflow engine DSLinks.

Executing `.\eff-windows broker`

```
INFO: EFF_ROOT = C:\Users\userid\iotdc taken neither from file, nor from environment variable, but
instead from platform default!
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 VM can execute and expose it's version?
[Y/n]:
INFO: dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 01:50:41 2017) on "windows_x64"
INFO: Sub-task install Dart VM (required by Dart broker) succeeded
INFO: Dart-Broker server configuration template copied successfully from C:\Users\userid\EFF-1-1-
0\Components\server.json
INTERACTION: Reconfigure the Dart-Broker server configuration? [y/n]: y
INFO: Dart-Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFF_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFF_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFF_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFF_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFF_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFF_BROKER_WORKERS to 1
INFO: Dart-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 EFF Dart
installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: Dart-Broker user configuration template copied ...
INTERACTION: Perform Dart-Broker custom user configuration now? [y/n]: y
INFO: Dart-Broker custom user configuration requested, continuing ...
INTERACTION: EFF GUI Admin login is (effAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the user configuration and grant admin rights ...
```

```
INFO: Creation of EFF GUI Admin User effAdmin succeeded with ["User 'effAdmin' was successfully
created."]
INFO: Grant admin rights for EFF GUI Admin User effAdmin succeeded with ["User 'effAdmin' was
sucessfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8<
- - - -
REPORT: Username is: effAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
0449RUwgavCGS7bGHraw5qqC7ZmdC9KZjvjjuFmDPZLqG8=eOG3hfCs2Lv9w7bwNExmiy+otNHkN+yAQlHikfjA1Do=
REPORT: - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the Dart-Broker can execute and expose it's
version? [Y/n]:
INFO: Dart-Broker check succeeded with ['DSA Version: 1.1.2', 'DGLux5 Build: r7712', 'DGLux Server
Build: 1159', 'Verifying Environment...', 'Verifying Configuration...']
INFO: The component EFF Message Broker has been installed
```

Installing the System Administrator:

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

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

Executing `.\eff-windows admin`

```
INFO: EFF_ROOT = C:\Users\userid\iotdc taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: The component EFF System Administrator has been installed
```

Installing the System Monitor:

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

Typing `.\eff-windows monitor`

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

```
INFO: EFF_ROOT = C:\Users\userid\iotdc taken neither from file, nor from environment variable, but
instead from platform default!
INFO: Interactive mode enabled (reason default)
INFO: Sub-task install Dart Broker (required by Admin) ...
INFO: Sub-task install Dart VM (required by Dart broker) ...
INTERACTION: Installed platform dart run-time. Check that the VM can execute and expose it's version?
[Y/n]:
INFO: dart VM check succeeded with Dart VM version: 1.21.1 (Fri Jan 13 01:50:41 2017) on "windows_x64"
INFO: Sub-task install Dart VM (required by Dart broker) succeeded
INFO: Dart-Broker server configuration template copied successfully from C:\Users\userid\EFF-1-1-
0\Components\server.json
```

```
INTERACTION: Reconfigure the Dart-Broker server configuration? [y/n]: y
INFO: Dart-Broker server configuration reconfiguration requested, continuing ...
INFO: Set server json value of certName from EFF_BROKER_FULL_CHAIN_PEM to cert.pem
INFO: Set server json value of certKeyName from EFF_BROKER_PRIV_KEY_PEM to key.pem
INFO: Set server json value of httpsPort from EFF_BROKER_SECURE_PORT 8443
INFO: Set server json value of isAlwaysOffline from EFF_BROKER_IS_ALWAYS_OFFLINE to false
INFO: Set server json value of port from EFF_BROKER_CLEARTEXT_PORT to 8080
INFO: Set server json value of workers from EFF_BROKER_WORKERS to 1
INFO: Dart-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 EFF Dart
installation? [Y/n]:
INFO: Setup kept secure as installed by default, good.
INFO: Dart-Broker user configuration template copied ...
INTERACTION: Perform Dart-Broker custom user configuration now? [y/n]: y
INFO: Dart-Broker custom user configuration requested, continuing ...
INTERACTION: EFF GUI Admin login is (effAdmin). Change? [y/N]:

INTERACTION: Enter Password:
INTERACTION: Enter Password (verify):
INFO: Map the given login credentials into the user configuration and grant admin rights ...
INFO: Creation of EFF GUI Admin User effAdmin succeeded with ["User 'effAdmin' was successfully
created."]
INFO: Grant admin rights for EFF GUI Admin User effAdmin succeeded with ["User 'effAdmin' was
sucessfully granted superuser."]
INFO: Users entry 1/1
REPORT: - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8< - - - - - 8<
- - - -
REPORT: Username is: effAdmin
REPORT: Admin rights granted: true
REPORT: Passphrase in PBKDF2 format:
044m6zE0aCqOE/OyCnIE2yYpfOxH8Y1AtLRi39LP9gP9DU=KE6esQMoTM9gNoXuNT9cUHW7kYV/9Br4NZfDx1p7AEQ=
REPORT: - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - - - - >8 - - -
- - >8 -
INTERACTION: Installed broker and run-time. Check that the Dart-Broker can execute and expose it's
version? [Y/n]:
INFO: Dart-Broker check succeeded with ['DSA Version: 1.1.2', 'DGLux5 Build: r7712', 'DGLux Server
Build: 1159', 'Verifying Environment...', 'Verifying Configuration...']
INFO: Sub-task install Dart Broker (required by Admin) succeeded
INFO: The component EFF System Administrator has been installed
```

Upgrading to EFF version 1.1.0 from an existing version 1.0.x installation

If an existing version 1.0.0 or 1.0.1 is installed and running on a Windows server it is possible to upgrade to the version 1.1.0. It is highly recommended performing a backup of the host prior to proceeding with the upgrade.

Assuming that the EFF version 1.0.x is already installed and running. Here are the pre-requisites to upgrading:

- Download the EFF 1.1.0 package
- Stop the running message broker (see below)

Perform the following steps to upgrade in a Command Window:

Prepare upgrade like a fresh install: Unpack the EFF-1-1-0.zip accordingly and change current working directory to EFF-1-1-0 folder all as described above in the install section (cf. there if unzip not present or something does not work):

```
C:\Users\userid\> unzip –q EFF-1-1-0.zip
C:\Users\userid\> cd %userprofile%\EFF-1-1-0
C:\Users\userid\EFF-1-1-0\>
```

Note directory of the destination as the default version 1.0.x installation, for example:

```
C:\Users\userid\EFF-1-1-0\> set EFF_ROOT=C:\cisco\iotdc\
```

Note: If your installation path differs from this default path, please adapt the corresponding paths in the following commands. If it is already %userprofile%\iotdc no setting of EFF_ROOT variable is required, but then below commands must be adapted to function properly.

Stop the broker:

```
C:\> %EFF_ROOT%\dart-sdk\bin\dart %EFF_ROOT%\eff_server\bin\daemon.dart stop
```

Upgrade the components:

```
C:\Users\userid\EFF-1-1-0\> .\eff-windows upgrade
```

Start the broker:

```
C:\> %EFF_ROOT%\dart-sdk\bin\dart %EFF_ROOT%\eff_server\bin\daemon.dart start
```

Starting and Stopping EFF Message Broker

Note: Assuming the default installation path root stored in variable EFF_ROOT in the following examples.

Start the broker:

```
C:\> %EFF_ROOT%\dart-sdk\bin\dart %EFF_ROOT%\eff_server\bin\daemon.dart start
```

To stop the broker (when demo or test complete):

```
C:\> %EFF_ROOT%\dart-sdk\bin\dart %EFF_ROOT%\eff_server\bin\daemon.dart stop
```

Connecting to the EFF System Components

All the EFF tools require a username and login for access. The administrator can add additional users in the System Administrator after the first login.

Application	Insecure Port (if supported)	Secure Port
EFF Data Flow Editor	`http://[Server IP Address]/dataflow.html`	`https://[Server IP Address]/dataflow.html`

server.json Configuration options

Example server.json configuration file.


```
{
    "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": "NO",
    "downstreamName": "downstream",
    "enableCertificateGeneration": true,
    "enableGit": false,
    "enableIPv6": false,
    "enableSingleSignOnServer": false,
    "enableUptimeChecker": true,
    "formatDg5": false,
    "generatedCertificateSubject": "/C=US/ST=California/L=Oakland/O=DGLogik Inc./OU=Customers/CN=*",
    "guestLoginRedirectPath": "/assets/",
    "hooks": {},
    "host": "0.0.0.0",
    "httpPathClassification": {},
    "httpsPort": 8443,
    "isAlwaysOffline": false,
    "javaRuntimeManagerVmFlags": [],
    "keepCustomAssets": true,
    "linkConfig": {},
    "linkManagerEnvironment": {},
    "linkRepositoryUrl": "https://dsa.s3.amazonaws.com/links/links.json",
    "logRotationInterval": 0,
    "loggers": [],
    "loginRedirectPath": "/",
    "observe": false,
    "passwordHasherIterations": 1000,
    "passwordHasherKeyLength": 32,
    "port": 8080,
    "proxies": {},
    "quarantine": false,
    "runBrokerInMain": true,
    "runPortChecks": true,
    "serverLogLevel": "INFO",
    "serverVmFlags": [],
    "ssoProviderUrl": null,
    "static": {
        "/.well-known": "/opt/cisco/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
}
```

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	For single-core machines, this is 1,	

	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.	for other devices, this is 2.	

static	Configures a static directory mapping. This is used to serve files and directories on the server.
	Example:
	{
	"/static": "/srv/http/static"
	}
		{"/.well-known": "/path/to/dsa/dglux-server/.well-known"}	

defaultPermission	Default permission setting for the root node. When this value is null, permissions are disabled, and everything has the config permission.		
allowAllLinks	When the value is true, all incoming DSLink connections will be accepted to /downstream. When the value is false, an incoming DSLink without proper authentication will be rejected unless quarantine is enabled.	true	
quarantine	** This setting has no effect when allowAllLinks is true ** When the value is true, a new incoming DSLink without a token will be put in /sys/quarantine. A quarantined DSLink can only work as a responder. Use the /sys/quarantine/authorize to move a quarantined DSlink to /downstream.	false	
isAlwaysOffline	Indicates that a server is expected to never have a full internet connection.	false	
useDartRuntimeManager	When the value is true, the Dart Runtime Manager is used for Dart DSLinks. The Dart runtime manager reduces resource consumption by merging Dart	false	

	DSLinks into a single process.		
useJavaRuntimeManager	When the value is true, the Java Runtime Manager is used for Java DSLinks. The Java runtime manager reduces resource consumption by merging Java DSLinks into a single process.	false	
guestLoginRedirectPath	Determines the URI that a user is redirected to when login is complete.	/	
authType	Determines the authentication provider to use.	file	
twoFactorAuth	Determines the two factor authentication provider to use. Supported Two-Factor Authentication Providers • none: Don't enable two factor authentication. • duo: Duo Two-Factor Authentication	none	
enableIPv6	Toggles support for IPv6 connections	false	
keepCustomAssets	When the value is true, custom assets in www/assets are kept upon updating 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.