



Installation Guide for Windows

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

Revised: August 25, 2017

These release notes provide an overview to Cisco IoT DataConnect – Edge and Fog Fabric version 1.0.1 and help you understand the product at a high level.

Contents

Introduction.....	2
Features and Functions	2
The Edge and Fog Fabric Components.....	3
Hardware Requirements.....	4
EFF Components Protocols and Ports	4
Licensing installation and requests	5
Required Libraries for Installation on Windows	5
Installation the EFF Broker, Dataflow Editor and links	5
Changing the EFF GUI Admin User password	6
Upgrading to EFF version 1.0.1 from an existing version 1.0 installation.....	6
Starting and Stopping EFF Message Broker.....	7
Connecting to the EFF System Components	7
server.json Configuration options.....	8
Obtaining documentation and submitting a service request	12

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	<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>
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	<p>DQL Link – DSA Query Language</p> <p>System Link – System Information</p> <p>ParStream Link v2.3 – ParStream Historian Database</p>
Smart License Agent Tool for Nodes	The Smart License Agent client that allows system users to manage license registration for Node Product IDs
Smart License Agent Tool for Devices	The Smart License Agent client that allows system users to manage license registration for Device Product IDs



Hardware Requirements

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 the EFF Broker, Dataflow Editor and links

The EFF Broker and links are compressed in the complete EFF software package. The following steps will install the EFF Windows Broker and links.

1. Create a destination folder where the EFF Broker and components will reside, e.g.
C:\cisco\iotdc\
2. Unpack the archive Components\eff-broker.zip into that folder e.g.
C:\cisco\iotdc\
sample execution with CMD:

```
$> unzip Components\eff-broker.zip -d C:\cisco\iotdc\
```
3. Edit the broker configuration file (using our sample folder):
C:\cisco\iotdc\eff_server\server.json
in a plain text editor like e.g. notepad and edit the following values:
 - 'httpsPort': 8443 (if the secure port shall be 8443)
 - 'certPassord': "" (if using https must be non-null value)
 - 'port': 8080 (if insecure port shall be 8080) or 0 or negative to disable
 - 'isAlwaysOffline': false (if internet shall be / is expected to be accessible to that machine)
 - 'workers': 2 (on a machine with say more than 6+ cores, or 1 otherwise)
4. Rename the folder (in our sample) C:\cisco\iotdc\dart-sdk-64-bit-windows\
as follows:

```
$> move C:\cisco\iotdc\dart-sdk-64-bit-windows C:\cisco\iotdc\dart-sdk
```

Changing the EFF GUI Admin User password

5. Change the password phrase of the EFF GUI Admin User from the command line, ... assuming the pass phrase would be "PW" (please choose a longer one!):

```
$> cd C:\cisco\iotdc\eff_server
$> ..\dart-sdk\bin\dart.exe bin\users.dart change-password -u effAdmin -p 'PW'
```

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 Windows 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 Windows
- Stop the running message broker (see below)

Perform the following steps to upgrade in a Command Window:

Change directory to the same destination as the current version 1.0 installation, for example:

```
$> cd C:\cisco\iotdc\
```

Note: If your installation path differs from this default path, please adapt the corresponding paths in the following commands.

Stop the broker:

```
$> cd C:\cisco\iotdc\eff_server
$> ..\dart-sdk\bin\dart.exe bin\daemon.dart stop
```

Unpack the archives:

```
$> unzip -o Components\eff-dataflow-1-0-1-patch.zip -d
C:\cisco\iotdc\eff_server\www
$> unzip -o Components\eff-job-engine-1-0-1-patch.zip -d
C:\cisco\iotdc\eff_server\dslinks\dataflow
$> unzip -o Components\parstream-dsa-package-2.4.0-binary.zip -d
C:\cisco\iotdc\eff_server\dslinks\parstream-dsa-link
```



Start the broker:

```
$> cd C:\cisco\iotdc\eff_server  
$> ..\dart-sdk\bin\dart.exe bin\daemon.dart start
```

Starting and Stopping EFF Message Broker

Note: Assuming the default installation path of C:\cisco\iotdc\eff_server in the following examples.

Start the broker:

```
$> cd C:\cisco\iotdc\eff_server  
$> ..\dart-sdk\bin\dart.exe bin\daemon.dart start
```

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

```
$> cd C:\cisco\iotdc\eff_server  
$> ..\dart-sdk\bin\dart.exe 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": "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  
}
```

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	



	DSLlinks 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 <ul style="list-style-type: none">• none: Don't enable two factor authentication.• duo: Duo Two-Factor Authentication	none	
enableIPv6	Toggles support for IPv6 connections	false	
keepCustomAssets	When the value is true, custom assets in www/assets are kept upon updating 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. 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.