Installation Guide

/ Identity Edge Controller 6.5.0

Latest update: 6.5

Lana Frost

ForgeRock AS.
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Guide to installing and upgrading ForgeRock® Identity Edge Controller software.

Table of Contents

Preface

This guide shows you how to install, upgrade, and remove IEC software components.

Chapter 1

Installing and Configuring IEC

This chapter covers the tasks required to prepare, install and run IEC software.

In addition to this documentation, ForgeRock provides a Docker training environment that you can use to quickly get started with the IEC installation and configuration. Follow the README in the GitHub project for instructions on using the training environment.

Before You Install

Before you install the IEC software you must install and configure a number of prerequisite components. For an understanding of how these components fit into an IoT deployment, see "IEC Components" in the *Getting Started*. The prerequisite steps are described in the following sections:

Installing ForgeRock Access Management (AM)

IEC requires AM to be installed and running. For instructions on installing AM for demonstration purposes, see the AM Quick Start Guide. Make sure that you have successfully logged in to the AM console before you continue.

Important

- To complete the procedures in this chapter, you must use Java 8. There are known issues with Java 11. See the following Knowledge Base article for more information.
- In production deployments, use HTTPS to protect network traffic. For information about securing AM, see Securing Communications in the *AM Installation Guide*. If you enable HTTPS, the IEC service must be able to access the certificate. To ensure this, place the certificate in the certificate store of the Operating System.

Installing the IEC AM Plugin

The IEC AM Plugin adds IoT-specific functionality to AM. The pugin provides a single, secure communication point for the IEC Service and allows the IEC Service to perform tasks such as registering edge nodes and retrieving OAuth2 tokens.

Install the plugin as follows:

1. Download the AM Plugin for IEC from the Edge Security section on the ForgeRock BackStage download site, and extract the zip archive to a new directory. For example:

```
mkdir ~/Downloads/am-iec-plugin
tar -xzvf iec-am-plugin-6.5.0.tgz -C ~/Downloads/am-iec-plugin
```

2. Copy the plugin and configuration to the AM web server. For example, if you are using Tomcat with its home directory stored in the variable `TOMCAT_HOME` and AM deployed to `${TOMCAT_HOME}/webapps/openam`, copy the plugin and configuration as follows:

```
cp ~/Downloads/am-iec-plugin/am-iec-plugin-6.5.0.jar ${TOMCAT_HOME}/webapps/openam/WEB-INF/lib/
cp ~/Downloads/am-iec-plugin/config/* ${TOMCAT_HOME}/webapps/openam/config/auth/default
```

3. Restart the Tomcat server so that the plugin and configuration are taken into account:

```
${TOMCAT_HOME}/bin/shutdown.sh
${TOMCAT_HOME}/bin/startup.sh
```

Configuring ForgeRock Directory Services (DS) For IoT Identities

AM stores identities in a DS repository. IoT identities are similar to user identities in AM. However, they have additional attributes and are stored alongside OAuth2 Clients.

To enable AM to manage IoT identities, you need to modify the embedded DS configuration as follows. Note that the `bindPassword` for the embedded DS server is the same as the password you configured for the `amadmin` user when you set up AM:

1. Configure DS to accept multiple structural object classes:

```
~/openam/opends/bin/dsconfig \
 set-global-configuration-prop \
 --hostname openam.example.com \
 --port 4444 \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --set single-structural-objectclass-behavior:accept \
 --no-prompt \
 --trustAll
```

2. Add the schema for an IoT device to DS. You will find the schema file in the `ds` directory that was created when you unzipped the AM IEC plugin:

```
~/openam/opends/bin/ldapmodify \
 --hostname openam.example.com \
 --port 50389 \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 ~/Downloads/am-iec-plugin/ds/iot-device.ldif
# MODIFY operation successful for DN cn=schema
```

Configuring AM for IoT

IEC communicates with AM through an IoT-enabled realm. This section describes how to create a new realm and configure it to store identities in the Application Store, and how to configure the OAuth 2.0 authorization service for the IoT realm.

In an evaluation deployment, use the default values unless instructed otherwise. In a production deployment, refer to the AM documentation for the appropriate values for your use case.

To Create and Configure the IoT Realm

1. In the AM admin console, select Realms > New Realm and name the realm `edge`.

   Leave all other fields blank and select Create.

2. Configure the `edge` realm so that identities are stored in the Application Store (AgentService), alongside OAuth2 clients.

   In the `edge` realm, select Identity Stores > embedded and change the following settings:

   On the Server Settings tab

   Change LDAP Organization DN to `ou=OrganizationConfig,ou=1.0,ou=AgentService,ou=services,o=edge,ou=services,dc=openam,dc=forgerock,dc=org` then select Load Schema and Save Changes.

   On the User Configuration tab

   - Change LDAP Users Search Attribute to `ou`
   - Change LDAP Users Search Filter to `(objectclass=sunservicecomponent)`
   - To LDAP User Object Class add the following object classes:

     ```
     sunservicecomponent
     forgerockIotDevice
     ```

   - To LDAP User Attributes add the following attributes:

     ```
     sunkeyvalue
     sunserviceID
     edgeControllerIdentifier
     edgeControllerVersion
     edgeControllerPlatform
     edgeNodeRegistrationStatus
     edgeNodeRegistrationTime
     edgeNodeRegistrationJwk
     edgeNodeEnvironmentData
     edgeClientIdentifier
     edgeNodeConfig
     edgeNodeUserConfig
     edgeNodeType
     edgeNodeDeviceCode
     edgeNodePairedUser
     ```

   - Change LDAP People Container Naming Attribute to `default`

- Select Load Schema then select Save Changes.

On the Authentication Configuration tab

Change Authentication Naming Attribute to `ou` then select Load Schema and Save Changes.

On the Persistent Search Controls tab

Change Persistent Search Base DN to `ou=OrganizationConfig,ou=1.0,ou=AgentService,ou=services,o=edge,ou=services,dc=openam,dc=forgerock,dc=org` then select Load Schema and Save Changes.

3. Configure the realm for OAuth2.

   In the `edge` realm, select Dashboard > Configure OAuth Provider > Configure OpenID Connect.

   Accept the default values and select Create.

4. Enable dynamic profile creation.

   You can create edge node identity profiles dynamically or manually, before they are registered. By default, IEC uses dynamic profile creation.

   In the `edge` realm, select Authentication > Settings > User Profile.

   Change the User Profile field from Required to Dynamic, then select Save Changes.

5. Set the Verification URL for device codes.

   For device pairing, users are sent to a verification URL with a specified code. To set that URL, navigate to Services > OAuth2 Provider > Device flow and change the Verification URL to `http://openam.example.com:8080/openam/oauth2/realms/root/realms/edge/device/user?nonce=0`.

To Add the IEC Service

Edge node tasks such as registration and token retrieval are achieved through scripts in AM. To install the default registration and command scripts, you must add the IEC Service to the `edge` realm. Adding the service also configures the default authentication modules and OAuth2 group.

Add the IEC Service to the realm as follows:

1. In the `edge` realm, select Services > Add a Service.
2. Select IEC Service as the Service Type.
3. Enter the following sample values and select Create:
   - ID Token Issuer: `edge-device` [1]

[1] If the value provided contains a colon character (**:**), it must be a valid URI. For more information, see StringOrURI in the *JSON Web Token specification*.

- ID Token Audience: `openam.example.com` [1]
- ID Token Client Secret: `letmein`
- Challenge Signing Key: `es256test`

  This signing key is one of the default test keys available in AM. In a production deployment, you should generate a new challenge signing key. The challenge signing key must be an ECDSA P-256 asymmetric key. For information on creating and adding keys to the AM keystore, see Setting Up Keys and Keystores in the *AM Maintenance Guide*.

  If you add a new key here, take note of the key alias.

Installing the Edge Identity Manager

The Edge Identity Manager is a basic User Interface to AM for viewing and managing device identities.

Before you install the Edge Identity Manager, note the following:

- The installation directory of AM must be `openam`.
- The IoT realm that you set up in "To Create and Configure the IoT Realm" *must* be named `edge`.

Install the Edge Identity Manager as follows:

1. Download the Edge Identity Manager WAR file from the Edge Security section on the ForgeRock BackStage download site.
2. Copy the WAR file to the same server that is running the AM web server. For example:

   ```
   cp ~/Downloads/edge-identity-manager-6.5.0.war ${TOMCAT_HOME}/webapps/identitymanager.war
   ```

3. Access the Edge Identity Manager at the context path `/identitymanager`, for example `http://openam.example.com:8080/identitymanager`.

   For the Edge Identity Manager to have sufficient privileges to access AM, an AM admin user must be logged into the AM Admin Console in the same browser session.

Installing the IEC Service

The IEC Service runs on a device on the local network and provides secure communications between client applications and AM.

Before you install the IEC service on your device, ensure that the device can communicate with the AM instance. To test the connection to AM, run the following REST request:

```
curl \
 --request GET \
 http://openam.example.com:8080/openam/json/serverinfo/*
{
  "_id": "*",
  "_rev": "1561602150",
  "domains": [
    "openam.example.com"
  ],
  "protectedUserAttributes": [],
  "cookieName": "iPlanetDirectoryPro",
  "secureCookie": false,
  "forgotPassword": "false",
  "forgotUsername": "false",
  "kbaEnabled": "false",
  "selfRegistration": "false",
  "lang": "en-US",
  "successfulUserRegistrationDestination": "default",
  "socialImplementations": [],
  "referralsEnabled": "false",
  "zeroPageLogin": {
    "enabled": false,
    "refererWhitelist": [],
    "allowedWithoutReferer": true
  },
  "realm": "/",
  "xuiUserSessionValidationEnabled": true,
  "fileBasedConfiguration": false
}
```

If the device can connect to the AM instance, output similar to that above is returned.

Install the IEC Service as follows:

1. Download the IEC Service binary from the Edge Security section on the ForgeRock BackStage download site.

   Choose the binary specific to your device operating system. Binaries are provided for the following operating systems:

   - ARM7
   - ARM8 - RichOS
   - ARM8 - OP-TEE

     This ARM TrustZone-enabled version of the IEC Service provides secure storage on devices that support OP-TEE.

   - x86_64

2. Unpack the tarball to a temporary directory:

   ```
   mkdir ~/Downloads/iec-service
   tar -xzvf ~/Downloads/iec-service-<OS>-6.5.0.tgz -C ~/Downloads/iec-service
   ```

3. In any text editor, open the IEC Service configuration file (`~/Downloads/iec-service/iec-config.json`) and set at least the following properties to match the Service configuration you set in "To Add the IEC Service".

```
{
  "iec_configuration": {
    ...
    "id_token_config.audience": "openam.example.com",
    ...
  },
  "am_configuration": {
    "url": "http://openam.example.com/openam",
    ...
  }...
```

Note

The value provided for *id_token_config.subject* is used as the name of the IEC identity and must be unique within the AM realm. [1]

4. Run the install script:

```
~/Downloads/iec-service/install.sh
iec util: Initialising service
iec util: Finished service initialisation
Created symlink /etc/systemd/system/multi-user.target.wants/iec.service - /lib/systemd/system/
iec.service.
```

The IEC Service is now installed and running as a daemon.

For a list of the files in the tarball, and where they are installed by the install script, see "IEC Service File Layout".

Note

If the target system is a Docker image or if the system does not support **systemctl**, you will see the following error:

```
"Failed to connect to bus: No such file or directory"
```

In this case, start the IEC Service manually by running the following command:

```
/opt/forgerock/iec/bin/iecservice &
```

The IEC Service will repeatedly attempt to register itself with AM. A registered IEC Service appears as an *identity* in the `edge` realm in the AM instance. You can check that the service has been registed in the AM admin console or in the Edge Identity Manager UI.

The following table lists the files in the IEC tarball, or created by the setup, and where these files are installed:

IEC Service File Layout

File	Installed To	Description
`iecservice`	`/opt/forgerock/iec/bin`	Main IEC Service executable
`iec.service`	`/lib/systemd/system`	Systemd unit file
`lib.*`	`/opt/forgerock/iec/lib`	3rd party libraries for IEC Service
`*.ta`	`/lib/optee_armtz`	IEC trusted application in default OP-TEE TA directory (OP-TEE installation only)
`iecutil`	–	IEC Utility executable used at install time and removed after system setup
`install.sh`	–	Bash script used to perform installation and removed after system setup
`iec-config.json`	–	IEC Service configuration; used at install time and removed after system setup
`iec-service.db`	`/var/opt/forgerock/iec`	IEC Service database, created at install time (RichOS installation only)
`TA secure storage`	`/data/tee`	TA secure storage files, created at install time (OP-TEE installation only)
`iecservice.log`	`/var/opt/forgerock/iec`	IEC Service log file

In a production deployment, after the IEC Service has been installed successfully, you should delete the files extracted from the tarball (the `~/Downloads/iec-service/` directory, in our example).

Managing the IEC Service

The IEC Service will automatically startup when the device is powered up. The following commands are useful to manage the service:

- To stop the service:

```
sudo systemctl stop iec.service
```

  If you stop the service, it is automatically restarted when the device is rebooted.

- To start the service:

```
sudo systemctl start iec.service
```

- To obtain the status of the service:

```
sudo systemctl status iec.service
iec.service - ForgeRock IEC Service
  Loaded: loaded (/lib/systemd/system/iec.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2019-04-11 21:11:10 SAST; 10s ago
Main PID: 11803 (iecservice)
   Tasks: 10 (limit: 4915)
  Memory: 6.4M
  CGroup: /system.slice/iec.service
    -11803 /opt/forgerock/iec/bin/iecservice

 Apr 11 21:11:10 lana-VirtualBox systemd[1]: Started ForgeRock IEC Service.
```

- To disable the service:

```
sudo systemctl disable iec.service
```

  Disabling the service stops it and the service does not restart when the device is rebooted. The service will only restart when it is enabled.

- To enable the service:

```
sudo systemctl enable iec.service
```

Installing a Client Application

When the IEC service is registered with AM, you can run client applications that use the IEC SDK.

To Install the IEC SDK

First, install the IEC SDK as follows:

1. Download the IEC SDK from the Edge Security section on the ForgeRock BackStage download site.

   Choose the binary specific to your device operating system. Binaries are provided for the following operating systems:

   - ARM7
   - ARM8
   - x86_64

2. Unpack the SDK tarball to an accessible directory, for example:

```
mkdir ~/forgerock
tar -xzvf ~/Downloads/iec-sdk-<OS>-6.5.0.tgz -C ~/forgerock
```

3. Add the extracted `lib` directory to your path. For example:

```
export LD_LIBRARY_PATH=~/forgerock/lib
export DYLD_LIBRARY_PATH=~/forgerock/lib
```

The SDK is now installed.

To Test the IEC SDK

The SDK binary includes a number of example applications.

To test the SDK, run the `simpleclient` example application as follows:

1. Copy the SDK configuration file (`sdk-config.json`) to the directory containing the `simpleclient` application:

```
cd ~/forgerock/examples/simpleclient
cp ~/forgerock/sdk-config.json .
```

2. Edit the SDK configuration file to specify the IP address on which the SDK runs. For example, if you are setting this up in the training environment, edit the file as follows:

```
zmq_client.endpoint: tcp://172.16.0.11:5556
```

Note

The value provided for *client_configuration/id* is used as the name of the client identity and must be unique within the AM realm. [1]

3. Use the IEC Utility (**iecutil**) to initialize the SDK:

```
~/forgerock/iecutil -file sdk-config.json -initialise sdk
iec util: Initialising sdk
iec util: Finished sdk initialisation
```

Note

If you change the configuration and need to reinitialize the SDK, remove the `iec-sdk.db` in the application directory, then run the initialization again. For example:

```
cd ~/forgerock/examples/simpleclient
rm iec-sdk.db
~/forgerock/iecutil -file sdk-config.json -initialise sdk
```

4. Run the example application:

```
./simpleclient
*** Running simpleclient

Setting attributes ... Done
Initialising sdk... Done
Registering device (id: Narwhal)... Done
Requesting configuration for device (id: Narwhal)... Done
Received configuration: {  }
Requesting tokens for device (id: Narwhal)... Done
Received tokens: {
 "access_token": "zcKlIqmDIgD70J7M0yukH3OmfbM",
 "id_token": "eyJ0e...Wgw",
 "token_type": "Bearer",
 "expires_in": "3599"
 }
Executing 'Hello World' custom command... Custom command request failed: Error: am_error, Description:
 no script found for command, URL:
```

The client application should register successfully and retrieve OAuth2 tokens.

Confirm that the registration has been successful by checking that the `simpleclient` identities were created in the AM Admin console or in the Edge Identity Manager. When you have successfully initialized the client application, delete the configuration file.

Use the following for help writing your own client applications:

- IEC SDK API

```
${SDK_DIR}/include/libiecclient.h
```

- Client example applications

```
${SDK_DIR}/examples/
```

Note

The IEC SDK requires the Sodium and ZMQ libraries for compilation. These are provided with the IEC SDK distribution, for example:

```
libs="-liecclient -lsodium -lzmq"
gcc ${ex_dir}/${ex}.c -I${SDK_DIR}/include -L${SDK_DIR}/lib ${libs}
```

IEC Glossary

Access Management (AM)	ForgeRock software (part of the ForgeRock Identity Platform) that provides access and identity management.
client	An edge node type representing a client application that uses the IEC SDK.
constrained device	A device that does not have the ability to connect securely across wide-area networks, due to cost and/or physical constraints. See RFC 7228.
device	An edge node type representing a physical device that can be onboarded via a client node.
Directory Services (DS)	ForgeRock software that is part of the ForgeRock Identity Platform and provides storage for identities and configuration.
edge	Industry term for the geographic distribution of IoT devices. *Edge computing* enables a connected device to process data closer to where it is created (on the *edge*).
edge gateway	Hardware and software deployed at the edge, through which devices communicate.
Edge Identity Manager	ForgeRock software that provides a User Interface to AM for viewing and managing device identities.
edge node	A physical or virtual object that exists at the edge and benefits from having an identity. Examples of edge nodes include a device, the IEC Service or a client application.

Identity Edge Controller (IEC)	ForgeRock software consisting of multiple components that securely provide devices with identity.
IEC AM Plugin	ForgeRock software plugin that adds IoT specific functionality to AM.
IEC SDK	ForgeRock client library that provides an API for client applications to invoke AM functionality via the IEC Service.
IEC Service	ForgeRock software that runs on the edge gateway and provides secure communication between client applications and AM.
IEC Utility	ForgeRock software used when installing the IEC Service or IEC SDK to configure the components.
OP-TEE	Open source implementation of the GlobalPlatform Trusted Execution Environment (TEE) specification.
Rich Execution Environment (REE)	GlobalPlatform term for the environment in which the user-facing operating system runs.
Rich OS	Operating system running in the Rich Execution Environment (REE), typically Linux.
Trusted Application (TA)	An application that can run in the Trusted Execution Environment (TEE).
Trusted Execution Environment (TEE)	GlobalPlatform term for a secure area of the main processor of a device that ensures data is stored and processed in an isolated and trusted environment.