eformsign

Installation Guide

eformsign v1.0

Document Information

Title: eformsign Installation Guide

Publication Date: 2020-05-31

Software Version: eformsign v1.0

Copyright Notice

Restricted Rights Legend

Trademarks

Open Source Software Disclaimer

This product includes various open source software developed and/or licensed by "OpenSSL", "RSA Data Security, Inc.", "Apache Foundation", and "JavaScript Object Notation (JSON)".

Table of Contents

List of Figures

List of Tables

About this Document

Intended Audience

This document is intended for users who want to install eformsign®(hereafter eformsign).

Required Knowledge

To successfully install eformsign, users must have a basic understanding of the following:

- Linux
- Web server, web application server (WAS), and DBMS

Document Scope

This guide only contains information needed for installing eformsign in your system. For information about how to use, manage, and operate eformsign, refer to **"eformsign User Guide"**.

Document Overview

This guide consists of 5 chapters.

Descriptions of each are as follows:

- Chapter 1: Installation Overview

  Describes the key features of eformsign and its system requirements.

- Chapter 2: Pre-installation Task 1 - Java Installation and Environment Variable Configuration

  Describes how to install Java and configure Java environment variables.

- Chapter 3: Pre-installation Task 2 - Server Environment Configuration

  Describes the process of installing server-side software (web server, WAS, and DB) required for installing eformsign.

- Chapter 4: eformsign Installation and Configuration

  Describes how to install eformsign and configure properties.

- Chapter 5: Company Registration

  Describes how to register your company before using eformsign.

Conventions

Convention	Meaning
<AaBbCc123>	Filename of a program source code
<Ctrl>+C	Hold the Control key and press C
Boldface	Emphasis of buttons, menu names, and input fields in UI
" " (Double quotation marks)	Reference to chapters or sections in the guide, or to other related documentation
`Hyperlink`	Email account or website to other chapters or sections
>	Progress order of menus
+----	Files or directories exist below
|----	Files or directories do not exist below
Note	Reference or note
[Figure 1.1]	Figure name
[Table 1.1]	Table name
`AaBbCc123`	Command, execution result, or example code
[ ]	Optional argument
|	Selective argument

Related Documents

Document	Description
eformsign User Guide	Describes how to use, manage, and operate eformsign.

Contact Information

Korea

FORCS Co. Ltd (HQ)
646, Nonhyeon-ro, Gangnam-gu, Seoul,
Republic of Korea, 06106
Tel: +82 2 6188 8200
Email: `eformsign@forcs.com`
Web: https://www.eformsign.com

Singapore

FORCS Singapore Pte Ltd
8 Shenton Way,
#04-01 AXA Tower,
Singapore, 068811
Tel: +65 6100 2007
Email: `singapore@forcs.com`
Web: http://www.forcs.com/en/

Japan

FORCS Japan
Tokyo-to, Shibuya-ku,
Sendagaya, 4 Chome-11-9,
#402, 151-0051, Japan
Email: `global@forcs.com`
Web: http://www.forcs.com/jp/

Europe

FORCS Europe
Cours Saint-Michel 30B,
Sint-Michielswarande,
1040 Etterbeek, Belgium
Email: singapore@forcs.com
Web: http://www.forcs.com/en/

Chapter 1. Installation Overview

This chapter describes the key features of eformsign and its system requirements.

1.1. System Module Configuration

eformsign consists of 5 modules: Home (front-end) and 4 server-side modules (Service server, Daemon server, Iam server, and DB).

[Table 1.1] eformsign Module Configuration

Module Name	Purpose	Remark
Home (front-end)	Displaying UI screens on the front-end	Installed on the web server and displays the front-end to users.
Service server	Handling synchronous user requests	Installed on the WAS and handles user requests.
Daemon server	Handling asynchronous user requests	Installed on the WAS and handles asynchronous batch tasks and scheduling requests such as email notification, bulk creation of documents, and generation of PDF documents.
Iam server	Managing accounts	Manages user accounts in the Oauth method.
DB	Storing collected data	The unified database used by the above 3 servers for storing and managing data. It is utilized for performing full-text searches of created documents.

Note

Installation and configuration of the web server, WAS, and DBMS are prerequisites for installing and using eformsign.

1.2. Installation Methods

eformsign supports both monolithic and distributed methods of server configuration according to the number of users. The monolithic method has the advantage of minimizing server construction cost, but has the disadvantage of not being able to perform fail-overs when error occurs.

When using the distributed method, the DB and WAS can be configured on the same server, or can be configured on separate servers for stability. Auto scaling is supported to automatically scale up or down according to user count and traffic size.

Note

For inquiry on how to construct servers according to user count, contact us at eformsign@forcs.com.

1.3. Minimum and Recommended System Requirements

The following tables describe the minimum and recommended requirements (hardware and software) for installing eformsign.

[Table 1.2] Minimum and Recommended Hardware Requirements

	Minimum	Recommended
CPU	4 Core Intel Xeon E5 2620 V3 or higher	8 Core Intel Xeon E5 2620 V3 or higher
Memory	8G	16G
Storage Device	HDD 1TB	SSD 1TB

[Table 1.3] Minimum and Recommended Software Requirements

	Minimum	Recommended
OS	Debian 10, Red Hat Enterprise Linux (RHEL) 7, or CentOS 7	Debian 10, RHEL 7, or CentOS 7
WAS	Any WAS system that supports JDK 1.8 or higher	Tomcat 8.5
Web Server	All web servers	nginx 1.14.2
DB		CouchDB 3.0

Note

When installing the DB on multiple machines, there is no restriction on the OS of the WAS and web server, but the OS of the DB server must be one of Debian, RHEL, or CentOS.

Also, the domain must be created and an SSL certificate must be installed.

Chapter 2. Pre-installation Task 1: Java Installation and Environment Variable Configuration

This chapter describes how to install Java and configure the environment variable as part of the pre-installation tasks for installing eformsign.

2.1. Installing Java

Java 8 installation is required before installing eformsign as it runs on Java 8.

The following describes how to install Java for each OS.

- RHEL 7

```
$ sudo -i
$ yum install java-1.8.0-openjdk -y
```

- CentOS 7

```
$ sudo -i
$ yum install -y wget
$ yum install java-1.8.0-openjdk-devel.x86_64 -y
```

- Debian 10

  1. Update the package list and then add repositories and install dependencies.

```
$ sudo -i
$ apt update
$ apt install apt-transport-https ca-certificates wget dirmngr gnupg
software-properties-common
```

Note

In Debian, the **apt** command can be used to automatically resolve dependency problems between packages. Use the **apt** (short for advanced package manager) command to automatically install and upgrade packages.

---

2. Add the AdoptOpenJDK APT repository.

```
$ wget -qO - https://adoptopenjdk.jfrog.io/adoptopenjdk/api/gpg/key/public
 | sudo apt-key add -
$ add-apt-repository --yes https://adoptopenjdk.jfrog.io/adoptopenjdk/deb/
```

3. Update and install Java 8.

```
$ apt update
$ apt install adoptopenjdk-8-hotspot
```

2.2. Verifying Java Installation

After installing Java, verify whether it's been installed properly by executing the following command.

```
$ java -version
```

If the following message is displayed, then Java is properly installed.

```
Openjdk version "1.8.0_242"
OpenJDK Runtime Environment (Build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (Build 25.242-b08, mixed mode)
```

Note

Messages can differ for each OS. If a message such as the one above is displayed, then the Java environment variable does not need to be configured. However, if such a message is not displayed, then the Java environment variable must be configured.

2.3. Configuring the Java Environment Variable

Configure the JAVA_HOME environment variable by OS as follows.

- RHEL 7

```
$ alternatives --config java
$ export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-11.0.1.13-3.el7_6.x86_64
$ echo $JAVA_HOME
```

- CentOS 7 / Debian 10

1. Execute the following command.

```
$ update-alternatives --config java
```

2. Add the following text.

```
$ vi /etc/environment
JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre"
```

3. Execute the following command.

```
$ source /etc/environment
$ echo $JAVA_HOME
```

Note

The method for configuring JAVA_HOME is the same in CentOS 7 and Debian 10.

Chapter 3. Pre-installation Task 2: Server Environment Configuration

This chapter describes how to install server-side software as part of the pre-installation tasks for installing eformsign.

The installation is command-based and proceeds in the order of web server, WAS, and DB. The installation order is the same for all OS.

[Figure 3.1] Installation Order

3.1. Installing the Web Server

This section describes how to install the web server and verify its installation.

eformsign uses nginx as the web server.

Note

The recommended nginx version is 1.14.2.

3.1.1. Installing nginx

The nginx installation process is as follows.

1. Create a repository.

   - RHEL 7

```
$ vi /etc/yum.repos.d/nginx.repo
[nginx]
name=nginx repo
baseurl=https://nginx.org/packages/rhel/7/$basearch/
```

```
gpgcheck=0
enabled=1
```

- CentOS 7

```
$ vi /etc/yum.repos.d/nginx.repo

[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/7/$basearch/
gpgcheck=0
enabled=1
```

Note

Create a repository only in RHEL 7 and CentOS 7.

---

2. Install nginx.

- RHEL 7 / CentOS 7

```
$ yum install nginx -y
```

- Debian 10

```
$ apt install nginx -y
```

Note

RHEL and CentOS (both are Red Hat-based) both use the **yum** command, and Debian (in which Ubuntu is derived from) uses the **apt** command.

---

3. Execute nginx.

   Execute nginx using the following command regardless of the OS .

```
$ service nginx start
```

4. Open the firewall.

   The method for opening the firewall and port differs for RHEL 7, CentOS, and Debian 10.

   Open the firewall in the method appropriate for each OS.

- RHEL 7/CentOS 7

1. Check if the firewall is running.

```
$ systemctl status firewalld.service
```

2. Check the default zone.

```
$ firewall-cmd --get-default-zone
```

3. Check for active zones.

```
$ firewall-cmd --get-active-zone
```

4. Check for open ports.

```
$ firewall-cmd --list-ports
```

5. Open port number 80.

```
$ firewall-cmd --permanent --zone=public --add-port=80/tcp
```

6. Restart the firewall.

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

- Debian 10

1. Check for open ports.

```
$ iptables -L
```

2. Open port number 80.

```
$ iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

3. Allow firewall settings to be applied automatically at boot time.

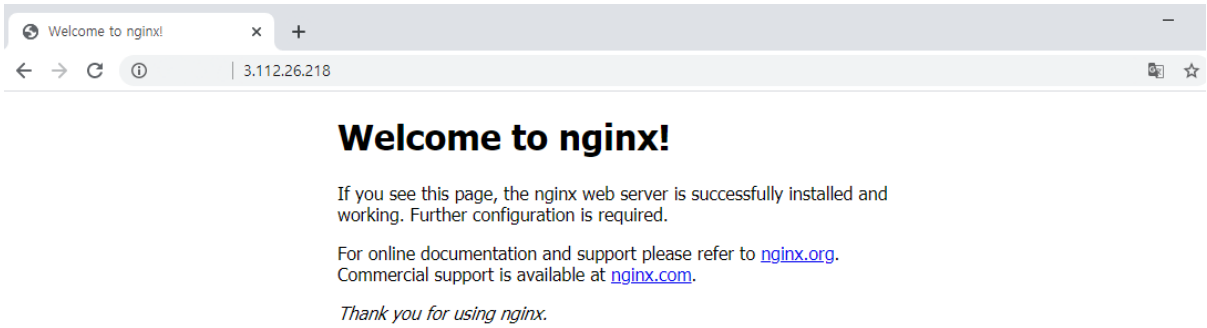
```
$ apt install iptables-persistent
$ netfilter-persistent save
$ netfilter-persistent reload
```

3.1.2. Verifying the Installation

Enter your IP address in a web browser.

If the following screen is displayed, then the installation is successfully completed.

[Figure 3.2] nginx Installation Verification Screen



3.2. Installing WAS

This section describes how to install WAS and verify its installation.

eformsign uses Tomcat as the WAS.

Note

The recommended Tomcat version is 8.5.

3.2.1. Installing Tomcat

The Tomcat installation process is as follows.

1. Install Tomcat by executing the following commands for each OS.

   - RHEL 7

     ```
     $ cd /opt
     $ yum install wget unzip -y
     $ wget
     https://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.54/bin/apache-tomcat-8.5.54.zip
     $ unzip apache-tomcat-8.5.54.zip
     ```

   - CentOS 7

     ```
     $ yum -y install tomcat
     $ yum install -y tomcat-webapps tomcat-admin-webapps
     ```

   - Debian 10

```
$ useradd -m -U -d /opt/tomcat -s /bin/false tomcat
$ cd /tmp
$ wget
http://apache.mirror.cdnetworks.com/tomcat/tomcat-8/v8.5.54/bin/apache-tomcat-8.5.54.tar.gz
$ tar -xf apache-tomcat-8.5.54.tar.gz
$ mv apache-tomcat-8.5.54 /opt/tomcat/
$ ln -s /opt/tomcat/apache-tomcat-8.5.54 /opt/tomcat/latest
```

2. Configure the basic settings and create a user.

- RHEL 7

```
$ ln -s /opt/apache-tomcat-8.5.54 /opt/tomcat
$ ln -s /opt/tomcat/logs /var/log/tomcat
$ useradd tomcat
$ chown –R tomcat:tomcat /opt/apache-tomcat-8.5.54
$ chown -R tomcat:tomcat /opt/tomcat
$ chown -R tomcat:tomcat /var/log/tomcat
$ chmod +x /opt/tomcat/bin/*.sh
```

- Debian 10

```
$ chown -R tomcat: /opt/tomcat
$ sh -c 'chmod +x /opt/tomcat/latest/bin/*.sh'
```

Note

This process is performed only in RHEL 7 and Debian 10.

---

3. Create a service file.

- RHEL 7

```
$vi /etc/systemd/system/tomcat.service
[Unit]
Description=Tomcat
After=syslog.target network.target

[Service]
Type=forking

User=tomcat
```

```
Group=tomcat

ExecStart=/opt/tomcat/bin/catalina.sh start
ExecStop=/opt/tomcat/bin/catalina.sh stop

[Install]
WantedBy=multi-user.target
```

- Debian 10

```
$vi /etc/systemd/system/tomcat.service
[Unit]
Description=Tomcat 8.5 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/adoptopenjdk-8-hotspot-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom"

Environment="CATALINA_BASE=/opt/tomcat/latest"
Environment="CATALINA_HOME=/opt/tomcat/latest"
Environment="CATALINA_PID=/opt/tomcat/latest/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"


ExecStart=/opt/tomcat/latest/bin/startup.sh
ExecStop=/opt/tomcat/latest/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Note

This process is performed only in RHEL 7 and Debian 10.

---

4. Reload the service file created with the following command.

```
$ systemctl daemon-reload
```

5. Open the firewall.

   The method for opening the firewall and port differs between RHEL 7/CentOS 7 and Debian 10, but the port number is 8080 for all of them.

   - RHEL 7/CentOS 7

     ```
     $ firewall-cmd --zone=public --add-port=8080/tcp --permanent
     $ firewall-cmd --reload
     ```

   - Debian 10

     ```
     $ iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
     $ netfilter-persistent save
     $ netfilter-persistent reload
     ```

6. Execute Tomcat.

   Execute Tomcat with the following command, regardless of the OS.

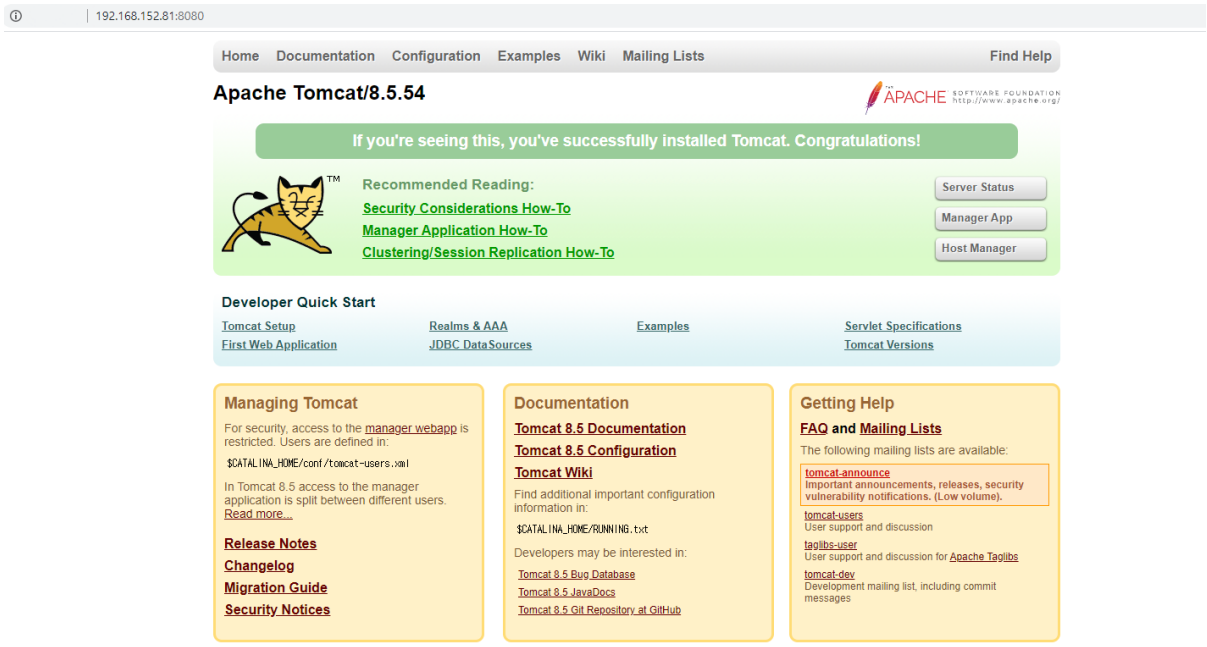
   ```
   $ service tomcat start
   ```

3.2.2. Verifying the Installation

Enter your IP address in a web browser.

If the following screen is displayed, then the installation is successfully completed.

[Figure 3.3] Tomcat Installation Verification Screen



3.3. Installing the DB

This section describes how to install the DB and verify its installation.

eformsign uses CouchDB as the DB.

Note

The recommended CouchDB version is 3.0.

3.3.1. Installing CouchDB

The CouchDB installation process is as follows.

1. Create the EPEL Repository.

   - RHEL 7/CentOS 7

```
$ yum update -y
$ wget
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
$ yum install -y epel-release-latest-7.noarch.rpm
```

Note

EPEL is short for Extra Packages of Enterprise Linux and is a repository provided by Fedora Project. It is a community-based storage that provides the latest version of various packages. It can be used to install and use the packages in the latest version that cannot be installed using the **yum** command.

2. Enable the CouchDB package repository.

   - RHEL 7/CentOS 7

```
$ vi /etc/yum.repos.d/bintray-apache-couchdb-rpm.repo
[bintray--apache-couchdb-rpm]
name=bintray--apache-couchdb-rpm
baseurl=http://apache.bintray.com/couchdb-rpm/el7/$basearch/
gpgcheck=0
repo_gpgcheck=0
enabled=1
```

   - Debian 10

```
$ apt-get install -y apt-transport-https gnupg ca-certificates
$ echo "deb https://apache.bintray.com/couchdb-deb buster main" \
    | sudo tee -a /etc/apt/sources.list.d/couchdb.list
```

3. Install the CouchDB package.

   - RHEL 7/CentOS 7

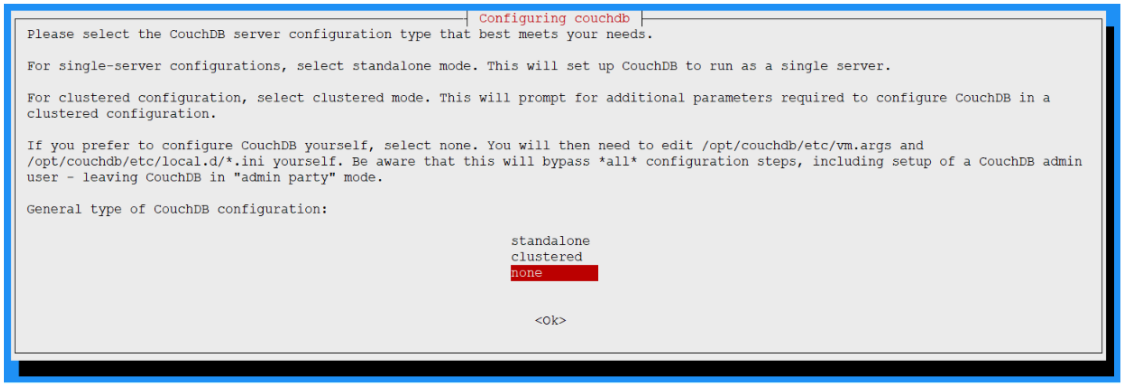
```
$ yum -y install epel-release && sudo yum -y install couchdb
```

   - Debian 10

```
$ apt-key adv --keyserver keyserver.ubuntu.com --recv-keys \
  8756C4F765C9AC3CB6B85D62379CE192D401AB61
$ apt update
$ apt install -y couchdb
```

4. When prompted to select a CouchDB type during installation, select **none**.

[Figure 3.4] The CouchDB Type Selection Window



3.3.2. Default Settings

Modify the local.ini file and the default.ini file as follows. The method is the same for all OS.

- local.ini file

```
$ vi /opt/couchdb/etc/local.ini
[admins]
admin = mysecretpassword // Remove the [admins] comment at the bottom and
 change the ID and PW to the ones you want.
```

- default.ini file

```
$ vi /opt/couchdb/etc/default.ini
[cluster]
q=2              // Change the q=2, n=3 in the middle [cluster] to q=2,
n=1(search using /cluster)
n=1
...
[chttpd]
port = 8308      // Change port = 5984 to port = 8308.
```

3.3.3. Verifying the Installation

Check the connection by running CouchDB.

- Run the DB.

```
$ service couchdb start
```

- Check the connection.

```
$ curl http://localhost:8308
```

Chapter 4. eformsign Installation and Configuration

This chapter describes how to install eformsign and configure its properties.

Note

To install and run eformsign, the web server, WAS, and DB must be installed beforehand.

For information about how to install them, refer to “Chapter 3. Pre-installation Task 2: Server Environment Configuration”.

4.1. Installing eformsign

This section describes how to install eformsign.

4.1.1. Pre-installation Tasks

Copy the default configuration file by executing the following command.

```
$ mkdir /opt/eformsign
$ cd /opt/eformsign
// Copy the compressed file you received to the current folder.
'10,2020,0429,300' is the build version and the file name may vary.
$ cp Onpremise-10,2020,0429,300.zip
```

4.1.2. Extracting eformsign

Extract the **Onpremise-10,2020,0429,300.zip** file by executing the following command.

Note

10,2020,0429,300 is the build version and the file name may differ.

```
$ unzip Onpremise-10,2020,0429,300.zip
```

4.1.3. Configuring the Storage

Configure the storage to be used by eformsign.

```
$ cd /opt
$ mkdir storage
$ chown -R tomcat:tomcat storage/
$ chmod -R 750 storage/
$ cd storage
$ unzip /opt/eformsign/storagefiles.zip
```

4.1.4. Extracting Daemon, Iam, and Service

Create a folder in webapps in Tomcat and extract the Daemon, Iam, and Service files.

The method of extraction differs for each OS.

- RHEL 7

```
$ cd /opt/tomcat/webapps/
$ rm -rf *
$ mkdir Daemon Iam Service
$ chown -R tomcat:tomcat *
$ chmod –R 750 *
$ cd Daemon
$ unzip /opt/eformsign/Daemon-10,2020,0429,300.war
$ cd ../Iam
$ unzip /opt/eformsign/Iam-10,2020,0429,300.war
$ cd ../Service
$ unzip /opt/eformsign/Service-10,2020,0429,300.war
```

- CentOS 7

```
$ cd /var/lib/tomcat/webapps
$ rm -rf *
$ mkdir Daemon Iam Service
$ chown -R tomcat:tomcat *
$ chmod –R 750 *
$ cd Daemon
$ unzip /opt/eformsign/Daemon-10,2020,0429,300.war
$ cd ../Iam
$ unzip /opt/eformsign/Iam-10,2020,0429,300.war
```

```
$ cd ../Service
$ unzip /opt/eformsign/Service-10,2020,0429,300.war
```

- Debian 10

```
$ cd /opt/tomcat/latest/webapps/
$ rm -rf *
$ mkdir Daemon Iam Service
$ chown -R tomcat:tomcat *
$ chmod –R 750 *
$ cd Daemon
$ unzip /opt/eformsign/Daemon-10,2020,0429,300.war
$ cd ../Iam
$ unzip /opt/eformsign/Iam-10,2020,0429,300.war
$ cd ../Service
$ unzip /opt/eformsign/Service-10,2020,0429,300.war
```

4.1.5. Stopping the Servers

To configure the server environments, stop the web server, WAS, and DB server that are installed and running.

```
$ service nginx stop
$ service tomcat stop
$ service couchdb stop
```

4.2. Encryption Settings

Encrypt your account and generate encryption key files.

4.2.1. Encrypting Your Account and Generating Key Files

Encrypt your account by executing the following command and write a memo of the result. The information of the encrypted account will be displayed and the key files, RS256.privateand RS256.pub, will be generated.

```
$ cd /opt/eformsign
$ java -jar eformsignGen.jar init
Enter couchdb ID : CouchDB ID
Enter couchdb Password : CouchDB PW
```

```
Enter email ID : email ID
Enter email Password : email PW
```

4.3. Configuring CouchDB

eformsign's data can be stored and managed in a DB by configuring CouchDB.

4.3.1. Creating the init-data Folder and Copying CouchDB Files

Create the init-data folder and copy CouchDB files to the eformsign folder.

```
$ cd /opt/couchdb
$ mkdir init-data
$ cd init-data/
$ cp /opt/eformsign/eformsign-couchdb-* .
```

4.3.2. Extracting Files

Extract CouchDB files by executing the following command.

```
$ tar -zxvf eformsign-couchdb-default-data.tar.gz
```

4.3.3. Copying the Configuration File

Copy the default configuration file by executing the following commands.

```
$ cp eformsign-couchdb-backup eformsign-couchdb-default-data
$ cp eformsign-couchdb-restore eformsign-couchdb-default-data
$ cd eformsign-couchdb-default-data
```

4.3.4. Modifying the eformsign-couchdb-restore File

Modify the eformsign-couchdb-restore file by executing the following command.

```
$ vi eformsign-couchdb-restore
./eformsign-couchdb-backup -r -H localhost -d $filename -f $f -u admin -p
password -q -c  // Enter couchdbID, couchdbPW in admin, password
```

4.3.5. Granting Execution Permission

Grant execution permission for CouchDB.

```
$ chmod +x eformsign-couchdb-*
```

4.3.6. Executing the Script

Restart CouchDB.

```
$ service couchdb restart
$ curl -X PUT http://admin:password@localhost:8308/_users // Enter
couchdbID:couchdbPW in admin:password.
$ curl -X PUT http://admin:password@localhost:8308/_replicator // Enter
couchdbID:couchdbPW in admin:password.
$ ./eformsign-couchdb-restore
```

4.4. Configuring Iam

Iam server is a server that manages user accounts in the Oauth method. Configure it in the following order.

4.4.1. Go to the Iam/WEB-INF Folder

Go to the WEB-INF folder in Iam by executing the following command for each OS.

- RHEL 7

```
$ cd /opt/tomcat/webapps/Iam/WEB-INF
```

- CentOS 7

```
$ cd /var/lib/tomcat/webapps/Iam/WEB-INF
```

- Debian 10

```
$ cd /opt/tomcat/latest/webapps/Iam/WEB-INF
```

4.4.2. Configuring Properties

Modify the config.properties file as follows.

```
$ vi config.properties
...
// Enter the mail part
mail.host=
mail.username=
mail.password=
mail.port=
mail.smtp.auth=
mail.smtp.starttls.enable=
mail.from.address=
mail.from.name=
mail.html=

mail.contacts.address=<enter the user account>
mail.redirect.address=<enter the domain or IP address>

############################################
## DataSource Configuration
############################################
couch.db.url=http://localhost:8308
couch.db.username= CouchDB ID       // Enter CouchDB Id, PW in
couch.db.username, couch.db.password
couch.db.password= CouchDB PW
couch.db.database=forcs_iam
couch.db.autoview=true
```

4.4.3. Copying the Encryption Key Files (RS256.private, RS256.pub)

Copy the “4.2.1. Encrypting Your Account and Generating Key Files”, RS256.private and RS256.pub files by executing the following commands.

- RHEL 7

```
$ cd /opt/tomcat/webapps/Iam/WEB-INF/classes/jwt-keys
$ rm -rf *
$ cp /opt/eformsign/app/RS256.* .
```

- CentOS 7

```
$ cd /var/lib/tomcat/webapps/Iam/WEB-INF/classes/jwt-keys
$ rm -rf *
$ cp /opt/eformsign/RS256.* .
```

- Debian 10

```
$ cd /opt/tomcat/latest/webapps/Iam/WEB-INF/classes/jwt-keys
$ rm -rf *
$ cp /opt/eformsign/RS256.* .
```

4.5. Configuring Service

Service server is a server that synchronously responds to user requests, and is run on the WAS. Configure it as follows.

4.5.1. Go to the Service/WEB-INF/classes Folder

Go to the Service/WEB-INF/classes folder in webapps in Tomcat.

- RHEL 7

```
$ cd /opt/tomcat/webapps/Service/WEB-INF/classes
```

- CentOS 7

```
$ cd /var/lib/tomcat/webapps/Service/WEB-INF/classes
```

- Debian 10

```
$ cd /opt/tomcat/latest/webapps/Service/WEB-INF/classes
```

4.5.2. Configuring Properties

Modify the aes256.properties, couchDB.properties, forcsCom.properties, storageConfig.xml, and email.properties files as follows.

- aes256.properties

```
$ vi aes256.properties
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'.
aeskey.couchdb=
aeskey.api=
aeskey.webhook=
aeskey.email=
```

- couchDB.properties

```
$ vi couchDB.properties
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'.
db.id_encrypted=
db.password_encrypted=
...
server.type=onpremise
server.service_url=http://Domain or ip/Service
server.daemon_url=http://Domain or ip/Daemon
server.home_url=http://Domain or ip
service.language=ko-kr,ja-jp,en-us
```

- forcsCom.properties

```
$ vi forcsCom.properties
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'.
db.id_encrypted=
db.password_encrypted=
```

- storageConfig.xml

```
$ vi storageConfig.xml
…
<!-- file system -->
        <bean id="storageService"
```

```
class="oz.service.storage.impl.FileSystemServiceImpl">
                <constructor-arg name="_root_path" value=""/>
 //Enter the storage path in the value of _root_path
                <constructor-arg name="_imgUrl" value=""/>
//Enter the domain/Service in the value in _imgUrl
        </bean>
…
```

- email.properties

```
$ vi email.properties
host : Enter the mail server host
port : Enter the mail server port
account : Enter the mail account
password : Enter the mail password
prefix : Enter the header(optional)
ssl : Configure by mail server (true / false)
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'
account_encrypted :
password_encrypted :
```

4.5.3. Copying the Encryption Key File (RS256.pub)

Copy the "4.2.1. Encrypting Your Account and Generating Key Files", RS256.pub file by executing the following command.

- RHEL 7

```
$ cd /opt/tomcat/webapps/Service/WEB-INF/classes/jwt-keys
$ rm -rf *
$ cp /opt/eformsign/RS256.pub .
```

- CentOS 7

```
$ cd /var/lib/tomcat/webapps/Service/WEB-INF/classes/jwt-keys
$ rm -rf *
$ cp /opt/eformsign/RS256.pub .
```

- Debian 10

```
$ cd /opt/tomcat/latest/webapps/Service/WEB-INF/classes/jwt-keys
$ rm -rf *
$ cp /opt/eformsign/RS256.pub .
```

4.6. Configuring the Daemon

The Daemon server is a server that asynchronously responds to batch and scheduler tasks such as email notification, bulk creation of documents, and generation of PDF documents. Same as the Service server, it is run on the WAS. Configure it as follows.

4.6.1. Go to the Daemon/WEB-INF/classes Folder

Go to the Daemon/WEB-INF/classes folder in webapps in Tomcat.

- RHEL 7

```
$ cd /opt/tomcat/webapps/Daemon/WEB-INF/classes
```

- CentOS 7

```
$ cd /var/lib/tomcat/webapps/Daemon/WEB-INF/classes
```

- Debian 10

```
$ cd /opt/tomcat/latest/webapps/Daemon/WEB-INF/classes
```

4.6.2. Configuring Properties

Modify each one of aes256.properties, couchDB.properties, forcsCom.properties, storageConfig.xml, and email.properties files as follows.

- aes256.properties

```
$ vi aes256.properties
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'
aeskey.couchdb=
aeskey.api=
aeskey.webhook=
aeskey.email=
```

- couchDB.properties

```
$ vi couchDB.properties
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'
db.id_encrypted=
db.password_encrypted=
...
server.type=onpremise
server.service_url=http://Domain or ip/Service
server.daemon_url=http://Domain or ip/Daemon
server.home_url=http://Domain or ip
service.language=ko-kr,ja-jp,en-us
```

- forcsCom.properties

```
$ vi forcsCom.properties
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'
db.id_encrypted=
db.password_encrypted=
```

- storageConfig.xml

```
$ vi storageConfig.xml
…
        <bean id="storageService"
class="oz.service.storage.impl.FileSystemServiceImpl">
                <constructor-arg name="_root_path" value=""/>
 //Enter the storage path in the value of _root_path.
                <constructor-arg name="_imgUrl" value=""/>
//Enter the domain/Service in the value of_imgUrl.
        </bean>
…
```

- email.properties

```
$ vi email.properties
host : Enter the mail server host
port : Enter the mail server port
account : Enter the mail account
password : Enter the mail password
prefix : Enter the header(optional)
```

```
ssl : Configure according to the mail server (true / false)
// Enter the encryption key generated in '4.2.1. Encrypting Your Account
and Generating a Key File'
account_encrypted :
password_encrypted :
```

4.7. Configuring Home

This is the screen displayed on the front-end to users and is installed on the web server.

4.7.1. Extracting the Home File

Extract the Home file in nginx.

- RHEL7/CentOS 7

```
$ cd /usr/share/nginx/html
$ rm -rf *
$ tar -zvxf /opt/eformsign/app/home-10,2020,0429,300.tar.gz
```

- Debian 10

```
$ cd /var/www/html
$ rm -rf *
$ tar -zvxf /opt/eformsign/app/home-10,2020,0429,300.tar.gz
```

4.7.2. Configuring the Version

1. To configure the Home version, go to the version folder.

- RHEL 7/CentOS 7

```
$ cd /usr/share/nginx/html/version
```

- Debian 10

```
$ cd /var/www/html/version
```

2. Open the version10,2020,0429,300.js file.

Note

10,2020,0429,300 is the build version, and the file name may differ.

```
$ vi version10,2020,****,***.js
```

3. Modify the file as follows.

```
var EFORMSIGN_VERSION="10,2020,****,***";
var OPTION_KO_BLOG_URL="https://www.eformsign.com/kr/blog/";
var OPTION_EN_BLOG_URL="https://www.eformsign.com/en/blog/";
var OPTION_JA_BLOG_URL="https://www.eformsign.com/jp/blog/";
var OPTION_KO_SUPPORT_URL="https://www.eformsign.com/kr/support/";
var OPTION_EN_SUPPORT_URL="https://www.eformsign.com/en/support/";
var OPTION_JA_SUPPORT_URL="https://www.eformsign.com/en/support/";
var OPTION_SERVICE_URL="http://domain/Service";
var OPTION_DAEMON_URL="http://domain/Daemon";
var OPTION_LOCATION_CODE="country code eg) kr";
```

4.8. Configuring ozconfig

1. To configure ozconfig, go to the html folder.

- RHEL 7/CentOS 7

```
$ cd /usr/share/nginx/html
```

- Debian 10

```
$ cd /var/www/html
```

2. Open the ozconfig.xml file.

```
$ vi ozconfig.xml
```

3. Modify the file as follows.

Note

The localhost must be changed to either the domain or IP.

```
<serviceInfo>
<browser>
 <preserveCache>false</preserveCache>
```

```
 </browser>
 <pageUrls>

<execution>http://localhost/eform/document/ozinoffice_view_service.html</execution>

 <download>http://localhost/eform/form/list_form_oz.html</download>
 <upload>http://localhost/eform/form/check_form_oz.html</upload>

<propertyPane>http://localhost/ozinoffice_pages/ozinoffice_prop_panel.html</propertyPane>

 </pageUrls>
 <update>
 <addInVersion>80.2020.0422.201</addInVersion>
 <productVersion>19081901</productVersion>

<downloadUrl>http://localhost/development_tools/office/eformsign_form_builder.exe</downloadUrl>

 </update>
 <ribbon>
 <documentInfo>true</documentInfo>
 </ribbon>
</serviceInfo>
```

4.9. Configuring the License

Go to the folder that contains the license file and set the license for each OS as follows.

- RHEL 7

```
$ cp ozlicense.xml eformsignlicense.xml
/opt/tomcat/webapps/Service/WEB-INF/classes/
$ cp ozlicense.xml eformsignlicense.xml
/opt/tomcat/webapps/Daemon/WEB-INF/classes/
$ cp ozlicense.xml /usr/share/nginx/html/plugins/ozhtml5/license/
$ cp ozuserservicelicense.xml /usr/share/nginx/html/
```

- CentOS 7

```
$ cp ozlicense.xml eformsignlicense.xml
/var/lib/tomcat/webapps/Service/WEB-INF/classes/
$ cp ozlicense.xml eformsignlicense.xml
/var/lib/tomcat/webapps/Daemon/WEB-INF/classes/
```

```
$ cp ozlicense.xml /usr/share/nginx/html/plugins/ozhtml5/license/
$ cp ozuserservicelicense.xml /usr/share/nginx/html/
```

- Debian 10

```
$ cp ozlicense.xml eformsignlicense.xml
/opt/tomcat/latest/webapps/Service/WEB-INF/classes/
$ cp ozlicense.xml eformsignlicense.xml
/opt/tomcat/latest/webapps/Daemon/WEB-INF/classes/
$ cp ozlicense.xml /var/www/html/plugins/ozhtml5/license/
$ cp ozuserservicelicense.xml /var/www/html/
```

4.10. Configuring the Proxy

4.10.1. Configuring nginx

In RHEL 7 and CentOS 7, modify the server{} part in the default.conf file.

In Debian 10, modify the server{} part in the default file.

1. Open either the default.conf or default file.

- RHEL 7/CentOS 7

```
$ vi /etc/nginx/conf.d/default.conf
```

- Debian 10

```
$ vi /etc/nginx/sites-available/defult
```

2. Modify the file as follows.

- RHEL 7/CentOS 7: default.conf file

```
server {
listen 80;
server_name localhost;

#charset koi8-r;
#access_log /var/log/nginx/host.access.log main;

location / {
```

```
root /usr/share/nginx/html;
#index index.html index.htm;
index login_private.html;
}

location /Iam {
# forward all request headers to backend
proxy_pass_request_headers on;
# these settings come from the CouchDB wiki
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

root /usr/share/nginx/html;
index index.html index.htm;
proxy_pass http://localhost:8080/Iam;
}

location /Service {
# forward all request headers to backend
proxy_pass_request_headers on;
# these settings come from the CouchDB wiki
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

root /usr/share/nginx/html;
index index.html index.htm;
proxy_pass http://localhost:8080/Service;
}

location /Daemon {
# forward all request headers to backend
proxy_pass_request_headers on;
# these settings come from the CouchDB wiki
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

root /usr/share/nginx/html;
index index.html index.htm;
proxy_pass http://localhost:8080/Daemon;
```

```
}

#error_page 404 /404.html;

# redirect server error pages to the static page /50x.html
#
```

- Debian 10: default file

```
server {

....

    location / {

        #index index.html index.htm;
        index login_private.html;

    }

    location /Iam {
        # forward all request headers to backend
        proxy_pass_request_headers on;
        # these settings come from the CouchDB wiki
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        index index.html index.htm;
        proxy_pass http://localhost:8080/Iam;
    }

    location /Service {
        # forward all request headers to backend
        proxy_pass_request_headers on;
        # these settings come from the CouchDB wiki
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        index index.html index.htm;
        proxy_pass http://localhost:8080/Service;
```

```
    }

    location /Daemon {
        # forward all request headers to backend
        proxy_pass_request_headers on;
        # these settings come from the CouchDB wiki
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        index index.html index.htm;
        proxy_pass http://localhost:8080/Daemon;

    }

...
```

4.10.2. Configuring Tomcat

1. Open server.xml in Tomcat.

- RHEL 7/CentOS 7

```
$ vi /opt/tomcat/conf/server.xml
```

- CentOS 7

```
$ vi /etc/tomcat/server.xml
```

- Debian 10

```
$ vi /opt/tomcat/latest/conf/server.xml
```

2. Modify the server.xml file as follows.

```
<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true">

<Context docBase="Iam" path="/Iam"/>
<Context docBase="Service" path="/Service"/>
```

```
<Context docBase="Daemon" path="/Daemon" />
<!-- SingleSignOn valve, share authentication between web applications
Documentation at: /docs/config/valve.html -->
<!--
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
-->

<!-- Access log processes all example.
Documentation at: /docs/config/valve.html
Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log" suffix=".txt"
pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />
</Host>
```

4.11. Configuring GZIP

1. Open the nginx.conf file in nginx.

```
$ vi /etc/nginx/nginx.conf
```

2. Modify the nginx.conf file as follows.

```
...
http {
...
    gzip on;
gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
    gzip_http_version 1.1;
    gzip_types text/html text/plain text/css application/json
application/javascript text/xml;
    client_max_body_size 100M;
...
}
```

4.12. Configuring Startup

Check whether the Service, Daemon, and Iam servers are installed and configured normally.

4.12.1. Running the Servers

Start each server.

```
Service nginx start
Service couchdb start
Service tomcat start
```

4.12.2. Service

Enter the IP address/Service in the web browser, and check whether the following execution result is displayed.

```
{"Service Name":"Service","eformsign version":"10,2020,0429,300","Viewer
version":"80,2020,0422,201","Server version":"80,2020,0422,1510","Cpu
Count":"any","IP List":"any","Request URL":"any",
"Expire Date":"2021-05-01"}
```

4.12.3. Daemon

Enter the IP address/Daemon in the web browser, and check whether the following execution result is displayed.

```
{"Service Name":"Service","eformsign version":"10,2020,0429,300","Viewer
version":"80,2020,0422,201","Server version":"80,2020,0422,1510","Cpu
Count":"any","IP List":"any","Request URL":"any",
"Expire Date":"2021-05-01"}
```

4.12.4. Iam

Enter the IP address/Iam in the web browser, and check whether the Iam login page is displayed.

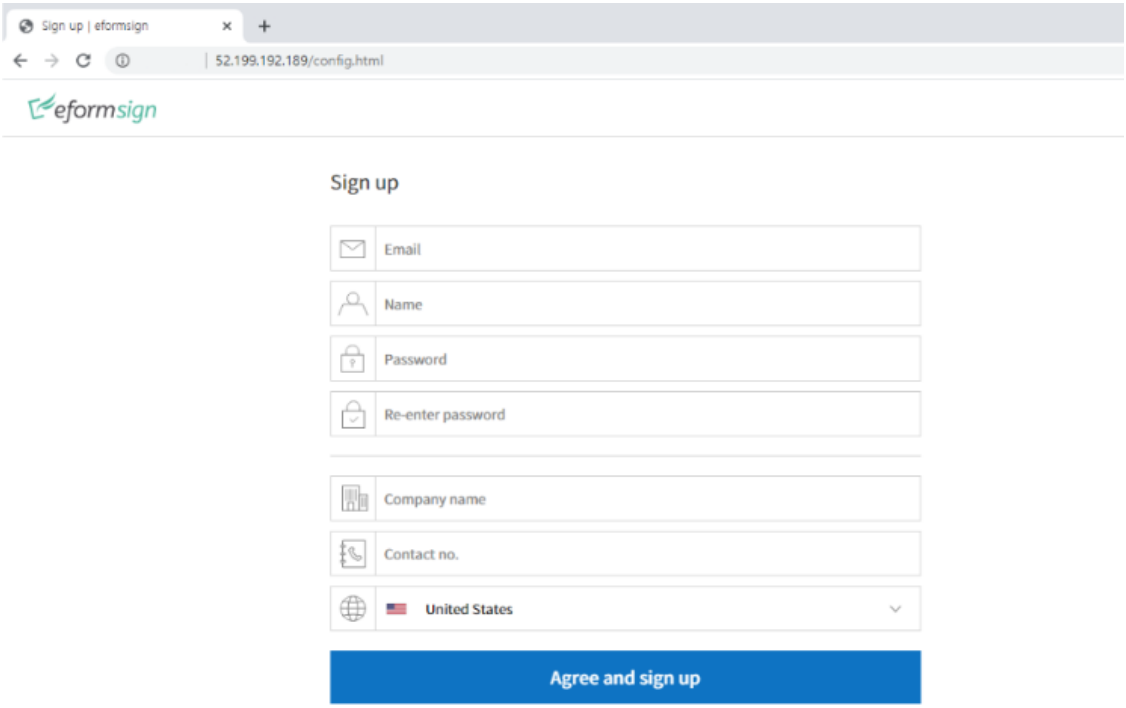
Chapter 5. Company Registration

5.1. Registering Your Company

The following screen will be displayed when entering the IP address/config.html in a browser.

Enter the email, user name, password, company name, and country to register your company.

[Figure 5.1] Registering Your Company



Index