Kofax Mobile ID Verification

Kofax Mobile ID Capture - ID Verification Installation Guide

Version: 2.5.0.6

Date: 2020-09-15

KOFAX

Table of Contents

Preface

This guide includes the information you need to successfully install the Kofax Mobile ID Verification and Facial Recognition components for your mobile project.

Related documentation

In addition to this guide, refer to the following documentation:

- *Kofax TotalAgility Administrator's Guide*: Contains essential information about installing and configuring Kofax TotalAgility.
- *Kofax Mobile SDK Developer's Guide*: Contains essential information about installing and configuring the Kofax Mobile SDK.
- *Kofax Mobile ID Extracted Field Tables*: This is an HTML document with a complete listing of all currently supported ID fields by region and by country.

Training

Kofax offers both classroom and computer-based training that will help you make the most of your Kofax Capture solution. Visit the Kofax website at www.kofax.com for complete details about the available training options and schedules.

Getting help with Kofax products

The Kofax Knowledge Base repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the Kofax website and select **Support** on the home page.

Note The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

- Powerful search capabilities to help you quickly locate the information you need.

  Type your search terms or phrase into the **Search** box, and then click the search icon.

- Product information, configuration details and documentation, including release news.

  Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.
- Access to the Kofax Customer Portal (for eligible customers).

  Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.
- Access to the Kofax Partner Portal (for eligible partners).

  Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.
- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.

  Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.

Product documentation

By default, the product documentation is available online. However, if necessary, you can also download the documentation to use offline.

Default online documentation

The product documentation is available at the following location.

https://docshield.kofax.com/Portal/Products/en_US/KMC/3.5.0-cs5i340uk7/KMID.htm

Configure offline documentation

To access the documentation offline, download the following files:

- KofaxMobileIDDocumentation-2.5.0_EN.zip
- KofaxMobileIDVerificationDocumentation_2.5.0_EN.zip

Download these files from the Kofax Fulfillment Site and extract it on a local drive available to your users.

The compressed files include the `print` folder that contains all guides, such as the Installation Guide and the Administrator's Guide. The KofaxMobileIDDocumentation-2.5.0_EN.zip file also contains a `help` folder with API references.

Chapter 1

Overview

Mobile ID Verification

The Mobile ID Verification product provides functionality that analyzes identification documents including ID Cards, Driver Licenses, and Residence Permits to determine if those documents are valid and authentic. ID Verification is comprised of the following components:

- Database Server
- Connect Application Server
- ID Rectification Server
- Tamper Server
- 2FA Server
- ID Authentication
- ID Extraction
- IDDI Web
- IDDI API
- IDDI Retention

Database Server

The product uses MySQL 5.7 and Mongo database to store data used during analysis. MongoDB is shipped as a docker image available in Linux installation folder.

Connect Application Server

This Windows-based server is responsible for saving verification transaction data in the database.

ID Rectification Server

This Linux-based service responsible for image cropping.

Tamper Server

This Linux-based server performs advanced image analysis and tamper detection checks to ensure the document and captured image have not been altered or tampered with in any way. The Tamper server performs a variety of checks with configurable thresholds that can be adjusted based on your verification requirements.

2FA Server

This Linux-based server is a security process for two-factor authentication. The user provides two different authentication factors to verify themselves to protect the user's credentials and accessible resources.

ID Authentication

This Linux-based service performs basic image analysis and verifies the document as authentic.

ID Extraction

This Linux-based service performs extraction-related analysis for authentication.

IDDI Web

This Linux-based sevice provides the web UI portal to view and maintain results.

IDDI API

This Linux-based service provides an API interface between end user and collaborate between other services.

IDDI Retention

This Linux-based service is used for data retention or purge.

Mobile ID Facial Recognition

The Mobile ID Facial Recognition product is a separately licensed add-on to the Mobile ID Verification product. It provides functionality that analyzes a live-captured selfie photo in concert with the Mobile ID Verification product to confirm that the live user matches the photograph of the identification document. Facial Recognition is comprised of the following additional components:

Facial Recognition Server

This Linux-based server performs the analysis of the live-captured photo comparing it with the analysis done by the Mobile ID Verification product

Chapter 2

Installing the Mobile ID Verification and Facial Recognition Components

System Requirements

Basic Server Hardware Requirements

Linux Services

Red Hat Enterprise Linux 7.6 64-Bit:

- CPU flag AVX must be enabled by default
- Minimum hardware supported:
  - 16 cores
  - 48 GB RAM
  - 150 GB for the `var` partition and 100 GB for `data` partition with XFS file system.
- Recommended hardware configuration:
  - 32 cores
  - 96 GB RAM
  - 150 GB for the `var` partition and 200 GB for `data` partition with XFS file system.

Connect Server

Windows Server 2012 R2 or higher

- 4-core CPU minimum
- 8 GB RAM minimum
- 100 GB disk storage minimum

Advanced Server Hardware Requirements

Note For an explanation of advanced servers, see the Advanced Setup diagram in Deployment diagrams.

Linux Services

Red Hat Enterprise Linux 7.6 64-Bit:

- CPU flag AVX must be enabled by default

Mongodb

- 4-core CPU minimum
- 16 GB RAM minimum
- 200 GB for `data` partition with XFS file system minimum
- Minimum servers count: 3 (2 data store nodes and 1 arbiter)

MySql

- 4-core CPU minimum
- 8 GB RAM minimum
- 100 GB for `data` partition minimum
- Minimum server count: 2

Services (IDR, IDE, IDA, IDDIApi, IDDIWeb, ID2FA, IDT256, IDT512, IDDIRet, IDFR)

- 32 cores minimum
- 96 GB RAM minimum
- 150 GB for the `var` partition
- 500GB disk storage
- Minimum server count: 2

Connect server

Windows Server 2012 R2 or higher

- 4-core CPU minimum
- 8 GB RAM minimum
- 100 GB disk storage minimum
- Minimum server count: 2

Minimum Software Requirements

Connect Application Server

- Microsoft .NET 4.6.2
- Microsoft IIS with ASP.NET
  - Including IIS Hostable Web Core

Note When installing in Azure, you must install the Connect component in Docker on a Windows Server 2016 Datacenter Azure environment. Windows Server 2016 - Version: 1607 (OS Build 14393.3504) is required.

Prerequisites

Before installing the product, ensure the following prerequisites are met.

Connect Application Server

- Install Microsoft .NET Framework 4.6.2.
- HTTPS certificates are available in Server Certificates in IIS Manager.

Note Reboot the server after installation of any missing prerequisites.

Linux servers

Facial Recognition server requires additional licensing. These steps assume you are logged in as a root user.

1. Connect to the server via SSH.
2. You must have an active Red Hat subscription before beginning the installation. If you don't have one, you can register for one at `https://developers.redhat.com`. If your Red Hat Enterprise Linux servers are not already registered with a Red Hat subscription, use the following command to register, entering your password at the prompt:

   ```
   sudo subscription-manager register --username <USERNAME> --auto-attach
   ```

3. Copy the contents of the <*Linux Installer*> on the installation media into a specific folder (such as `/tmp/KofaxMobileIDVerification`), and then navigate to the same folder in a terminal window.
4. Install the required type of the Docker Engine (CE/EE).
5. Install the required MySQL client to execute the database scripts.

   ```
   sudo yum install -y mysql
   ```

6. Reboot the server. You can do this from the terminal by running the following command:

   ```
   sudo reboot
   ```

Nginx configuration

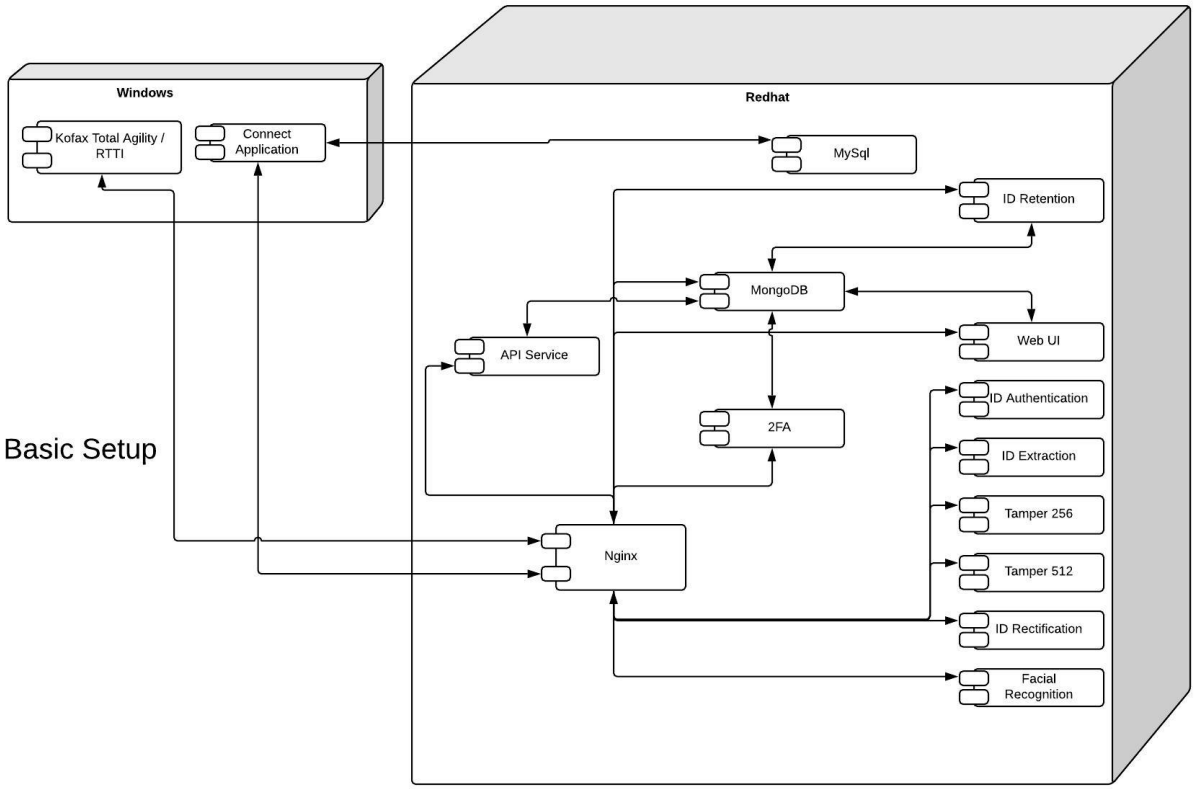
When setting up Linux services, Nginx should also be configured.

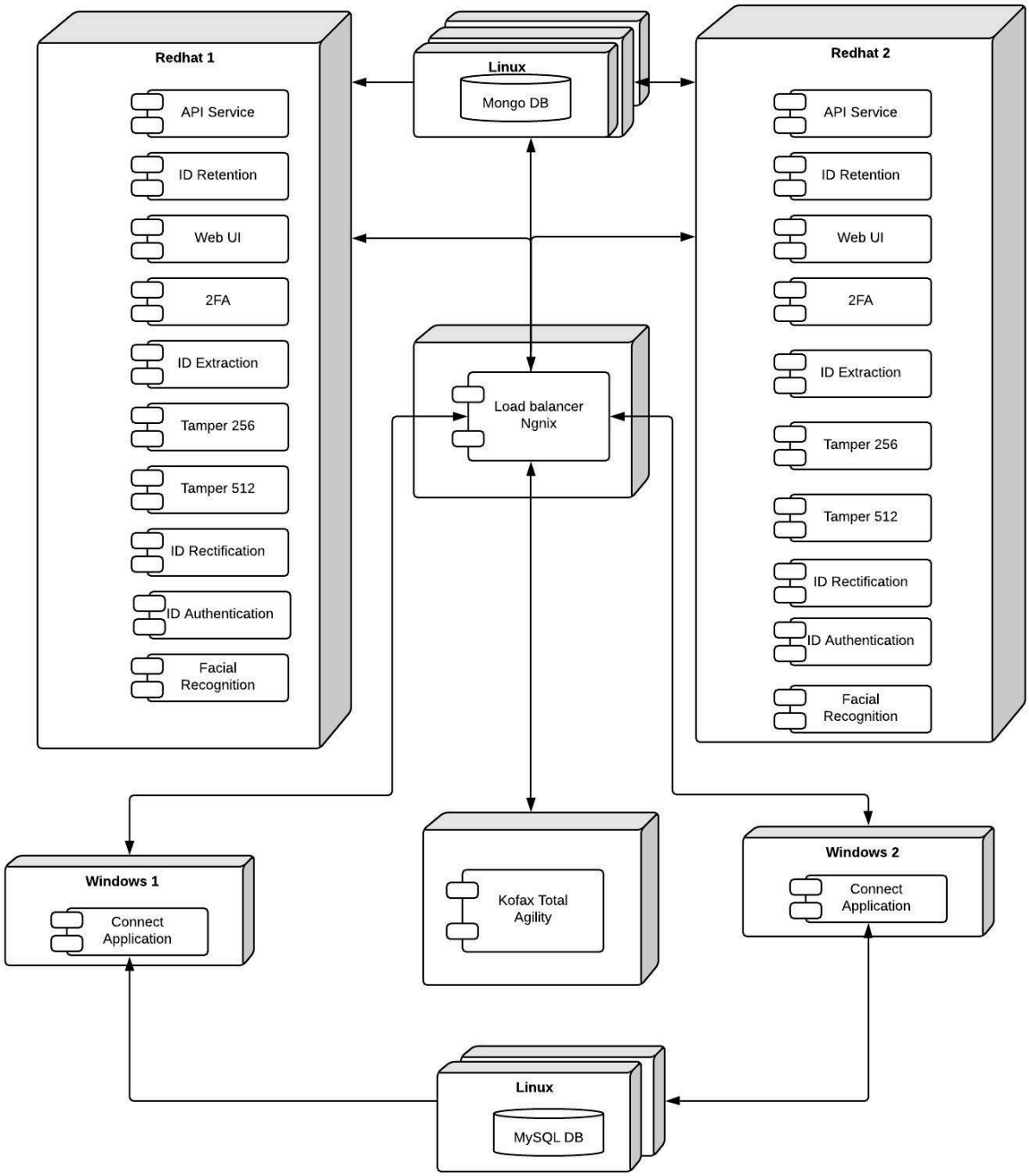
1. Install Nginx to support SSL communication.
2. Replace the nginx.conf file at `/etc/nginx` with the nginx.conf file provided with Kofax Mobile ID Verification.
3. Replace all server IP addresses and certificate paths in the Nginx file.
4. Restart the Nginx service.
5. Reboot the Linux server as shown in the *Kofax Mobile ID Verification Kofax Mobile ID Capture - ID Verification Installation Guide*.

Chapter 3

Deployment diagrams

The following diagrams show a sample setup configuration. We recommend scaling horizontally.





Advanced Setup

Chapter 4

Installation

Installation proceeds in the following order.

Server	OS	Comments
Database Server	Linux	Required. (MongoDB ,MySQL)
IDDIAPI/IDDI Retention and WebUI Server	Linux	Required. Coordinates document verification across the component systems.
2FA Server	Linux	Required. Enables two-factor authentication to improve security for the user's credentials and accessible resources.
Rectification/IDA/IDE Servers	Linux	Required. Performs cropping ,authentication and extraction analysis on a document.
Tamper Server	Linux	Required. Performs advanced image analysis to detect tampered physical and digital documents.
Facial Recognition Server	Linux	Optional. Analyzes live-captured photos of the user and compares them to a previously verified document. This functionality requires an additional license.
Connect Application Server	Windows	Required. Saves verification transaction data in the database.

Note If you receive a message during installation that an unsupported version of Red Hat Enterprise Linux was detected, contact Kofax Support for assistance.

Collect environment information

Before you begin the installation, gather the following information as you will use it during the installation and configuration of various components.

- Hostnames or IP addresses of each server in your environment.
- Certificates used for HTTPS communication.

You also need to install the Google Authenticator application on your mobile device.

Linux Services

1. Run the following commands to enable outbound and read access:

```
setsebool -P httpd_can_network_connect 1
```

```
setsebool -P httpd_read_user_content 1
```

2. Connect to your server using SSH and navigate to the folder containing the Linux installation files.
3. Pull the MySQL docker image from docker hub and run using the following commands.

```
$sudo docker pull mysql:5.7

$firewall-cmd --permanent --zone=trusted --add-port=3306/tcp

$firewall-cmd  --reload

$docker run -dti -p 3306:3306 -v /data/mysql:/var/lib/mysql -e
 MYSQL_ROOT_PASSWORD=<password> --name mysql57 mysql:5.7
```

Note Log onto MySQL using the MySQL client and grant root user access from remote computer. For example:

```
grant all privileges on *.* to 'root'@'%' with grant option;
```

4. Execute the following commands to install Mongo db.

```
$sudo chmod +x Linux_install.sh
$sudo bash -c "bash Linux_install.sh mongodb <mongoDBPassword>| tee
 mongodbdb_install.log"
```

Note the following:

- Replace <mongoDBPassword> with the password. Only alphanumeric characters.
- This step may take several minutes. The message "Installation Complete" will be displayed when the installation is finished.

5. Install mongodb-org-tools.

   Refer to the documentation on the Mongo DB website for insturctions on installing on Linux.
6. Copy the globalMerchantConfiguration.zip file and extract it.
7. Restore the Globalmerchantconfig date with the following command. Replace the correct values with the IP and password.

```
$mongorestore --ssl --sslAllowInvalidCertificates --host <MongodbHostIP>
 --authenticationDatabase=dataIntelligence --username admin --password
 <MongoDBPassword> globalMerchantConfiguration
```

Note MonogoDBHostIP is the static IP of the system where the MongoDB Docker image is installed and running.

8. Execute the following commands to install few Linux Services (Rectification, extraction, authentication and Tamper).

```
$sudo bash -c "bash Linux_install.sh verification | tee verification.log"
```

9. Execute the following commands to install few Linux Services (WebUI, DataIntelligence API, Data Retention, 2FA).

```
$sudo bash -c "bash Linux_install.sh webservice <MongodbHostIP> <mongoDBPassword>
 http://<DataIntelligenceHost IP>:<DataIntelligencePort> | tee webservices.log"
```

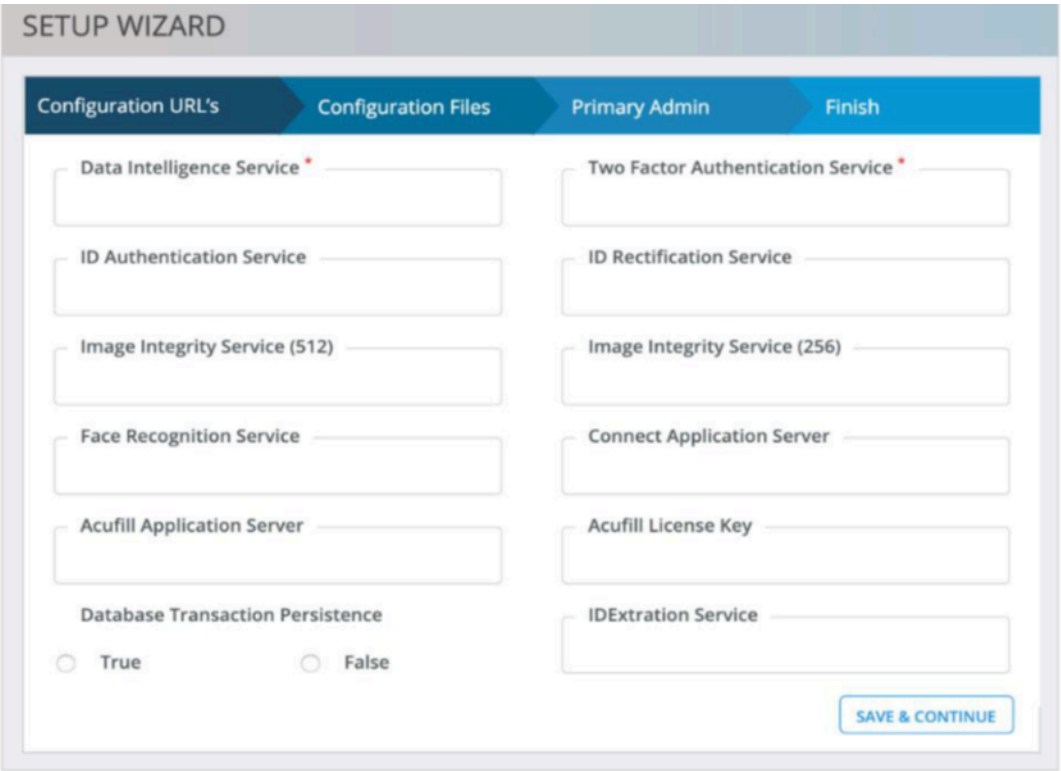
Note The DataIntelligenceHost IP refers to the static IP address of the host system where the web services are installed. The port is 8080.

10. Locate the logs by using the following command:

```
$docker logs iddiweb
```

11. Access Data Intelligence web URL `http://<hostname/IPaddress>` to configure all predefined endpoints

    The Setup Wizard appears.



12. On the Configuration URL's tab, enter the mandatory services data intelligence and two-factor authentication URLs.

Image	Port
IDR (Rectification)	8083
IDA (Authentication)	8081
IDE (Extraction)	8082
ID2FA (Two Factor Authentication)	8088

Image	Port
IDT256 (Tamper 256)	8085
IDT512 (Tamper 512)	8086
IDDI (Data Intelligence)	8080
Connect	8082
Face Recognition Service	8084

For example, the Data Intelligence Service:

- http://<HostName/IPaddress>:<Port>
- http://172.31.72.111:8080

13. Select an option for **Database Transaction Persistence** and click **SAVE & CONTINUE**.

    If the settings are saved correctly, the message appears, **Global configurations are saved successfully**. If the data is incorrect or incomplete, provide the correct information before continuing.

14. On the **Configuration Files** tab, upload .json files for the listed services.

    Note the following:

    - ImageIntegrity json files can be uploaded directly, but you must modify the Mongo Database password in 2famongo json file before uploading.
    - Do not modify mongodb_database and mongodb_username in 2famongo.json.

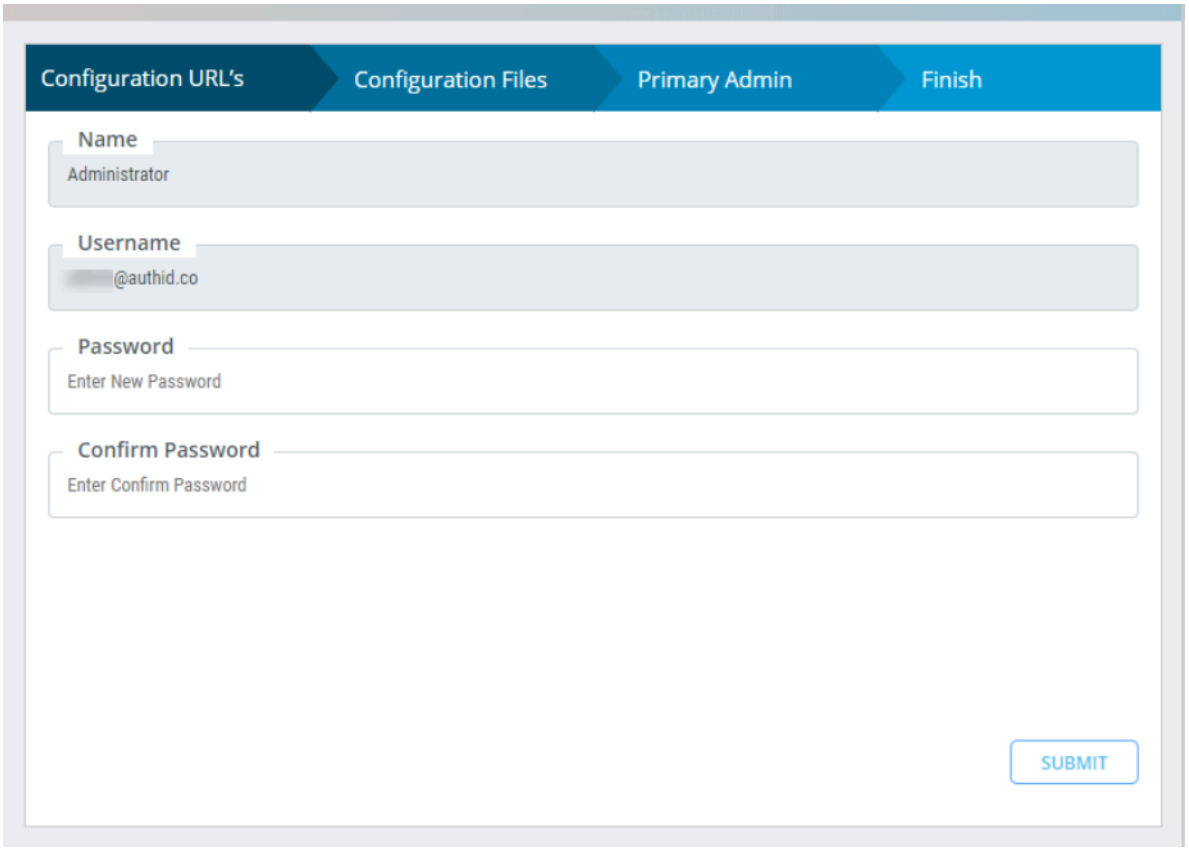
    Do the following:

    a. Click **UPLOAD FILE** to upload the .json file.

    b. Once you upload files for all the services, click **SAVE & CONTINUE**.

15. If the Data Intelligence and 2FA services are down, the **Please Start your Services** page appears. If so, do the following:
    a. Restart the API and 2FA services from the system.
    b. Return to the Setup Wizard and click **Refresh** to ensure the services are running.

    When the services are running, the **Primary Admin** tab appears.



16. Complete the **Primary Admin** tab as follows:
    a. Enter the administrator's password in the **Password** and **Confirm Password** fields.
    b. Click **SUBMIT**.

    If the configuration is successful, **CONGRATULATIONS Your Application is Configured Successfully** appears

17. On the **CONGRATULATIONS** page, click **Click here to Login** to log onto the **Data Intelligence Web UI** service.

18. If you had not set up the configuration URLs at the beginning of setup, do the following:
    a. Log onto the portal with user admin@authid.co,
    b. Navigate to **Settings** > **Global config**, and enter the remaining URL to submit changes:

    ```
    http://<hostname>:80/di
    ```

    c. Restart the data Intelligence API service after updating the URL in the global configuration with the command:

    ```
    $docker restart iddi
    ```

19. Disable two-factor authentication when logging into the portal. Do the following:
    a. Install mongodb-org-shell. See the following URL for instructions:
    https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/
    b. Run the following command to disable two-factor authentication. Replace the text in angle brackets with the appropriate values.

    ```
    $ mongo --ssl --sslAllowInvalidCertificates --authenticationDatabase admin
     --host <MongoDB Host IP>:<Mongo Port> -u master -p<MongoDB Password> --
    eval'db.getSiblingDB("dataIntelligence").getCollection("catfishidusers").
     update({"UserEmail" : "<Email address>"},{$set: {"MFAEnabled":
     NumberInt(0)}},{"multi" : false, "upsert" : false });'
    ```

Facial Recognition Server installation

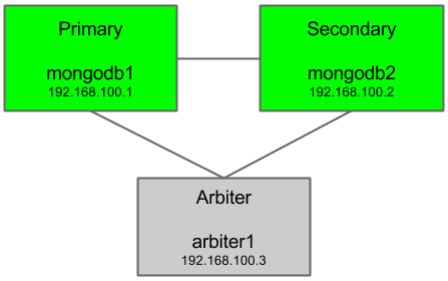
1. Connect to your server using SSH and navigate to the folder containing the Linux installation files.
2. Create a folder name `lic` on the server under `/opt/idmetrics/idverification/lic` and copy the FR license file you received to it.
3. Run the following command to install the Facial Recognition server. Replace the MAC address with the MAC address of the host.

   ```
   $sudo chmod +x Linux_install.sh
   $sudo bash -c "bash Linux_install.sh fr <mac_address> | tee fr_install.log"
   ```

   Note This step may take several minutes. The message "Installation Complete" will be displayed when the installation is finished.

High availability setup

Linux servers can be set up for high availability. You will need three server to set up the high availability configuration for MongoDB.



Then, follow these steps.

1. Run the following commands to enable outbound and read access:

```
setsebool -P httpd_can_network_connect 1

setsebool -P httpd_read_user_content 1
```

2. Connect to your server using SSH and navigate to the folder containing the Linux installation files.
3. Pull the MySQL docker image from the Docker hub and run using the following commands.

```
$sudo docker pull mysql:5.7 $firewall-cmd --permanent --zone=trusted --add-
port=3306/tcp

$firewall-cmd --reload

$docker run -dti -p 3306:3306 -v /data/mysql:/var/lib/mysql -e
 MYSQL_ROOT_PASSWORD=<password> --name mysql57 mysql:5.7
```

Note Log onto MySQL using the MySQL client and grant root user access from remote computer. For example:

```
grant all privileges on *.* to 'root'@'%' with grant option;
```

4. Set up MongoDB on the three servers you are using, Primary, Secondary, and Arbiter.
   a. Run the following commands on the Secondary server (node1.mongod server).

```
$sudo chmod +x HA_Linux_install.sh
```

```
$ sudo bash -c "bash HA_Linux_install.sh node1mongo master.mongod:<IPAddress>
 node1.mongod:<IPAddress> node2.mongod:<IPAddress> <dbpassword> | tee
 node1mongo.log"
```

Use the following values in these commands:

- For Add-Host:
  - Master.mongod (master server IP)
  - Node1.Mongod (node1 server IP)
  - Node2.mongod (Node2 server IP)
- For ENV variables:
  - dbpassword= MongoDB password (Only alphanumeric characters)

This step may take several minutes. The message **Installation Complete** appears when the installation is finished.

b. Run the following commands on the Arbiter server (node2.mongod server).

```
$sudo chmod +x HA_Linux_install.sh

$ sudo bash -c "bash HA_Linux_install.sh node2mongo node1.mongod:<IPAddress>
 node2.mongod:<IPAddress> master.mongod:<IPAddress> <dbpassword> | tee
 node2mongo.log"
```

Use the following values in these commands:

- For Add-Host:
  - Master.mongod (master server IP)
  - Node1.Mongod (node1 server IP)
  - Node2.mongod (Node2 server IP)
- For ENV variables:
  - dbpassword= MongoDB password (Only alphanumeric characters)

This step may take several minutes. The message **Installation Complete** appears when the installation is finished.

c. Run the following commands on the Primary server (master.mongod server).

```
$sudo chmod +x HA_Linux_install.sh
```

```
$ sudo bash -c "bash HA_Linux_install.sh mastermongo node1.mongod:<IPAddress>
 node2.mongod:<IPAddress> master.mongod:<IPAddress> <dbpassword> | tee
 mastermongo.log"
```

Use the following values in these commands:

- For Add-Host:
  - Master.mongod (master server IP)
  - Node1.Mongod (node1 server IP)
  - Node2.mongod (Node2 server IP)
- For ENV variables:
  - dbpassword= MongoDB password (Only alphanumeric characters)

This step may take several minutes. The message **Installation Complete** appears when the installation is finished.

d. Review the logs for any exceptions. Use the following command:

```
$ sudo docker logs tail 200 master
```

5. Log on to Node1 and Install mongodb-org-tools.

   See the documentation on the Mongo DB website for instructions on installing on Linux.

6. Copy the globalMerchantConfiguration.zip file and extract it.

7. Restore the Globalmerchantconfig date with the following command.

```
$mongorestore --ssl --sslAllowInvalidCertificates --host <masterMongoHostIP> --
authenticationDatabase=dataIntelligence
--username admin --password  <masterMongoDBPassword>
globalMerchantConfiguration
```

   Note the following:

   - Replace the values with the IP and password.
   - masterMonogoDBHostIP is the static IP of the system where the MongoDB Docker image is installed and running.

8. Execute the following commands to install the Linux Services Rectification, extraction, authentication and Tamper on the Redhat 1 and Redhat 2 nodes.

```
$sudo bash -c "bash HA_Linux_install.sh verification | tee verification.log"
```

9. Execute the following commands to install WebUI on the Redhat 1 and Redhat 2 nodes and Data retention on the Redhat 1 node.

   a. Run the following commands on the Redhat 1 node:

```
$sudo chmod +x HA_Linux_install.sh

$ sudo bash -c "bash HA_Linux_install.sh webservice master.mongod:<IPAddress>
 node1.mongod:<IPAddress> <dbpassword> | tee webservices.log"

$ sudo bash -c "bash HA_Linux_install.sh retention master.mongod:<IPAddress>
 node1.mongod:<IPAddress> <dbpassword> | tee retention.log"
```

   b. Run the following commands on the Redhat 2 node:

```
$sudo chmod +x HA_Linux_install.sh
```

```
$ sudo bash -c "bash HA_Linux_install.sh webservice master.mongod:<IPAddress>
 node1.mongod:<IPAddress> <dbpassword> | tee webservices.log"
```

Use the following values in these commands:

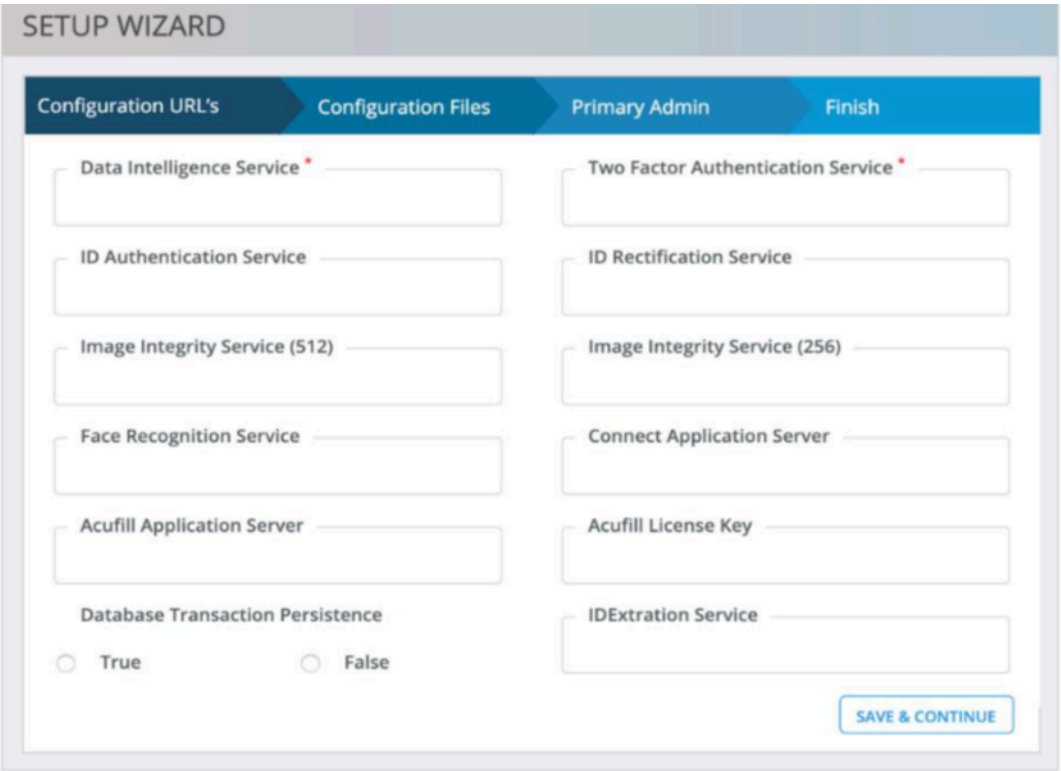
- For Add-Host:
  - Master.mongod= ipaddress(mongoDB master server ip address)
  - Node1.mongod= ipaddress(mongoDB node1 server ip address)
- For ENV variables:
  - dbuserpassword =(mongodb user passwd)

10. Locate the logs by using the following command:

```
$docker logs iddiweb
```

11. Configure all predefined endpoints by accessing the Data Intelligence at the following URL: http://<hostname/IPaddress>

The Setup Wizard appears.



12. On the **Configuration URL's** tab, enter the mandatory services data intelligence and two-factor authentication URLs.

Image	Port
IDR (Rectification)	443
IDA (Authentication)	443
IDE (Extraction)	443
ID2FA (Two Factor Authentication)	443

Image	Port
IDT256 (Tamper 256)	443
IDT512 (Tamper 512)	8443
IDDI (Data Intelligence)	8443
Connect	443
Face Recognition Service	443

Here are some examples for the Data Intelligence Service:

- https://<HostName>:<Port>
- https://datainteligence.kofax.com:8443

13. Select an option for **Database Transaction Persistence** and click **Save & Continue**.

    If the settings are saved correctly, the message appears, **Global configurations are saved successfully**. If the data is incorrect or incomplete, provide the correct information before continuing.

14. On the **Configuration Files** tab, upload the .json files for the listed services.

    Note the following:

    - ImageIntegrity .json files can be uploaded directly, but you must modify the Mongo Database password in the 2famongo .json file before uploading.
    - Do not modify mongodb_database and mongodb_username in 2famongo.json.
    - Update the mongodb host value to "master.mongod:27017,node1.mongod:27017" in 2famongo.json.

    Follow these steps:

    a. Click **Upload File** to upload a .json file.

    b. When you have uploaded files for all the services, click **Save & Continue**.

15. Execute the following commands to install the remaining services, IDDIAPI and 2FA, on the Redhat 1 and Redhat 2 nodes.

    a. Run the following commands on the Redhat 1 node.

    ```
    $sudo chmod +x HA_Linux_install.sh

    $ sudo bash -c "bash HA_Linux_install.sh masterApi master.mongod:<IPAddress>
     node1.mongod:<IPAddress> node1:<IPAddress> <dbpassword> http://
    <DataIntelligenceHostIP>:<DataIntelligencePort> | tee apinode.log"
    ```

    Use the following values in these commands:

    - The DataIntelligenceHost IP refers to the static IP address of the host system where the web services are installed. The port is 8080/8443.
    - For Add-Host:
      - Master.mongod= IPAddress (mongoDB master server ip address)
      - Node1.mongod= IPAddress (mongoDB node1 server ip address)
    - For ENV variables:
      - dbuserpassword =(mongodb user passwd)

    b. Run the following commands on the Redhat 2 node:

    ```
    $sudo chmod +x HA_Linux_install.sh
    ```

```
$ sudo bash -c "bash HA_Linux_install.sh masterApi master.mongod:<IPAddress>
 node1.mongod:<IPAddress> node1:<IPAddress> <dbpassword> http://
<DataIntelligenceHostIP>:<DataIntelligencePort> | tee apinode.log"
```

Use the following values in these commands:

- For Add-Host:
  - Master.mongod= IPAddress (mongoDB master server ip address)
  - Node1.mongod= IPAddress (mongoDB node1 server ip address)
  - Node1= IPAddress (DI master server IP address)
- For ENV variables:
  - dbuserpassword =(mongodb user passwd)

16. Review the logs for any exceptions. Use the following command:

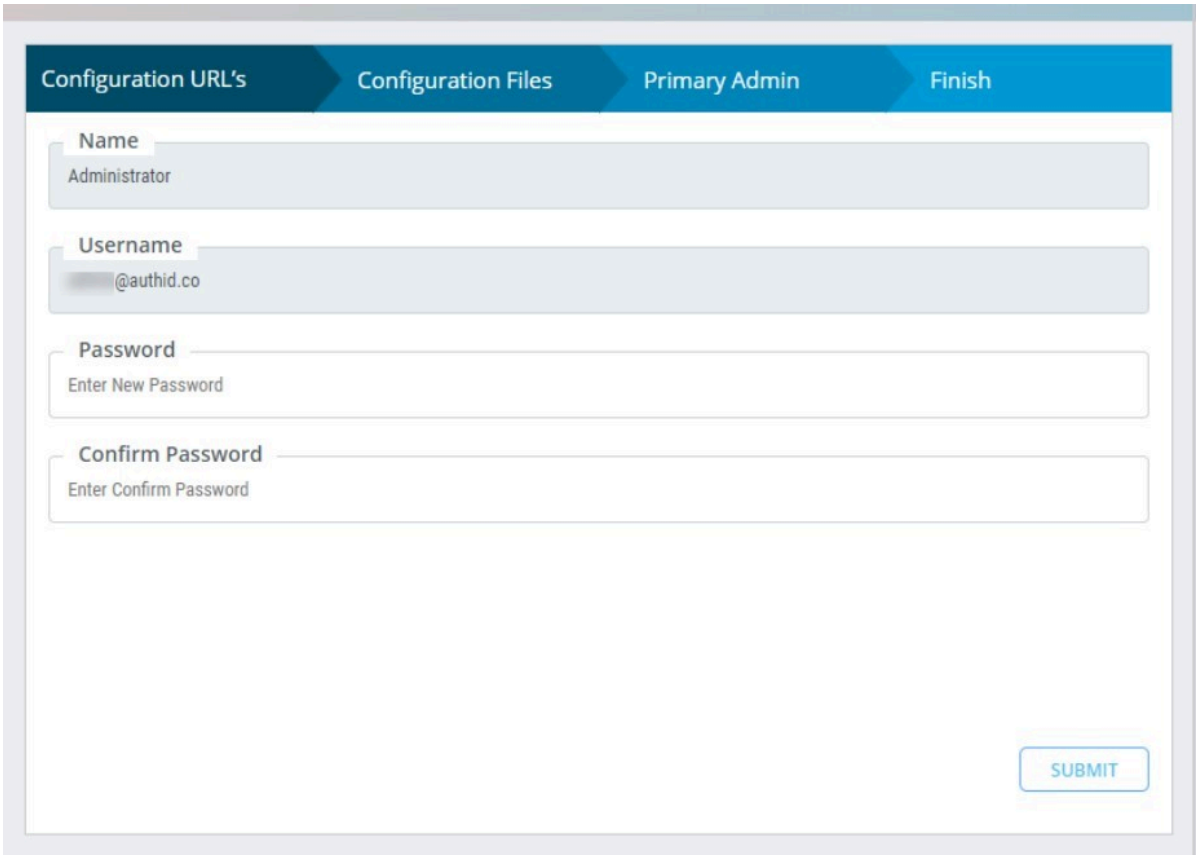
```
$ sudo docker logs --tail 200 iddi
```

17. If the Data Intelligence and 2FA services are down, the **Please Start your Services** page appears. If so, do the following:

a. Restart the API and 2FA services from the system.

b. Return to the Setup Wizard and click **Refresh** to ensure the services are running.

When the services are running, the **Primary Admin** tab appears.



18. Complete the **Primary Admin** tab as follows:
    a. Enter the administrator's password in the **Password** and **Confirm Password** fields.
    b. Click **Submit**.

    If the configuration is successful, **Congratulations, Your Application is Configured Successfully** appears.
19. On the **Congratulations** page, click **Click here to Login** to log onto the Data Intelligence Web UI service.
20. If you had not set up the configuration URLs at the beginning of setup, do the following:
    a. Log onto the portal as user admin@authid.co.
    b. Select **Settings** > **Global config**.
    c. Enter the remaining URLs to submit changes:

       http://<hostname>:80/di
    d. Restart the data Intelligence API service after updating the URL in the global configuration by using this command:

       ```
       $docker restart iddi
       ```
21. Disable two-factor authentication when logging into the portal by doing the following:
    a. Install mongodb-org-shell. See the following URL for instructions:

       https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/
    b. Run the following command to disable two-factor authentication. Replace the text in angle brackets with the appropriate values.

       ```
       $ mongo --ssl --sslAllowInvalidCertificates --authenticationDatabase admin
       --host <MongoDB Host IP>:<Mongo Port> -u master -p<MongoDB Password> --
       eval'db.getSiblingDB("dataIntelligence").getCollection("catfishidusers").
       update({"UserEmail" : "<Email address>"},{$set: {"MFAEnabled":
       NumberInt(0)}},{"multi" : false, "upsert" : false });'
       ```

Facial Recognition Server installation for high availability

1. Connect to your server using SSH and navigate to the folder containing the Linux installation files.
2. Create a folder name `lic` on the server under `/opt/idmetrics/idverification/` and copy the FR license file you received to it.
3. Run the following command to install the Facial Recognition server. Replace the MAC address with the MAC address of the host.

   ```
   $sudo chmod +x HA_Linux_install.sh
   $sudo bash -c "bash HA_Linux_install.sh fr <mac_address> | tee fr_install.log"
   ```

   Note This step may take several minutes. The message "Installation Complete" will be displayed when the installation is finished.

Connect Application Server

The required files are located in the provided `WindowsInstaller` folder.

1. Copy the contents of the `WindowsInstaller` folder to a folder on the server.
2. Launch PowerShell as the administrator, navigate to the destination folder specified during installation, and execute the following commands. Enter "y" when prompted.

   ```
   Set-ExecutionPolicy Unrestricted
   ```

3. Run the following command:

   ```
   .\pre-req.ps1
   ```

4. Run KofaxMobileIDVerificationConnectServer.exe from the folder above and follow the on-screen instructions. Note the following:
   - Enter mySQL hostIPAddress and Password details as showing in the instructions.
   - For UserName,enter "root".
5. Navigate to the destination folder specified during installation and execute the following batch file.

   ```
   .\setup.bat
   ```

6. Verify the installation was successful by browsing to `https://<fully qualified hostname>:8082/AssureIDService/ping`. The following should be returned. (This may take several minutes.)

   ```
   <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/"> AssureID
    Connect Automated Document Authentication Service Acuant Inc. Copyright (c) 2018
    - All Rights Reserved Web engine V2.2.0.24 Document library V19.6.26.113 Current
    server time: 2019-04-12T07:54:54.9826167Z
   </string>
   ```

7. Go to the Linux system and run the following command to update the connect database

   ```
   $ chmod 777 Connectsql.sh
   $ ./Connectsql.sh <mySQL HostIPAddress> <mySQL root Password>
   ```

Chapter 5

Docker support

Mobile ID Verification supports the execution of facial recognition and ID verification within Docker containers. This simplifies the deployment of complex configurations.

Docker installation for Windows

1. Enable the following Windows features and their subfeatures where applicable:
   - Containers
   - Hyper-V
   - IIS

   If you are using a virtual machine, open the Windows PowerShell as an Administrator and run the following commands:

   ```
   Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All -NoRestart

   Install-WindowsFeature RSAT-Hyper-V-Tools -IncludeAllSubFeature
   ```
2. Install Docker and enable virtualization in the BIOS settings.

   See the Docker website for instructions.
3. Change the Docker container to the Windows container.
4. Restart the computer.
5. In the Command Prompt window, verify that Docker is installed by running the following command:

   ```
   docker info
   ```

   Messages should indicate the information of the running Docker and that your installation is working correctly, and there are no errors.

Create Connect Server image

Create a Connect Server image to be used for Docker.

1. Copy the contents of the `WindowInstaller` folder to a folder on the server.
2. From the folder on the server, run KofaxMobileIDVerificationConnectServer.exe. Do the following:
   a. Select the option to configure for Docker.
   b. Follow the on-screen instructions to complete the installation.
   c. Note the destination folder to be used in the next step.

3. Copy all of the files installed by KofaxMobileIDVerificationConnectServer.exe from the destination `Deploy` folder and paste them into the `Connect` folder provided in the `docker` folder along with the software (`Docker\Connect\Connect`).
4. Open PowerShell as administrator and navigate to the `Docker\Connect` folder.
5. Build the Dockerfile with the following command:

   ```
   docker build -t connect .
   ```

   Where -t indicates the tag name. The period (.) indicates the current folder.

   Note Creating the containers for IIS and connect can take a long time. Any warnings that appear during installation can be ignored.

6. Create the Docker network by using the following command:

   ```
   docker network create -d nat --subnet=<ip>/24 --gateway=<ip> <networkname>
   ```

7. Run the Docker image with following command:

   ```
   docker run -it -net <networkname> --ip <staticIPAddress> -p <public port>:443 --
   name <containername> <Docker Image Name>
   ```

   The following is an example:

   ```
   docker run -it –net connectnet --ip 172.17.71.114 -p 4434:443 --name connectServer
    connect
   ```

   The example uses the following settings:
   - 443: Docker Container internal port
   - 4434: External port for accessing the connect service.
   - connectServer: Windows Docker container name.
   - connect: Image name.
   - connectnet: Network name.
   - 172.17.71.114: Docker static IP address. (If you get an error, provide another Docker support IP address.)
8. Run the following command to install the connect server in Docker.

   ```
   >docker exec connectServer powershell -command "c:\bin\install.ps1
    <staticIPAddress>"
   ```

   The following is an example:

   ```
   >docker exec connectServer powershell -command "c:\bin\install.ps1 172.17.71.114"
   ```

9. Install the .pfx-type certificate for the HTTPS port using the following commands:

   ```
   docker cp <cert.pfx> <ContainerName>: bin\
   docker exec <containerName> powershell -command "c:\bin\installCert.ps1 -Name

   connect -FileName 'c:\bin\<cert.pfx>' -Password <password> -Port 443"
   ```

10. Verify that the installation was successful by entering the following address in a web browser:

    ```
    https://<fully qualified hostname>:<External port>/AssureIDService/ping
    ```

    After several minutes, the following message should appear.

    ```
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
    AssureID Connect Automated Document Authentication Service Acuant Inc. Copyright
     (c) 2018 - All Rights Reserved Web engine V2.2.0.24 Document library V19.6.26.113
     Current server time: ……
    </string>
    ```

11. To get the connect logs, run the following command to copy the `logs` folder to the destination folder.

```
docker cp <ContainerName>:\Programdata\Assuretec\Logs\ <DestinationDirectoryPath>
```

Import and export Docker images

Do the following to import and export the Docker images. For more information, see the Docker website.

Export Docker images

Export the Docker container as an image .tar file with the following command:

```
docker save -o <fileName>.tar  <Repository>:<TAG>
```

For example:

```
docker save -o connectServer.tar  connect:latest
```

The Docker Image file created from the image.

Import Docker images

Import Docker images with the following command:

```
docker load -i <DockerImage.tar>
```

Where <DockerImage.tar> is the Docker Image file.

Connect Docker image build messages to ignore

While building connect Docker image, the following messages may appear. These can be safely ignored:

- add-windowsfeature : The request to add or remove features on the specified server failed. add-windowsfeature web-server -includeallsubfeature
- Copy-Item : Cannot find path 'C:\ProgramData\Assuretec\AssureID Document Library \DocumentLibrary.bak' because it does not exist. Copy-Item $ConnectDocumentLibrary $appData -Force
- Exception calling "AddAccessRule" with "1" argument(s): "This access control list is not in canonical form and therefore cannot be modified." At C:\bin\access.ps1:4 $acl.AddAccessRule($accessRule)
- PowerShell DSC resource '[WindowsFeature]NETCore' with SourceInfo 'C:\ProgramData\Assuretec \AssureID\ConnectInstall\InstanceSetup\baseconfig.ps1::45::8::WindowsFeature' threw one or more non-terminating errors while running the Set-TargetResource functionality. FullyQualifiedErrorId : NonTerminatingErrorFromProvider
- The PowerShell DSC resource '[WindowsFeature]HttpActivation' with SourceInfo 'C:\ProgramData \Assuretec\AssureID\ConnectInstall\InstanceSetup\baseconfig.ps1::51::9::WindowsFeature' threw one or more non-terminating errors while running the Set-TargetResource functionality. FullyQualifiedErrorId : NonTerminatingErrorFromProvider
- The request to add or remove features on the specified server failed. FullyQualifiedErrorId : DISMAPI_Error__Failed_To_Enable_Updates,Microsoft.Windows.ServerManager.Commands. AddWindowsFeatureCommand
- Restore-WDPackage : This access control list is not in canonical form and therefore cannot be modified. At C:\programdata\Assuretec\AssureID\ConnectInstall\ApplicationSetup \AssureIDConnectAdministration.psm1:1243 Restore-WDPackage -package $WebSitePackage -Parameters $packageP ...

- Set-WebConfiguration : There is no configuration defined for object at path IIS: \SslBindings. At C:\programdata\Assuretec\AssureID\ConnectInstall\ApplicationSetup \AssureIDConnectAdministration.psm1:2394 Set-WebConfiguration -Filter '/system.applicationHost/ serviceAuto ...
- Web-Lgcy-Mgmt-Console is not found on the target machine. FullyQualifiedErrorId : ProviderOperationExecutionFailure

Chapter 6

Upgrade

To upgrade Kofax Mobile ID Verification, do the following:

1. If a previous version of Kofax Mobile ID Verification and Kofax Mobile ID Facial Recognition is installed, follow the uninstallation instructions from this guide included with the previous version. All settings and transaction records are deleted.
2. Follow the steps in Installation to install the new version.

Chapter 7

Uninstallation

Connect Application Server

The required files are located in the provided `Uninstall` folder.

1. Navigate to `C:\ProgramData\Assuretec\AssureID\ConnectInstall\` and run uninstall.bat
2. Back up and delete the AssuredIDConnect database from the MySQL Server.
   This database is not removed automatically during uninstallation.
3. Run the KofaxMobileIDVerificationConnectServer.exe file used during installation and follow instructions to remove the program.
4. To verify uninstallation was successful, check the following items.
   - The site assureid_connect does not exist on IIS Manager
   - The folder `C:\inetpub\wwwroot\connect` does not exist
5. Remove the `ConnectInstall` folder from `C:\ProgramData\Assuretec\AssureID\` to delete the intermediate installer files.

Linux Servers

The required file is located in the provided `Uninstall` folder.

1. Connect to your server using SSH.
2. Copy the file Linux-Uninstall.sh to the server.
3. Navigate to the location of the Linux-Uninstall.sh file and execute the following commands:

```
$sudo chmod +x Linux-Uninstall.sh
$sudo bash Linux-Uninstall.sh
```

Note This procedure does not remove the MySQL Docker container because it is not installed by the installer. Additionally, the MongoDB database and MySQL are not deleted. Back these up before uninstalling. You can find database data in the `/data` partition.

Connect uninstallation messages to ignore

For uninstallation, the following messages can be ignored.

- Stop-Process : Cannot find a process with the name "FREngineService". Verify the process name and call the cmdlet again. At C:\Temp\Umaster.ps1:126 char:2
- Stop-Process : Cannot find a process with the name 'redis-server'. Verify the process name and call the cmdlet again. At C:\Temp\Umaster.ps1:127 char:2

Chapter 8

Licenses

In order to fully license your Mobile ID Verification and Mobile ID Facial Recognition products, Kofax requires some additional information about your system.

Gathering system information

Facial Recognition Servers

Facial Recognition is a separately licensed product from Kofax. Please contact your sales representative if you are unsure if you are entitled to use this feature . To license the Facial Recognition component, a FR_MAC.txt file needs to be generated for Facial Recognition server in your environment and provided to Kofax.

Follow these steps on each Facial Recognition server:

1. Connect to your server using SSH.
2. Navigate to the folder containing the Linux installation files and run the following command.

   ```
   sudo bash -c "bash Linux_install.sh getMACAddress"
   ```

   A FR_MAC.txt file should be generated after completion of the Facial Recognition server installation.
3. Copy the generated FR_MAC.txt file in preparation to send to Kofax.

Sending to Kofax

Once you have collected the FR_MAC.txt file (if applicable), submit it to Kofax using the information provided in your order confirmation email.

Applying the licenses

After submitting your information to Kofax, you will receive one or more license files. These files need to be applied to the system in order for it to function.

Facial Recognition

Facial Recognition is a separately licensed product from Kofax. Contact your sales representative if you are unsure if you are entitled to use this feature.

To apply your Facial Recognition license(s), follow these steps on each Facial Recognition server.

1. Connect to your server using SSH.
2. Copy the `ROC.lic` file to the following folder: `/opt/idmetrics/idverification/lic/`.
3. Restart the Facial Recognition server.

Chapter 9

Logging

If you need to refer to logs for each server, check the following files.

Type	Location
Rectification Server Log	`/opt/idmetrics/idrectification/logs/IDRectification.log`
ID Authentication Service Log	`/opt/idmetrics/idauthentication/logs/IDAuthentication.log`
Facial Recognition Server Log	`/opt/idmetrics/idverification/logs/VerificationService.log`

Tamper Server Logs	Tamper: 256:
	`/opt/idmetrics/idimageintegrity/service256/logs/application.log`
	`/opt/idmetrics/idimageintegrity/tamper256/logs/gunicorn_stdout.log`
	Tamper: 512:
	`/opt/idmetrics/idimageintegrity/service512/logs/application.log`
	`/opt/idmetrics/idimageintegrity/tamper512/logs/gunicorn_stdout.log`

ID Extraction Service Logs	`/opt/idmetrics/idextraction/logs/IDExtraction.log`
Data Intelligence Service Logs	`/opt/idmetrics/iddataIntelligence/logs/application.log`
Connect Server Logs	`C:\ProgramData\Assuretec\Logs`
2FA Server Logs	`/opt/idmetrics/id2fa/logs/application.log`
Data Intelligence Web Logs	`/opt/idmetrics/iddiweb/logs/`
Data Retention Logs	`/opt/idmetrics/iddireten/logs`

Chapter 10

Tests for services

Use the following URLs to test services. For each of these URLs, replace <hostname>:<port> with the name or IP address of the server and the required port number.

Connect

URL: `http://<hostname>:<port>/AssureIDService/ping`

Expected result: Displays assure ID connect information.

Webui

URL: `http://<hostname>:80`

Expected result: Displays the setup or log on screen.

Authentication

URL: `http://<hostname>:8081/IDAuthenticationService/getVersion`

Expected result: Displays the IDA version.

Extraction

URL: `http://<hostname>:8082/IDExtractionService/getVersion`

Expected result: Displays the IDE version.

Data Intelligence

URL: `http://<hostname>:8080/`

Expected result: Displays the IDDI version.

Rectification

URL: `http://<hostname>:8083/IDRectificationService/getVersion`

Expected result: Displays the IDR version

Tamper

Tamper 256 URL: `http://<hostname>:8085`

Tamper 512 URL: `http://<hostname>:8086`

Expected result: Displays the Tamper version.

2FA

URL: `http://<hostname>:8088`

Expected result: Displays the TwoFactorAuthentication service version.

FR

URL: `http://<hostname>:8084`

Expected result: Displays the FR service version.