

## OpenText™ Intelligent Capture

### **Cloud Deployment Guide**

This guide is intended for users who need to deploy Intelligent Capture Web Client/REST Services or Real Time Services using Docker Compose. It also describes how to deploy Intelligent Capture to cloud services.

ECPCORE220300-ICD-EN-01

---

**OpenText™ Intelligent Capture  
Cloud Deployment Guide**  
ECPCORE220300-ICD-EN-01  
Rev.: 2022-Apr-14

**This documentation has been created for OpenText™ Intelligent Capture CE 22.3.**  
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

**Open Text Corporation**

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <https://support.opentext.com>

For more information, visit <https://www.opentext.com>

**Copyright © 2022 Open Text. All Rights Reserved.**

Trademarks owned by Open Text.

Adobe and Adobe PDF Library are trademarks or registered trademarks of Adobe Systems Inc. in the U.S. and other countries.

One or more patents may cover this product. For more information, please visit <https://www.opentext.com/patents>.

**Disclaimer**

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

---

# Table of Contents

<b>1</b>	<b>Deploying Capture Web Client (CWC) and REST Services subsystems containers using Docker Compose .....</b>	<b>5</b>
1.1	Prerequisites for deploying the Web Client/REST Services using Docker compose .....	5
1.2	Container image tags .....	6
1.3	Pulling and installing the Docker image into a container registry .....	7
1.4	Deploying the Web Client/REST Services Docker image .....	8
1.5	Creating a custom Docker image .....	8
1.5.1	Building Docker images from Docker Compose artifacts .....	8
1.6	Configuring logging .....	8
1.7	Docker Compose environment file parameter reference .....	9
1.8	Troubleshooting .....	10
1.8.1	The Recognition project path is not found .....	10
1.8.2	Unable to load the Capture Web Client (CWC) Web login page .....	11
<b>2</b>	<b>Deploying Intelligent Capture on Microsoft Azure .....</b>	<b>13</b>
<b>3</b>	<b>Deploying Intelligent Capture on the Google Cloud Platform .....</b>	<b>15</b>
<b>4</b>	<b>Deploying Intelligent Capture on Amazon Web Services ....</b>	<b>17</b>



## Chapter 1

# Deploying Capture Web Client (CWC) and REST Services subsystems containers using Docker Compose

You can deploy the Intelligent Capture Web Client/REST Services Docker image using the provided Docker Compose file.

Real Time Services is a product offering based on Intelligent Capture REST Services, which are a set of RESTful web service interfaces that custom client applications can use to call the services of the Intelligent Capture server or the Module server. An example of an Intelligent Capture REST Services client is Intelligent Capture Web client.



**Note:** For brevity, Web Client/REST Services hereafter refers to both Intelligent Capture Web Client/REST Services and Real Time Services.

After installing the Web Client/REST Services chart and Docker image, use Intelligent Capture Designer to create and upload capture profiles and artifacts to the Intelligent Capture Server. The REST Services regularly pulls the capture profiles and artifacts from the Intelligent Capture server. For more information, see *OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)*. You can use Intelligent Capture Administrator to install licenses.

The Web Client and REST Server as well as the Module server and Module Service instances run as docker containers. The other Intelligent Capture components, Intelligent Capture Server, SQL Server (optional), Designer, Administrator, and the shared data folder reside on their own virtual machines.

## 1.1 Prerequisites for deploying the Web Client/REST Services using Docker compose

The following software must be installed and configured before deploying the Web Client/REST Services Docker image:

- gMSA (group Managed Service Account)  
Intelligent Capture runs on Windows only and thus Microsoft gMSA is required for domain access. You must add all Windows nodes to the same gMSA (which will be in the same Active Directory domain) by creating a gMSA credential spec that references the gMSA account and then creating the gMSA credential spec on each Windows node.
- Windows servers for the Web Client/REST Server and Module server nodes.

- Windows Server 2019 Long-Term Servicing Channel (LTSC) is required. For more information, see <https://docs.microsoft.com/en-us/windows-server/getstarted/servicing-channels-comparison#long-term-servicing-channel-ltsc>.



**Note:** Hyper-V and Containers Windows features must be enabled.

- Intelligent Capture

The following Intelligent Capture components must be installed and set up. For more information, see *OpenText Intelligent Capture - Installation Guide (ECPCORE-IGD)* and the *Intelligent Capture Release Notes* (available in My Support (<http://support.opentext.com>)).

- Intelligent Capture server



**Note:** SQL Server is optional.

- Web Client/REST Services must be configured as follows:

For more information, see *OpenText Intelligent Capture - Installation Guide (ECPCORE-IGD)*.

- Shared data folder
  - The REST Server must be able to connect to the shared data folder.
  - The gMSA account must have read/write/delete/create permissions for the shared data folder.
- The REST Server must be able to connect to the Intelligent Capture server. Specify the Intelligent Capture server connection properties by running Start > OpenText Intelligent Capture > REST Service Config and specifying the connection properties on the Intelligent Capture server tab. (The connection properties are saved to the `server.config` file, which resides in the shared data folder.)
- Intelligent Capture Designer
- Intelligent Capture Administrator

## 1.2 Container image tags

Container images in the OpenText Container Registry are identified by the base container image name and a tag (version). You can obtain container images using two types of tags:

---

### YY.Q

In a tag of this style, the dot-separated numbers represent the year and quarter. This image tag identifies the latest version of a container image available in the OpenText Container Registry.

Use a <YY.Q> tag to obtain the latest image available that has the most recent patches and configurations. To download the latest available version of the `otic-cwc-server` container image, use the following command:

```
docker pull registry.opentext.com/<container_image_server>:<YY.Q>
```

where *<container\_image\_server>* represents the name of the Intelligent Capture web client server and *<YY.Q>* represents the year and quarter of the release to obtain. For example:

```
docker pull registry.opentext.com/otic-cwc-server:22.1
```

#### Notes

- OpenText recommends that you use the *<YY.Q>* tag in your docker pull command to make sure that you have the latest available image including all patches available at the time the image was built.

## 1.3 Pulling and installing the Docker image into a container registry

To pull the Web Client/REST Services and Module Server Docker images from the OpenText Container Registry and install them into your container registry, use the following sequence of commands:

```
docker pull registry.opentext.com/otic-cwc-server:<version tag>
docker pull registry.opentext.com/otic-module-server:<version tag>
docker tag<your_container_registry_url>/<your_web_client_rest_services_image>/
docker push<your_container_registry_url>/<your_web_client_rest_services_image>/
```

where:

***<version tag>***

The version tag of the Web Client/REST Services and Module Server Docker images in the OpenText Container Registry. For example:

22.1

***<your\_container\_registry\_url>***

The URL to your container registry.

***<your\_web\_client\_rest\_services\_image>***

The name of the Web Client/REST Services Docker image in your container registry.

## 1.4 Deploying the Web Client/REST Services Docker image

To deploy the Web Client/REST Services, specify your Web Client/REST Services configuration in the `.env` file and then use the Docker Compose command.

Example:

```
docker-compose up -d --no-build
```

## 1.5 Creating a custom Docker image

The Web Client/REST Services Docker image is also provided as an archive file. You can create a custom Docker image using the Docker file and artifacts. Download the archive file from <https://registry.opentext.com/otic-cwc-server.tar.gz>.

After you have created your custom Docker image, to install it into your own container registry, execute the following sequence of commands:

```
docker load <your_image.tar.gz>
docker tag <your_container_registry_url>/<your_web_client_rest_services_image>
docker push <your_container_registry_url>/<your_web_client_rest_services_image>
```

### 1.5.1 Building Docker images from Docker Compose artifacts

To build Docker images from Docker Compose artifacts, download the `Intelligent_Capture_22.1_docker_compose_rest_cwc.zip` file from MySupport, and execute the following command from the root folder:

```
docker-compose -f docker-compose.yml -f build.yml build
```

## 1.6 Configuring logging

The following relevant logs are sent to `stdout` and `stderr` streams:

- Modular Server and Module Services Windows Services application events
- Web Client/REST Services IIS logs
- Cluster and node logs

You can save the logs to a network share using the `<SHARED_LOG_FOLDER>` Docker Compose environment variable. The `gMSA` account must have `read/write/delete/ create` permissions for the folder specified in `<SHARED_LOG_FOLDER>`.

## 1.7 Docker Compose environment file parameter reference

This section describes the Docker Compose environment parameters included in the file (.env) specific to Web Client/REST Services and Module server.

### ***CERT\_FILE\_NAME***

The PFX format of the certificate file name with extension.

### ***CERT\_FOLDER***

The path to the folder containing the HTTPS certificates.

### ***CREDENTIAL\_SPEC\_FILE***

Name of the JSON file containing gMSA account information. By default it is located in the following folder: C:\ProgramData\docker\credentialspecs

### ***CWCPORT***

The port for HTTP/HTTPS requests. Default: 443.

### ***CWCURL***

The Web Client/REST Services IIS web application URL used for JSON requests.

### ***DOCKER\_REG***

The name of the container registry of Web Client/REST Services and Module server Docker images.

### ***ICTAG***

Any arbitrary string that can be used to further specify the image as specified in Web Client/REST Services and Module server images. Default: latest, which means the most recent image.

### ***SHARED\_DATA\_FOLDER***

The path to the shared data folder where server configuration information resides. This folder contains temporary image capture files and other state information as well as a shared configuration file.

### ***SHARED\_LOG\_FOLDER***

The path to the folder in which all logs are stored. Otherwise, the logs are stored in their respective directories. For information, see “Configuring logging” on page 8.

### ***configure***

The Configure module service.

***cwc-server***

The Web Client/REST Services.

***extraction***

The Extraction module service.

***fpocr***

The full-page OCR module service.

***image-converter***

The Image Converter module service.

***image-processor***

The Image Processor module service.

***scripting***

The Scripting module service.

## 1.8 Troubleshooting

This section describes troubleshooting scenarios for deploying Capture Web Client (CWC) and REST Services subsystems containers using Docker Compose.

### 1.8.1 The Recognition project path is not found

#### Symptom

The following error message is displayed:

```
Path Project path not found for project: . Extraction project shared directory.
```

This error is displayed in the Web Client and is logged in the trace file on invoking the Extraction Service module.

#### Possible cause

File not found. This affects classification and extraction imaging services.

#### Solution

Ensure the gMSA has access to the Recognition project folder. You must also copy the Recognition project folder to the shared data folder.

## 1.8.2 Unable to load the Capture Web Client (CWC) Web login page

### Symptom

The following error message is displayed in the log:

```
Check Shared Directory Write Access .....
```

This error is displayed in the Web Client and is logged in the trace file on invoking the Extraction Service module.

### Possible cause

The gMSA service account does not have sufficient permissions for the Shared Data folder.

### Solution

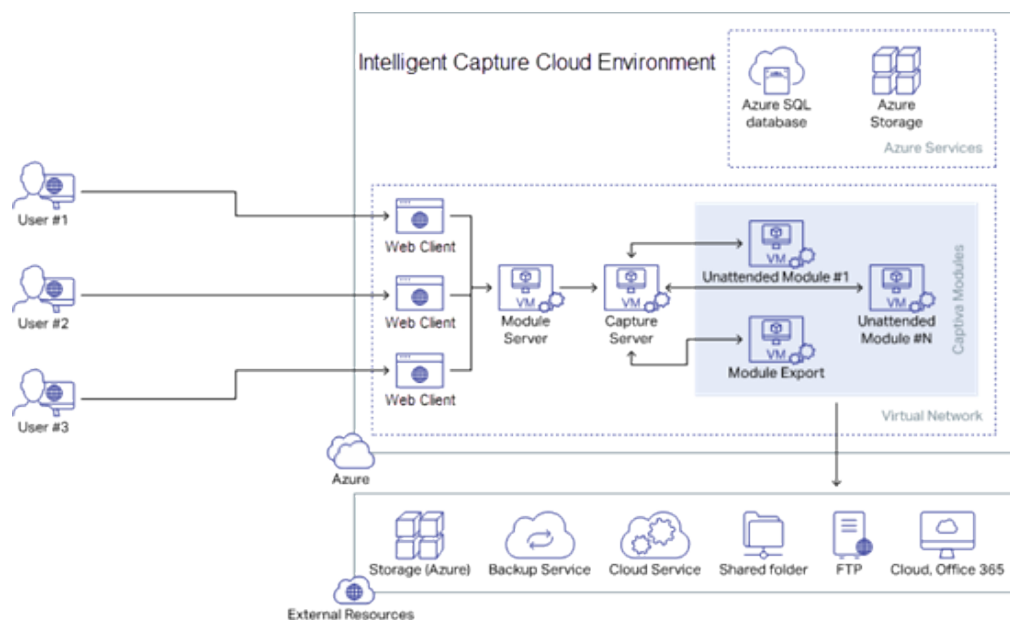
Ensure the gMSA service account has read, write, delete, and create permissions to the Shared Data folder.



## Chapter 2

# Deploying Intelligent Capture on Microsoft Azure

The following diagram shows a typical example of an Intelligent Capture architecture on Microsoft Azure. With this all-in-one approach, the Intelligent Capture Server and all components are installed on Microsoft Azure virtual machines. Users connect to the system using the Remote Desktop Protocol (RDP) for configuration and Intelligent Capture Web Client for working with the system.



**Figure 2-1: Intelligent Capture deployment on Microsoft Azure**

Prepare the Microsoft Azure environment before deploying Intelligent Capture on Microsoft Azure.

### To deploy Intelligent Capture on Microsoft Azure:

1. Prepare the Microsoft Azure environment by performing the following tasks in Microsoft Azure:
  - a. Create a resource group.
  - b. Create a virtual network.
  - c. Create a storage account.
  - d. (Optional) If the Intelligent Capture Server is going to use an external SQL database instead of the file-based internal database, create a SQL database.

- e. Create all the virtual machines required for the Intelligent Capture Server and client modules.

For each client module, set a public IP address.

2. On each virtual machine, run the Intelligent Capture installer.



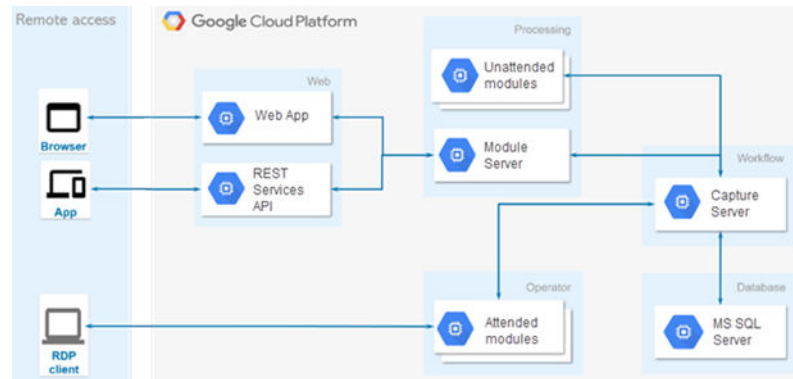
**Tip:** You can also perform the installation from the command line.

For more information, see the *OpenText Intelligent Capture Azure Cloud Deployment Guide* white paper on My Support (<http://support.opentext.com>).

## Chapter 3

# Deploying Intelligent Capture on the Google Cloud Platform

Intelligent Capture can be fully deployed on the Google Cloud Platform (GCP). With this all-in-one approach, the Intelligent Capture Server and all its components are installed on GCP virtual machines. Although operators can run attended modules such as Identification and Completion via the Remote Desktop Protocol (RDP), it is a best practice for operators to use Intelligent Capture Web Client or the REST Services API to perform capture processing.




**Figure 3-1: Intelligent Capture deployment on Google Cloud platform**

### To deploy Intelligent Capture on GCP:


1. Prepare the GCP environment by performing the following tasks in GCP:
  - a. Create a project.
  - b. Create a virtual network also known as a **VPC** (Virtual private cloud) network.
  - c. Create virtual machines (also known as **Compute engine VM instances**).

It is recommended that each of the following components reside on its own dedicated virtual machine:

- Intelligent Capture Server
- Microsoft SQL Server (for the external Microsoft SQL Server option, which is recommended instead of the internal database option)
- Each high-load module, such as Image Converter or an OCR module

 **Note:** For each virtual machine, use the Intelligent Capture component's recommended hardware configuration.

- d. Specify the networking parameters for the connection of VMs to the VPC network.
2. On each virtual machine, mount the installation image and run the installer, following the recommended order of installation for the components.

 **Tip:** You can also perform the installation from the command line.

You must install Intelligent Capture components that use a graphical user interface (GUI), such as Administrator, on a virtual machine operating system with a GUI.

3. (Optional) If you are migrating existing capture processes, convert them to use Intelligent Capture Web Client as follows and then deploy them to your Intelligent Capture GCP environment:
  - Replace the ScanPlus module with the Standard Import module and create a corresponding **Distributed Capture** profile.
  - As appropriate, convert the Identification, Completion, Extraction, and Standard Export modules to **Distributed Web Client** modules and create corresponding **Distributed Capture** profiles.

For more information, see *OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)*.

 **Notes**

- Web Client supports JavaScript client scripting but not .NET scripting.
- If a capture process uses the *OutputXMLData* IA value (from the Classification module), then migrate to the Distributed Web Client module and **Distributed Capture** profile to perform both document import and classification. Web Client does not support *OutputXMLData* IA value.

For more information and examples, see the *OpenText Intelligent Capture Google Cloud Deployment Guide* white paper on My Support (<http://support.opentext.com>). This white paper also describes migrating existing on-premise capture processes to run on GCP.

## Chapter 4

# Deploying Intelligent Capture on Amazon Web Services

Intelligent Capture can be fully deployed on Amazon Web Services (AWS). To perform any actions in AWS, you must have an established AWS project and an operator familiar with Web Services procedures must be logged in to AWS with valid credentials.

 **Note:** Before you deploy Intelligent Capture software on the virtual machines (VM), ensure you have created all the required cloud resources.

As a best practice, install Intelligent Capture in the following sequence:

- Intelligent Capture Database
- Intelligent Capture Server
- Client modules
- Web components

 **Note:** The Intelligent Capture installation requires Microsoft .NET Framework 4.8 which must be installed prior to the installation of any Intelligent Capture component. A VM image doesn't include the preinstalled .NET Framework 4.8. To download the .NET Framework installation package from the Microsoft download site, go to <https://www.microsoft.com/en-us/download/details.aspx?id=56116>.

### To deploy Intelligent Capture on AWS:

1. Copy the Intelligent Capture installation image to all VMs.
2. Mount the Intelligent Capture image to the virtual drive.
3. Open RDP with the target VM, go to the directory with the Intelligent Capture image.
4. Launch PowerShell from the command line.
5. From the PowerShell command line, type the command for mounting the ISO image as a virtual drive. `Mount-DiskImage -ImagePath "C:\Captiva\Captiva-20.2.0.iso"`

For more information and examples, see the *OpenText Intelligent Capture Amazon Web Services Deployment Guide* white paper on My Support (<http://support.opentext.com>). This white paper also describes migrating existing on-premise capture processes to run on AWS.

