



myInsight for Documentum

myInsight Cloud Deployment Guide

Contents

- 1. Version History..... 3**
- 2. Product Description..... 4**
 - 2.1. Document Description..... 5
- 3. Deploying myInsight in a Kubernetes environment..... 6**
 - 3.1. Planning the installation..... 6
 - 3.2. Requirements..... 6
 - 3.3. Deploying myInsight-installer pod..... 6
 - 3.4. Configure Documentum to include myInsight..... 7
- 4. How it works..... 9**
 - 4.1. Persistent Volume..... 9
 - 4.2. Pods..... 9
 - 4.2.1. myinsight-installer..... 9
 - 4.2.2. dcs-pg..... 9
 - 4.2.3. d2client..... 10
 - 4.3. D2 Ingress..... 10
- Index..... 11**

1. Version History

Date	Changes	Version number
26-March-2021	First version of the myInsight Cloud Deployment Guide	1.0
30-March-2023	Update for myInsight 8.0.	1.1
22-September-2023	Update for myInsight 8.1.	1.2

2. Product Description

With myInsight for Documentum, end-users can request reports from predefined report definitions. They can see the reports displayed on their computer screen or receive reports automatically in their e-mail or at a specified location inside or outside the Documentum repository. The preferred format can be chosen by the end-user, without the end-user needing any knowledge about DQL, HTML or XSL.

myInsight for Documentum categorizes functionalities according to the user's role in the report generator. There are 3 predefined roles:

- Users in the *Report User* role, which supplies predefined reports from the system, require no knowledge of either DQL or style sheets.
- Users in the *Report Administrator* role can schedule reports so that they are generated automatically at a predefined time and location.
- Users in the *Report Builder* role can define new report specifications by configuring the DQL statements, and they can identify and compose the desired style sheets.

myInsight for Documentum can be accessed by anyone who has been given one of these default roles.

myInsight for Documentum can produce output in any format that can be generated using XSL style sheets. For example: reports can be presented in PDF, HTML, text file, Microsoft Excel spread sheet or Microsoft Word format. myInsight for Documentum can also e-mail the output file automatically. In this case the recipient does not need to be a Documentum user and can even be someone from outside the organisation.

myInsight for Documentum is integrated into Documentum Webtop, Documentum Administrator, Documentum D2 and Documentum xCP, in line with the corporate philosophy of Documentum. This enables end users to work in an environment that they are already familiar with. Due to its full integration within the Documentum environment, no additional components need to be installed on the end user's local machine.

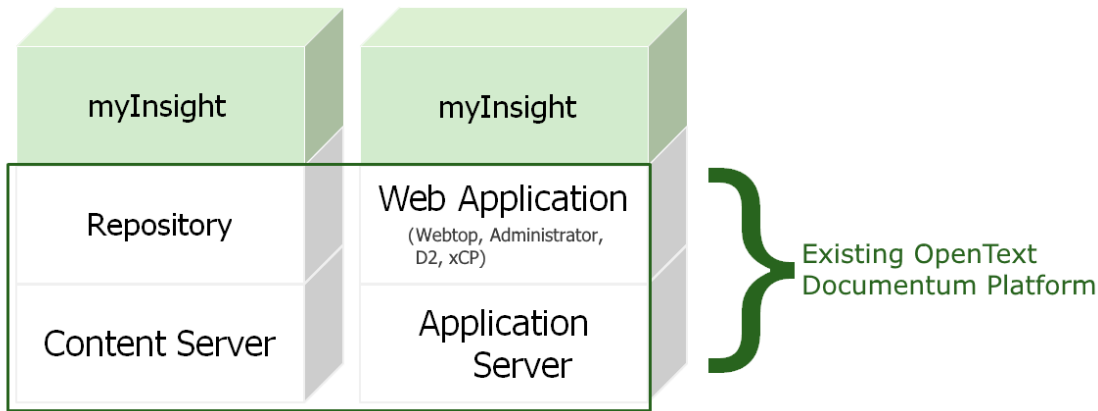


Figure 1: myInsight integration into the OpenText Documentum platform

As shown in the previous figure, myInsight for Documentum components are located on both Repository level as Web Application Level. Both use standard OpenText | Documentum subcomponents for its functionality.

As of version 7.0, myInsight for Documentum can be installed separate from the Content Server.

In December 2015 eurosript has been renamed to Amplexor. The product eurosript Documentum Report Generator has been renamed to myInsight for Documentum. In 2020, Amplexor joined the Acolad Group under the name Acolad Digital. In October 2023, Acolad Digital became AmeXio.

For the reader's convenience, the abbreviation 'myInsight' will be used in this document, instead of the full product title 'myInsight for Documentum'.

2.1. Document Description

This document is intended for System Administrators who are installing or upgrading myInsight on a Kubernetes cloud environment.

Installing myInsight using Kubernetes is supported on Documentum D2 cloud installations which were installed using the Kubernetes Helm charts provided by OpenText.

All cloud providers supported for Documentum D2 by OpenText are also supported for myInsight.

The myInsight Kubernetes Helm charts and scripts are similar to the ones from OpenText.

3. Deploying myInsight in a Kubernetes environment

3.1. Planning the installation

Consider the following when deploying myInsight in the cloud.



Note

When myInsight is added to an already running environment the Documentum pods will be restarted.

3.2. Requirements

- myInsight must be installed in the same namespace as Documentum D2
- myInsight needs one ReadWriteMany Persistent Volume in addition to the ones already required by Documentum. Default size for the PV is 1 Gi. This PV needs to be available before installation.
- The myInsight-installer Docker image which can be downloaded from the website must be available in a local registry.

3.3. Deploying myInsight-installer pod



Note

The license file is required the first time you install myInsight but can be omitted on upgrades.

1. Extract the myInsight HelmCharts tar.

2. Modify the values.yaml

- Update the repository key with the location of the Docker repository you are using.
- Update the storageClass key with the storage class that should be used for the PV.
- To install a myInsight license set the import value to true and specify the copy the content of the content of the license file to the licenseKey.

```
license:
  import: true
  licenseKey: | <base64 encoded content of the license file>
```

3. Deploy the helm chart using the following command.

```
helm -n <namespace used for D2> install myInsight-installer
```

3.4. Configure Documentum to include myInsight

1. Update the d2/values.yaml from the D2 Helm charts used to deploy D2

- Change the value of content-server.customer.scriptExecute to true

```
### Custom Script Execution ###
custom:
  scriptExecute: true
```

- Change the value of d2client.custom.scriptinPV to true
- Change the value of d2.client.custom.markerFiles to include "myInsight-installer-copy-succeeded-7.4"

```
custom:
  scriptinPVC: true
  scriptPVCname: d2client-shared-pvc
  PVCSubPath: customscripts
  ### Provide the marker file names of client products in
  any order separated by comma. Before CS copying the custom
  scripts, these marker files presence are verified.
  # e.g. markerFiles: CustomApp1-installer-copy-succeeded-1
  markerFiles: myInsight-installer-copy-succeeded-7.4
```

- Add the extra myInsight path to the d2ingress configuration

```
d2ingress:
  enabled: true

  ingress:
    hostShortName: d2-ingress
    hostDomainName: *ingress_domain
    proxyBodySize: 8m
    proxyConnectTimeout: 300
    proxySendTimeout: 300
    proxyReadTimeout: 300

    # List of valid values are CFCR, Azure, AWS, GCP
    cloudPlatform: CFCR

    extraPaths:
      - backend:
          serviceName: d2client
          servicePort: 8080
          path: /myInsight
```

2. Update the d2/dockerimages-values.yaml from the D2 Helm charts used to deploy D2

- Change the value of content-server.customer.markerFiles to include "myInsight-installer-copy-succeeded-7.4"

```
custom:
  versions: d2.20.4.0000.0240
  ### Provide the marker file names of client products in
  any order separated by comma. Before CS copying the custom
  scripts, these marker files presence are verified.
  markerFiles: d2-copy-succeeded-20.4.0000.0240,myInsight-
  installer-copy-succeeded-7.4
```

3. Run the helm command to install or upgrade the D2 deployment. Refer to the OpenText documentation for more information.

4. How it works

This chapter describes how the Docker image and the Helm chart deploy myInsight to the Documentum pods.

4.1. Persistent Volume

The myInsight deployment for Kubernetes uses two Persistent Volumes:

1. The shared-pvc Persistent Volume which is accessed by the dcs-pg pods and created by the d2installer deployment.
2. The d2client-shared-pvc Persistent Volume which is accessed by the d2client pods and created by the myInsight-installer deployment.

4.2. Pods

myInsight gets deployed to the dcs-pg and d2client pods. To do this, a myinsight-installer pod is deployed to the same namespace as Documentum.

4.2.1. myinsight-installer

This is a small pod similar to the d2installer pod which copies the myInsight installation software and configurations to the d2client-shared-pvc and the shared-pvc Persistent Volumes.

A marker file named "myInsight-installer-copy-succeeded-7.4" is also placed on the Persistent Volumes.

4.2.2. dcs-pg

This is the Content Server pod from OpenText Documentum. Because the dockerimagesvalues.yaml file was updated this pod will wait for the myInsight marker file to be present on the shared-pvc Persistent Volume before continuing deployment.

After the Content Server and D2 installations have been completed the myInsight script 30myinsightcsdeploy.sh will run.

On the first dcs-pg pod (dcs-pg-0) this script will check which version of myInsight is installed in the repository and will install or upgrade myInsight if needed.

On the (optional) other HA dcs-pg pods the script will exit immediately.

4.2.3. d2client

This is the pod from OpenText Documentum with the D2 webapplication. Because the values.yaml file was updated, this pod will wait for the myInsight marker to be present on the d2client-shared-pvc Persistent Volume before continuing deployment.

After the D2 webapplication is configured, the myInsight script 10myinsightwebappdeploy.sh will run.

This script will install the myInsight web application to the same Tomcat instance as the D2 webapplication is installed on.

4.3. D2 Ingress

The D2 Ingress configuration is updated to include the /myInsight path, to make myInsight available on the same URL as D2.

Index

A

Acolad [5](#)
AmeXio [5](#)
AMPLEXOR [5](#)

C

Configure Documentum to include myInsight [7](#)

D

D2 Ingress [10](#)
d2client pod [10](#)
dcs-pg pod [9](#)
Deploying myInsight-installer pod [6](#)
Document Description [5](#)

E

eDRG [5](#)
euroscript [5](#)

H

How it works [9](#)

M

myinsight-installer pod [9](#)

P

Persistent Volume [9](#)
Planning the installation [6](#)
Pods [9](#)
Product description [4](#)

R

Rebranding [5](#)
Requirements [6](#)
Roles [4](#)

V

Version history [3](#)