**Standard Operating Procedure (SOP)**

*Solution Deployment*

*May, 2019*

*On hold awaiting upcoming new procedures based on Azure DevOps CI/CD pipelines and Any2DataCatalog solution.*

Contents

[Introduction 3](#_Toc8898381)

[SOP steps overview 3](#_Toc8898382)

[1 Prepare deployment scripts 3](#_Toc8898383)

[2 Get deployment approval 4](#_Toc8898384)

[3 Deploy security components 4](#_Toc8898385)

[3.1 Active Directory 4](#_Toc8898386)

[3.2 Key vault 4](#_Toc8898387)

[4 Deploy storage components 4](#_Toc8898388)

[4.1 Blob storage 4](#_Toc8898389)

[4.2 SQL database 4](#_Toc8898390)

[4.3 Data Lake Storage (DLS) directories 4](#_Toc8898391)

[5 Deploy ELT (extract, load and transform) components 5](#_Toc8898392)

[5.1 Databricks Notebooks 5](#_Toc8898393)

[5.2 Function app 5](#_Toc8898394)

[5.3 Data factory pipeline 5](#_Toc8898395)

[6 Deploy UI (user interface) & API components 5](#_Toc8898396)

[6.1 Web app 5](#_Toc8898397)

[6.2 API 5](#_Toc8898398)

[7 Deploy Data Catalog additions and updates 5](#_Toc8898399)

[7.1 Register or update 5](#_Toc8898400)

[7.2 Delete 6](#_Toc8898401)

[8 References 6](#_Toc8898402)

# Introduction

**Purpose of this template**

This **Standard Operating Procedure (SOP)** contains step-by-step instructions on **how to deploy data flow solutions in OMNIA.** The primary audience for this document is data engineering personnel responsible for deployment tasks.

**Legal**

This document contains information that is proprietary to Equinor ASA. Neither the document nor the information contained therein should be disclosed or reproduced in whole or in part, without express written consent of Equinor ASA. The document and the information it contains shall be handled according to Equinor’s information classification scheme.

# SOP steps overview

Deployment of a data flow solution to a specific OMNIA test or production environment consists of the following steps:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool(s) used* | ***Requirements*** |
| **1** | [**Prepare deployment scripts**](#_Prepare_all_deployment) | SQL database components, key vault secrets, function apps and data factory pipelines | Script editor(s) e.g. in Visual Studio | * Prepare all required scripts for deployment (upgrade) and undeployment (downgrade) from/to the current component version. |
| **2** | [**Get deployment approval**](#_Get_deployment_approval) | All solution components | [Data flow delivery checklist](https://statoilsrm.sharepoint.com/:f:/r/sites/DataPlatformTeamTwo/Shared%20Documents/Checklists,%20templates%20and%20standard%20operating%20procedures?csf=1&e=08cEYF) | * See [checklist template](https://statoilsrm.sharepoint.com/:f:/r/sites/DataPlatformTeamTwo/Shared%20Documents/Checklists,%20templates%20and%20standard%20operating%20procedures?csf=1&e=bYLX3Q). Approval is required to continue with step 3. |
| **3** | [**Deploy security components**](#_Deploy_all_storage) | AD, Key Vault | Azure portal | * Deploy required new security components in Azure AD and specific key vaults |
| **4** | [**Deploy storage components**](#_Deploy_all_storage_1) | SQL DB, Data Lake Storage (DLS), Blobs etc. | SQL Server Management Studio (SSMS) for SQL DB components, Windows PowerShell for other | * Deployment must be completed before continuing to step 5. |
| **5** | [**Deploy ELT components**](#_Deploy_all_ELT) | Databricks Notebooks, Function apps, Data Factory pipelines | Windows PowerShell | * Deployment must be completed before continuing to step 6. Use the deployment generator. See guidelines on GitHub. |
| **6** | [**Deploy UI & API components**](#_Deploy_all_UI) | Web apps, APIs | Windows PowerShell | * Deployment must be completed before continuing to step 7. |
| **7** | [**Deploy data catalog updates**](#_Deploy_all_security) | DB views, DLS directories | Any2DataCatalog data flow solution | * Deployment must be done right after step 6 is completed, preferably the same day. |

# Prepare deployment scripts

Prepare all required scripts for upgrade and downgrade (deploy/undeploy) from/to the current component version.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool(s) used* | ***SOP*** |
| **1.1** | **SQL database components** | All components except the database creation | SSMS | * Prepare one deploy script and one undeploy script, each covering all SQL database components * Test the scripts in the development environment |
| **1.2** | **Data Lake Storage (DLS) directories** | All DLS directories used in solution | PowerShell script editor | * Prepare one deploy script and one undeploy script, each covering all DLS directories * Test the scripts in the development environment |
| **1.3** | **Key vault secrets** | All secrets used by the solution | PowerShell script editor | * Prepare one deploy script and one undeploy script, each covering all secrets * Test the scripts in the development environment |
| **1.4** | **Function apps** | All function apps in the solution | PowerShell script editor | * Prepare one deploy script and one undeploy script, each covering all function apps * Test the scripts in the development environment |
| **1.4** | **Data factory pipelines** | All data factory pipeline components | * Custom app for retrieval of JSON files with definitions of data factory components * [Deployment generator](https://github.com/equinor/DataLakers/tree/master/tools/scripts/DeploymentGenerator) (PowerShell script) * PowerShell script editor | * Get all data pipeline definitions (JSON files). * Create deployment script template. See [guidelines on GitHub](https://github.com/equinor/DataLakers/tree/master/tools/scripts/DeploymentGenerator). * Create one deploy script and one undeploy script each covering all pipeline components. * Test the scripts in the development environment |

Note: Include links to DF deploy tools in table above

# Get deployment approval

The data flow checklist must be completed and approved for the relevant implementation stage.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **2.1** | **Get approvers** | Reviewer(s) for the specific review (development and/or proction review) | N/A | * Request reviewer(s) from ODET2 leader/PO |
| **2.2** | **Review meeting** | All checklist items for the specific review (development or production) | Outlook | * Schedule and conduct review meeting(s) with the assigned reviewer(s) |

# Deploy security components

## Active Directory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **3.1.1** | **Users** | All new solution users | Azure portal |  |
| **3.1.2** | **Groups** | All new solution groups | Azure portal |  |
| **3.1.3** | **Apps** | All new apps to be used in solution | Azure portal |  |

## Key vault

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **3.2.1** | **Keys** | All keys used in the solution | Azure portal | * Access the relevant key vault and add all keys to be used in the solution |
| **3.2.2** | **Secrets** | All secrets used in the solution | PowerShell | * Run deploy/undeploy script prepared in step 1 |

# Deploy storage components

## Blob storage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **4.1.1** | **Accounts** | All blob storage accounts used by the solution | Azure portal | * Create the account(s) |
| **4.1.2** | **Containers** | All blob containers used in the solution | Azure portal | * Create the container(s) |

## SQL database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool(s) used* | ***SOP*** | |
| **4.2.1** | **Database** | All databases used in solution | Azure portal | * Create database(s) default configuration |
| **4.2.2** | **Schemas** | All schemas | SSMS and SQL script | * Run the single deploy/undeploy script covering all database components |
| **4.2.3** | **Users** | All users |
| **4.2.4** | **Permissions** | All permissions |
| **4.2.5** | **Tables** | All tables |
| **4.2.6** | **Views** | All views |
| **4.2.7** | **Procedures** | All stored procedures and functions |

## Data Lake Storage (DLS) directories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **4.3.1** | **Directories** | All blob storage accounts used by the solution | Azure portal | * Run deploy/undeploy script prepared in step 1 |

# Deploy ELT (extract, load and transform) components

## Databricks Notebooks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **5.1.1** | **Environment** | Cluster version, code packages/libraries | Databricks workspace | * Ensure that the cluster version is correct and that the required code packages/libraries are installed |
| **5.1.2** | **Notebooks** | All notebooks used in solution | Databricks workspace | * Import all notebooks used in solution |

## Function app

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **5.2.1** | **App** | All apps used in solution | PowerShell | * Run the deploy/undeploy script prepared in step 1 to publish/unpublish the app(s) |
| **5.2.2** | **Configuration** | Application settings | Azure portal | * Find the published app, select Appliction settings and configure the settings |
| **5.2.3** | **Key** | Function key for app | Azure portal |  |
| **5.2.3** | **Access** | Grant the function app required access permissions in previously deployed components and in the Data Lake Storage (DLS) | Azure portal |  |

## Data factory pipeline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **5.3.1** | **Script** | All data pipeline components in solution | PowerShell | * Run the deploy (upgrade) or undeploy (downgrade) script |
| **5.3.2** | **Access** | Data factory pipeline access in all relevant components in solution | Azure portal | * Grant the pipeline required access in previously deployed components and in the Data Lake Storage (DLS) |

# Deploy UI (user interface) & API components

## Web app

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **6.1.1** |  |  |  |  |
| **6.1.n** |  |  |  |  |

## API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **6.2.1** |  |  |  |  |
| **6.2.n** |  |  |  |  |

# Deploy Data Catalog additions and updates

## Register or update

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **7.1.1** | **Register/ update data catalog item(s)** | All catalog items (assets and/or glossary terms) related to the solution | SQL Server Management Studio connected to database Omnia\_Common on server dataengineeringsqlprod.database.windows.net | * Enter/update content in the table Any2DataCatalog.Assets. * Use existing records as guidelines/examples. * If the content of a column is a tag and the term exists in the data catalog glossary, copy the termId from the Link field in the glossary item. Else specify the new required term, and it will be included automatically in the data catalog glossary. * All updates will be transferred to the data catalog by the Any2DataCatalog data flow at <time>. |

## Delete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | | *Scope* | *Tool used* | ***SOP*** |
| **7.2.1** | **Delete data catalog item(s)** | All catalog items (assets) related to the solution, that should to be deleted | SQL Server Management Studio connected to database Omnia\_Common on server dataengineeringsqlprod.database.windows.net | * Set the value of the column ToBeDeleted to TRUE for relevant record(s) in the table Any2DataCatalog.Assets. * The data catalog will be updated by the Any2DataCatalog data flow at <time>. |

# References

Recently used deployment scripts can be used as references. These are available in the [Projects folder in the Equinor GitHub repository DataLakers](https://github.com/equinor/DataLakers/tree/master/projects).