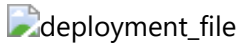


Introduction FMD_FRAMEWORK

Fabric Metadata-Driven Framework (FMD)

Efficient data management is a cornerstone of modern organizations, and leveraging the right tools can make all the difference. The Fabric Metadata-Driven Framework (FMD) is a cutting-edge solution designed to optimize data handling and utilization. This innovative framework harnesses the powerful capabilities of the Fabric SQL Database to build a robust, scalable, and flexible metadata-driven architecture.



Key Features:

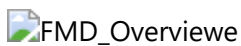
-**Enhanced Data Governance:** FMD ensures comprehensive data governance by maintaining detailed metadata, enabling better data quality, consistency, and compliance.

-**Scalability and Flexibility:** The framework is designed to scale seamlessly with your organization's growth, adapting to evolving data needs without compromising performance.

-**Streamlined Data Integration:** FMD simplifies the integration of diverse data sources, providing a unified view of your data landscape.

-**Cost Efficiency:** By optimizing data processes and reducing redundancy, FMD helps organizations achieve significant cost savings.

FMD_FRAMEWORK Deployment



Requirements

The admin settings below needs to be enabled:

- Users can create Fabric items
- SQL database (preview)

The following steps need to be done:

1. Download Files

Download the three files

- FMD_deployment.json : Contains all definitions to create all artifacts for the FMD_FRAMEWORK (Input for the notebook below)
- NB_FMD_DEPLOYMENT_MULTI_ENV.ipynb : Creates all artifacts for the FMD_FRAMEWORK within Fabric Data Platform(baed on your configuration)
- NB_FMD_DEPLOYMENT_UTILS.ipynb: Contains all definition to Creates all artifacts to your local machine.

or

- Clone this repository:

```
git clone https://github.com/edkreuk/FMD_FRAMEWORK.git
```

3. Create Connection

Create the following two connections and Write down the Connection ID for later usage.:

Connection Name	CON_FMD_FABRICPIPELINES
Connection Type	Fabric Data Pipelines
Authentification	OAuth2
-	-
Connection Name	CON_FMD_FABRICSQL
Connection Type	Fabric SQL database
Authentification	OAuth2

4. Create Workspace and Config Lakehouse

- Create a new workspace "FMD_FRAMEWORK_CONFIGURATION" (Name is up to you)
- Import Notebooks (NB: Make sure your are in the Fabric Experience)
 - Import the notebook "NB_FMD_DEPLOYMENT_MULTI_ENV.ipynb" into the workspace
 - Import the notebook "NB_FMD_DEPLOYMENT_UTILS.ipynb" into the workspace



6. Notebook Configuration

Deployment Configuration Script This script is used to configure deployment settings for different environments (development and production) in a Fabric SQL Framework. Below is a detailed explanation of each section of the script.

Open the notebook NB_FMD_DEPLOYMENT_MULTI_ENV and scroll to the third cell. This is the deployment configuration.

1. Capacity ID capacity_id = '075e5656-1234-5678-a36c8c0e8bca' Purpose: Unique identifier for the capacity to be used.
2. Workspace Roles Purpose: Defines roles (or keep empty []) to be added to the workspace. Each role includes: principal: The group or user to which the role is assigned. role: The role assigned to the principal (e.g., Member, Admin)

Example with roles:

```

    ...
    workspace_roles = [
        {
            "principal": {
                "id": "00000000-0000-0000-0000-000000000000",
                "displayName": "sg-fabric-contributor",
                "type": "Group"
            },
            "role": "Member"
        },
        {
            "principal": {
                "id": "00000000-0000-0000-0000-000000000000",
                "displayName": "sg-fabric-admin",
                "type": "Group"
            },
            "role": "Admin"
        }
    ]
    ...

```

Example without roles:

```

    ...
    workspace_roles = []
    ...

```

3. Logging settings. Purpose: Defines the configuration for the logging items. This can be a different workspace or the same workspace as of the following items.
4. Configuration settings. Purpose: Defines the configuration for the Configuration items. This can be a different workspace or the same workspace for the other items.
5. Environments. Purpose: Defines configurations for different environments (development and production). Each environment includes:
 - environment_name: Name of the environment.
 - workspaces: Workspaces for data and code, each with:
 - name: Name of the workspace.
 - roles: Roles to be assigned.
 - capacity_id: Capacity ID to be used. connections: Various connection identifiers for the environment.
 - CON_FMD_FSQ (is created in a earlier stage): fill in the guid
 - CON_FMD_FABRICPIPELINES (is created in a earlier stage): fill in the guid

- CON_FMD_ASQL_01: Optional connection. Used if you want to connect to an Azure SQL database (default: None)
- CON_FMD_ASQL_02: optional connection. Used if you want to connect to an second Azure SQL database (default: None)
- CON_FMD_ADLS_02: optional connection. Used if you want to connect to an Azure Datalake storage (default: None)
- CON_FMD_ADF_PIPELINES: optional connection. Used if you want to run an Azure Datafactory Pipeline (default: None)

```
environments = [
    {
        'environment_name' : 'development',
        'workspaces': {
            'data' : {
                'name' : 'FMD_FRAMEWORK_DATA (D)',
                'roles' : workspace_roles,
                'capacity_id' : capacity_id_dv1m
            },
            'code' : {
                'name' : 'FMD_FRAMEWORK_CODE (D)',
                'roles' : workspace_roles,
                'capacity_id' : capacity_id_dv1m
            }
        },
        'connections' : {
            'CON_FMD_FSQ' : '372237f9-709a-48f8-8fb2-ce06940c990e',
            'CON_FMD_ASQL_01' : 'cf673e6a-13f6-4ebb-9cbb-4ba4ab390818',
            'CON_FMD_ASQL_02' : '11a8e5fe-fbca-4822-9ba4-9162cf56e6dd',
            'CON_FMD_ADLS_01' : 'a0581b6e-5e38-46eb-bab2-7f08e9a35c30',
            'CON_FMD_FABRICPIPELINES' : '6d8146c6-a438-47df-94e2-540c552eb6d7',
            'CON_FMD_ADF_PIPELINES' : '02e107b8-e97e-4b00-a28c-668cf9ce3d9a'
        }
    },
    {
        'environment_name' : 'production',
        'workspaces': {
            'data' : {
                'name' : 'FMD_FRAMEWORK_DATA',
                'roles' : workspace_roles,
                'capacity_id' : capacity_id_prod
            },
            'code' : {
                'name' : 'FMD_FRAMEWORK_CODE (prod)',
                'roles' : workspace_roles,
                'capacity_id' : capacity_id_prod
            }
        }
    }
],
```


```


    'connections' : {
      'CON_FMD_FSQ' : '372237f9-709a-48f8-8fb2-ce06940c990e',
      'CON_FMD_ASQ_01' : 'cf673e6a-13f6-4ebb-9cbb-4ba4ab390818',
      'CON_FMD_ASQ_02' : '11a8e5fe-fbca-4822-9ba4-9162cf56e6dd',
      'CON_FMD_ADLS_01' : 'a0581b6e-5e38-46eb-bab2-7f08e9a35c30',
      'CON_FMD_FABRICPIPELINES' : '6d8146c6-a438-47df-94e2-540c552eb6d7',
      'CON_FMD_ADF_PIPELINES' : '02e107b8-e97e-4b00-a28c-668cf9ce3d9a'
    }
  }
]

```

6. Deployment File deployment_file = 'deployment/FMD_deployment.json' Purpose: Specifies the source file to read the deployment manifest from.

Remarks:

- Fabric SQL Database can fail. Mostly this will be caused of to many Fabric Databases in your tenant. Try to create the Fabric Database manually, you will directly see if this is the case. If not you can add the manual added database setting to the Deployment Notebook in cell 3.
- In a trial capacity you can't create more than 3 databases.
- In the deployment notebook you can receive the following error  Fabric Experience

You check this by creating the database manually  Fabric Database Error

Contributing

Contributions are welcome! If you have suggestions or improvements, feel free to open an issue or submit a pull request.

License

This project is licensed under the MIT License.