Cognite Data Fusion and Azure Digital Twin Plug-in

Technical Documentation

This is a technical documentation of the Cognite Data Fusion (CDF) and Azure Digital Twin (ADT) Plug-in.

The purpose of the plug-in is to synchronize the industrial knowledge graph between the CDF and ADT, using Azure Functions written in Python, i.e. to translate the CDF asset hierarchy nodes together with contextualized operational and engineering data into Digital Twin Definition Language (DTDL) ontologies.

For development **Python 3.9.7** was used, as this was the latest version supported by Azure functions.

In the first phase the following CDF resource types are mapped:

* Assets
* Asset-to-asset relationships
* Timeseries with the value of the latest datapoint

The up-to-date source code for the plug-in is stored in [this Github repository](https://github.com/muradsater/azure-digital-twin-cdf-sync).

The project contains two main features:

1. a timer-triggered Azure function to create/update the knowledge graph in the CDF->ADT direction
2. an event-triggered Azure function to update changes in the ADT->CDF direction

# Dependencies

In order to deploy and run the plugin, the following resources are required:

* CDF tenant, which contains the initial industrial knowledge graph(s) to be mapped
* Microsoft Azure tenant, where the Azure functions will be deployed to replicate and synchronize the graph(s). The Azure resources below need to be created beforehand:
  + 2 function apps (one timer-triggered and one event-triggered)
  + 2 blob storage accounts (one for each function)
  + Key vault
  + Azure Digital Twins
  + Event Hub.

# Timer-triggered Azure Function

The timer-triggered function replicates the CDF graph in ADT and synchronizes any changes in the CDF 🡪ADT direction. It is an [Azure function](https://docs.microsoft.com/en-us/azure/azure-functions/) and there are [various ways for deployment](https://docs.microsoft.com/en-us/azure/azure-functions/functions-deployment-technologies).

The function is dependent on the Python libraries listed in the table below.

|  |  |
| --- | --- |
| **Python library** | **Version** |
| azure-core | 1.23.1 |
| azure-digitaltwins-core | 1.1.0 |
| azure-functions | 1.10.1 |
| azure-identity | 1.9.0 |
| azure-storage-blob | 12.11.0 |
| cognite-sdk | 2.38.6 |

The function implements the following features:

* Mapping of CDF assets, asset-to-asset relationships and time series resources to their corresponding DTDL models (also uploaded to the Github repository) and instantiating the digital twins in ADT. This constitutes the replication of the CDF graph in ADT.
* Updating all the changes on these three CDF resources inside ADT:
  + asset property changes including metadata,
  + asset parent change,
  + new asset creation,
  + old asset delete,
  + relationship property updates,
  + new relationship creation,
  + time series property updates,
  + time series linked asset change,
  + new time series creation,
  + latest datapoint change in time series.

All the inputs for the timer-triggered function must be defined as environment variables in the Azure function configuration settings. The keys below must be set:

* "ADT\_URL": URL of the ADT resource,
* "AzureWebJobsStorage": connection string to the blob storage linked to this Azure function,
* "CDF\_CLIENT\_SECRET" client secret of the Cognite tenant,
* "CDF\_CLIENTID": the client ID of the Cognite tenant,
* "CDF\_CLUSTER": cluster of the Cognite tenant,
* "CDF\_TENANTID": ID of the Cognite tenant,
* "CDF\_PROJECT": Cognite project inside the Cognite tenant,
* "ROOT\_ASSET\_EXTERNAL\_ID": the external ID of the root asset node of the knowledge graph to be instantiated and synchronized,
* "FUNCTIONS\_WORKER\_RUNTIME": this defaults to "*python*" in our case.

# Known Issues, Limitations

During development the following issues and limitations were found:

* in ADT the digital twin ID ($dtId) cannot contain whitespace and colon characters
* in ADT map keys cannot contain these characters: $, ., <space>
* in CDF different type of resources can have the same external ID (e.g., one asset and one timeseries can have the same external ID)
* in CDF the external ID can be changed (it can even be switched between 2 resources of the same type)
* in CDF relationships can exist by their own, in ADT not
* resources (assets, timeseries) in CDF, and twins in ADT can exist by their own, i.e. not linked to any other resource/twin so their synchronization might stop at one point,
* in ADT a timeseries can be linked to multiple assets, while in CDF only one.