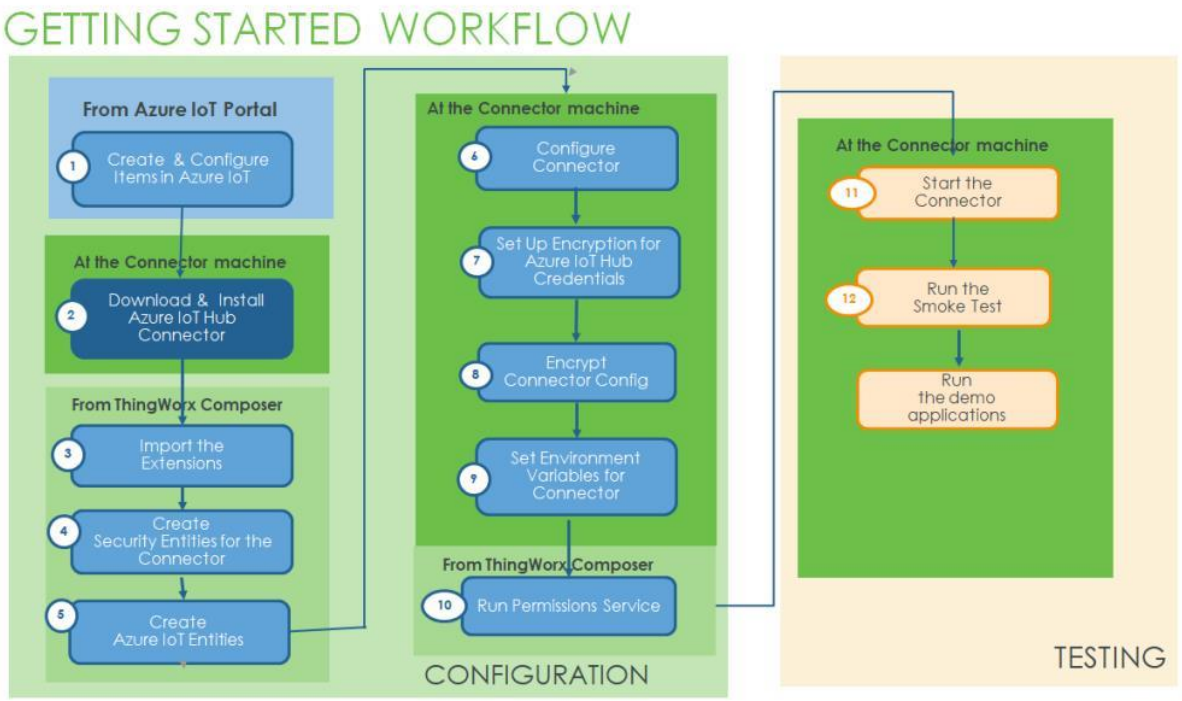
HOW TO MAKE CONNECTIVITY BETWEEN THINGWORX 8.5.2 & AZURE IoT HUB With AZURE IOT HUB CONNECTOR 3.0.0.20

By FAROUK LOUALI

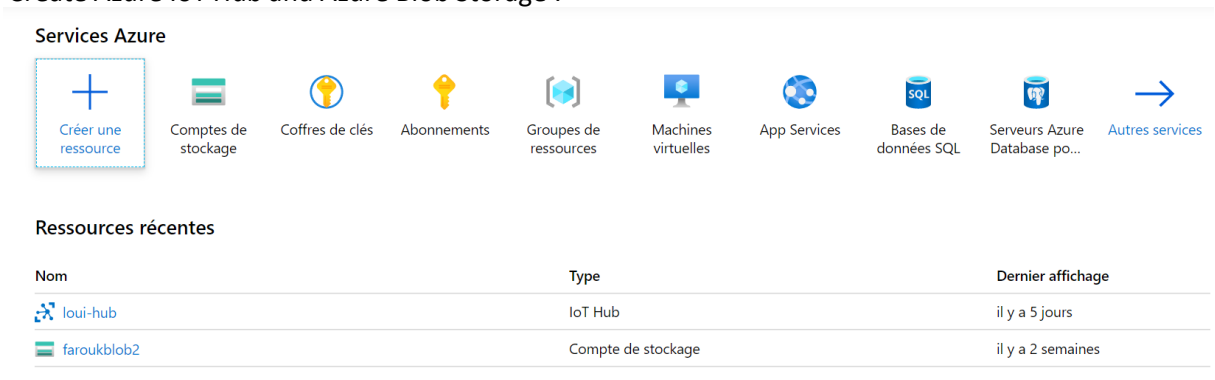
Here are the detailed notes of my work with ThingWorx Azure IoT Hub Connector, including the configuration and the operations I performed to set it up and run it. I also provided information on the use of ThingWorx and azure, in addition to the directives provided in the documentation ptc on :

https://support.ptc.com/help/thingworx_scm_azure/r3.0/en/#page/thingworx_scm_azure%2FWelcome.html%23)

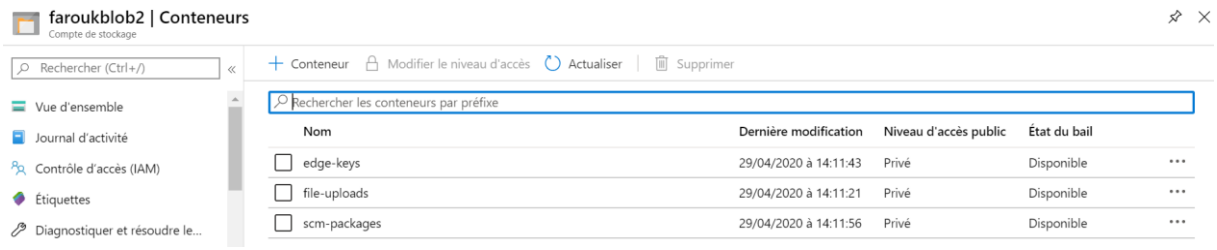


1. set up Azure IoT environment using the Azure IoT portal :

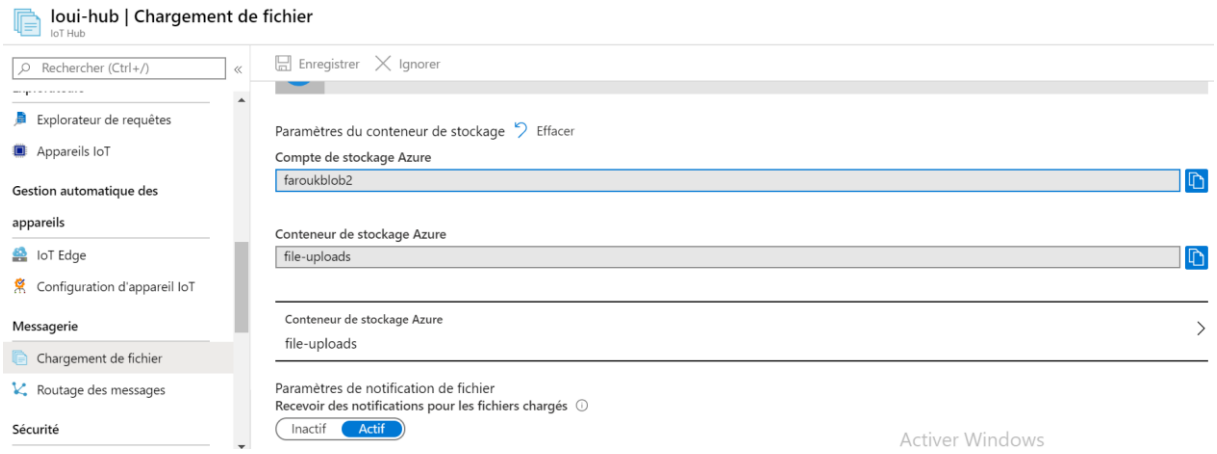
-Create Azure IoT Hub and Azure Blob Storage :



Ideally, your IoT Hub and storage account are in the same location, as specified in the Current Location settings for the Hub and for the storage account in Azure IoT.
-Create containers for storing files :
◦ Azure IoT Edge device upload container. For example, file-uploads.
◦ Software Content Management (SCM) module storage container. For example, edge-keys.
◦ SCM packages container. For example, scm-packages.



-Associated file uploads with your blob storage container by enabling file notifications and specifying the Edge device upload container that you created

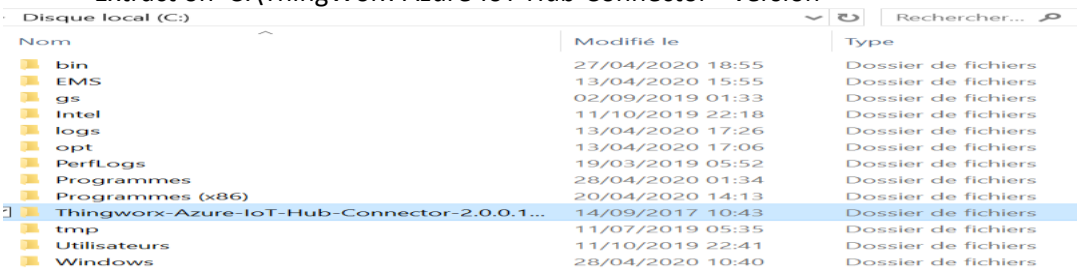


If you skip this step, file transfer notifications will not be sent. You need to have file transfer notifications enabled in your Azure configuration and in your Azure IoT Hub. The Connector will try repeatedly to connect but cannot open the endpoint if these settings are not in place.

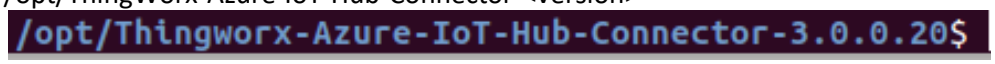
2. Download and Install the Azure IoT Hub Connector

-ThingWorx environment must meet the following requirements:
• ThingWorx Platform is installed and running. For the version required (8.5.2)
• have a separate computer that meets the following system requirements for the Connector:
◦ Linux, 64-bit operating system, or Windows Server, 64-bit.
◦ Oracle JDK 8, update 141 or later, installed and PATH set.
-Download and extract the distribution bundle :
For windows :

Extract on C:\ThingWorx-Azure-IoT-Hub-Connector-<version>



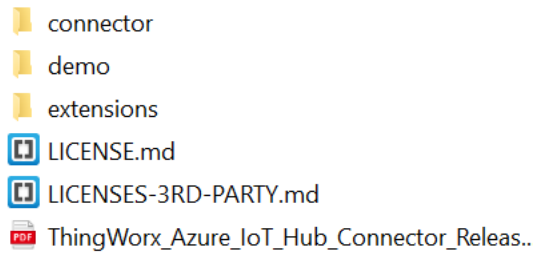
For Linux :
/opt/ThingWorx-Azure-IoT-Hub-Connector-<version>

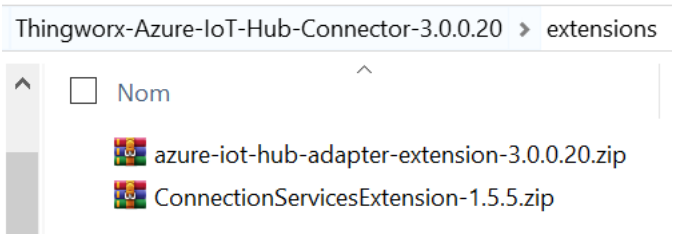


3. Import the extension :

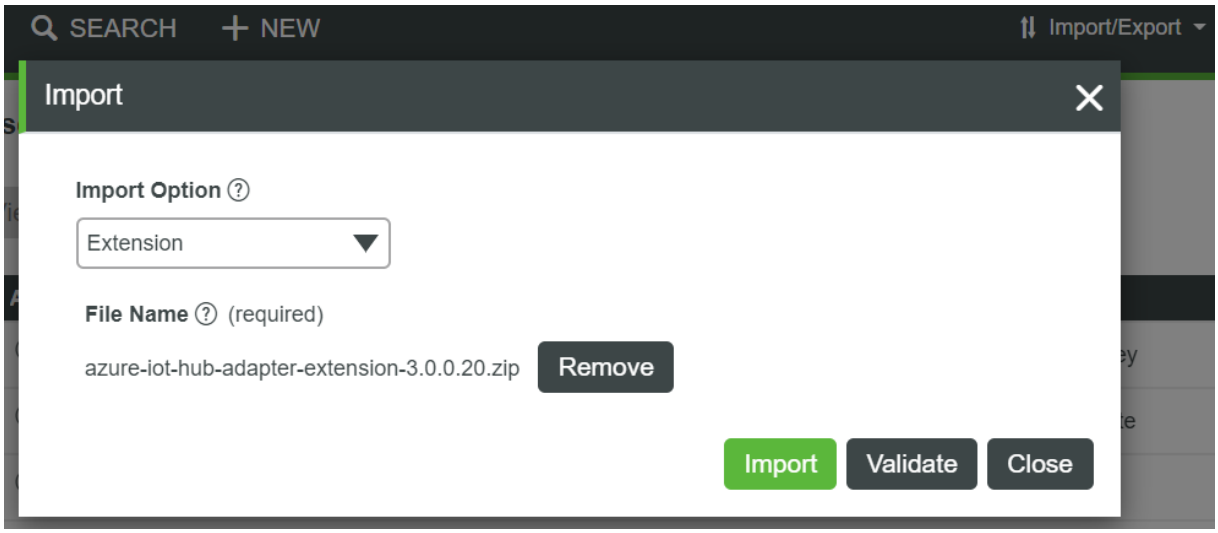
The ThingWorx Azure IoT Connectors requires that you import into your ThingWorx Platform the two extensions that are provided within the distribution bundle, namely the ThingWorx Connection Services Extension (CSE) and the ThingWorx Azure IoT Extension (AIE)

You can Find the extension on Azure Iot connector folder :





Go to thingworx composer :



4. Create Security Entities for the Connector :

The Azure IoT Hub Connector requires a user and application key to access the ThingWorx Platform. In addition, running the services that grant visibility and permissions to a Connector requires that you specify a user, user group, and organization
Follow the step described on the link below :
https://support.ptc.com/help/thingworx_scm_azure/r3.0/en/#page/thingworx_scm_azure%2Fazure_connector%2Fc_azure_connector_create_security_entities_for_connector.html%23

5. Create Azure IoT Entities in ThingWorx Composer :

Refer to the link
:http://support.ptc.com/help/thingworx_scm_azure/r3.0/en/#page/thingworx_scm_azure%2Fazure_connector%2Fc_azure_connector_create_azure_entities_in_thingworx.html%23
Be carful If you enter the wrong connection key here, you will get the error

6. Configure the Connector

https://support.ptc.com/help/thingworx_scm_azure/r3.0/en/#page/thingworx_scm_azure%2Fazure_connector%2Fc_azure_connector_configure.html%23

on azure iot connector folder go to :
connector\conf → save the file azure-iot-sample.conf as azure-iot.conf and configure it like this

```
cx-server {
        thing {
                name = " ‘put the  name of azure IoT hub’ -cxserver"
//              name = "azure-iothub-cxserver"
        }

        protocol {
```

```
                ssl {
                        enabled = false
                        protocol = "TLSv1.2"
//                      verify-host = false
//                      trust-all-certificates = true
                        key-store {
                                file = ".keystore"
                                password = "NotRealUnencryptedKeystorePass9843754938"
                        }
                }

                file-upload-events {
                        // Should the receiver for file-upload notifications be started?
                        enabled = true

                        // Name of the Thing (using template
AzureStorageContainerFileRepository) which is associated to the storage container
                        // linked with this IoT Hub (within Azure Portal:  Hub > File upload
(under Messaging) > Storage container).
                        //
                        // This thing does not need to exist prior to starting the connector,
but any file-upload events received
                        // before this Thing exists will fail and not be retried.
                        repository-thing-name = " put the name you created on thingworx "
                }

                // This is the thing name that represents your Azure IoT Hub in Thingworx
                hub-thing-name = " put the name you created on thingworx "
        }

        transport.websockets {
                // ThingWorx platform application key
                app-key = "put the appkey"

                // One or more platforms can be specified here, comma separated
//              platforms = "ws://@ip:8080/Thingworx/WS"
//              platforms = "wss://@ip:443/Thingworx/WS"
                platforms = "wss://platformURL:443/Thingworx/WS"
        }
}
```

7. Set Up Encryption for Azure IoT Hub Credentials :

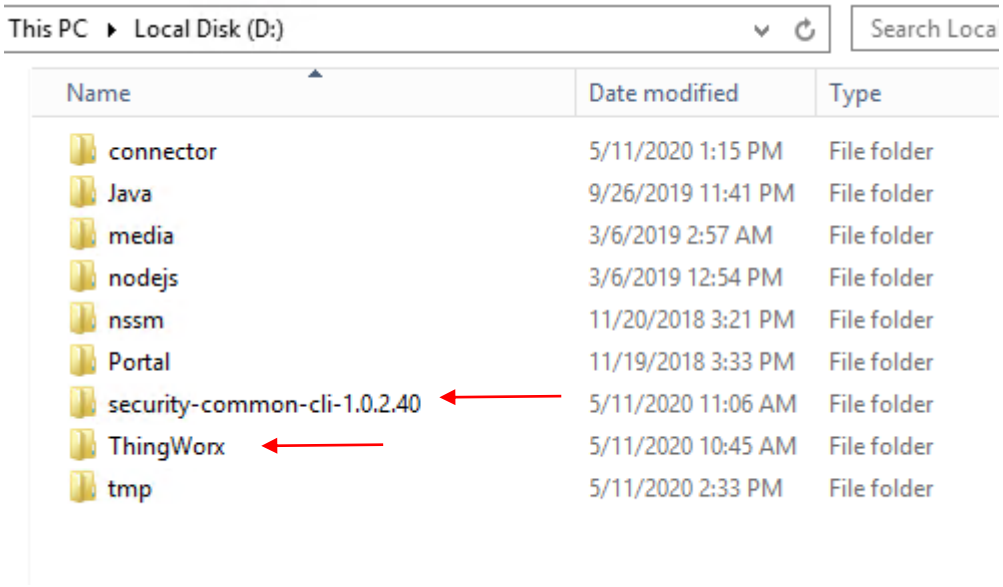
For the Azure IoT Hub Connector to be able to communicate with the Azure IoT Hub, you must set up the Connector and ThingWorx Platform to support encryption/decryption of the Azure Hub credentials. The Azure IoT Hub Connector encryption key must be added to the keystore of your ThingWorx Platform

ON WINDOWS :

-First start the thingworx platform

-Download the Security Management Tool from the Software Downloads page

-extract the Security Management Tool on the same directory of thingworx platform



On \bin directory of Security management tool :

- Create file encryption.conf :

```
{
  security {
    secret-provider = "com.thingworx.security.provider.keystore.KeyStoreProvider"
    default-encryption-key-length = 256

    keystore {
      password-file-path = "D:\\ThingWorx"
      password-file-name = "keystore-password"
      path = "D:\\ThingWorx\\ThingworxStorage"
      name = "keystore"
    }
  }
}
```

- Open a terminal and execute the command below :

1- Security-common-cli.bat
2- Init encryption.conf
3- generate-key azure.connector.key

You have Generate a new encryption key for the Connector

4- get azure.connector.key

In a secure location, save the output value of the encryption key. Next, you will add it to the keystore of the Azure IoT Hub Connector.

5- exit

To support decryption of the Azure IoT Hub credentials, the encryption key generated in the procedure above must be added to the keystore of the Azure IoT Hub Connector

On \bin directory of Security management tool :

- Create file encryption.conf :

```
{
   security {
      secret-provider = "com.thingworx.security.provider.keystore.KeyStoreProvider"
      default-encryption-key-length = 256

      keystore {
         password-file-path = "D:\\connector\\conf"
         password-file-name = "keystore-password"
         path = "D:\\connector"
         name = "keystore.pfx"
      }
   }
}
```

- Open a terminal and execute the command below :

1- Security-common-cli.bat
2- Init encryption.conf
3- set azure.connector.key « add the encryption key value that you saved in the ThingWorx Platform setup above »
4- exit

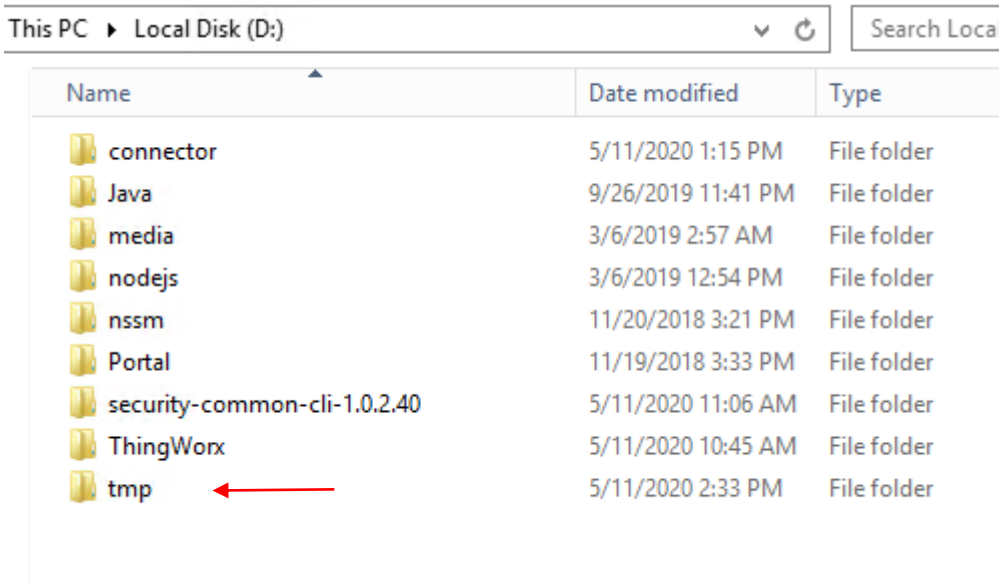
At this point, the encryption key for the Azure IoT Hub should exist with an identical value in both the ThingWorx Platform keystore and the Azure IoT Hub Connector keystore.
For best security practices, delete the output value for the encryption key from the location where you saved it.

8. Encrypt the Configuration File

Security for assets and data is a top priority for companies. For best security practices for your ThingWorx Azure IoT Hub Connector, set up its configuration file to be encrypted, using the Security Management Tool. Encrypting the configuration file provides security for sensitive information, such as network addresses and application keys

On \bin directory of Security management tool :

Create a tmp folder :



- Create file encryption.conf :
-

```
{
 security {
  secret-provider = "com.thingworx.security.provider.keystore.KeyStoreProvider"
  default-encryption-key-length = 256

  keystore {
     password-file-path = " D:\\tmp"
     password-file-name = "keystore-password"
     path = " D:\\tmp"
     name = "keystore.pfx"
  }
 }
}
```

- Open a terminal and execute the command below :

1- Security-common-cli.bat
2- Init encryption.conf
3- encryptFile [pathTo]azure-iot.conf [pathTo]azure-iot.conf.encrypted

Verify that the encryption was successful by using the decryptFile command:

4- decryptFile [pathTo]azure-iot.conf.encrypted [pathTo]azure-iot.conf.decrypted
5- exit

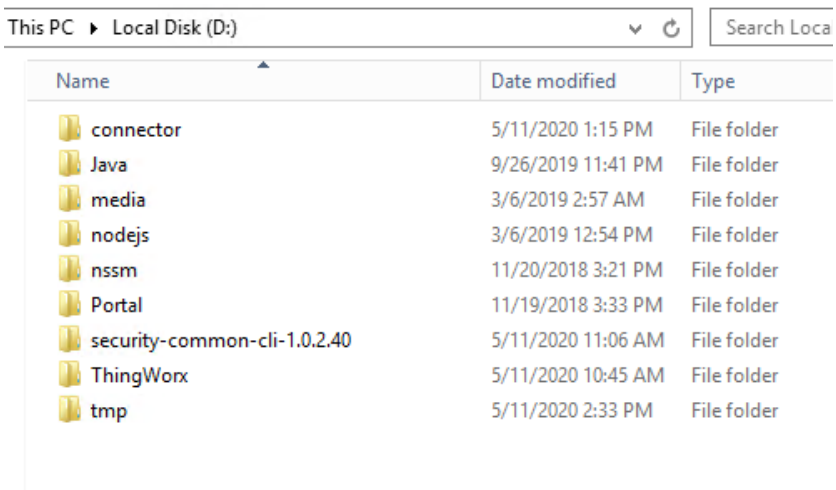
9. Set the Environment Variables for the Azure IoT Hub Connector

The Connector needs to know where its configuration file is located when it starts up. In addition, it needs to know where the encryption configuration file is located when decrypting credentials for the Azure Hub that are sent from the ThingWorx Platform

Refer to this link :https://support.ptc.com/help/thingworx_scm_azure/r3.0/en/#page/thingworx_scm_azure%2Fazure_connector%2Fc_azure_connector_set_environment_variable.html%23

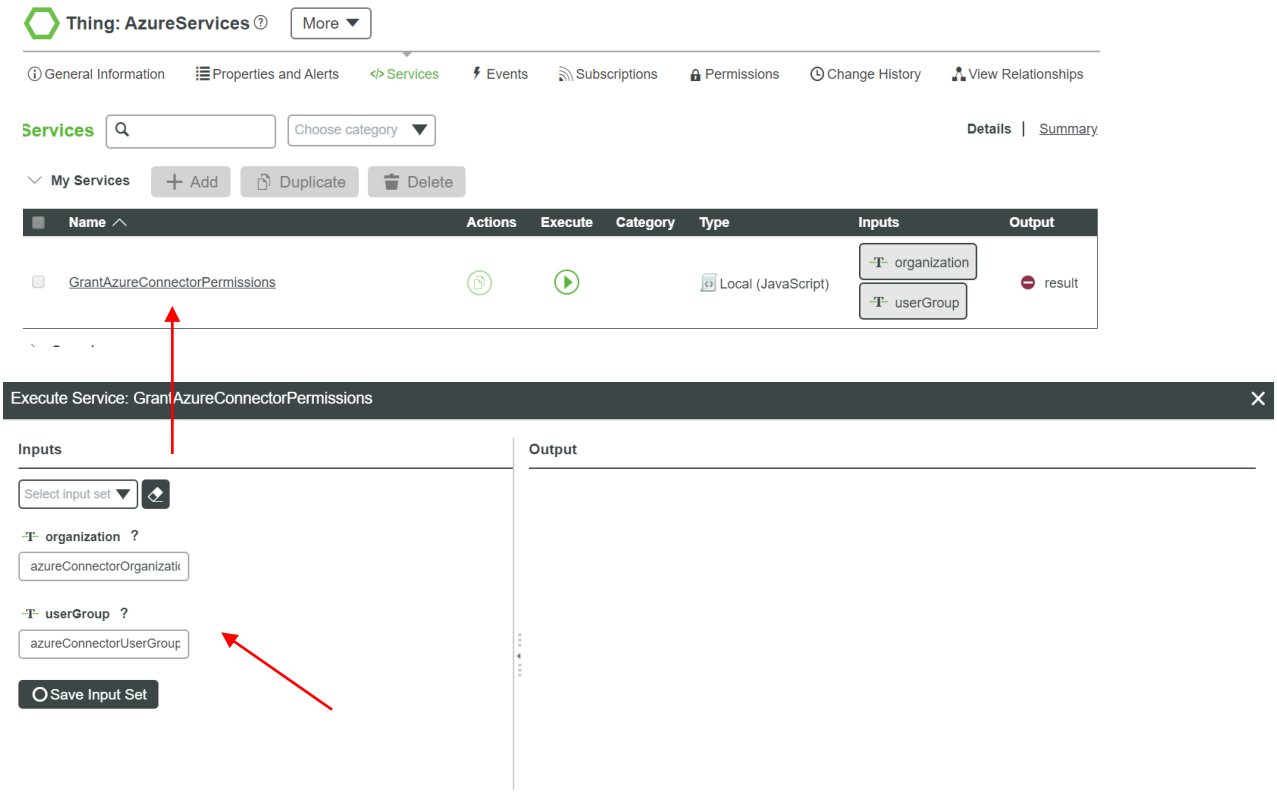
from windows environement :

D:\>set AZURE_IOT_OPTS=-Dsecret.management.config.file=D:\security-common-cli-1.0.2.40\bin\encryption.conf -Dconfig.file=D:\connector\conf\azure-iot.conf.encrypted -Dlogback.configurationFile=D:\connector\conf\logback.xml



10. Run the Service to Grant Permissions and Visibility to the Connector

The following link provide more explaination and list the entities and their visibility and permissions requirements that are granted by the GrantAzureConnectorPermissions service https://support.ptc.com/help/thingworx_scm_azure/r3.0/en/#page/thingworx_scm_azure%2Fazure_connector%2Fc_azure_connector_running_as_non_admin.html%23



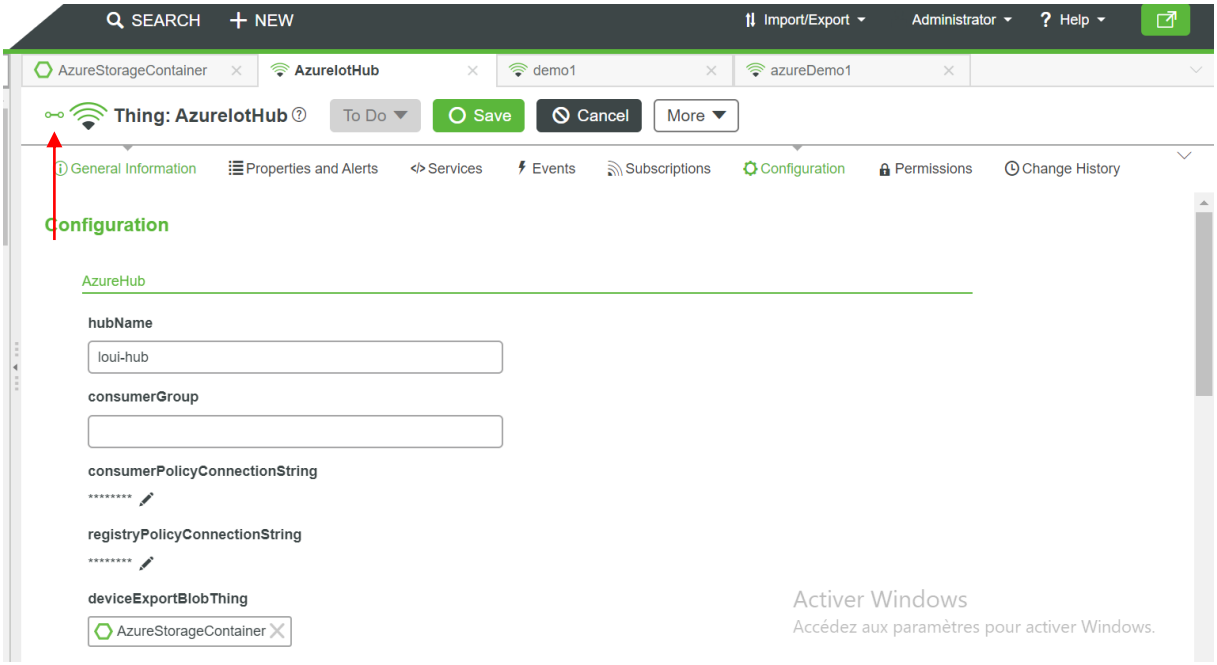
11. Start the Azure IoT Hub Connector

On \bin directory of the connector (windows) :

- Open a terminal and execute the command below :
  Azure-iot.bat

Congrats The Connector starts up and begins processing data, displaying messages in the console where you started it.

To check that the Connector is running and communicating with ThingWorx and Azure, log in to ThingWorx Composer, and navigate to the AzureIotHub Thing that represents your Azure IoT Hub. An icon next to the Thing name shows whether it is connected:



12. Supplementary tips

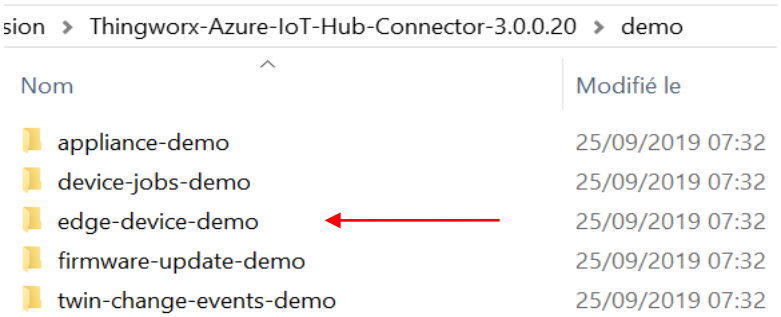
After checking that the Connector is running and communicating with ThingWorx and Azure, i will show you how to run and test edge device demo wich is provided with Azure IoT Hub connector package.

-First start the thingworx platform & Azure IoT Hub Connector
-go to your azure portal :

- Create device with azure CLI :

  Az extension add –name azure-cli-iot-ext
  Az iot hub device-identity create –hub-name « your hub name » --device-id « your device name »

-Go to your connector folder on demo directory



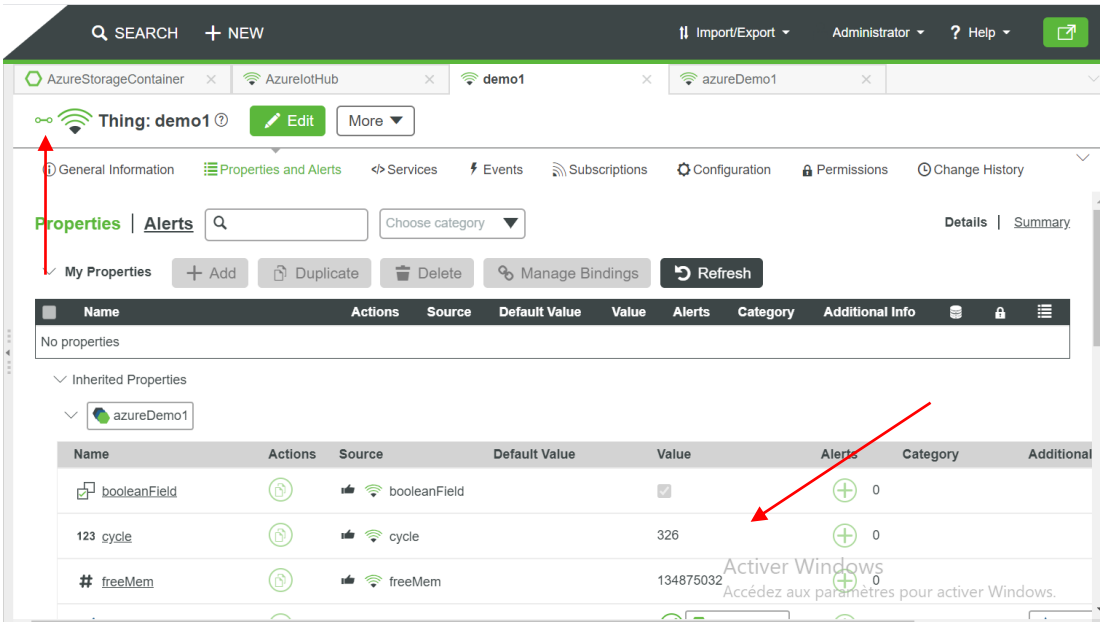
- Set file edge-device.conf :

```
// Azure Edge Device Demo configuration
azure-edge-device {
  // Name of the remote thing on the ThingWorx platform, which should match the
Azure Device ID
  deviceId = "put the name of your device"

  // Name of the hub host in Azure
  iotHubHostname = "put the hub host name .azure-devices.net "

  // Policy name used by this thing (could require services as well in future)
  registryPolicyName = "registryReadWrite"

  // The Key related to the policy above
  registryPolicyKey = "KEY"
}
```

-On \bin directory of edge-device-demo:

- Open a terminal and execute the command below :
  Edge-device-demo.bat

The edge devices starts up and messages flow from it to the Azure IoT Hub, then to the ThingWorx Azure IoT Hub Connector and the Azure Storage. From the Azure Storage, messages can be sent to the Connector. All messages sent to the Connector are forwarded to a ThingWorx Platform.


SEARCH
NEW
Import/Export
Administrator
Help
AzureStorageContainer
AzureIotHub
demo1
azureDemo1
Thing: demo1
Edit
More
General Information
Properties and Alerts
Services
Events
Subscriptions
Configuration
Permissions
Change History
Properties
Alerts
Choose category
Details
Summary
My Properties
Add
Duplicate
Delete
Manage Bindings
Refresh
Name
Actions
Source
Default Value
Value
Alerts
Category
Additional Info
No properties
Inherited Properties
azureDemo1
Name
Actions
Source
Default Value
Value
Alerts
Category
Additional
booleanField
booleanField
123 cycle
cycle
326
0
freeMem
freeMem
134875032
0
Activer Windows
Accédez aux paramètres pour activer Windows.