ICONICS

Make the Invisible Visible™

Microsoft

Installing IoTWorX on IoT Edge on Windows

An ICONICS Whitepaper

February 2022

Installing IoTWorX on IoT Edge on Windows

Contents

Copyright and Confidentiality

By accessing and using the installation instructions (the “instructions”) you acknowledge and agree, on your behalf and on behalf of the person, entity or other organization on whose behalf you are accessing the instructions, that neither Microsoft nor ICONICS, nor any of its service providers, including, without limitation, any system integrator or independent software vendor: (1) makes any representations or warranties of any kind, either express, implied, statuary or otherwise with respect to the instructions, including the accuracy, completeness or usefulness thereof; and (2) shall be liable for damages of any kind, under any legal theory, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions, including any direct, indirect, incidental, special, punitive or consequential damages, or for loss of use, loss of profits, loss of data, loss of business, or loss of privacy or security, even if foreseeable, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions. You further acknowledge and agree that your use of the instructions, whether directly or indirectly, is at your own risk and that you expressly assume all risks in connection with your use of the instructions. If you do not agree to the foregoing, you may not access or use the instructions.

Authors

- Spyros Sakellariadis, Global Director, IoT Business Development, ICONICS, Inc.
- Richard Henderson, US QA Manager, ICONICS, Inc.
- Roberto Vercelli, Director of Engineering, ICONICS, Inc.
- Zhi Wei Li, Director of Innovation and Engineering, US, ICONICS
- Terry Warwick, Program Manager, Microsoft Corporation
- Francisco Cabrera, Program Manager, Microsoft Corporation

Installing IoTWorX on IoT Edge on Windows

Introduction

ICONICS IoTWorX is a piece of software that supports collecting data from equipment and devices and sends it to the cloud for analyzing. The features and benefits of IoTWorX have been well documented and will not be repeated here. Furthermore, instructions for installing IoTWorX on a Linux computer, or on IoT Edge on Linux, have also been fully documented previously, and we will not repeat those instructions here either. The current whitepaper shows you only how to install IoTWorX on a Windows computer.

The ICONICS Suite is a collection of modules that can be used to ingest, visualize, and analyze the data from IoTWorX. Instructions for configuring and using the many features of the ICONICS Suite are well documented elsewhere, and in this whitepaper, we will show you only how to use it to configure IoTWorX remotely.

IoT Edge is a piece of software from Microsoft that runs on a Linux or Windows computer and supports edge computing functionality, for example software designed specifically to run on IoT Edge such as IoTWorX. Installing it on a Windows computer requires first installing a component from Microsoft known as Azure IoT Edge for Linux on Windows (EFLOW) to support the Linux functionality required by IoT Edge.

As stated in the Microsoft documentation, Azure IoT Edge extends IoT Hub, and allows you to "analyze device data locally instead of in the cloud to send less data to the cloud, react to events quickly, and operate offline." The advantage of installing it on a Windows computer is that you can manage the device using Microsoft Update, following the instructions here, which eliminates the need to remote into the machine to install operating system updates.

Prerequisites

The prerequisites for installing EFLOW on a Windows computer are specified in this document from Microsoft. Note that the current whitepaper was written based upon IoT Edge 1.1, and the prerequisites for IoT Edge may change with subsequent versions of IoT Edge. For the latest information, users should check the release notes here.

In this whitepaper we will install IoTWorX on the EFLOW and use it to pull data from an on-premises device and then send it to Azure for visualization and analytics. To follow the steps in this whitepaper, in addition to the EFLOW-capable Windows computer, you need the following:

- An Azure subscription with a storage account, an Azure IoT Hub, and a virtual machine (VM) with at least 2 GB of memory running Windows. Instructions for creating a storage account in Azure can be found here, for creating an Azure IoT Hub here, and for creating a VM here. The VM will be used for the instance of GENESIS64 that will be used to provision IoTWorX.
- Windows and ICONICS licenses.

This whitepaper documents how to install IoTWorX in a Windows VM running on a Windows computer and send data to ICONICS Suite running in Azure. It presupposes that the reader has a good working knowledge of both the ICONICS products and with Azure services. The following document will be referenced for more details:

- IoTWorX Quick Start. Provides instructions for setting up a basic instance of IoTWorX. We will refer to this document as *Quick Start.*

While the steps in the document listed above, and those in the current whitepaper, are generally applicable for most users, results may vary depending upon a user's software and hardware systems.

Overall design

In *Quick Start*, you will find instructions for how to provision IoTWorX on IoT Edge and configure it to send data to IoT Hub. The high-level architecture is as follows:

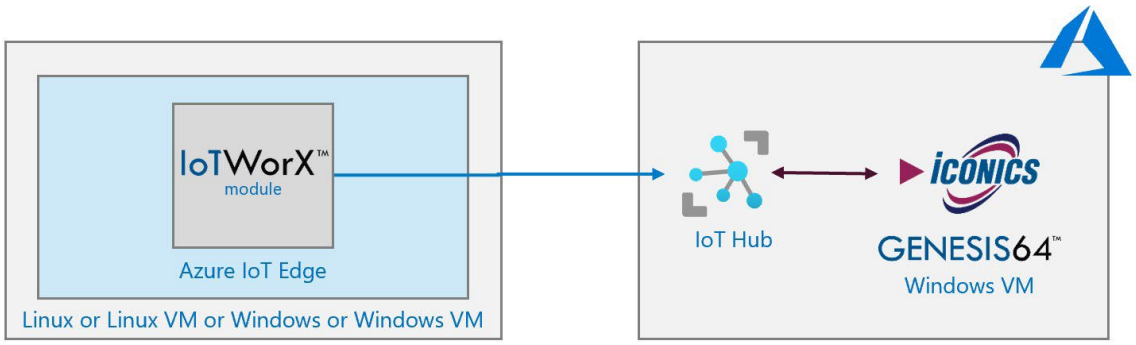


Figure 1: High-level architecture used in Quick Start

The instructions include a simulated temperature sensor running within the IoTWorX module.

The current whitepaper expands specifically on, and only on, the instructions for installing IoTWorX in a Windows VM and uses a physical sensor instead of a simulated one:

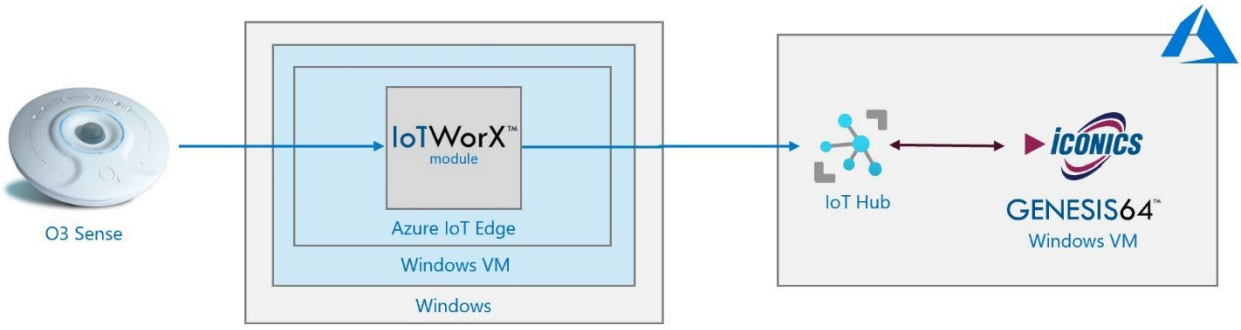


Figure 2: High-level architecture used for this whitepaper

To aid in visualizing what we are doing, the various element names are shown in the following diagram:

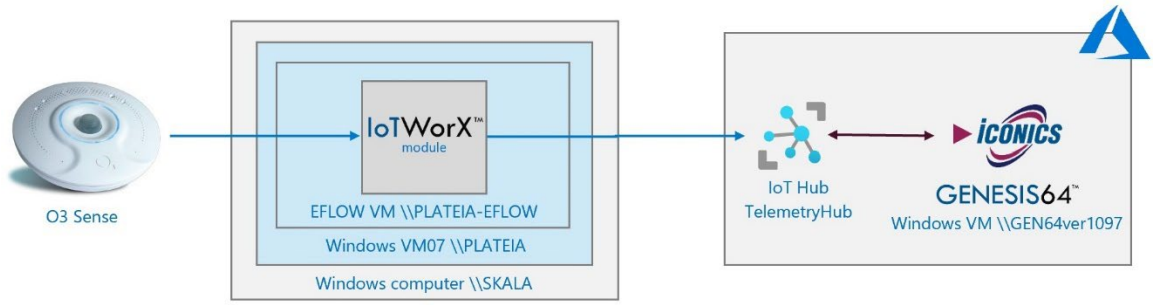


Figure 3:High-level architecture, annotated

The reader can follow the instructions in this paper to install EFLOW directly on \\SKALA by omitting the sections on creating the \\PLATEIA VM. The reason for adding this extra VM in this paper is to provide additional details regarding the nuances of deploying an IoT Edge module on Windows in a nested VM for those users who need that extra layer in their infrastructure – for example, the enterprise customers to whom we have talked that have an existing infrastructure server and have IT policies that require isolation of all applications by installing these in VMs rather than on the server itself.

Creating the required Azure resources

You need to collect some information about the target IoT Hub, so you can provision the IoT Edge device to send data to it. Follow these instructions from *Quick Start,* modified to use the names we show in this whitepaper. In the Azure portal:

1. Select your Azure storage account and then **Access Keys**. Copy the **Connection string** for **key 1** to Notepad. In this whitepaper, the storage account was called **devicetelemetrydata**.
2. Select your IoT Hub, select **Built-in endpoints**. Copy the **Event Hub compatible endpoint** to Notepad. In this whitepaper, the IoT Hub was called **TelemetryHub**.
3. Go to **Shared access policies** in the **Security Settings** section. Select the **iothubowner** policy. Copy the **Primary connection string** to Notepad.
4. Still on the IoT Hub page, select **IoT Edge** in the **Device management** section and click **Add IoT Edge Device**. Name your device and then click **Save**. For this whitepaper, we named the device `vm07`.
5. Click `vm07` and copy the **Primary Connection String** to Notepad.

Creating an instance of the ICONICS Suite

You do not need an instance of the ICONICS Suite to install IoTWorX, though you can use it to do so, but you do need it to configure IoTWorX to collect data. Before starting, install **ICONICS Suite 10.97.1** available from the **Azure Marketplace** here. The instructions for what to do next to create an IoT project are fully documented in the short *Quick Start* section **Creating an IoT Project** and need not be repeated here. To be consistent with the names in this whitepaper, provide the following values as you follow the steps to deploy **IoT Suite** in *Quick Start*:

Field	Value
Server/VM name	\\GEN64ver1097
Project name	SPYROS-HUB
Subscriber connection	IoTWorX configuration

As you follow the directions, remember that the IoT Hub is called **TelemetryHub**, and that you copied the required connection strings while following the instructions at the beginning of this whitepaper.

Getting the IoTWorX software

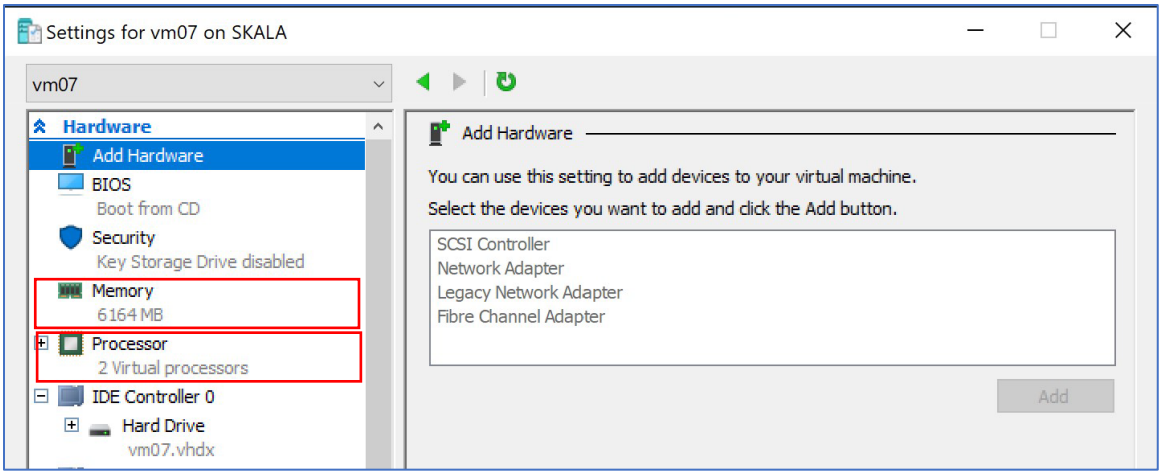
You do not need to download IoTWorX software – it will be deployed from the cloud once you have configured IoT Edge. However, the local Windows VM does need to be configured first as an IoT Edge device. The simplest way to do this is to download and run a script provided by ICONICS, following these instructions from *Quick Start*:

> ICONICS customers with a current support plan can download the zip file from the Customer Connection Portal.
>
> 1. Go to https://partners.iconics.com .
> 2. Log in.
> 3. Select **Support** > **Product Downloads**.
> 4. Download the IoTWorX Tools file.
>
> New users should request an IoTWorX trial at https://iconics.com/Products/IoTWorX, upon which they will be sent a link to download the IoTWorX Tools file.

Preparing the host VM for IoTWorX

Since IoT Edge requires certain Linux capabilities, Microsoft implemented IoT Edge on Windows by providing a technology called EFLOW, which deploys a Microsoft managed Linux VM on top of Windows or a Windows VM. In the environment used for this whitepaper, we chose to run the EFLOW VM on top of a Windows VM. The physical host is a Windows computer named \\SKALA. It has Hyper-V enabled, several VMs used for other applications, and a VM called **vm07** running Windows with the system name \\PLATEIA, into which we are going to install an EFLOW VM. (Hence why we called the Edge device in the Azure portal **vm07**.)

Before continuing, note that an EFLOW VM on IoT Edge 1.1 requires a minimum of 2 CPU and 2048 MB of memory. In our case, that means that **vm07** itself needs to meet or exceed these requirements. In the environment used for this whitepaper, we created **vm07** with 2 CPU and 6164 MB:

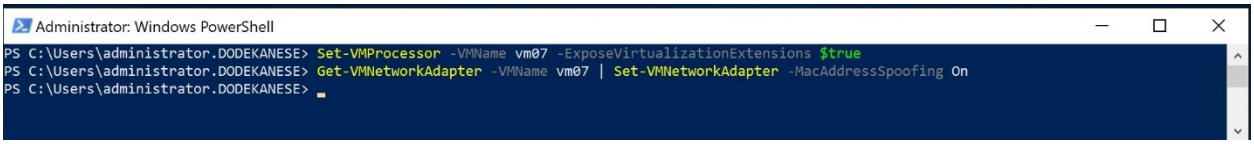


Furthermore, if you are installing an EFLOW VM with IoT Edge 1.1 inside a Windows VM, you need to configure the host VM to support nested VMs. To do this, you can run the following PowerShell commands in the host VM:

```
Set-VMProcessor -VMName <VMname> -ExposeVirtualizationExtensions $true
```

```
Get-VMNetworkAdapter -VMName <VMname> | Set-VMNetworkAdapter -MacAddressSpoofing On
```

So, to enable nested VMs in **vm07** we used Windows Remote Desktop (RDP) to remote into \\SKALA, launched PowerShell with Administrative privileges, and ran those commands:



Subsequent releases of IoT Edge may change the processor and memory requirements, as well as add or eliminate other requirements such as the need to run commands to support nested VMs. The reader should check the latest requirements on the Microsoft website before continuing.

At this point in the process, we have the following infrastructure:

1. A host machine running Windows, \\SKALA with IP 192.168.1.10, with Hyper-V enabled, hosting
2. VM **vm07** running Windows, \\PLATEIA with IP 192.168.1.103, with Hyper-V nesting enabled

Installing IoT Edge and EFLOW on the host VM

We are now ready to create the EFLOW VM in **vm07**. RDP into \\PLATEIA and copy the IoTWorX tools file into a location that can be accessed in the VM. In the deployment used for this whitepaper, we copied to a folder in the C:\ drive:

```
C:\Software\ICONICS\IoTWorX\IoTWorX_Tools_V10.97.1
```

Installing IoTWorX on IoT Edge on Windows

Before continuing, launch a command shell in \\PLATEIA and run `ipconfig /all` to note the Windows host DNS server address – you will need it later. In the configuration used for this whitepaper, the DNS is running on \\SKALA, at 192.168.1.10. Next, launch PowerShell with Administrative privileges and enter the following commands to allow script execution and run the PowerShell script to create an EFLOW VM with IoT Edge inside **vm07**:

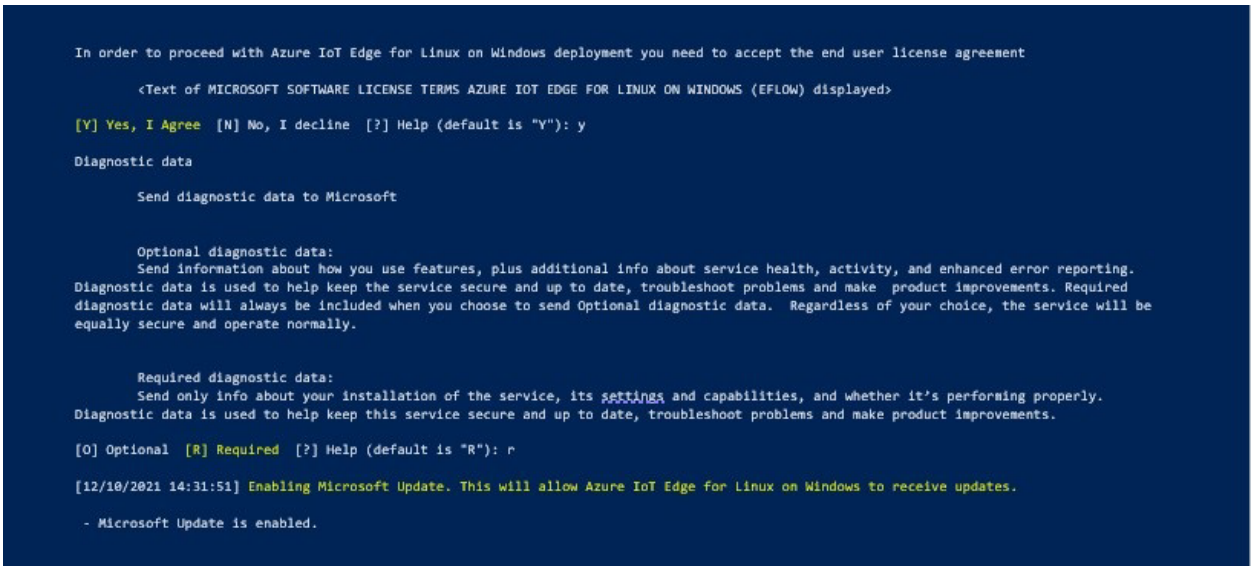
```
Set-ExecutionPolicy -ExecutionPolicy AllSigned -Force
```

```
PS C:\Users\Administrator> cd C:\Software\ICONICS\IoTWorX\IoTWorX_Tools_V10.97.1\
```

```
PS C:\Software\ICONICS\IoTWorX\IoTWorX_Tools_V10.97.1>  & '.\IoTWorX_inst_win.ps1'
```

When asked for the number of cores and memory, enter **2** and **2048,** respectively. Enter **Y** to confirm that you want the script to provision an EFLOW VM as an Edge device. Finally paste in the IoT Edge Device Connection String copied to Notepad earlier. The script will next check to see if Hyper-V is enabled in \\PLATEIA, and if it is not, Hyper-V is enabled and the system reboots.

If the system has rebooted, RDP back into it, launch PowerShell and restart the PowerShell script. Next the system will display the Microsoft software license and request the user to accept the terms, and whether to send optional or only required diagnostic data to Microsoft. Answer **Y** and then **R** to continue:



At this point the installation script is broken into several steps. In Step 1, the script checks for whether the required features to host VMs are enabled in \\PLATEIA:

```
[12/10/2021 14:31:51] Step 1: Preparing host for Azure IoT Edge for Linux on Windows


[12/10/2021 14:31:52] Checking host for required features

 - Checking the status of 'Microsoft-Hyper-V'
 - Checking the status of 'Microsoft-Hyper-V-Management-PowerShell'
 - Checking the status of 'Microsoft-Hyper-V-Hypervisor'
 - Checking the status of 'OpenSSH.Client*'
 - Verifying whether Hyper-V is enabled and functional
 - Hyper-V core services are active

[12/10/2021 14:31:56] Checking for virtual switch with name 'EFLOWSwitch'

 - The virtual switch 'EFLOWSwitch' of type 'External' is present
```

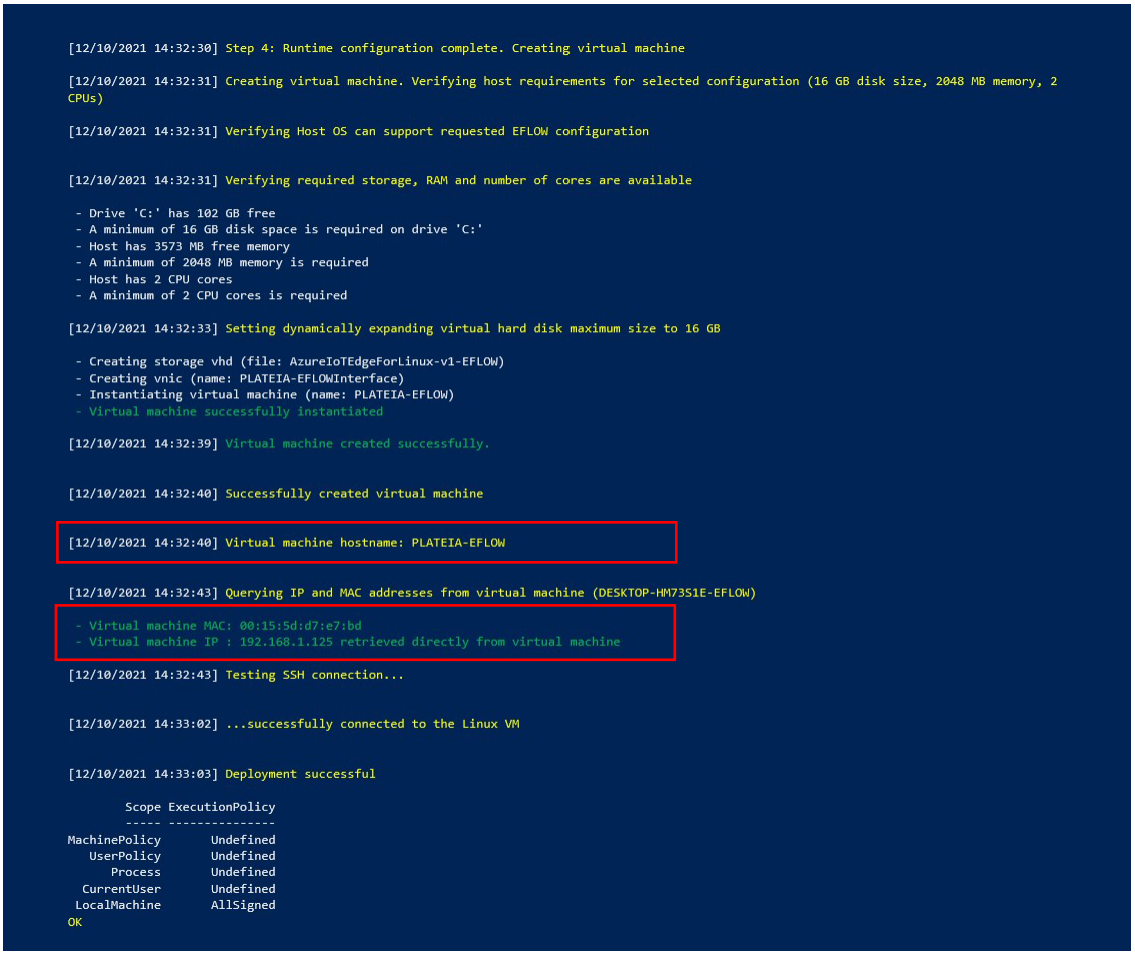
Then it checks for whether **vm07** has the prerequisite CPU, memory, and disk space:

```
[12/10/2021 14:31:56] Step 2: Verifying minimum host requirements


[12/10/2021 14:31:56] Verifying Host OS can support requested EFLOW configuration


[12/10/2021 14:31:56] Verifying required storage, RAM and number of cores are available

 - Drive 'C:' has 105 GB free
 - A minimum of 10 GB disk space is required on drive 'C:'
 - Host has 3934 MB free memory
 - A minimum of 1024 MB memory is required
 - Host has 2 CPU cores
 - A minimum of 2 CPU cores is required
```

Step 3 configures the requisite networking, including creating a network switch that can communicate externally if one does not already exist:

```
[12/10/2021 14:31:56] Step 3: Configuring Azure IoT Edge for Linux on Windows runtime


[12/10/2021 14:31:58] Associating wssdagent service with nodectl

[12/10/2021 14:31:59] Creating vnet (name: EFLOWSwitch)
[12/10/2021 14:32:01] Verifying installation

 - Verifying expected Windows host binaries
 - Verifying expected Linux VM image
 - Extracting Linux VM image
 - Testing for ssh key
 - Generating SSH key...
 - Testing for wssdagent service
 - Testing if wssdagent is running
 - Testing if container resource is provisioned
 - Testing if vnet resource 'EFLOWSwitch' is provisioned
```

Then in Step 4 the script starts to install the EFLOW VM inside **vm07**. Note the EFLOW VM name constructed is \\PLATEIA-EFLOW, and the IP address assigned is 192.168.1.125:



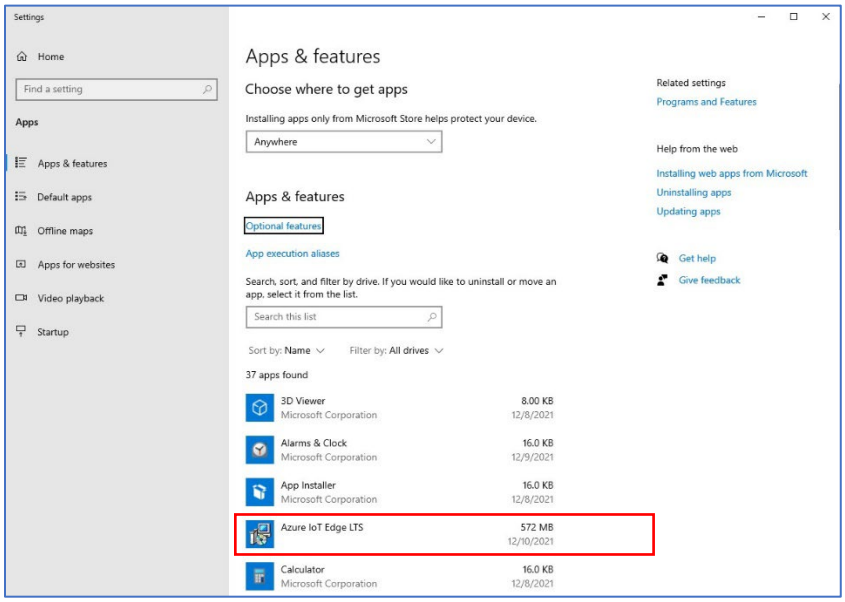
Once the EFLOW VM is created, the script updates firewall settings, including forwarding rules, to allow communication between applications inside the EFLOW VM and those outside of it:

```
~~Updating EFLOW VM firewall settings...~~
Open TCP port 8443.
Open TCP port 8843.
Open TCP port 8080.
Open TCP port 80.
Open TCP port 47808.
Save IPv4 settings
# Generated by iptables-save v1.8.3 on Fri Dec 10 22:33:06 2021
*mangle
:PREROUTING ACCEPT [2398:215698]
:INPUT ACCEPT [2384:212800]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2381:214607]
:POSTROUTING ACCEPT [2380:214531]
COMMIT
# Completed on Fri Dec 10 22:33:06 2021
# Generated by iptables-save v1.8.3 on Fri Dec 10 22:33:06 2021
*nat
:PREROUTING ACCEPT [19:2125]
:INPUT ACCEPT [11:572]
:OUTPUT ACCEPT [1:40]
:POSTROUTING ACCEPT [1:40]
:DOCKER - [0:0]
-A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER
-A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER
-A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE
-A DOCKER -i docker0 -j RETURN
COMMIT
# Completed on Fri Dec 10 22:33:06 2021
# Generated by iptables-save v1.8.3 on Fri Dec 10 22:33:06 2021
*filter
:INPUT DROP [1:172]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:DOCKER - [0:0]
:DOCKER-ISOLATION-STAGE-1 - [0:0]
:DOCKER-ISOLATION-STAGE-2 - [0:0]
:DOCKER-USER - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8843 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8080 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p udp -m udp --dport 47808 -j ACCEPT
-A FORWARD -j DOCKER-USER
-A FORWARD -j DOCKER-ISOLATION-STAGE-1
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A OUTPUT -j ACCEPT
-A DOCKER-ISOLATION-STAGE-1 -i docker0 ! -o docker0 -j DOCKER-ISOLATION-STAGE-2
-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
COMMIT
# Completed on Fri Dec 10 22:33:06 2021
```

Finally, IoT Edge is installed in the EFLOW VM, and the script terminates:

```
~~Provisioning EFLOW VM...~~

[12/10/2021 14:33:06] Provisioning virtual machine...


[12/10/2021 14:33:06] Provisioning iotedge for the first time.


[12/10/2021 14:33:06] Setting provisioning information for provisioning type: ManualConnectionString


[12/10/2021 14:33:12] Provisioning successful. iotedge service running.

OK
~~EFLOW VM installation complete and provisioned as IoT Edge device.~~
```

At this point, you can see the IoT Edge application in Add/Remove Programs in the Windows Control Panel of \\PLATEIA:



Once EFLOW is set up, launch and elevate PowerShell session and type the command **`Connect-EflowVm`** to connect to the EFLOW VM and run Linux commands to query or configure the environment:

```
PS C:\Software\ICONICS\IoTWorX\IoTWorX_Tools_V10.97.1> Connect-EflowVm
```

This will bring up a bash terminal within PowerShell, where you can run Linux commands in the EFLOW VM, including utilities like **`ifconfig`** or **`iotedge list`**.

At this stage, the Linux computer in the EFLOW VM is called \\PLATEIA-EFLOW:

```
iotedge-user@PLATEIA-EFLOW [ ~ ]$ hostname
PLATEIA-EFLOW
```

It has an IP address of 192.168.1.125:

```
iotedge-user@PLATEIA-EFLOW [ ~ ]$ ifconfig
br-2773b3635600 Link encap:Ethernet  HWaddr 02:42:ed:59:25:f6
          inet addr:172.18.0.1  Bcast:172.18.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:edff:fe59:25f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:127 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:18918 (18.9 KB)  TX bytes:28065 (28.0 KB)

docker0   Link encap:Ethernet  HWaddr 02:42:7f:55:9c:1a
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0      Link encap:Ethernet  HWaddr 00:15:5d:d7:e7:bd
          inet addr:192.168.1.125  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fed7:e7bd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:49118 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7036 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:71399465 (71.3 MB)  TX bytes:484325 (484.3 KB)
```

Some networking configurations prevent the EFLOW VM from getting the DNS server address from the host, so it is worth checking the DNS server in the EFLOW VM and configuring it manually if not there. Remembering the Windows host DNS server address noted earlier, follow the instructions in https://aka.ms/AzEFLOW-DNS to check and configure the DNS in the EFLOW VM manually if needed.

At this point, the IoT Edge edgeAgent should be running:

```
iotedge-user@PLATEIA-EFLOW [ ~ ]$ sudo iotedge list
NAME            STATUS          DESCRIPTION      CONFIG
edgeAgent       running         Up 22 minutes    mcr.microsoft.com/azureiotedge-agent:1.1
```

To recap, at this point in the process we have the following:

1. A host machine running Windows, \\SKALA with IP 192.168.1.10, with Hyper-V enabled, hosting
2. VM **vm07** running Windows, \\PLATEIA with IP 192.168.1.103, running Azure IoT Edge LTS, with Hyper-V enabled, hosting
3. EFLOW VM running Linux, \\PLATEIA-EFLOW with IP 192.168.1.125, running the Azure IoT Edge edgeAgent

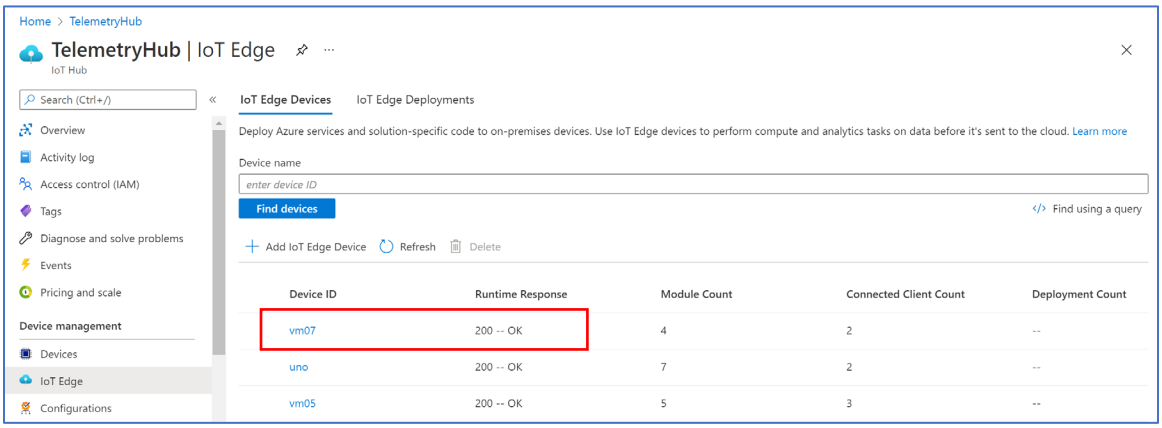
Installing IoTWorX on the EFLOW VM

Now that \\PLATEIA is running Azure IoT Edge and registered as an Edge device in IoT Hub, we can install ICONICS edge modules on it.

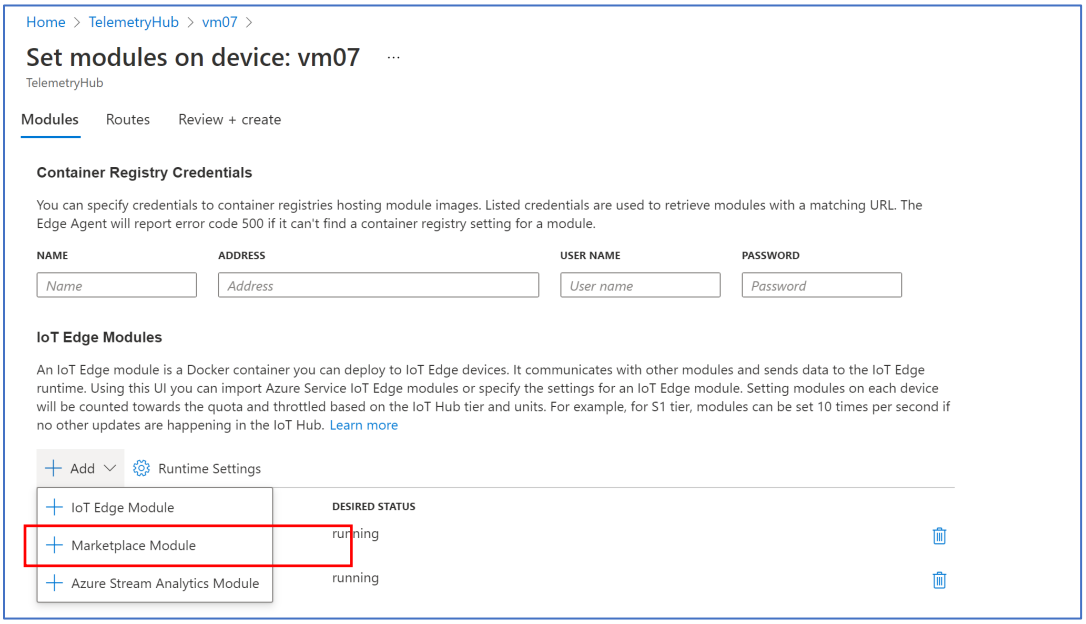
There are two ways you can do this. The easiest is to install from the Azure **IoT Edge Module Marketplace**, and the other way is to deploy from an instance of the **ICONICS Suite** following the instructions in *Quick Start.* Both methods are described here, but the reader only needs to follow the steps in one of them.

Option 1: Deploying from the Azure IoT Edge Module Marketplace

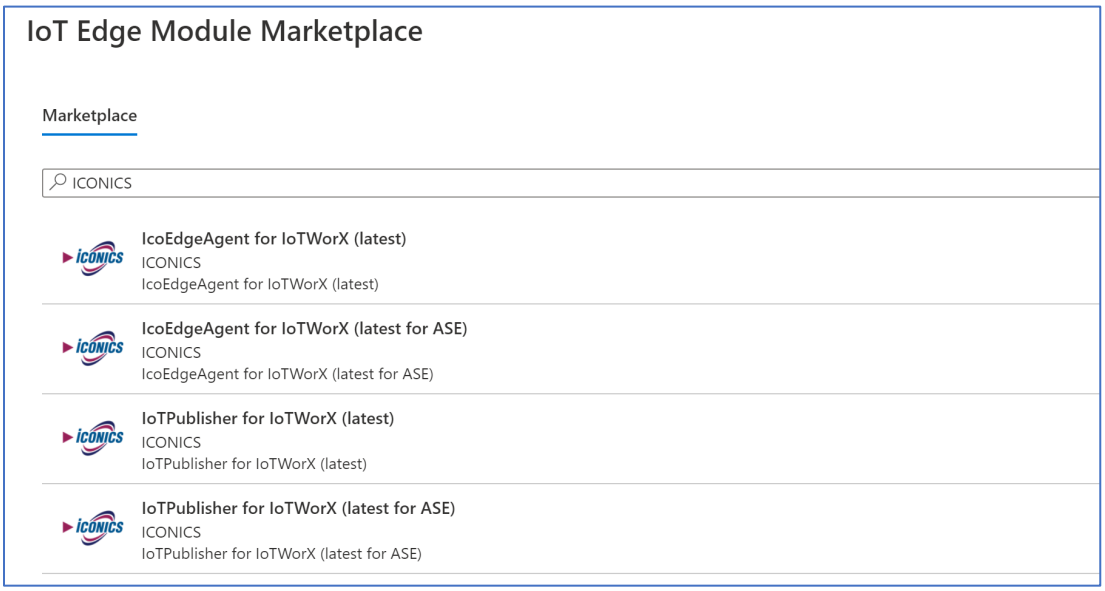
To install the ICONICS edge modules from the Azure **IoT Edge Module Marketplace**, bring up the Azure portal in a browser and select your IoT Hub – in our case **TelemetryHub**. The instructions in the previous section will have created an Edge ID for **vm07**, which you can see by navigating to the IoT Edge section under Device management:



Select **vm07** and then the **Set modules** tab to bring you to a screen where you can select modules to deploy to the Edge device **vm07**. Click **Add**, and select **Marketplace Module:**



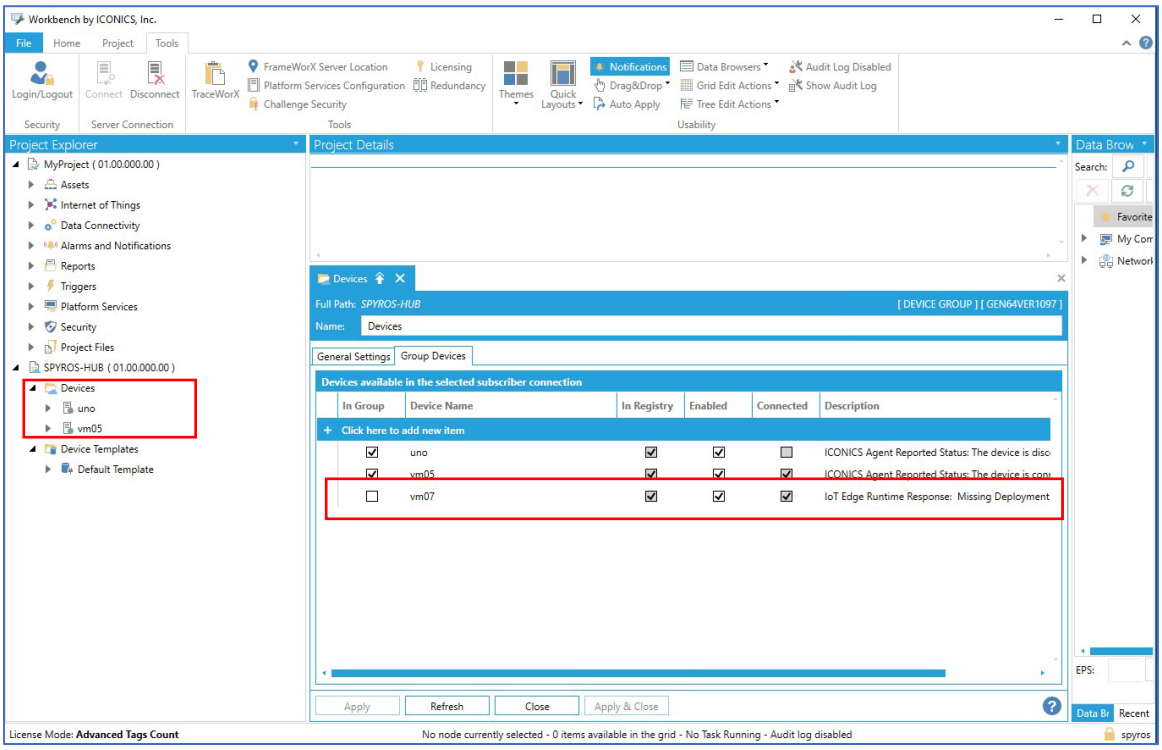
In the subsequent screen, search for ICONICS to bring up IoT Edge modules that can be deployed:



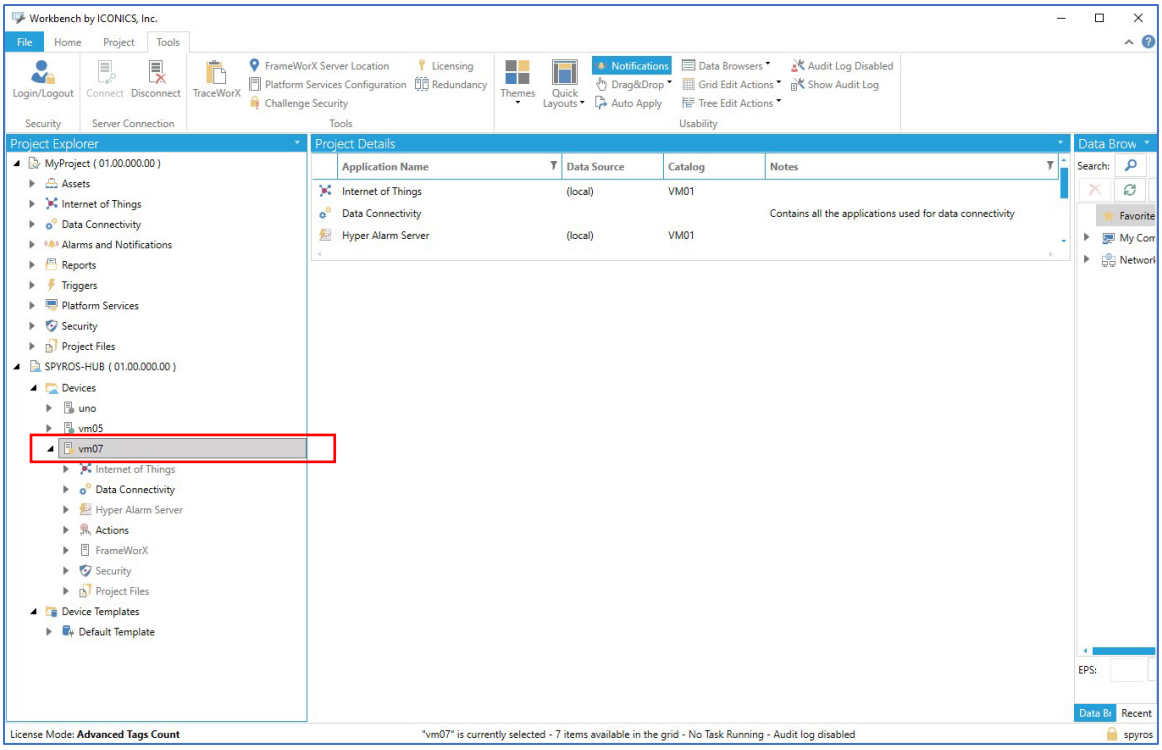
Select first **IcoEdgeAgent for IoTWorX (latest)** and then **IoTPublisher for IoTWorX (latest)** and then **Review + create** to deploy those modules to **vm07**. Once you have competed these steps, jump to Confirming deployment below.

Option 2: Deploying from an instance of ICONICS Suite

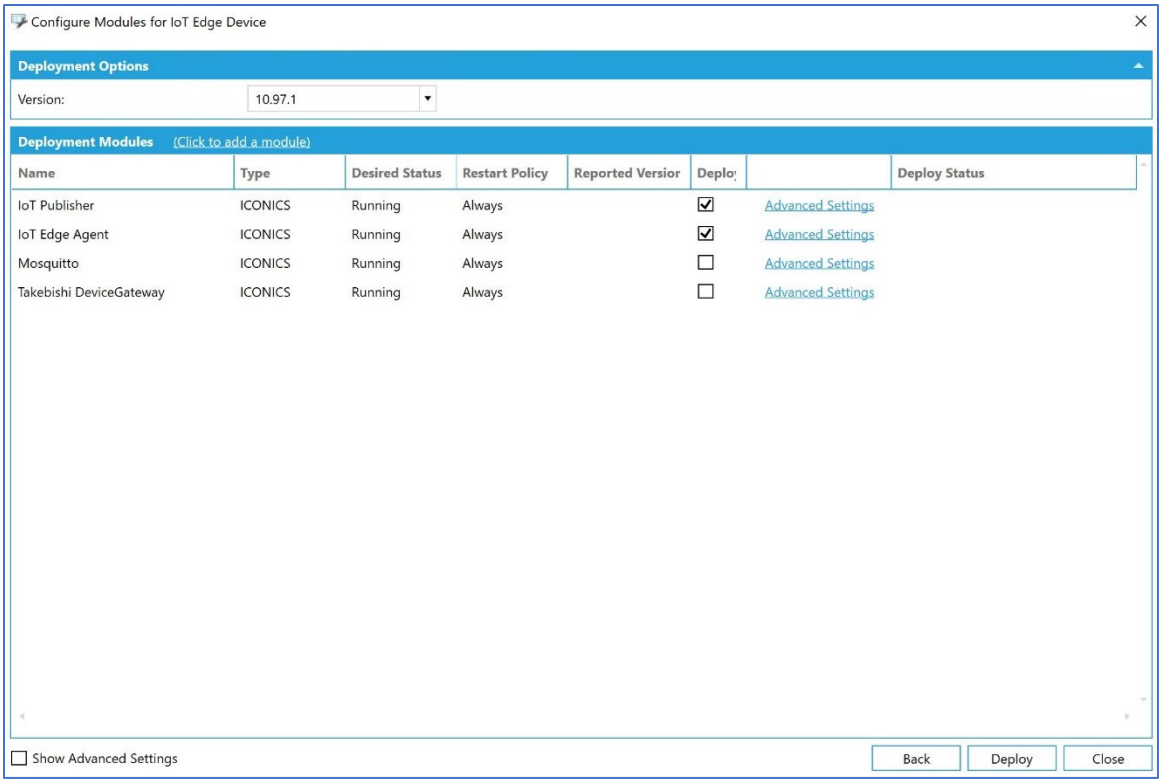
Another way to install **IoTWorX** is to do so from an instance of **ICONICS Suite**. Use RDP to access \\GEN64ver1097, and launch **ICONICS Workbench**. In the left navigation menu, expand **Devices** under your IoT project, in our case the project called **SPYROS-HUB**. The Edge device, **vm07**, will not yet be visible in the list of devices, though in our case two other devices (**uno** and **vm05**) we added previously are showing. To add **vm07** to the list of available devices, right click on **Devices** and select **Configure Device Group Settings**. Devices registered in IoT Hub are listed in the **Group Devices** tab, and **vm07** can be seen there:



Select device **vm07** and click **Apply & Close**. In the left navigation menu, refresh the **Devices** folder, and you should see **vm07**:



Next, we want to specify which Edge modules we want to deploy to \\PLATEIA. Right click on **vm07** and click **Configure Modules** to open the **Configure Modules for IoT Edge Device** window. Click on **Click to Configure Modules** to bring up the **Deployment Modules** pane:



Select **IoT Publisher** and **IoT Edge Agent** to deploy those two Edge modules and click **Deploy**.

Confirming deployment

After a minute or two, RDP into the Edge device \\PLATEIA, launch PowerShell, and list the modules deployed:

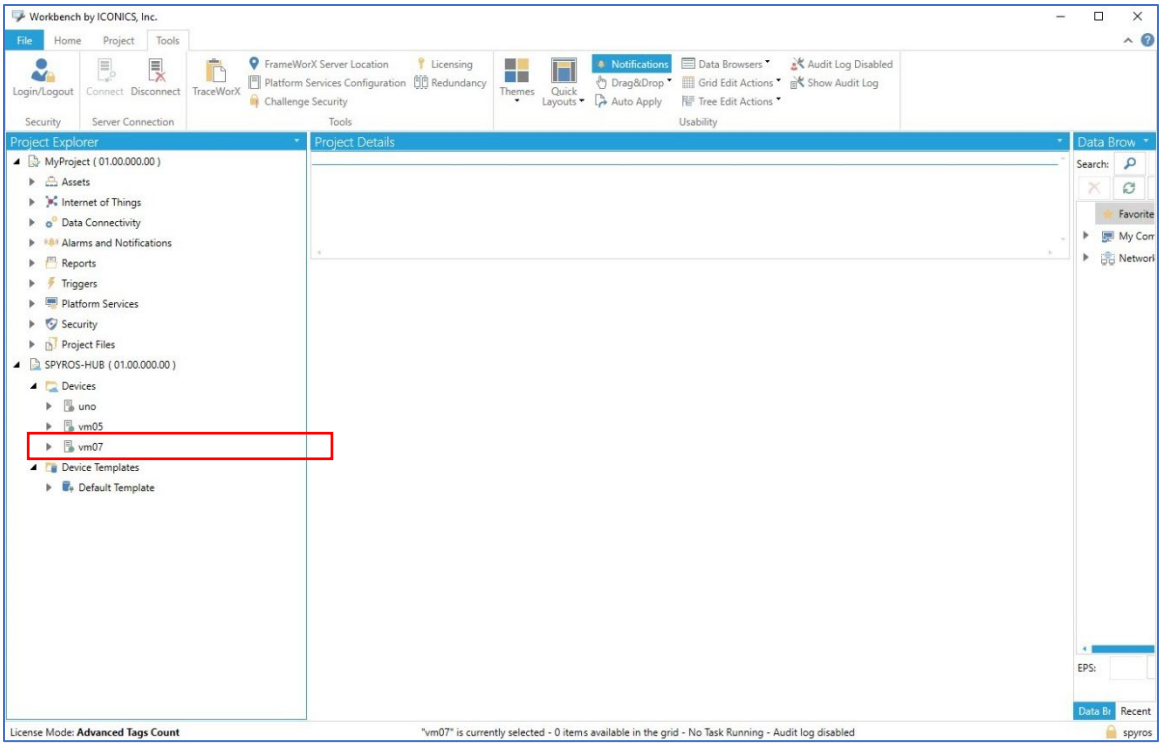
```
iotedge-user@PLATEIA-EFLOW [ ~ ]$ sudo iotedge list
NAME            STATUS          DESCRIPTION      CONFIG
IcoEdgeAgent    running         Up 35 seconds    iconics/icoedgeagent:10.97.1
IoTPublisher    running         Up a minute      iconics/iotworx:10.97.1
edgeAgent       running         Up a minute      mcr.microsoft.com/azureiotedge-agent:1.1.0
edgeHub         running         Up 44 seconds    mcr.microsoft.com/azureiotedge-hub:1.1.0
```

The modules you specified in either Option 1 or Option 2 above are visible and running.

Configuring IoTWorX to get and send device data

To this point you have installed ICONICS IoTWorX on the IoT Edge device **vm07**, and it is connected to the IoT Hub **TelemetryHub**. However, you have not configured IoTWorX to do anything, so it is not sending any data to IoT Hub. To keep IoTWorX lean, there is no local user interface on **vm07** to configure IoTWorX, so you need to do that from an instance of ICONICS Suite.

To do this, first RDP into \\GEN64ver1097 and launch the ICONICS Workbench application. In the left navigation pane, you will see that the Edge device **vm07** icon is green, meaning it is receiving data from IoTWorX:

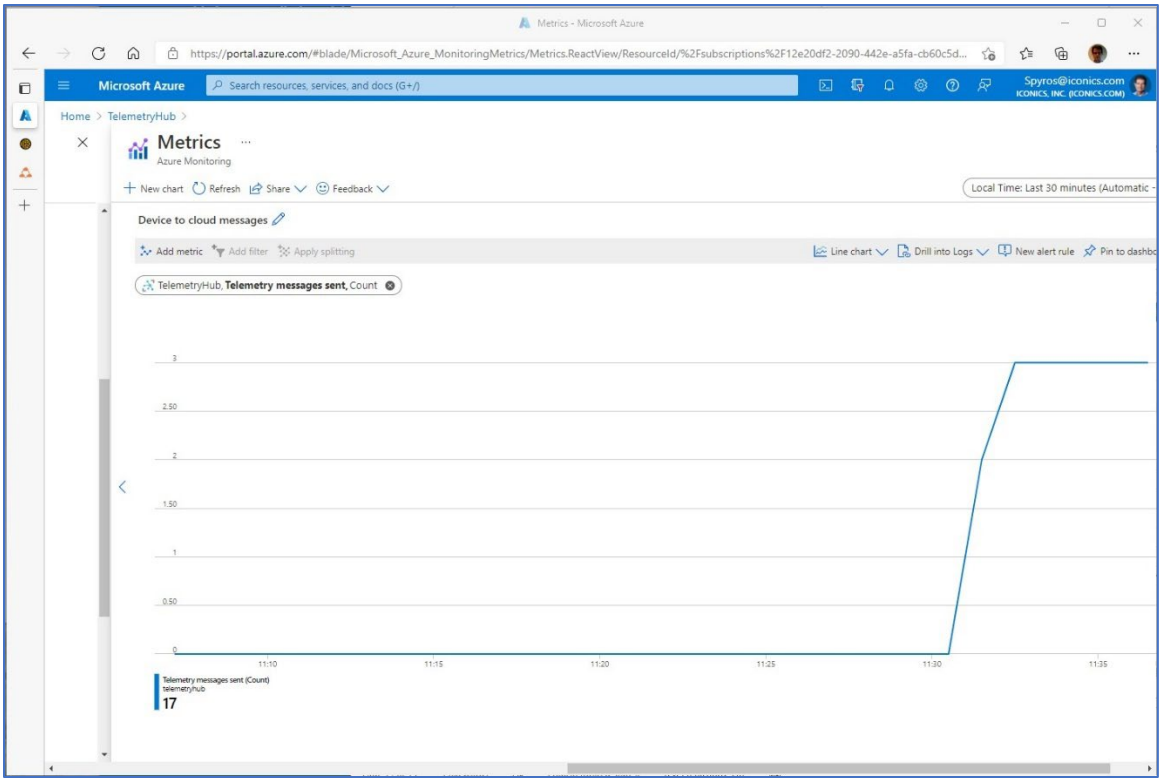


No device data is being sent by **vm07** yet, as we have not deployed any instructions to the device on polling any local devices. If you follow the instructions in *Quick Start* to deploy a simulator in IoTWorX, **vm07** will send the simulated data to \\GEN64ver1097. Similarly, if you have a physical device that supports BACnet or Modbus communications, you can follow the instructions in Connecting a Delta VFD to Microsoft Azure using ICONICS IoTWorX or Monitoring Building Power Consumption to get **vm07** sending data from those devices.

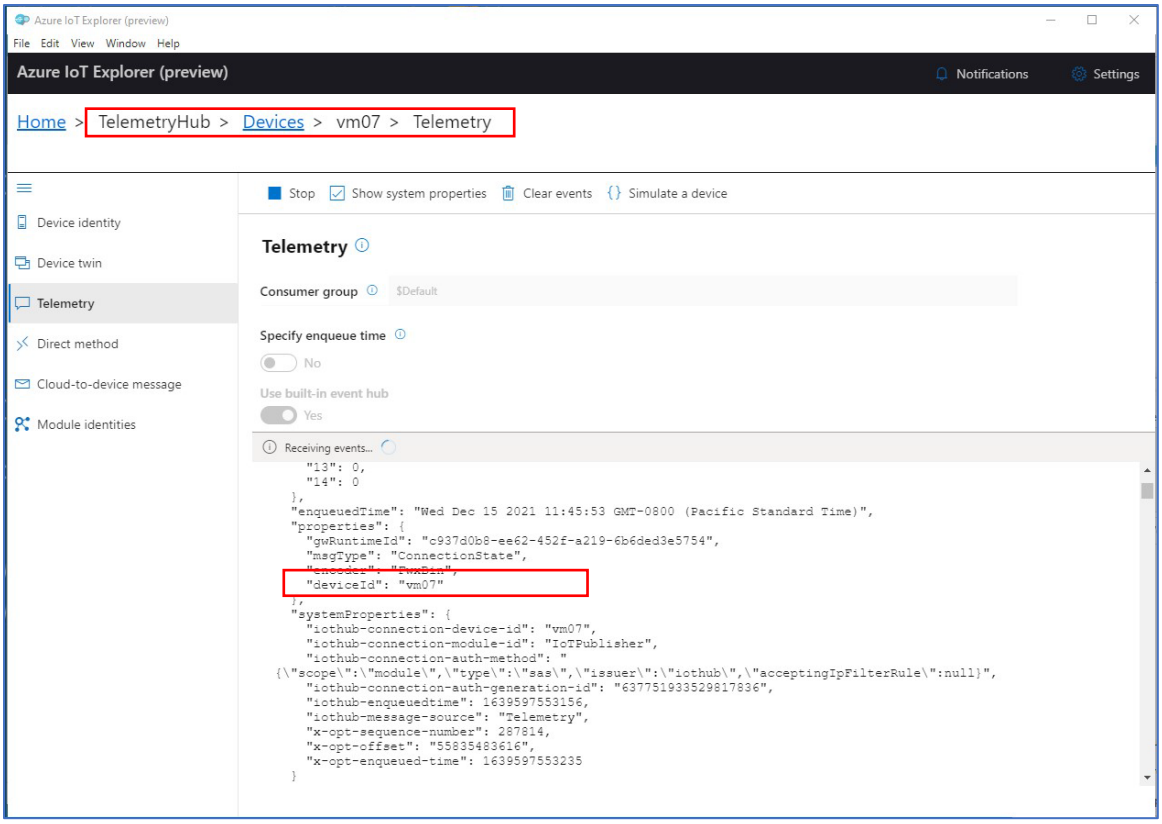
To see the data, we need to take some additional steps. The following additional whitepapers include instructions for some options to display and analyze the data in **ICONICS Suite**, and should prove helpful:

- Combining IoTWorX and Azure Stream Analytics on Azure IoT Edge
- Measuring Occupancy with Delta Controls O3 Sense, Azure IoT, and ICONICS
- Air Quality Monitoring Solution with ICONICS and Sysinno

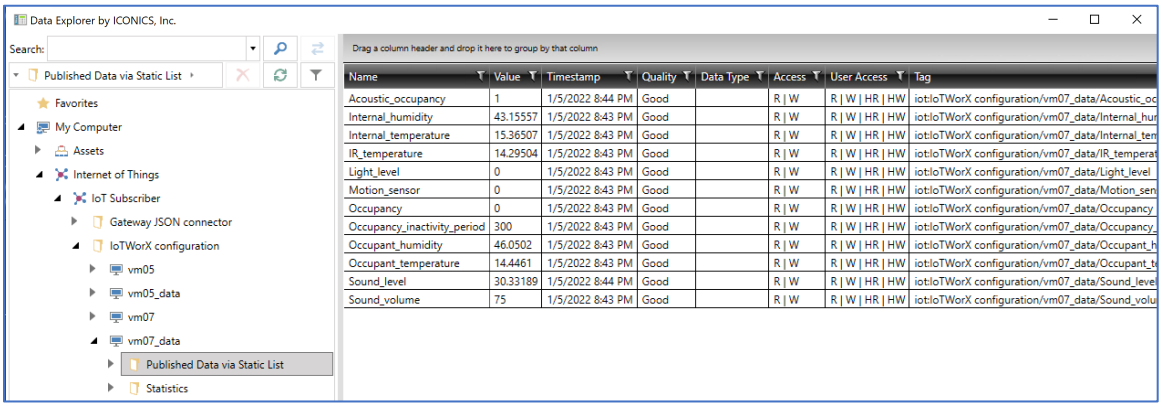
To follow up, if you followed the instructions in this whitepaper along with the referenced whitepapers, you would now be able to see data coming from a BACnet device arriving in IoT Hub in the Azure portal:



Next, we were able to see the raw data coming into IoT Hub using Azure IoT:



And finally, because we subscribed ICONICS Suite in \\GEN64ver1097, we were able to see the interpreted data in ICONICS Data Explorer:



Next Steps

At this point, we have done what we set out to do in this whitepaper: we created an Edge device, deployed IoTWorX modules to the Edge device, and the device is sending data to the ICONICS application in Azure.

Founded in 1986, ICONICS is an award-winning global software provider offering real-time visualization, HMI/SCADA, energy management, fault detection, manufacturing intelligence, MES, and a suite of analytics solutions for operational excellence. ICONICS solutions are installed in 70 percent of the Global 500 companies around the world, helping customers to be more profitable, agile and efficient, to improve quality, and to be more sustainable.

ICONICS is leading the way in cloud-based solutions with its HMI/SCADA, analytics, mobile and data historian to help its customers embrace the Internet of Things (IoT). ICONICS products are used in manufacturing, building automation, oil and gas, renewable energy, utilities, water and wastewater, pharmaceuticals, automotive, and many other industries. ICONICS' advanced visualization, productivity, and sustainability solutions are built on its flagship products: GENESIS64™ HMI/SCADA, Hyper Historian™ plant historian, AnalytiX® solution suite, and MobileHMI™ mobile apps. Delivering information anytime, anywhere, ICONICS' solutions scale from the smallest standalone embedded projects to the largest enterprise applications.

ICONICS promotes an international culture of innovation, creativity, and excellence in product design, development, technical support, training, sales, and consulting services for end users, systems integrators, OEMs, and channel partners. ICONICS has over 375,000 applications installed in multiple industries worldwide.

ICONICS Sales Offices

World Headquarters

100 Foxborough Blvd.
Foxborough, MA, USA, 02035
+1 508 543 8600
us@iconics.com

European Headquarters

Netherlands
+31 252 228 588
holland@iconics.com

Australia
+61 2 9605 1333
australia@iconics.com

China
+86 10 8494 2570
china@iconics.com

Czech Republic
+420 377 183 420
czech@iconics.com

France
+33 4 50 19 11 80
france@iconics.com

Germany
+49 2241 16 508 0
germany@iconics.com

India
+91 265 6700821
india@iconics.com

Italy
+39 010 46 0626
italy@iconics.com

Singapore
+65 6667 8295
singapore@iconics.com

UK
+44 1384 246 700
uk@iconics.com

For more, visit iconics.com

Gold
Microsoft Partner
Ten-time Microsoft Partner of the Year

VISUALIZE HISTORIZE MOBILIZE ANALYZE IoT CLOUD