

zenon IoT Edge Module

Quick Start Tutorial

www.copadata.com
sales@copadata.com



zenon
by COPA-DATA

Content

| | |
|--|----|
| 1. INTRODUCTION | 2 |
| 1.1. What are we going to do? | 2 |
| 1.2. Prerequisites | 3 |
| 2. CREATING AN IOT HUB AND AN EDGE DEVICE | 4 |
| 3. INSTALL THE IOT EDGE RUNTIME..... | 6 |
| 4. UPLOAD THE ZENON PROJECT TO THE BLOB STORAGE..... | 7 |
| 4.1. Create an Azure storage account..... | 8 |
| 4.2. Install the Upload Wizard | 8 |
| 4.3. Upload the demo project..... | 9 |
| 5. DEPLOY THE ZENON IOT EDGE MODULE | 11 |
| 6. HAVE A LOOK AT YOUR DATA..... | 13 |
| 7. HOW TO PROCEED FROM HERE? | 15 |

1. Introduction

Cloud platforms like Microsoft Azure offer a lot of easy to use, scalable functionality. This is especially true, when you look at industrial automation. The production process creates huge amounts of data from machines or other systems, which has to be stored and visualized. Additionally, there are all kinds of different services from machine learning to order management and mobile apps, where a centralized platform can offer big benefits.

Nevertheless, there is a rising need to have “cloud-compatible” services deployed on premises. This could be to aggregate or harmonize data before you send it to the cloud, to make some calculations or to access that data from your industrial equipment in the first place.

The zenon IoT Edge module is a deployable component for the Microsoft IoT Edge Runtime, which offers you the data acquisition functionality of a zenon Supervisor inside the IoT Edge Runtime.

In this tutorial you will learn, how you can install and run this zenon IoT Edge module and how to realize your own zenon IoT Edge projects.

1.1. What are we going to do?

During this tutorial you will:

- Install a Microsoft IoT Edge Runtime on one of your (on-premises) machines
- Deploy and configure a zenon IoT Edge module into that runtime
- Optionally upload a zenon demo project for that module into a MS Azure Blob Storage
- Run the zenon IoT Edge module and see data coming into your MS Azure IoT Hub

1.2. Prerequisites

All the components, that you need, can be downloaded from COPA-DATAs GitHub Repository from:

<https://github.com/COPA-DATA>

Additionally, you will need:

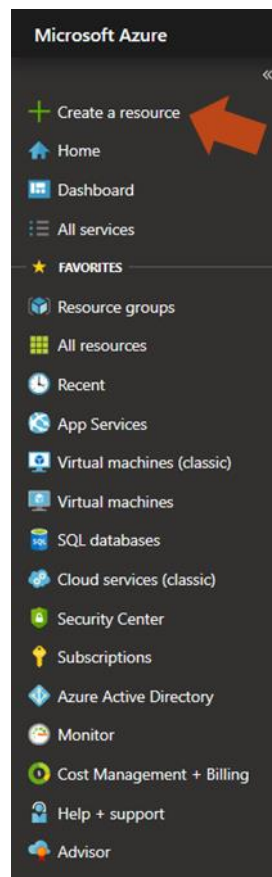
- A Windows 10 Enterprise / Windows Server 2019 Version 1809 / Build 17763 machine to host the IoT Edge Gateway (see [Supported Operating Systems](#))
- A licensed zenon Editor installation (Version 8.20 or higher)
- An MS Azure Account

In MS Azure your will need:

- An IoT Hub
- A storage account

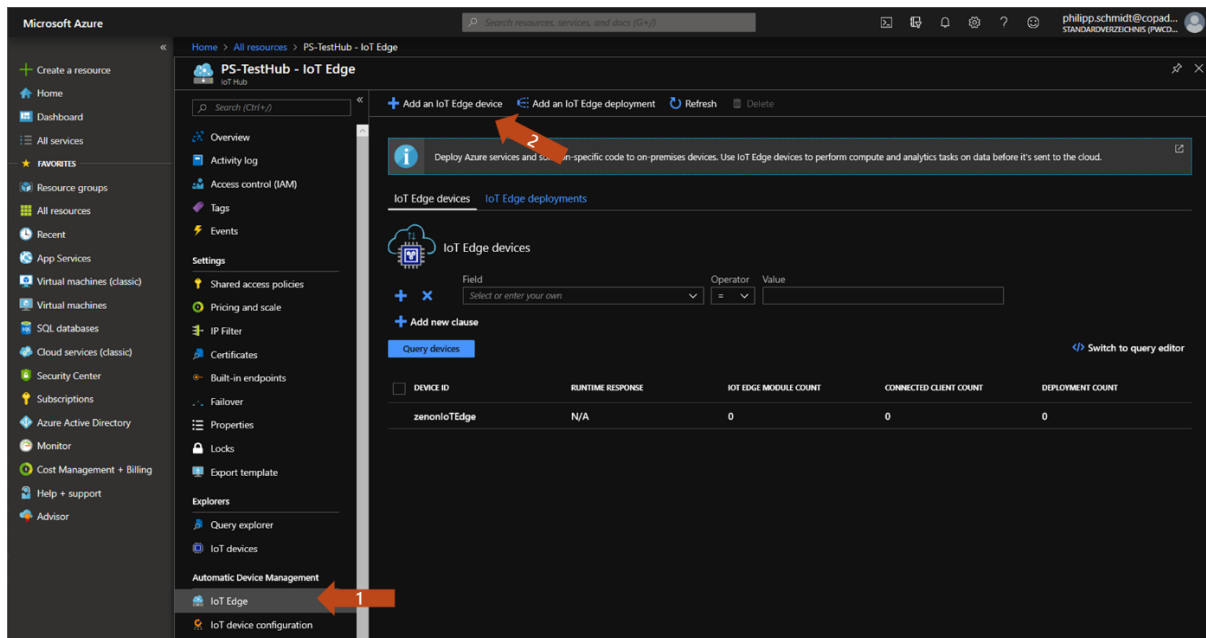
2. Creating an IoT Hub and an Edge Device

First of all, you will need an IoT Hub in your MS Azure account. So, if you don't have one already, please create it now. You can do so by clicking on "Create a resource" in your MS Azure portal.



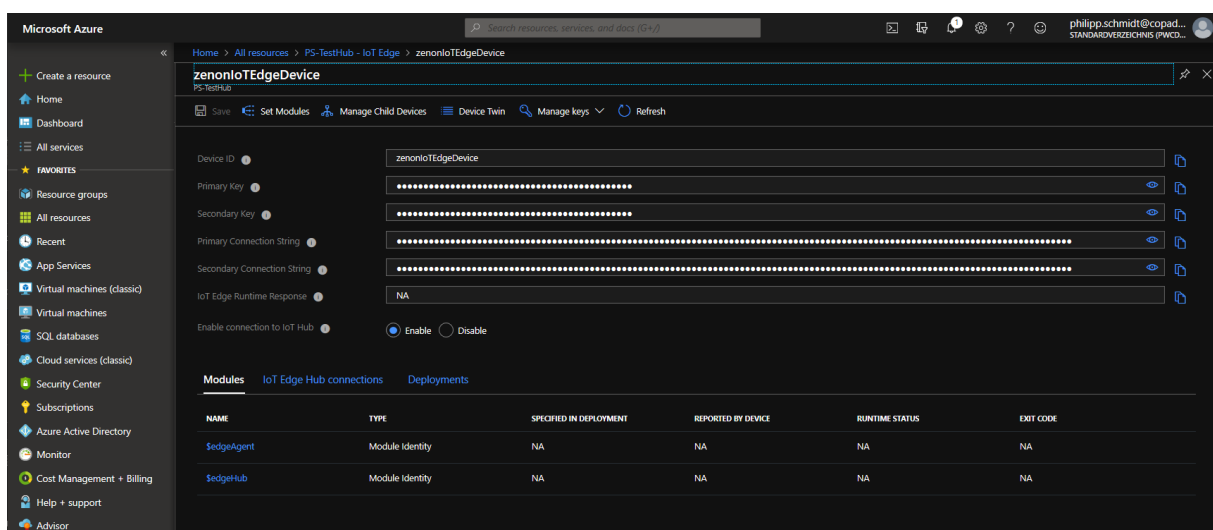
For our purposes the free starter IoT Hub "F1 – Free" should be enough.

Now that we have the hub, we will create an edge device for it. To do that, please select your IoT Hub, in its properties pane go to "IoT Edge" and click on "Add an IoT Edge device".



In the upcoming dialog you only have to give a name to your device. Mine is called “zenonIoTEdgeDevice”. All other settings can be kept at their defaults.

When clicking on your device you will find a screen like this:



Please copy your Primary Connection String, we will need it later.

3. Install the IoT Edge Runtime¹

On your local machine, that you want to configure as edge device, please start a 64bit PowerShell as an administrator and execute the following command:

| PowerShell |
|---|
| <pre>. {Invoke-WebRequest -useb https://aka.ms/iotedge-win} Invoke-Expression; ` Deploy-IoTEdge</pre> |

This will install the windows based IoT Edge Runtime on your machine. If prompted for a reboot, allow it.

After the reboot open PowerShell as an administrator again and initialize your IoT Edge runtime:

| PowerShell |
|---|
| <pre>. {Invoke-WebRequest -useb https://aka.ms/iotedge-win} Invoke-Expression; ` Initialize-IoTEdge</pre> |

When prompted please provide the device connection string to your IoT Edge device, that you have copied earlier.

You can check whether your installation was successful by typing:

| PowerShell |
|--------------------------------|
| <pre>Get-Service iotedge</pre> |

¹ The following text is just a short version of Microsofts installation instruction. If you have any problems, please refer to: [Install the Azure IoT Edge runtime on Windows](#)



...and have a look at the logs:

| PowerShell |
|--|
| . {Invoke-WebRequest -useb https://aka.ms/iotedge-win} Invoke-Expression; Get-IoTEdgeLog |

List all running modules:

| PowerShell |
|--------------|
| iotedge list |

For now, this will just list the edgeAgent. Once we have deployed the zenon IoT Edge module you will it and the edgeHub, too.

4. Upload the zenon project to the blob storage

The zenon IoT Edge module, that we're going to deploy in the next step, will execute a zenon runtime internally and of course, that zenon runtime needs a zenon project to function properly. The module is preconfigured to automatically download and run a simple demonstration project, so it will work "out of the box". If you just want to get on with your deployment, you can skip this step for now and come back later, when you want to use your own projects.

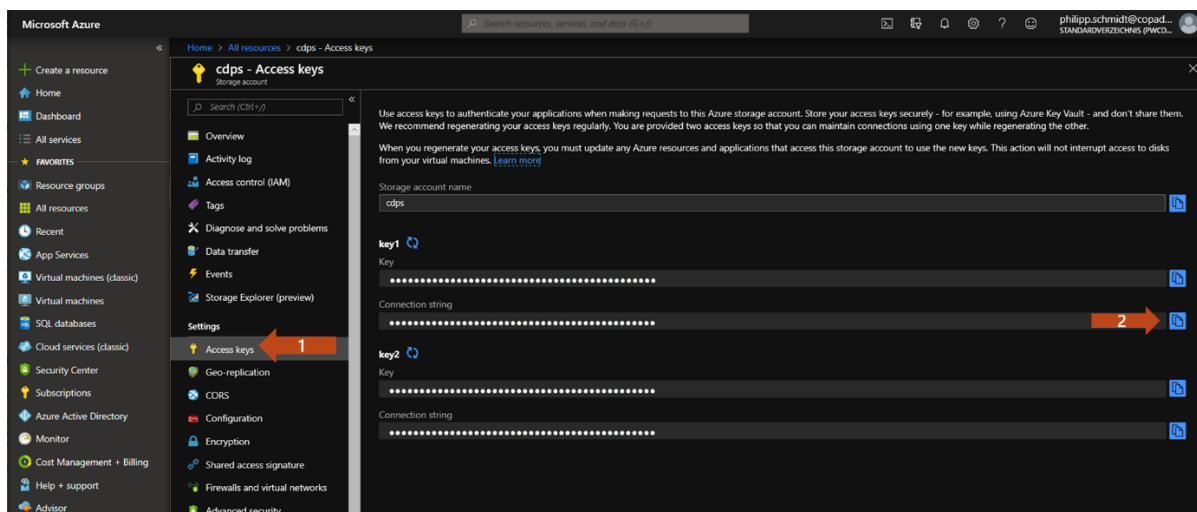
The zenon IoT Edge Module, like most modules, is configured through environment variables, that can be set directly in the device configuration in your Azure Portal. This way, after setting up the IoT Edge Device itself, all the configuration for all your devices can be done easily from one central point.

In order to stick to this centralized approach, the zenon IoT Edge Module will download its zenon project from an Azure Blob Storage, which you will configure using its environment variables.

4.1. Create an Azure storage account

If you don't have one already, please create a storage account in your MS Azure account now. Just like before, click on "Create a resource" and choose "Storage account". You can, of course, choose the settings to you liking, but for a simple test a "Standard" account with LRS Replication should be enough (and cheap). But we will need a public endpoint, so that your storage is accessible from your edge device. For our purposes we will only need a few MB of data.

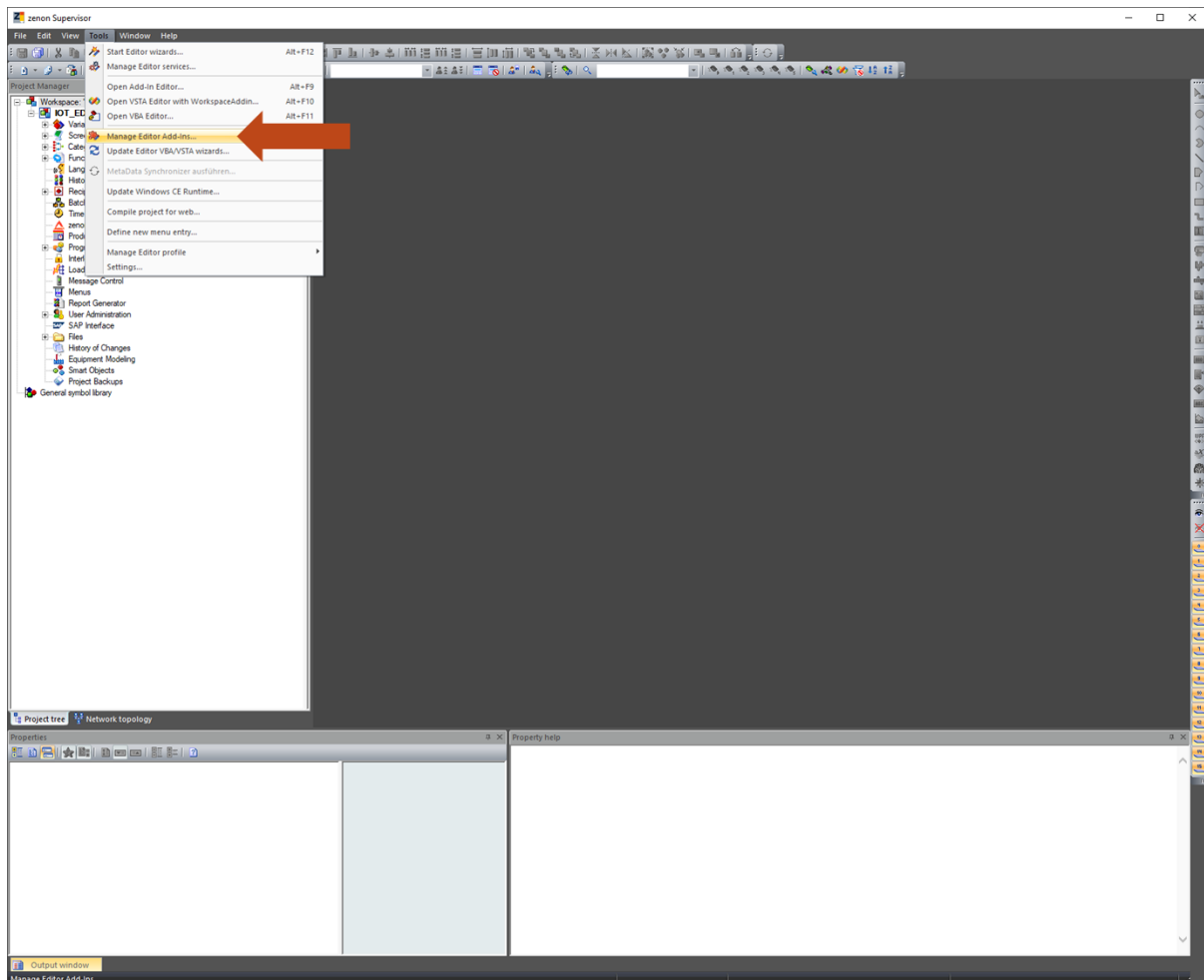
Please select that account and open the "Access keys" page. Copy the primary access key, as we will need it later.



4.2. Install the Upload Wizard

To make life a little bit easier we provide a wizard, that will automatically upload a zenon project to your MS Azure Blob Storage in the correct format. To install this wizard please download it from COPA-DATAs GitHub.

Next open your zenon Editor and select "Manage Editor Add-Ins..." in the Tools menu:

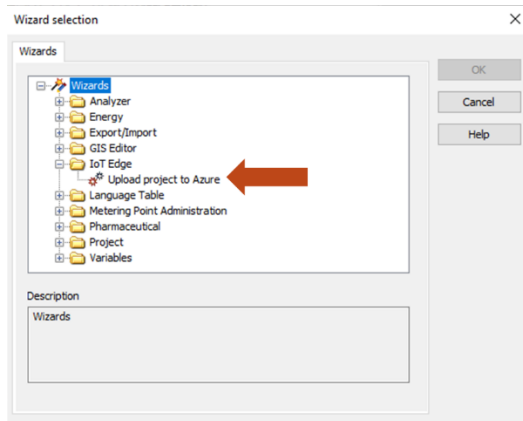


In the upcoming dialog click on the Icon "Import and install..." and select the "UploadProjectToAzure.scadaAddIn" file you just downloaded.

4.3. Upload the demo project

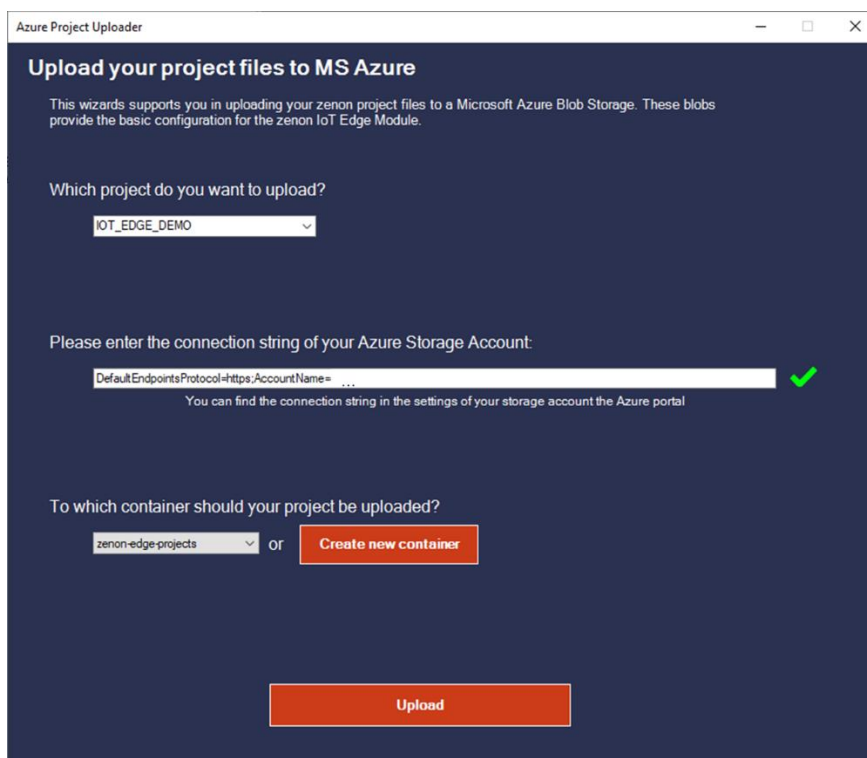
In the GitHub repository you will find a zenon project backup named "iot_edge_demo.zip". We will use that project for now. If you want to use your own project, please mind the hints at the end of this tutorial.

Now just start the wizard, that we have installed during the last step, by hitting ALT+F12 or selecting the menu-entry in the Tools menu. Select the entry "Upload project to Azure" in the category "IoT Edge".



In the upcoming dialog select the IOT_EDGE_DEMO as the project you want to upload and paste the connection string to your storage account, that we have copied earlier into the corresponding textbox.

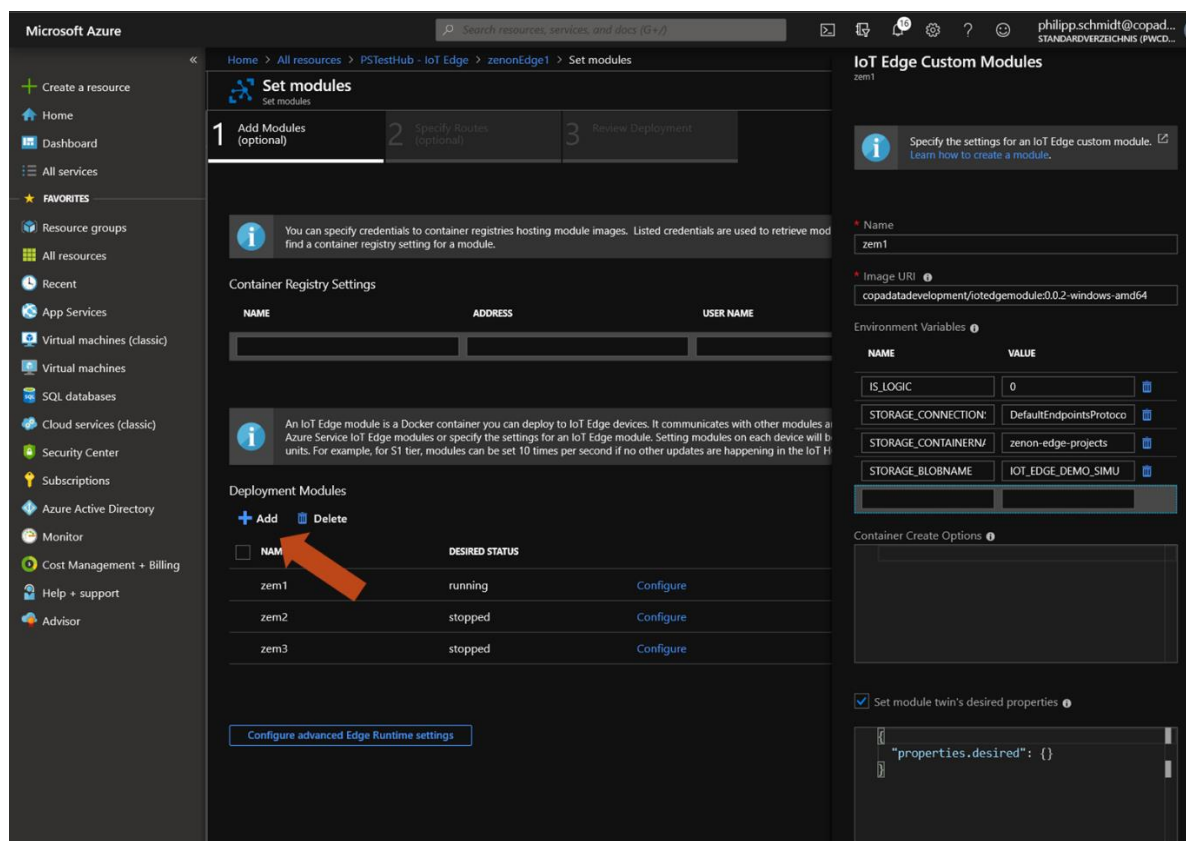
Finally, we have to choose or create a blob storage container to which we want to upload the project. If you don't have one, just create a new one by clicking on the button "Create container".



As you can see, my container has the name “zenon-edge-projects”. Now all we have to do is hit the “Upload”- Button, and the project files will be upload to the selected blob storage container to a blob named like the project itself “iot_edge_demo”.

5. Deploy the zenon IoT Edge module

As we have connected the IoT Edge runtime to our IoT Hub the rest of its management can be done right from the Azure portal. Please select your IoT Hub and navigate to “IoT Edge”, there you will find the IoT Edge device, that we have created earlier. After clicking on that device, select “Set Modules” and add the “zenon on IoT Edge” Module as seen below.



Microsoft Azure

Home > All resources > PStestHub - IoT Edge > zenonEdge1 > Set modules

Set modules

1 Add Modules (optional) 2 Specify Routes (optional) 3 Review Deployment

You can specify credentials to container registries hosting module images. Listed credentials are used to retrieve mod find a container registry setting for a module.

Container Registry Settings

| NAME | ADDRESS | USER NAME |
|------|---------|-----------|
| | | |

An IoT Edge module is a Docker container you can deploy to IoT Edge devices. It communicates with other modules a Azure Service IoT Edge modules or specify the settings for an IoT Edge module. Setting modules on each device will b units. For example, for S1 tier, modules can be set 10 times per second if no other updates are happening in the IoT H

Deployment Modules

+ Add - Delete

| NAME | DESIRED STATUS | |
|------|----------------|-----------|
| zem1 | running | Configure |
| zem2 | stopped | Configure |
| zem3 | stopped | Configure |

Configure advanced Edge Runtime settings

IoT Edge Custom Modules

zem1

Specify the settings for an IoT Edge custom module. [Learn how to create a module.](#)

Name: zem1

Image URI: copadatadevelopment/iotedgmodule:0.0.2-windows-amd64

Environment Variables

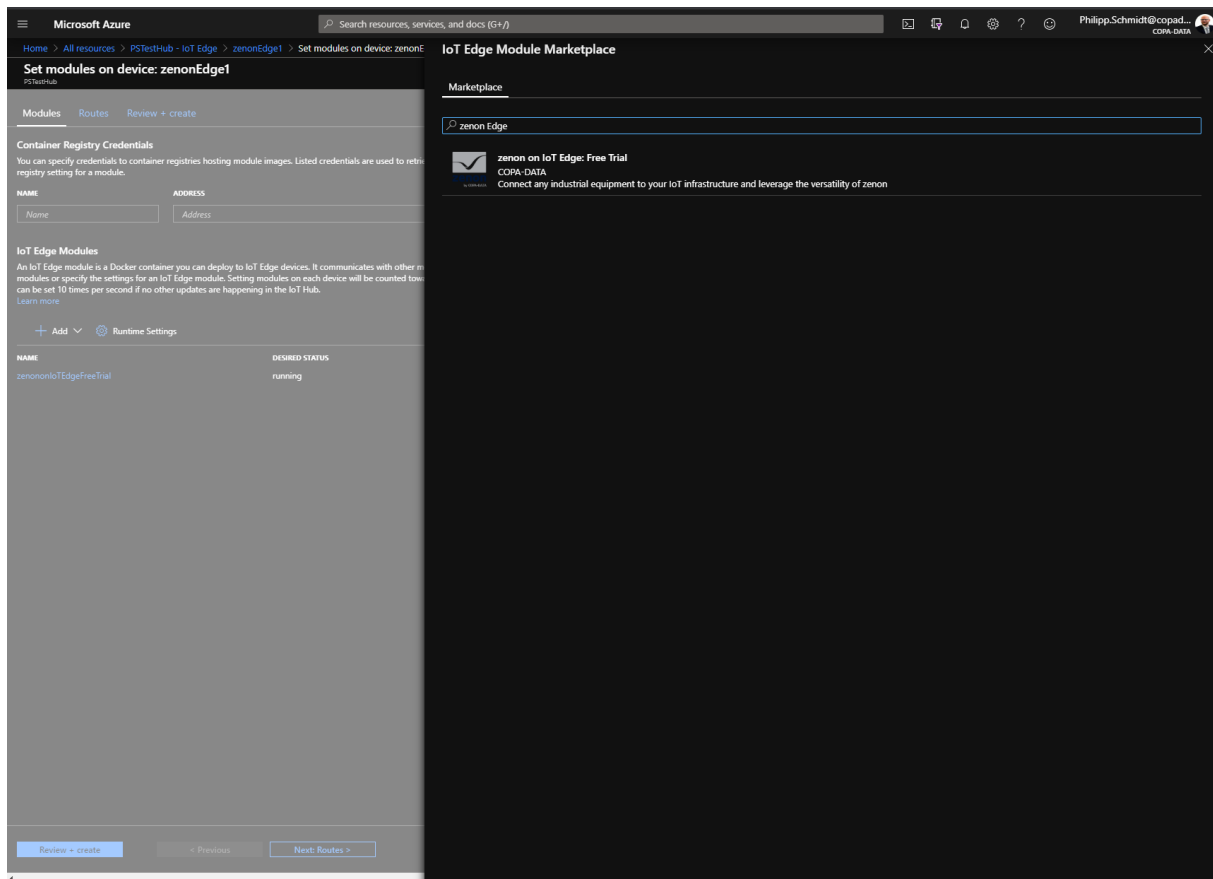
| NAME | VALUE |
|--------------------|-----------------------|
| IS_LOGIC | 0 |
| STORAGE_CONNECTION | DefaultEndpointsProto |
| STORAGE_CONTAINERN | zenon-edge-projects |
| STORAGE_BLOBNAME | IOT_EDGE_DEMO_SIMU |

Container Create Options

☒ Set module twin's desired properties

```

{
  "properties.desired": {}
}
  
```



You can set the following environment variables on the zenon IoT Edge Module:

| Setting | Description | Example |
|--------------------------|---|---|
| IS_LOGIC | Environment variable to switch between zenon logic and zenon supervisor. In this tutorial we only addressed zenon Supervisor... zenon logic will follow soon. | 0 |
| STORAGE_CONNECTIONSTRING | Connection String to the blob storage, where your zenon project files are stored | DefaultEndpointsProtocol=https;AccountName=<YOUR_STORAGE_ACCOUNT_NAME>;AccountKey=<YOUR_ACCOUNT_KEY>;Endpoint |

| | | |
|-----------------------|----------------------------------|--------------------------------|
| | | Suffix= <YOUR_ENDPOINT_SUFFIX> |
| STORAGE_CONTAINERNAME | Name of the storage container | zenon-edge-projects |
| STORAGE_BLOBNAME | Name of the blob / zenon project | IOT_EDGE_DEMO |

After adding your module configuration, you can leave the route settings at their default and submit the deployment to your edge device.

It will take some time for the module to go online, since the docker image and zenon project have to be downloaded.

Please note: Your first deployment of the zenon docker image is likely to take some time (up to 60 minutes). This is because the image is about 4.2 Gb in size and has to be downloaded by the edge runtime. If you plan to have several edge devices it could be beneficial to use a local docker registry. (see <https://docs.docker.com/registry/deploying/> for more information)

6. Have a look at your data

Congratulations, you just put your first IoT Edge device into service! Now it is time to have a look at the results. The zenon demo project, that you deployed, uses a very simple simulation to provide a handful of “real” sensor values. Those values will be sent to your IoT Hub and can be processed as you like.

If you just want to monitor the messages in your IoT Hub to see, if a deployment was successful, you can use Visual Studio Code and its Extension “Azure IoT Hub Toolkit”. In that extension you can provide the connection string to your IoT Hub and then monitor all the different devices, that are connected to it. In our case I get the following output for my device “zenonIoTEdgeDevice”:

```
[IoTHubMonitor] [10:58:13 AM] Message received from [zenonEdge1/zem1]:
{
  "Variable": "SimulVar1",
  "Value": {
    "Value": 1,
    "SourceTimestamp": "2019-10-04T08:58:13.389Z",
    "StatusCode": 4325376
  }
}
[IoTHubMonitor] [10:58:13 AM] Message received from [zenonEdge1/zem1]:
{
  "Variable": "SimulVar2",
  "Value": {
    "Value": 1,
    "SourceTimestamp": "2019-10-04T08:58:13.389Z",
    "StatusCode": 4325376
  }
}
[IoTHubMonitor] [10:58:13 AM] Message received from [zenonEdge1/zem1]:
{
  "Variable": "SimulVar3",
  "Value": {
    "Value": 1,
    "SourceTimestamp": "2019-10-04T08:58:13.389Z",
    "StatusCode": 4325376
  }
}
[IoTHubMonitor] [10:58:13 AM] Message received from [zenonEdge1/zem1]:
{
  "Variable": "SimulVar4",
  "Value": {
    "Value": 1,
    "SourceTimestamp": "2019-10-04T08:58:13.389Z",
    "StatusCode": 4325376
  }
}
[IoTHubMonitor] [10:58:13 AM] Message received from [zenonEdge1/zem1]:
{
  "Variable": "SimulVar5",
  "Value": {
    "Value": 1,
    "SourceTimestamp": "2019-10-04T08:58:13.389Z",
    "StatusCode": 4325376
  }
}
```

```
}  
}
```

7. How to proceed from here?

This tutorial was meant to get you started. We set everything up to get data from a simple zenon demo project in the IoT Edge Module into your IoT Hub.

To make first experiences with your own use case, you can, of course, build your own zenon project and upload it to the blob storage for the IoT Edge module to use. There are only a few things you have to keep in mind for this:

- 1) The free starter edition of the zenon IoT Edge module comes with a limited license. In your project you can only use one external driver and 25 Tags. Otherwise no additional zenon modules are licensed. So keep it simple!
- 2) In the Git repository you can find the `AddInProjectServiceIoTEdge.scadaAddIn`. It is responsible for publishing all the variables in your project to the internal endpoint of the IoT Edge runtime. Please include that AddIn in your own project to forward the data.

Have fun with building your own project!

When you start hitting the boundary of one driver/ 25 Tags or need more modules and want to start with "real projects", please contact us: sales@copadata.com



© Ing. Punzenberger COPA-DATA GmbH.

All rights reserved. This document is protected by copyright and may not be reproduced, utilized or photocopied in any form or by any means without permission in writing from Ing. Punzenberger COPA-DATA GmbH. The technical data contained herein have been provided solely for informational purposes and are not legally binding. The COPA-DATA logo, zenon, zenon Analyzer, zenon Supervisor, zenon Operator, zenon Logic and straton are registered trademarks of Ing. Punzenberger COPA-DATA GmbH. All other brands and product names may be the trademarks or registered trademarks of their representative owners. Subject to change, technical or otherwise.