

 master 

[IoT\\_Academy / Month\\_3 / Day\\_1 /](#)

...



[mpram](#) Windows IoT HOL ...

9 minutes ago  [History](#)

..



[media](#)

4 hours ago



[pdf files](#)

12 days ago



[README.MD](#)

9 minutes ago

 [README.MD](#)



## Azure IoT Academy: Windows IoT Lab

In this lab we will be setting up Azure IoT Edge for Linux on Windows (EFLOW) with a Live Video Analytics module to showcase how Windows IoT OS enables various retail and safety scenarios.

### Content:

- [Exercise 1: Set up Environment](#)
  - [Task 1: Virtual Network](#)
  - [Task 2: Virtual Machine](#)
  - [Task 3: Connect to Virtual Machine](#)
- [Exercise 2: Set up Azure Edge For Linux on Windows](#)
  - [Task 1: Enable Hyper-V](#)
  - [Task 2: Set up Azure IoT Hub](#)
  - [Task 3: Register an IoT Hub Device](#)
  - [Task 4: Download Windows Admin Center](#)
  - [Task 5: Create a new deployment](#)

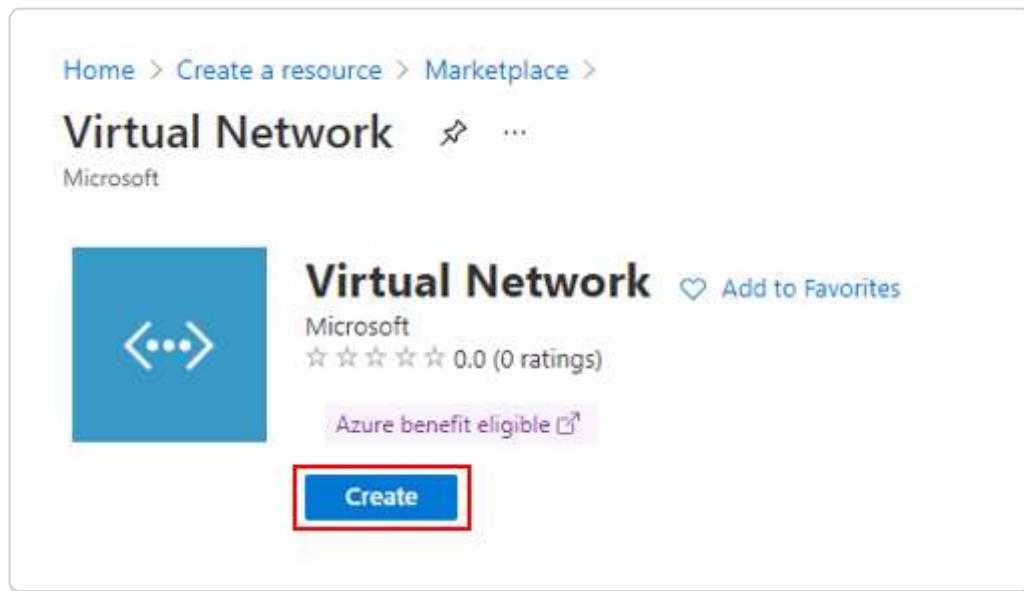
- Task 6: Verify successful configuration
- Exercise 3: Set up Live Video Analytics
  - Task 1: Download Tools and Resources
  - Task 2: Create Azure Resources
  - Task 3: Configure the Azure IoT Tools extension
  - Task 4: Deploy Modules on Windows Host
  - Task 5: Provision Azure IoT Edge for Linux Configuration
  - Task 6: Test Video Stream
  - Task 7: Enable Live Video Analytics: Inferencing
  - Task 8: Connect Windows Video with Linux Inferencing
- Exercise 4: Clean up
  - Task 1: Delete resources

## Exercise 1: Set up Environment

During this exercise you will be setting up your Windows 10 IoT Enterprise environment for this lab in an Azure Virtual Machine.

### Task 1: Virtual Network

1. In your browser, navigate to the [Azure portal](#), select **+Create a resource** in the navigation pane, enter **virtual Network** into the **Search the Marketplace** box.
2. Select **Virtual Network** from the results, and then select **Create**.



3. In **Create virtual network**, enter or select this information in the **Basics** tab:

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select <b>Create new</b> . Enter a name for the Resource Group. Select <b>OK</b> .
Instance details	
Name	Enter <b>myVNetwork</b> .
Region	Select <b>(US) East US</b> .

Home > Create a resource > Marketplace > Virtual Network >

## Create virtual network

Basics IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

**Project details**

Subscription \* ⓘ

Resource group \* ⓘ  [Create new](#)

**Instance details**

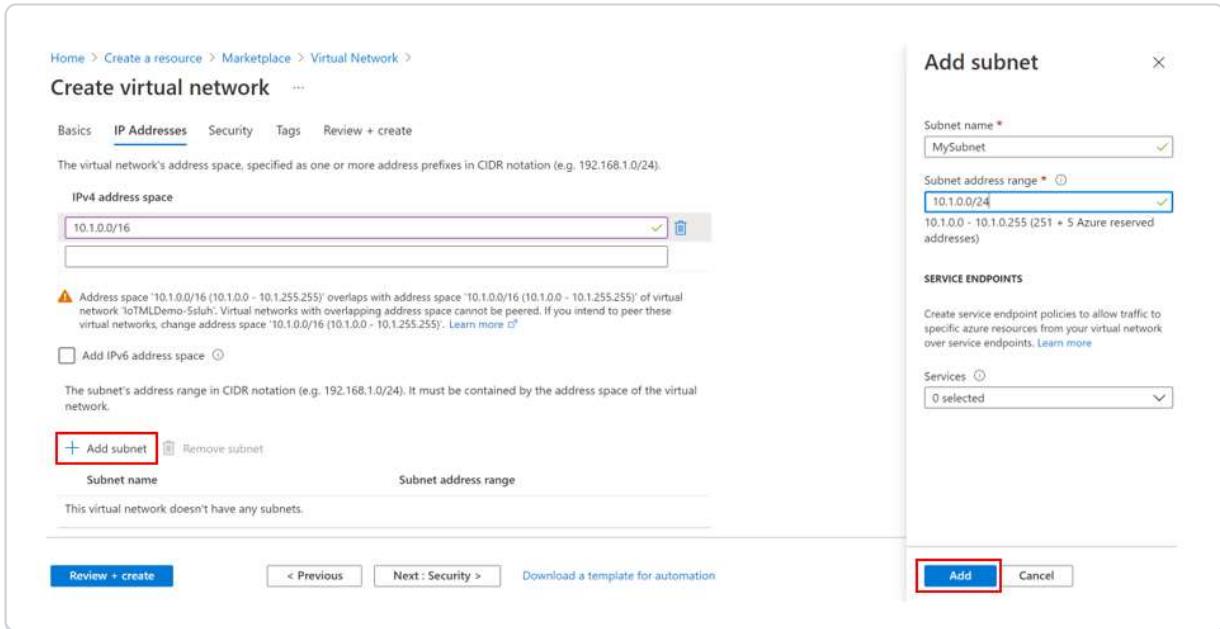
Name \*  ✓

Region \*

[Review + create](#) [< Previous](#) [Next : IP Addresses >](#) [Download a template for automation](#)

4. Select the IP Addresses tab, or select the **Next : IP Addresses >** button at the bottom of the page.
5. In **IPv4 address space**, select the existing address space and change it to **10.1.0.0/16**.
6. Select **+ Add subnet**, then enter **MySubnet** for **Subnet name** and **10.1.0.0/24** for **Subnet address range**.

## 7. Select Add.



The screenshot shows the 'Create virtual network' wizard in the Azure portal. The 'IP Addresses' tab is selected. A modal window titled 'Add subnet' is open on the right. In the modal, the 'Subnet name' field is set to 'MySubnet'. The 'Subnet address range' field is set to '10.1.0.0/24'. The 'Add' button in the modal is highlighted with a red box.

[!NOTE] You can ignore the warning as we are not intending to peer virtual networks.

## 8. Select the **Security** tab, or select the **Next: Security** button at the bottom of the page.

## 9. Under **BastionHost**, select **Enable**. Enter this information:

Setting	Value
Bastion name	Enter <b>myBastionHost</b>
AzureBastionSubnet address space	Enter <b>10.1.1.0/24</b>
Public IP Address	Select <b>Create new</b> . For <b>Name</b> , enter <b>myBastionIP</b> . Select <b>OK</b> .

## 10. Select the **Review + create** tab or select the **Review + create** button.

## Create virtual network

Basics IP Addresses **Security** Tags Review + create

BastionHost  ⓘ  Disable  Enable

Bastion name \* myBastionHost  ✓

AzureBastionSubnet address space \* 10.1.1.0/24  ✓  
10.1.1.0 ~ 10.1.1.255 (256 addresses)

Public IP address \* (New) myBastionIP  ↴  
[Create new](#)

DDoS Protection Standard  ⓘ  Disable  Enable

Firewall  ⓘ  Disable  Enable

---

**Review + create**  < Previous  Next : Tags > [Download a template for automation](#)

11. Select **Create**.

## Create virtual network

 Validation passed

Basics IP Addresses Security Tags Review + create

### Basics

Subscription [REDACTED]  
Resource group (new) IoT-Academy-Demo  
Name myVNetwork  
Region East US

### IP addresses

Address space 10.1.0.0/16  
Subnet mySubnet (10.1.0.0/24), AzureBastionSubnet (10.1.1.0/24)

### Tags

None

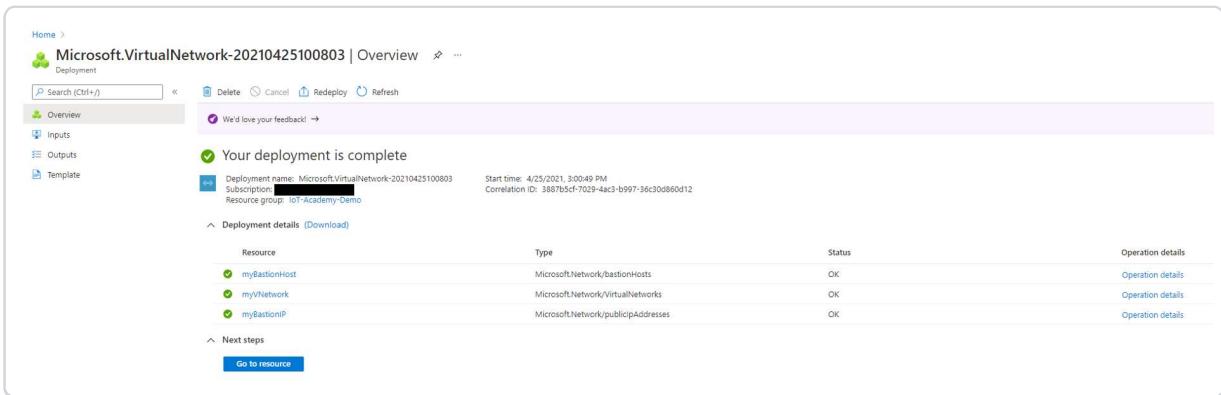
**Create**

< Previous

Next >

Download a template for automation

12. It will take a few minutes to deploy. At the end you should see the your resources deployed.



Microsoft.VirtualNetwork-20210425100803 | Overview

Deployment

Search (Ctrl+ /)

Deployment

Deployment name: Microsoft.VirtualNetwork-20210425100803

Start time: 4/25/2021, 3:00:49 PM

Correlation ID: 3887b5cf-7029-4ac3-b997-36c30d860d12

Subscription [REDACTED]

Resource group: (new) IoT-Academy-Demo

Deployment details (Download)

Resource	Type	Status	Operation details
myBastionHost	Microsoft.Network/bastionHosts	OK	Operation details
myVNetwork	Microsoft.Network/virtualNetworks	OK	Operation details
myBastionIP	Microsoft.Network/publicIPAddresses	OK	Operation details

Next steps

Go to resource

## Task 2: Virtual Machine

1. On the upper-left side of the portal, select: **Create a resource > Compute > Virtual machine >> Create**

## Create a resource ...

Get started

Search services and marketplace

Getting Started? Try our Quickstart center

Recently created

Featured See all

## Categories

AI + Machine Learning



Virtual machine

Create Learn more

Analytics



Virtual machine scale set

Create | Learn more

Blockchain



Kubernetes Service

Create | Docs | MS Learn

Compute

Containers

2. In **Create a virtual machine**, type or select the values in the **Basics** tab:

Setting	Value
<b>Project Details</b>	
Subscription	Select your Azure subscription
Resource Group	Select Your Resource Group
<b>Instance details</b>	
Virtual machine name	Enter <b>myVM1</b>
Region	Select <b>(US) East US</b>
Availability Options	Select <b>No infrastructure redundancy required</b>
Image	Select <b>Windows 10 Pro, vNext - Gen1</b>
Azure Spot instance	Select <b>No</b>
Size	<b>Standard_D4s_v3 - 4 vcpus, 16 GiB memory</b>
<b>Administrator Account</b>	<b>Use the following Credentials</b>
Username	AIOTA
Password	iotacademyDay3!
Confirm password	Reenter password
<b>Inbound port rules</b>	
Public inbound ports	Select <b>None.</b>

## Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

[REDACTED]

Resource group \* ⓘ

IoT-Academy-Demo

[Create new](#)

### Instance details

Virtual machine name \* ⓘ

myVM1

Region \* ⓘ

(US) East US

Availability options ⓘ

Availability zone

Availability zone \* ⓘ

1

Image \* ⓘ

Windows 10 Pro, vNext - Gen1

[See all images](#)

Azure Spot instance ⓘ

Size \* ⓘ

Standard\_D4s\_v3 - 4 vcpus, 16 GiB memory (\$274.48/month)

[See all sizes](#)

### Administrator account

Username \* ⓘ

AIOTA

Password \* ⓘ

\*\*\*\*\*

Confirm password \* ⓘ

\*\*\*\*\*

### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ

None

Allow selected ports

Select inbound ports

Select one or more ports

 All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

[Review + create](#)

[< Previous](#)

[Next : Disks >](#)

3. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

4. In the Networking tab, select or enter:

Setting	Value
<b>Network interface</b>	
Virtual network	Select <b>myVNetwork</b> .
Subnet	Select <b>mySubnet</b>
Public IP	Select <b>None</b>
NIC network security group	Select <b>Basic</b>
Public inbound ports network	Select <b>None</b> .

## Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

### Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network \* ⓘ

Subnet \* ⓘ

Public IP ⓘ

NIC network security group ⓘ

None  
 Basic  
 Advanced

**i** The selected subnet 'mySubnet (10.1.0.0/24)' is already associated to a network security group 'rg-cleanupservice-nsg2'. We recommend managing connectivity to this virtual machine via the existing network security group instead of creating a new one here.

Public inbound ports \* ⓘ

None  
 Allow selected ports

Select inbound ports

**i** All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Accelerated networking ⓘ

The selected image does not support accelerated networking.

### Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Place this virtual machine behind an existing load balancing solution?

[Review + create](#)

[< Previous](#)

[Next : Management >](#)

5. Select the **Review + create** tab, or select the blue **Review + create** button at the bottom of the page.

6. Review the settings, and then select **Create**.

Home >

## Create a virtual machine

Validation passed

**Basics**

Subscription	[REDACTED]
Resource group	IoT-Academy-Demo
Virtual machine name	myVM1
Region	East US
Availability options	Availability zone
Availability zone	1
Image	Windows 10 Pro, vNext - Gen1
Size	Standard D4s v3 (4 vcpus, 16 GiB memory)
Username	AIOTA
Public inbound ports	None
Already have a Windows license?	No
Azure Spot	No

**Disks**

OS disk type	Premium SSD LRS
Use managed disks	Yes
Ephemeral OS disk	No

**Networking**

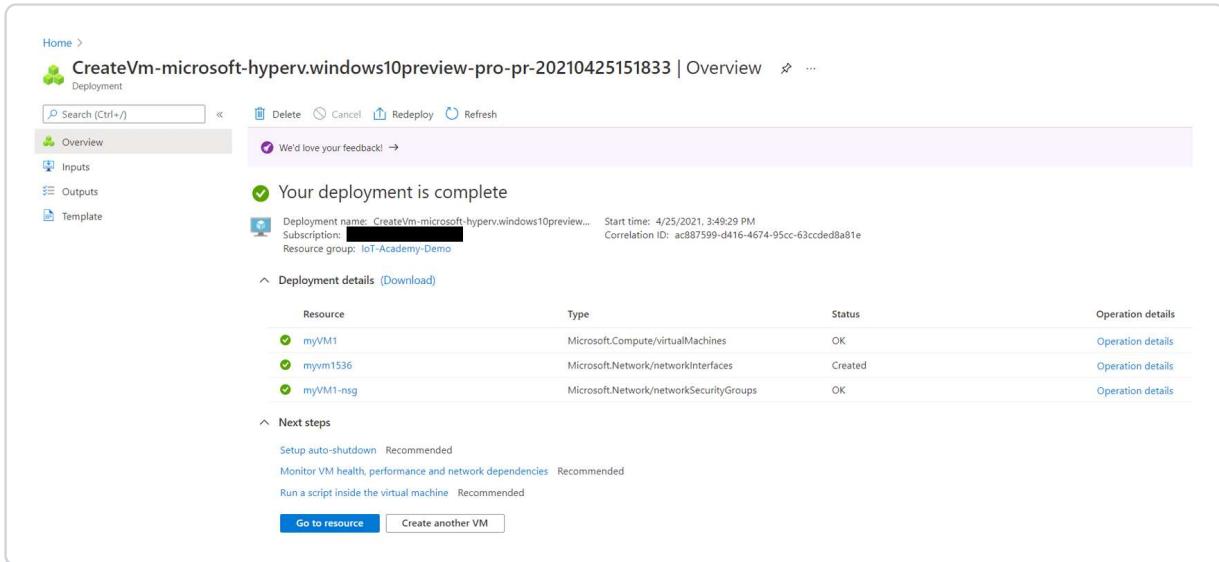
Virtual network	myVNetwork
Subnet	mySubnet (10.1.0.0/24)
Public IP	None
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No

**Management**

Azure Security Center	None
Enable detailed monitoring	Off
Boot diagnostics	On
Enable OS guest diagnostics	Off

**Create** < Previous Next > [Download a template for automation](#)

7. It will take a few minutes to deploy. At the end you should see the your resources deployed.



Home > CreateVm-microsoft-hyperv.windows10preview-pro-pr-20210425151833 | Overview

Deployment

Search (Ctrl+ /)

Delete Cancel Redeploy Refresh

We'd love your feedback!

Your deployment is complete

Deployment name: CreateVm-microsoft-hyperv.windows10preview... Start time: 4/25/2021, 3:49:29 PM

Subscription: [REDACTED] Correlation ID: ac887599-d416-4674-95cc-63ccded8a81e

Resource group: IoT-Academy-Demo

Deployment details (Download)

Resource	Type	Status	Operation details
myVM1	Microsoft.Compute/virtualMachines	OK	Operation details
myvm1536	Microsoft.Network/networkInterfaces	Created	Operation details
myVM1-nsg	Microsoft.Network/networkSecurityGroups	OK	Operation details

Next steps

Setup auto-shutdown Recommended

Monitor VM health, performance and network dependencies Recommended

Run a script inside the virtual machine Recommended

Go to resource Create another VM

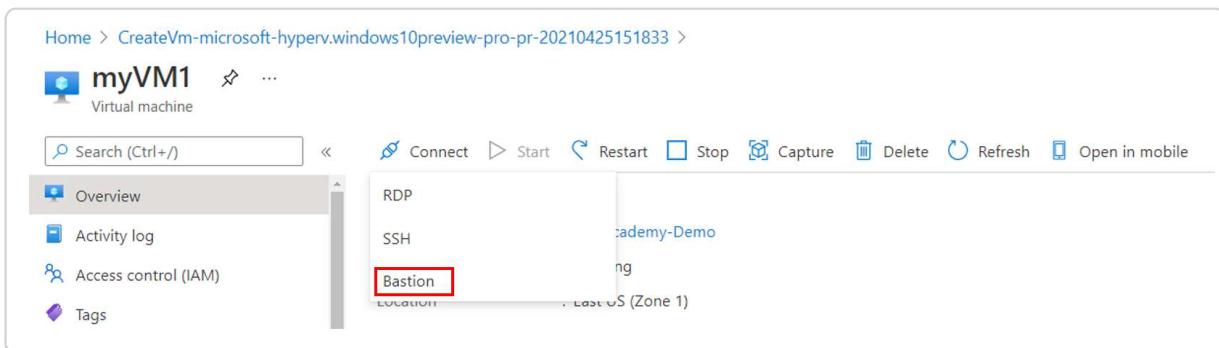
## Task 3: Connect to Virtual Machine

1. Navigate to the Azure Portal Home and select your newly created virtual machine.

2. Make sure you that you start your Virtual Machine and that the status is **Running**.

[!TIP] You will not be able to start the Bastion connection if the VM has not been started and is running. So give it a minute or two to finish updating and wait for the status to say "Running".

3. In the VM menu bar, select **Connect**, then select **Bastion**.



Home > CreateVm-microsoft-hyperv.windows10preview-pro-pr-20210425151833 >

myVM1

Virtual machine

Search (Ctrl+ /)

Connect Start Restart Stop Capture Delete Refresh Open in mobile

Overview

Activity log

Access control (IAM)

Tags

RDP

SSH

Bastion

academy-Demo

ng

LastS (Zone 1)

4. In the **Connect** page, select the blue **Use Bastion** button.

RDP    SSH    **BASTION**

**!** Bastion is an Azure service that allows fast, secure connections to any VM within a VNet. [Learn more](#)

**Use Bastion**

5. In the **Bastion** page, enter the username and password for the virtual machine.

Field	Enter
Username	AIOTA
Password	<i>iotacademyDay3!</i>

6. Select **Connect**.

Using Bastion: **myBastionHost**, Provisioning State: **Succeeded**

Please enter username and password to your virtual machine to connect using Bastion.

Open in new window

Username \* ⓘ

AIOTA

Password \* ⓘ

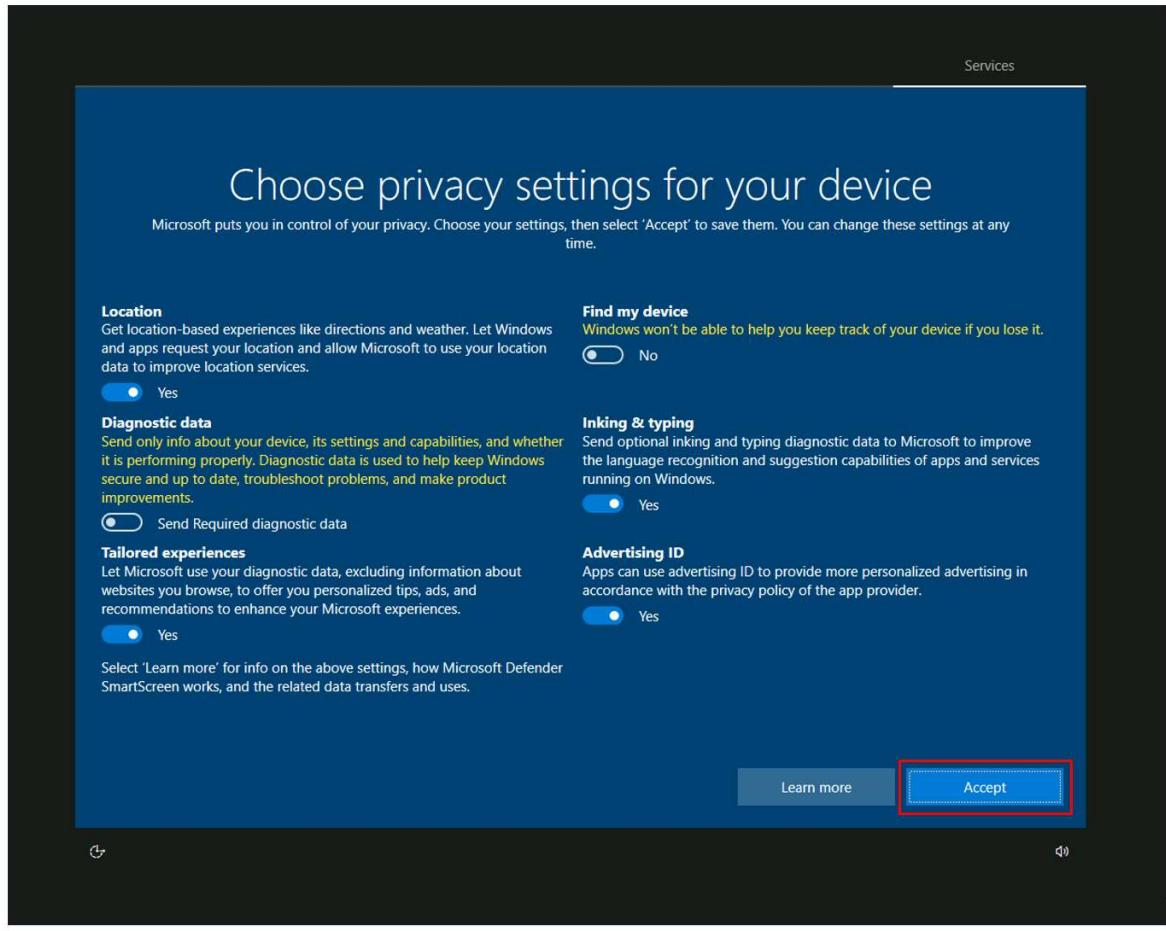
\*\*\*\*\*

Show

**Connect**

7. A new tab should open, and you should be connected to your virtual machine.

8. Accept the default settings.



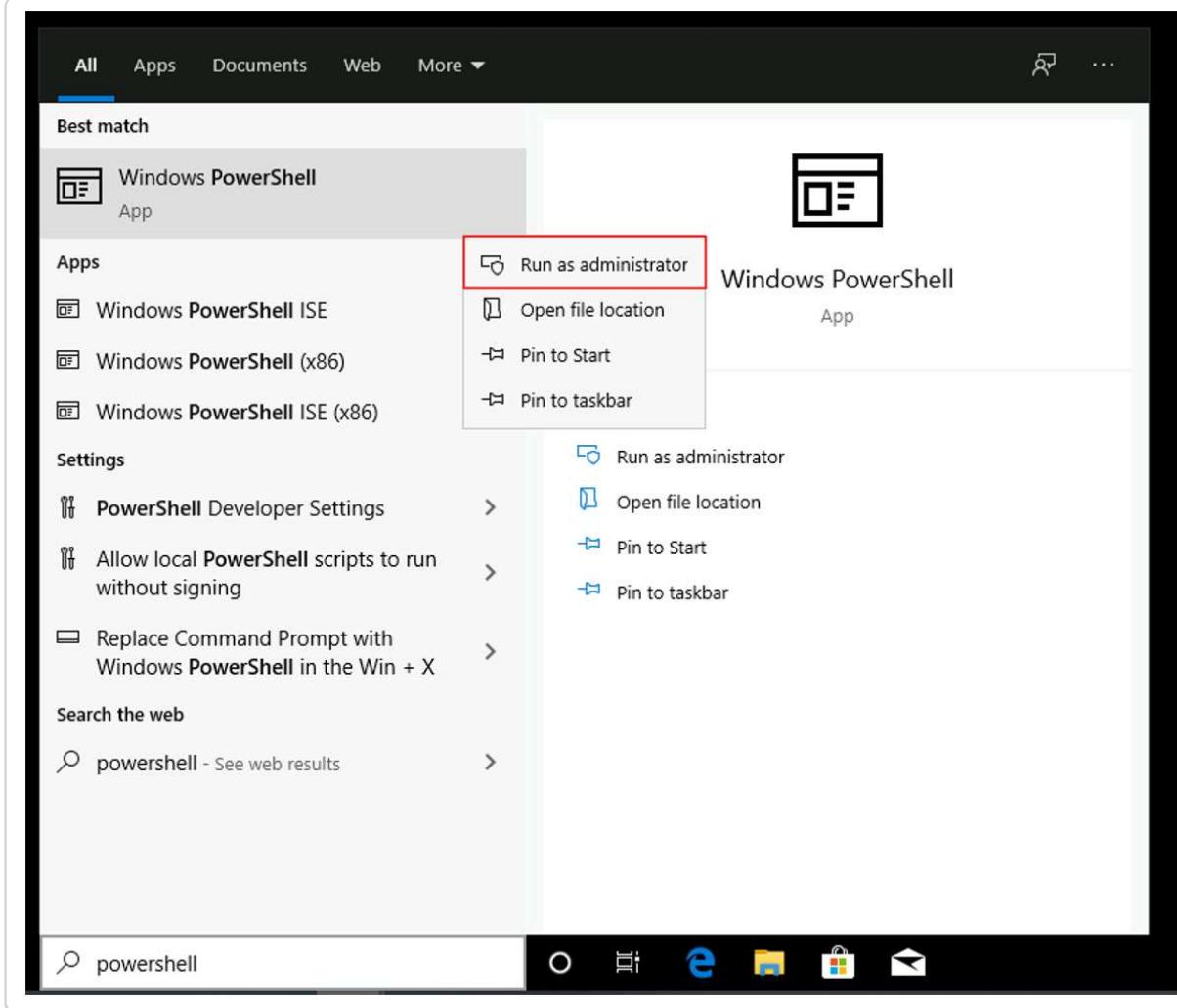
9. We will be using this virtual machine for the remaining two exercises.

## Exercise 2: Set up Azure Edge For Linux on Windows

### Task 1: Enable Hyper-V

We are going to enable Hyper-V via PowerShell in the newly created VM.

1. Search for **PowerShell** and right click to select **Run as Administrator**.

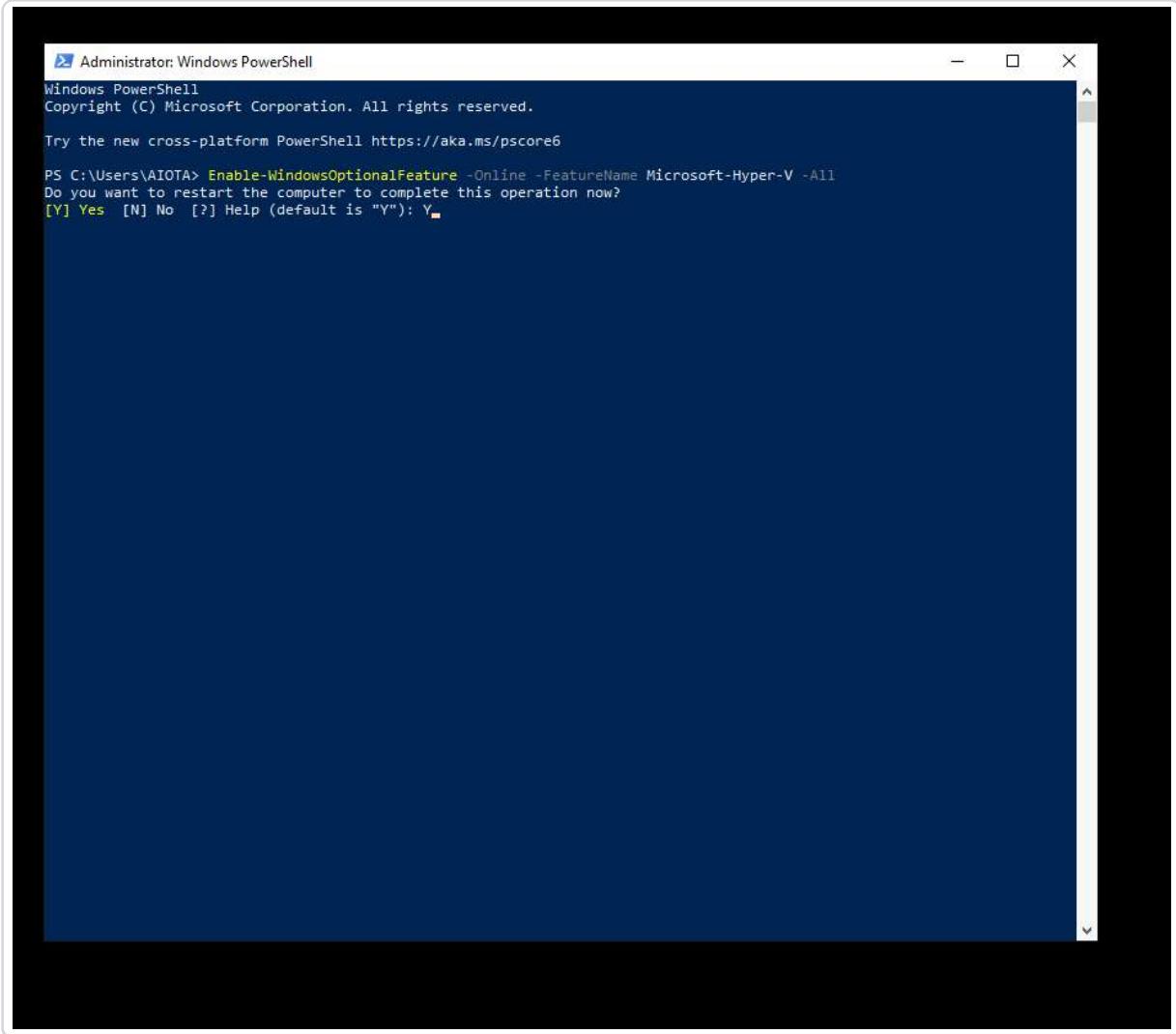


2. Run the following command:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

If the command couldn't be found, make sure you're running PowerShell as an **Administrator**.

3. When the installation has completed, reboot the VM by typing in Y.



4. Reconnect to the VM.

[!NOTE] If you are not prompted to restart the VM within PowerShell. Please close the Bastion Host tab, and return to the Azure Portal, and select your VM. At this point you can either "restart your VM" and reconnect via Bastion. OR you can *STOP* the VM and *Start* the VM again.

## Task 2: Set up Azure IoT Hub

*These steps can be done outside of the VM or you can open up a browser window in the Azure Window and do everything from within the VM.*

1. Navigate to the Azure Portal, select the **+ Create a resource** button, and then enter *IoT Hub* in the **Search the Marketplace** field.

2. Select IoT Hub from the search results, and then select **Create**.

The screenshot shows the Azure Marketplace page for the IoT Hub. At the top, the breadcrumb navigation is "Home > Create a resource > Marketplace > IoT Hub". The page title is "IoT Hub" with a Microsoft logo and a "..." button. Below the title is a blue icon representing a network or hub. To the right of the icon, the text "IoT Hub" is displayed in large bold letters, followed by a "Microsoft" publisher logo, a "4.2 (348 ratings)" rating, and a "Azure benefit eligible" badge with a checkmark. A large blue "Create" button is centered at the bottom of the card.

3. On the **Basics** tab, complete the fields as follows:

- **Subscription:** Select the subscription to use for your hub.
- **Resource Group:** Select the resource group you've created for exercise 1 above.
- **Region:** Select the region in which you want your hub to be located. Select the location closest to you.
- **IoT Hub Name:** Enter a name for your hub. This name must be globally unique.

[!IMPORTANT]

Because the IoT hub will be publicly discoverable as a DNS endpoint, be sure to avoid entering any sensitive or personally identifiable information when you name it.

## IoT hub

...

Microsoft

Basics Networking Management Tags Review + create

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#)

### Project details

Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription \* ⓘ

IOT-Athens-SLFHST



Resource group \* ⓘ

IoT-Academy-Demo



[Create new](#)

Region \* ⓘ

East US



IoT hub name \* ⓘ

AIOTA-Demo-Hub



[Review + create](#)

< Previous

Next: Networking >

4. Select **Next: Networking** to continue creating your hub.

Choose the endpoints that can connect to your IoT Hub. You can select the default setting **Public endpoint (all networks)**.

# IoT hub

...

Microsoft

Basics

**Networking**

Management

Tags

Review + create

## Network connectivity

Connect to your IoT Hub using public or private endpoints.

Connectivity method \* i

Public endpoint (all networks)

Private endpoint

i All networks will have access to this IoT hub.  
[Learn more about connectivity methods.](#)

**Review + create**

< Previous: Basics

Next: Management >

5. Select **Next: Management** to continue creating your hub.

You can accept the default settings here.

6. Select **Next: Tags** to continue to the next screen.

7. Select **Next: Review + create** to review your choices. You see something similar to this screen, but with the values you selected when creating the hub.

8. Select **Create** to create your new hub.

## IoT hub

Microsoft

Validation passed.

Basics Networking Management Tags **Review + create**

### Basics

Subscription	[REDACTED]
Resource group	IoT-Academy-Demo
Region	East US
IoT hub name	AIOTA-Demo-Hub

### Networking

Connectivity method	Public endpoint (all networks)
Private endpoint connections	None

### Management

Pricing and scale tier	S1
Number of S1 IoT hub units	1

**Create**

< Previous: Tags

Next >

Automation options

9. It will take a few minutes for your Hub to deploy.

AIOTA-Demo-Hub-425172834 | Overview

Deployment

Search (Ctrl+ /) < Delete Cancel Redeploy Refresh

Overview Inputs Outputs Template

We'd love your feedback! →

✓ Your deployment is complete

Deployment name: AIOTA-Demo-Hub-425172834 Start time: 4/25/2021, 5:28:38 PM  
Subscription: [REDACTED] Correlation ID: 067dd5b1-4f37-47d9-9848-97efaf95a25e  
Resource group: IoT-Academy-Demo

Deployment details (Download)

Resource	Type	Status	Operation details
AIOTA-Demo-Hub	Microsoft.Security/IoTSecurity	OK	<a href="#">Operation details</a>
AIOTA-Demo-Hub	Microsoft.Devices/IotHub	OK	<a href="#">Operation details</a>

Next steps

[Add and configure IoT Devices](#) Recommended  
[Configure routing rules for device messaging](#) Recommended

Go to resource

## Task 3: Register an IoT Hub Device

1. Navigate to your IoT Hub and select **IoT Edge** from the menu.

Home >

## AIOTA-Demo-Hub IoT Hub

Search (Ctrl+ /) Move Delete Refresh

**Pricing and scale**

- Networking
- Certificates
- Built-in endpoints
- Failover
- Properties
- Locks

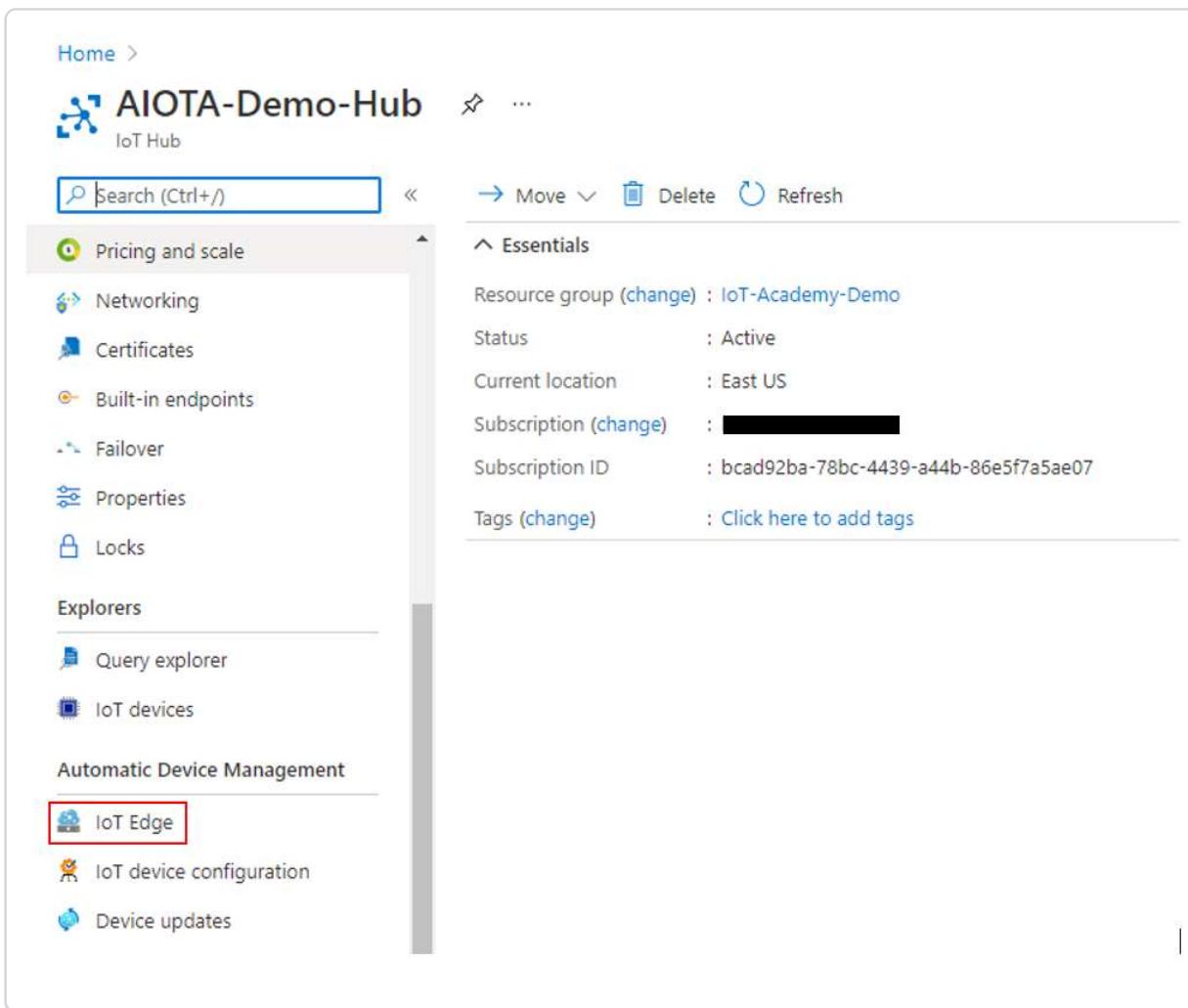
**Explorers**

- Query explorer
- IoT devices

**Automatic Device Management**

- IoT Edge**
- IoT device configuration
- Device updates

Resource group (change) : IoT-Academy-Demo  
Status : Active  
Current location : East US  
Subscription (change) : [REDACTED]  
Subscription ID : bcad92ba-78bc-4439-a44b-86e5f7a5ae07  
Tags (change) : Click here to add tags



## 2. Select Add an IoT Edge device.

Home > AIOTA-Demo-Hub

## AIOTA-Demo-Hub | IoT Edge

Search (Ctrl+ /) + Add an IoT Edge device Create Deployment Create Layered Deployment Refresh Delete

Deploy Azure services and solution-specific code to on-premises devices. Use IoT Edge devices to perform compute and analytics tasks on data before it's sent to the cloud.

**IoT Edge devices** **IoT Edge deployments**

**IoT Edge devices**

Field Operator Value

+ X =

+ Add new clause

Query devices

Device ID Runtime Response IoT Edge Module Count

No results

**Explorers**

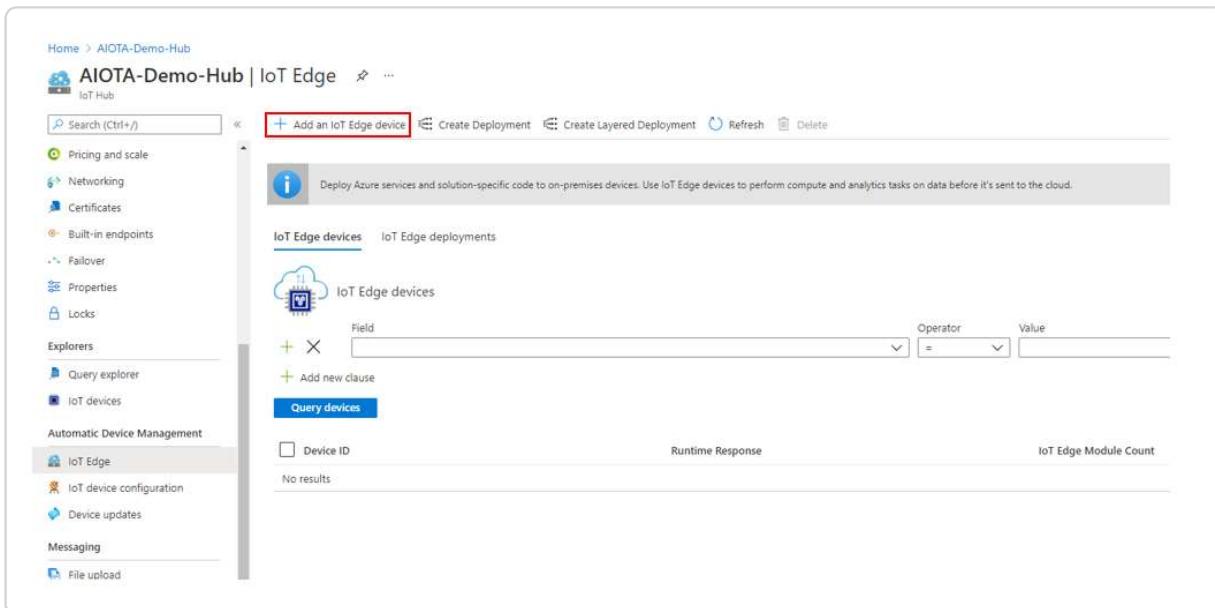
- Query explorer
- IoT devices

**Automatic Device Management**

- IoT Edge**
- IoT device configuration
- Device updates

**Messaging**

File upload



## 3. Create a Device - Select or Enter:

Setting	Value
---------	-------

Setting	Value
Device ID	Provide a descriptive Device ID
Authentication type	<b>Symmetric Key</b>
Auto-generate keys	Default - Keep Box Checked
Connect this device to an IoT Hub	Select <b>Enable</b>

4. Select **Save**.

## Create a device



Find Certified for Azure IoT devices in the Device Catalog



Device ID \* ⓘ

AIOTA-Device



Authentication type ⓘ

Symmetric key X.509 Self-Signed

Primary key ⓘ

Enter your primary key

Secondary key ⓘ

Enter your secondary key

Auto-generate keys ⓘ



Connect this device to an IoT hub ⓘ

Enable

Disable

Parent device ⓘ

**No parent device**

[Set a parent device](#)

Child devices ⓘ

0

[Choose child devices](#)

[Save](#)

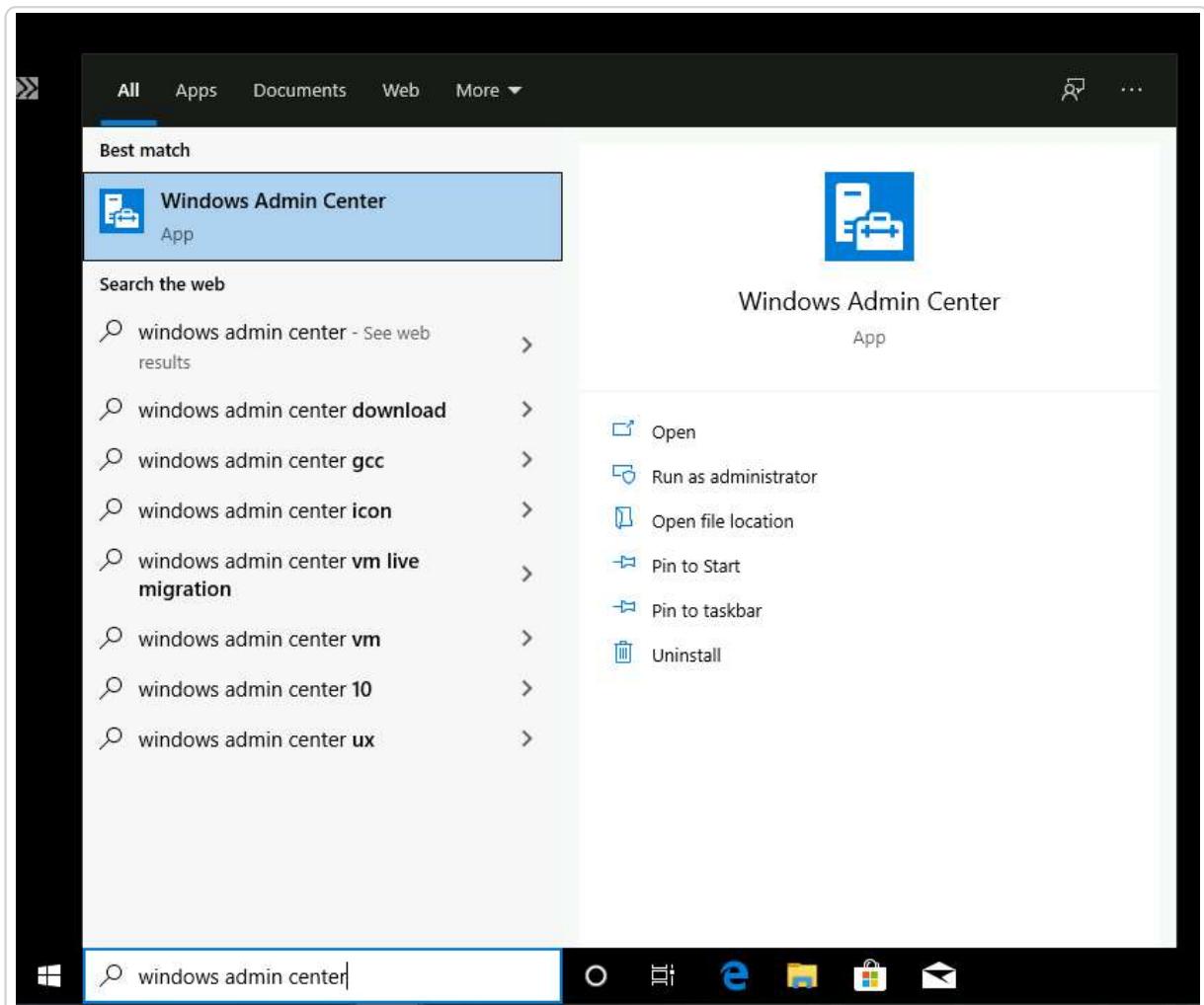
## Task 4: Download Windows Admin Center

Back to the Windows VM, we need to download Windows Admin Center.

1. To download Windows Admin Center installer, type the following address in the browser: aka.ms/wacdownload
2. A prompt at the bottom of the screen should appear. Select Run



3. Run the downloaded installer and follow the install wizard prompts to install Windows Admin Center.
4. Once installed, use a supported browser to open Windows Admin Center. Supported browsers include Microsoft Edge (Windows 10, version 1709 or later), Google Chrome, and Microsoft Edge Insider. Search **Windows Admin Center**



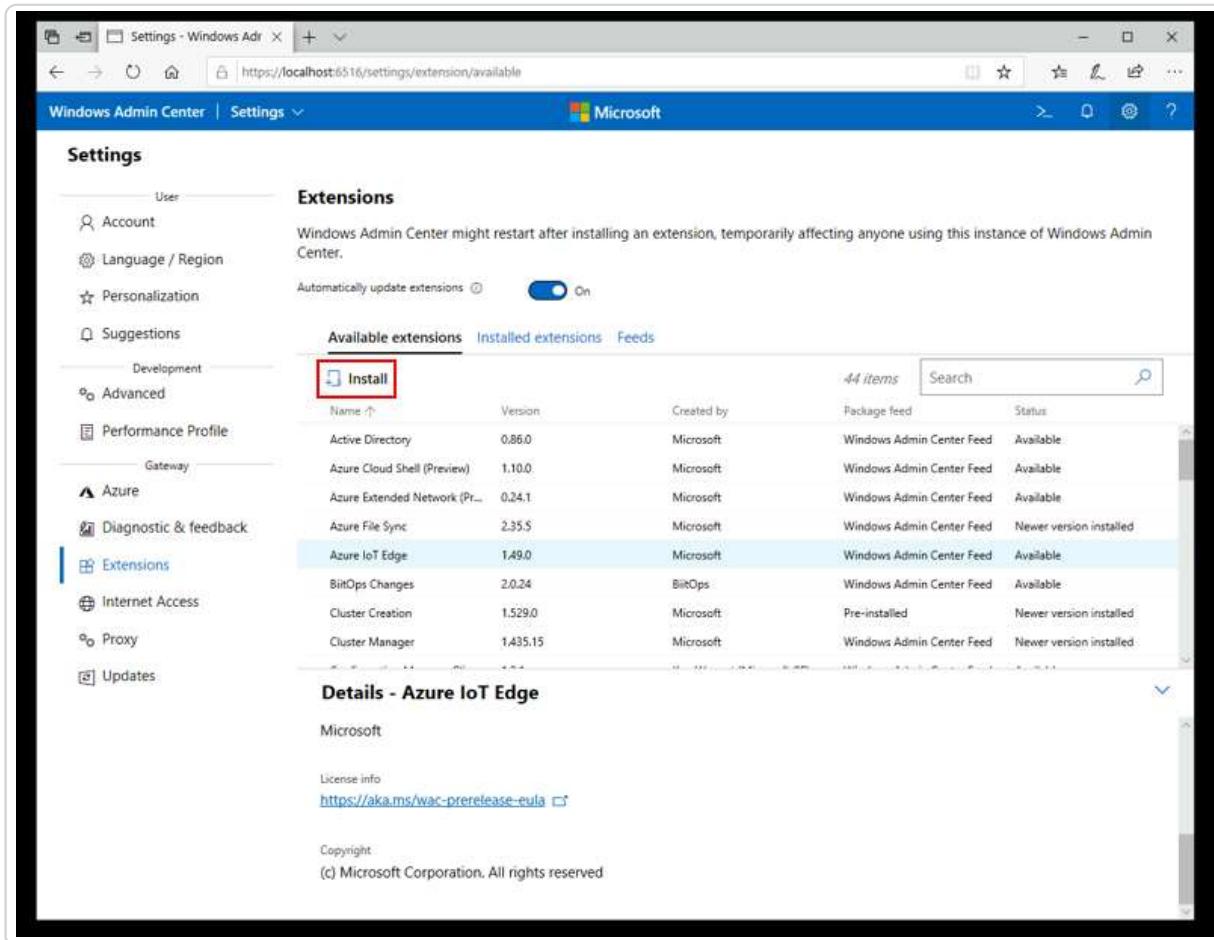
5. On the first use of Windows Admin Center, you will be prompted to select a certificate to use. Select **Windows Admin Center Client** as your certificate.

6. It is time to install the Azure IoT Edge extension. Select the gear icon in the top right of the Windows Admin Center dashboard.

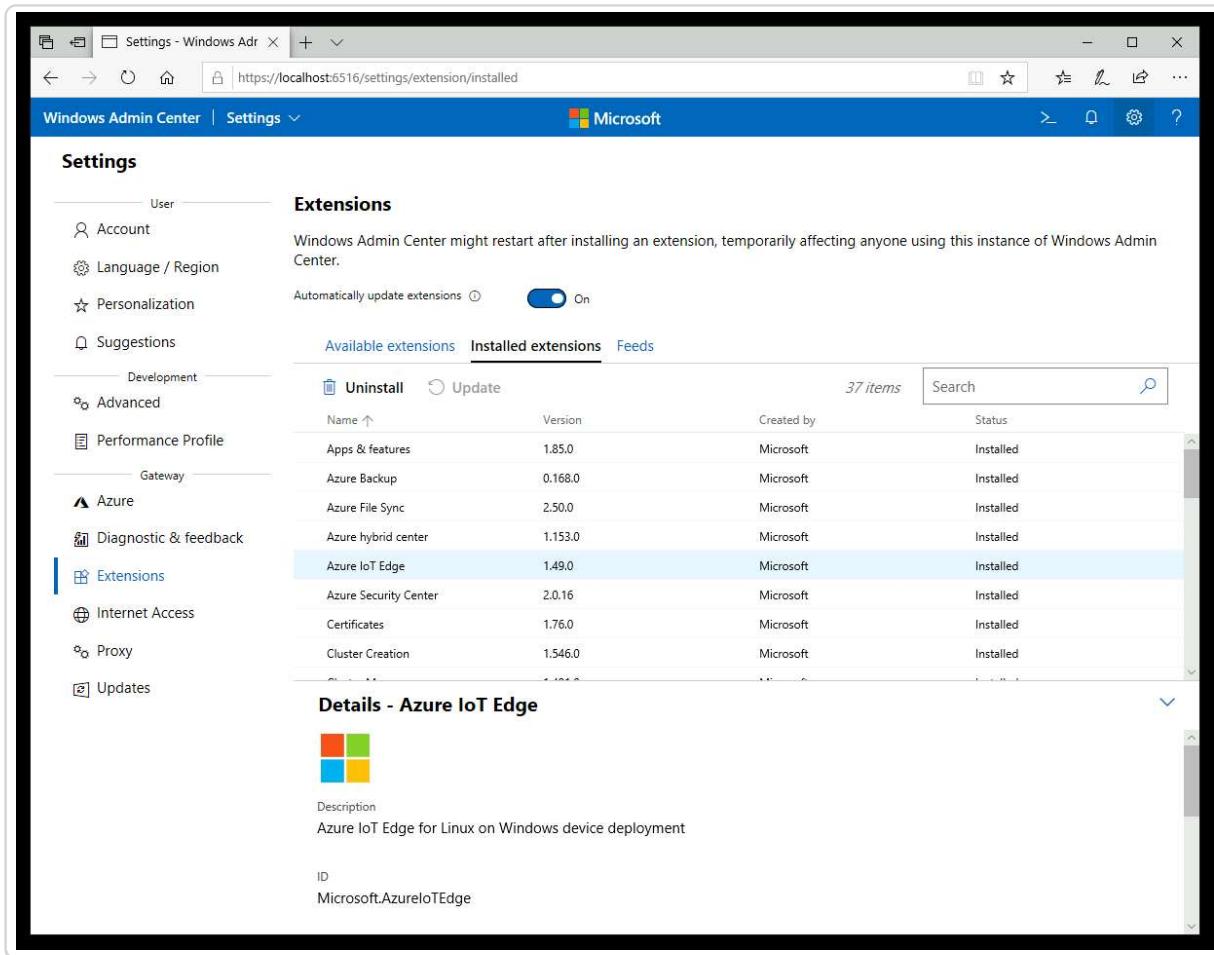


7. On the **Settings** menu, under **Gateway**, select **Extensions**.

8. On the **Available extensions** tab, find **Azure IoT Edge** in the list of extensions. Choose it, and select the **Install** prompt above the list of extensions.



9. After the installation completes, you should see Azure IoT Edge in the list of installed extensions on the **Installed extensions** tab.

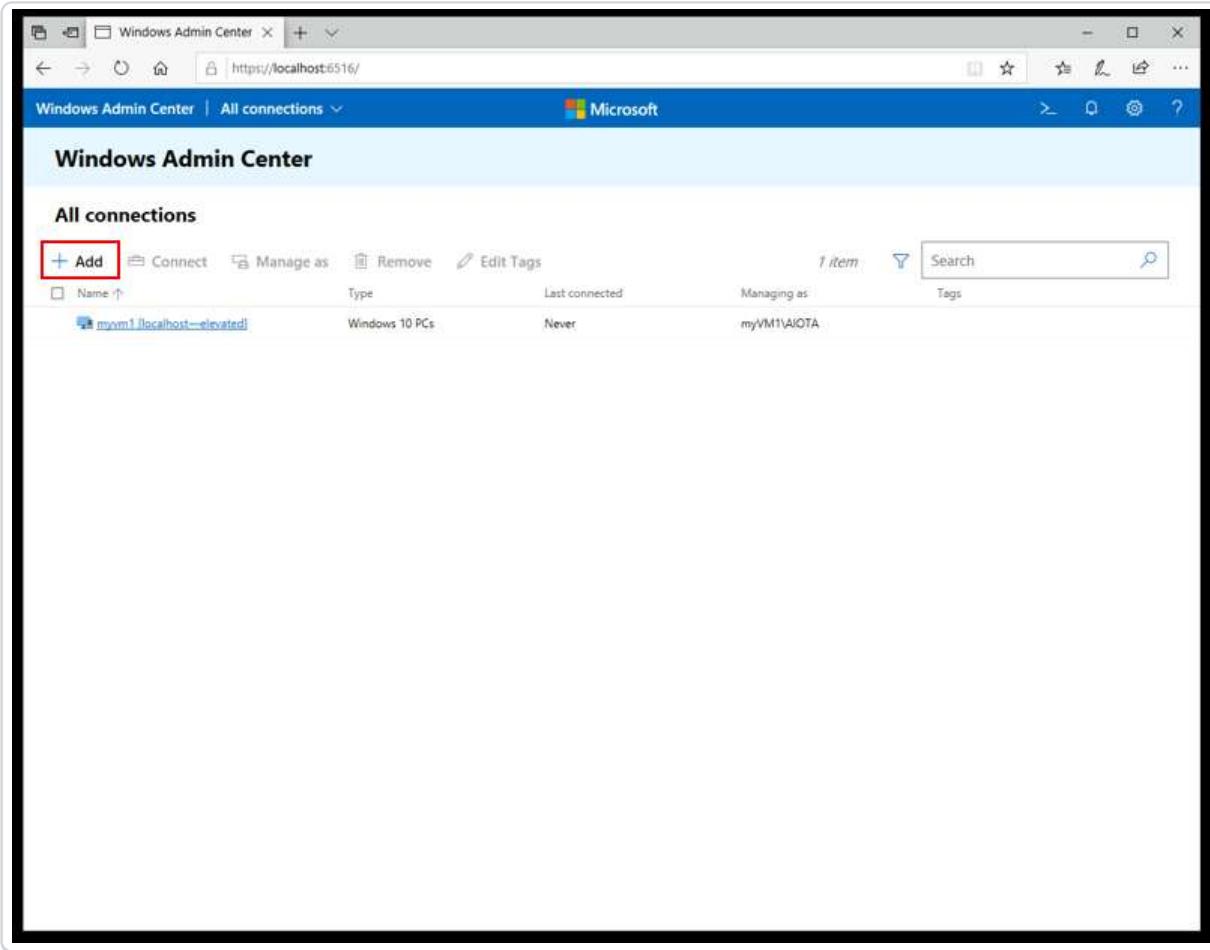


## Task 5: Create a new deployment

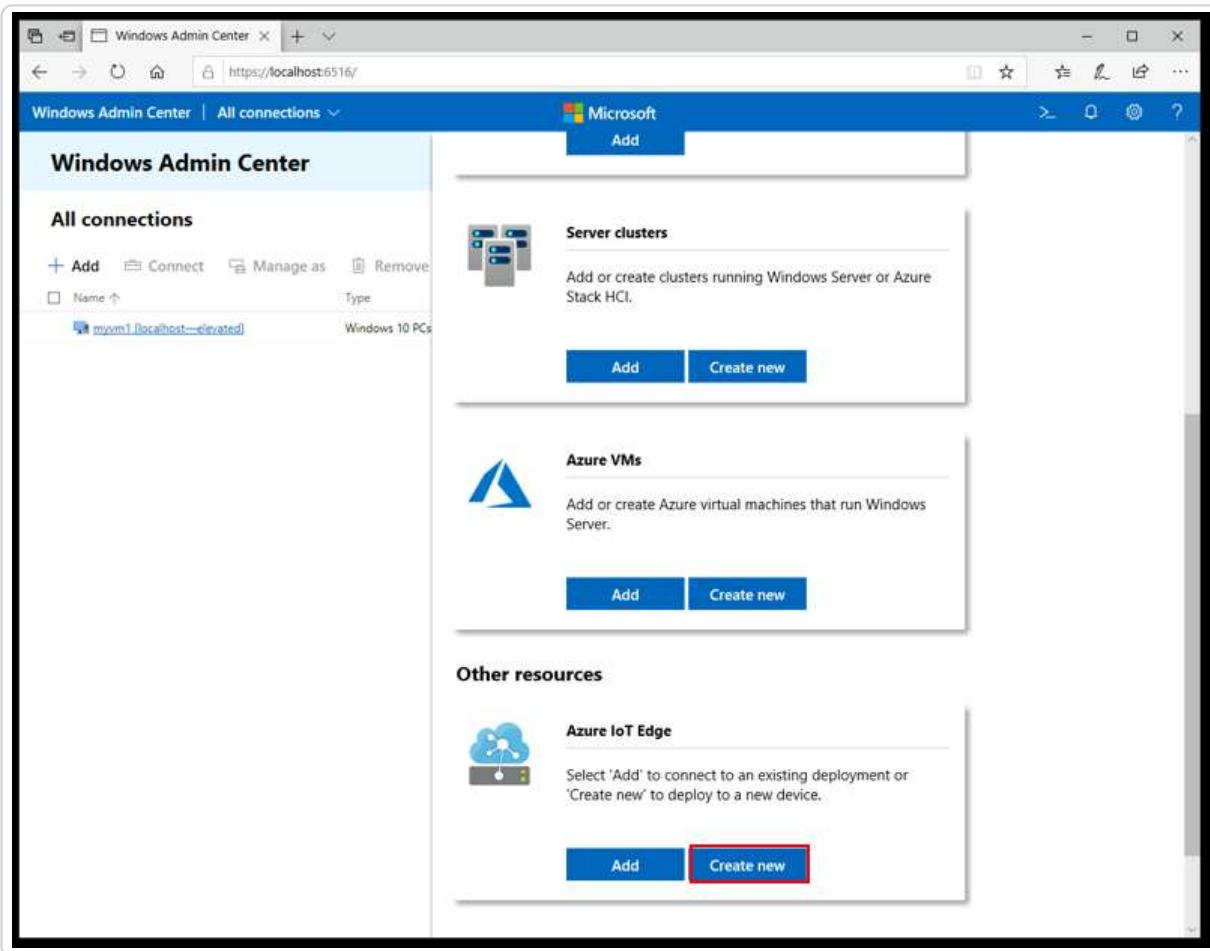
On the Windows Admin Center start page, under the list of connections, you will see a local host connection representing the PC where you are running Windows Admin Center. Any additional servers, PCs, or clusters that you manage will also show up here.

You can use Windows Admin Center to install and manage Azure IoT Edge for Linux on Windows on either your local device or remote managed devices. In this guide, the local host connection will serve as the target device for the deployment of Azure IoT Edge for Linux on Windows.

1. Select Add.



2. On the **Add or create resources** pane, locate the **Azure IoT Edge** tile. Select **Create new** to install a new instance of Azure IoT Edge for Linux on Windows on a device.



3. The **Create an Azure IoT Edge for Linux on Windows deployment** pane will open. On the **1. Getting Started** tab, verify that your target device meets the minimum requirements, and select **Next**.

## Create an Azure IoT Edge for Linux on Windows deployment

1 Getting Started 2 Deploy 3 Connect

### 1.1 Prerequisites

- 1.2 License Terms
- 1.3 Diagnostic Data



### Welcome to Azure IoT Edge for Linux on Windows

This wizard guides you through deployment of Azure IoT Edge for Linux on Windows and connecting it to your Azure IoT Hub. [Learn more](#)

#### Prerequisites

Please ensure your target device meets the minimum system requirements before continuing with your installation.

#### Minimum system requirements on target device

- Windows 10 or Windows Server build 17763 or later
- At least 1 GB of free memory
- 10 GB of free disk space

#### This instance of Windows Admin Center

- Must already have a PC or Server connection to your target device

Back

Next

Exit

4. Review the license terms, check **I Accept**, and select **Next**.
5. You can toggle **Optional diagnostic data** on or off, depending on your preference.
6. Select **Next: Deploy**.
7. On the **2.1 Deploy** tab, under **Select a target device**, click on your listed device to validate it meets the minimum requirements. Once its status is confirmed as supported, select **Next**.

Create an Azure IoT Edge for Linux on Windows deployment

1 Getting Started 2 Deploy 3 Connect

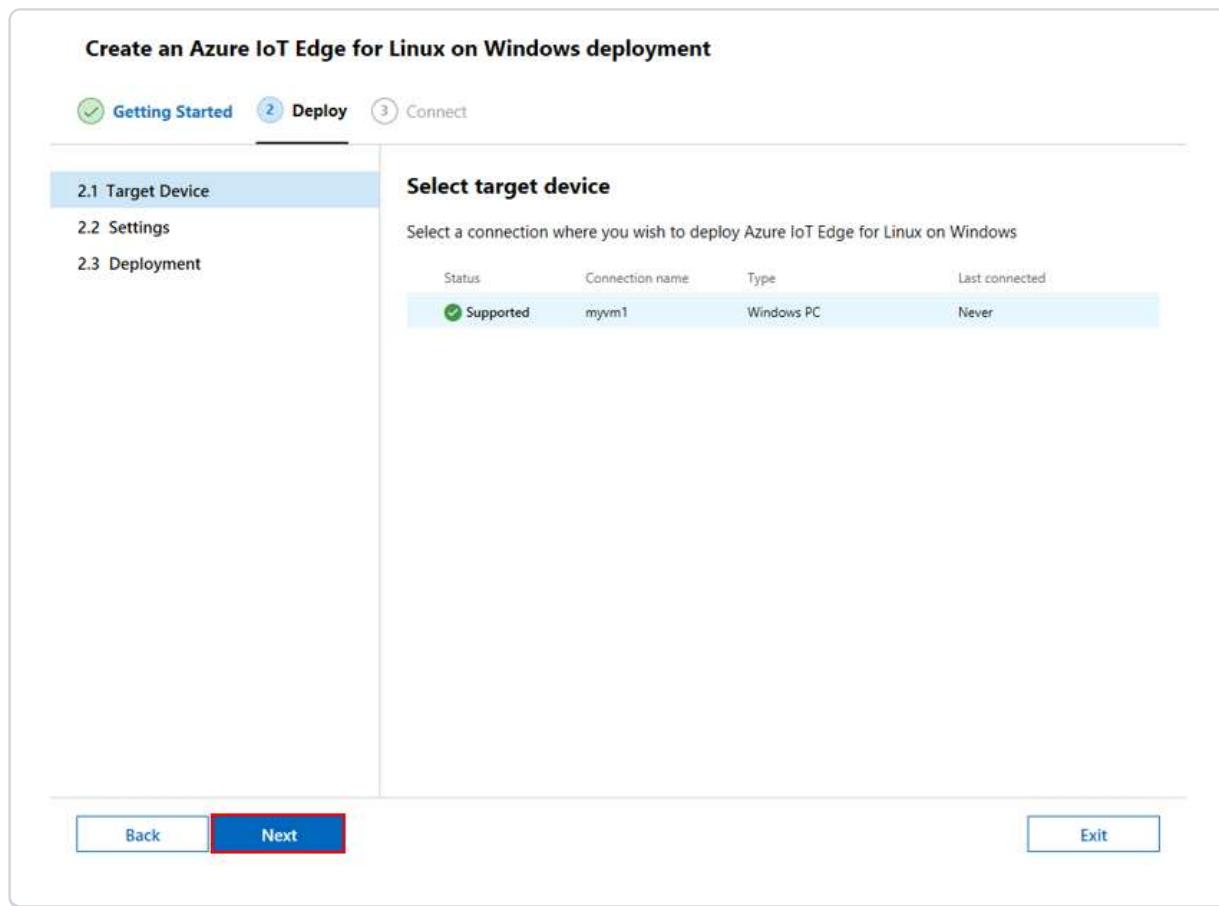
2.1 Target Device 2.2 Settings 2.3 Deployment

Select target device

Select a connection where you wish to deploy Azure IoT Edge for Linux on Windows

Status	Connection name	Type	Last connected
Supported	myvm1	Windows PC	Never

Back Next Exit



8. On the 2.2 Settings tab, change the **Memory and cores** configuration to **4 GB, 2 cores (Standard\_A2\_v2)** and select **Next**.

Create an Azure IoT Edge for Linux on Windows deployment

1 Getting Started 2 Deploy 3 Connect

2.1 Target Device 2.2 Settings 2.3 Deployment

Configuration settings

Please review the proposed settings and adjust for your specific workload needs

Virtual disk size\* 16

Memory and cores\* 4 GB, 2 cores (Standard\_A2\_v2) 4 GB, 2 cores (Standard\_A2\_v2)

Enable vTPM  Off

Switch type\* Transparent (External)

Switch name\* External

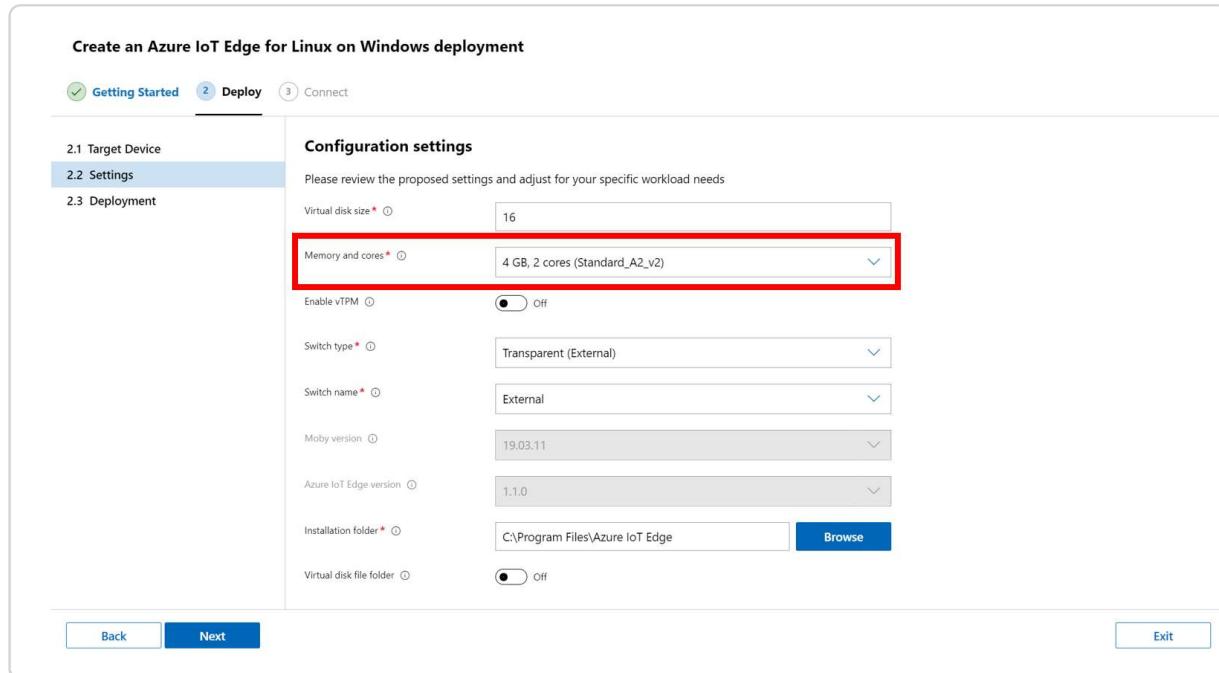
Moby version\* 19.03.11

Azure IoT Edge version\* 1.1.0

Installation folder\* C:\Program Files\Azure IoT Edge

Virtual disk file folder  Off

Back Next Exit



9. On the **2.3 Deployment** tab, you can watch the progress of the deployment. The full process includes downloading the Azure IoT Edge for Linux on Windows package, installing the package, configuring the host device, and setting up the Linux virtual machine. This process may take several minutes to complete. A successful deployment is pictured below.

The screenshot shows the 'Create an Azure IoT Edge for Linux on Windows deployment' interface. The 'Getting Started' tab is selected. The 'Deploy' tab is active, indicated by a blue bar and a '2' icon. The 'Connect' tab is shown with a '3' icon. The 'Deployment' tab is also visible. The main content area shows the following steps:

- Setup is complete** (Duration: 3 minutes 20 seconds):
  - Package download**: Shows a table with one item: 'Azure IoT Edge for Linux on Windows MSI' (310 MB) with a 'Complete' status.
  - Package installation**: Shows a table with one item: 'Azure IoT Edge for Linux on Windows MSI' (C:\Program Files\Azure IoT Edge) with a 'Complete' status.
  - Host configuration**: Shows a table with one item: 'Installing dependencies and configuring host' with a 'Complete' status.
  - Setup Linux Virtual Machine**: Shows a table with one item: 'Azure IoT Edge for Linux on Windows runtime and Linux virtual machine setup' with a 'Complete' status.

At the bottom, there are 'Back' and 'Next: Connect' buttons, and an 'Exit' button on the right.

10. Select **Next: Connect**

11. Go back to the Azure Portal in another browser tab (outside of the VM) and navigate to the **IoT Edge** tab of your **IoT Hub**.

Home > AIOTA-Demo-Hub

## AIOTA-Demo-Hub | IoT Edge

IoT Hub

Search (Ctrl+ /) < Add an IoT Edge device Create Deployment Create Layered Deployment Refresh Delete

Automatic Device Management

- IoT Edge
- IoT device configuration
- Device updates

Messaging

- File upload
- Message routing

Security

- Overview
- Security Alerts

Recommendations

- Settings

Monitoring

Deploy Azure services and solution-specific code to on-premises devices. Use IoT Edge devices to perform compute and analytics tasks.

### IoT Edge devices

IoT Edge deployments

IoT Edge devices

Field	Operator	Value
+	=	

Add new clause

Query devices

Device ID	Runtime Response	IoT Edge Module Count
AIOTA-Device	N/A	0

12. Click on the device ID of your device. Copy the Primary Connection String field.

AIOTA-Device

AIOTA-Demo-Hub

Save Set modules Manage child devices Device twin Manage keys Refresh

Device ID	AIOTA-Device
Primary Key	.....
Secondary Key	.....
Primary Connection String	.....
Secondary Connection String	.....
IoT Edge Runtime Response	N/A
Enable connection to IoT Hub	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
Parent device	No parent device

13. Paste it into the device connection string field in the Windows Admin Center. Then, choose Provisioning with the selected method.

Create an Azure IoT Edge for Linux on Windows deployment

Getting Started Deploy 3 Connect

3.1 Provisioning

Azure IoT Edge device provisioning

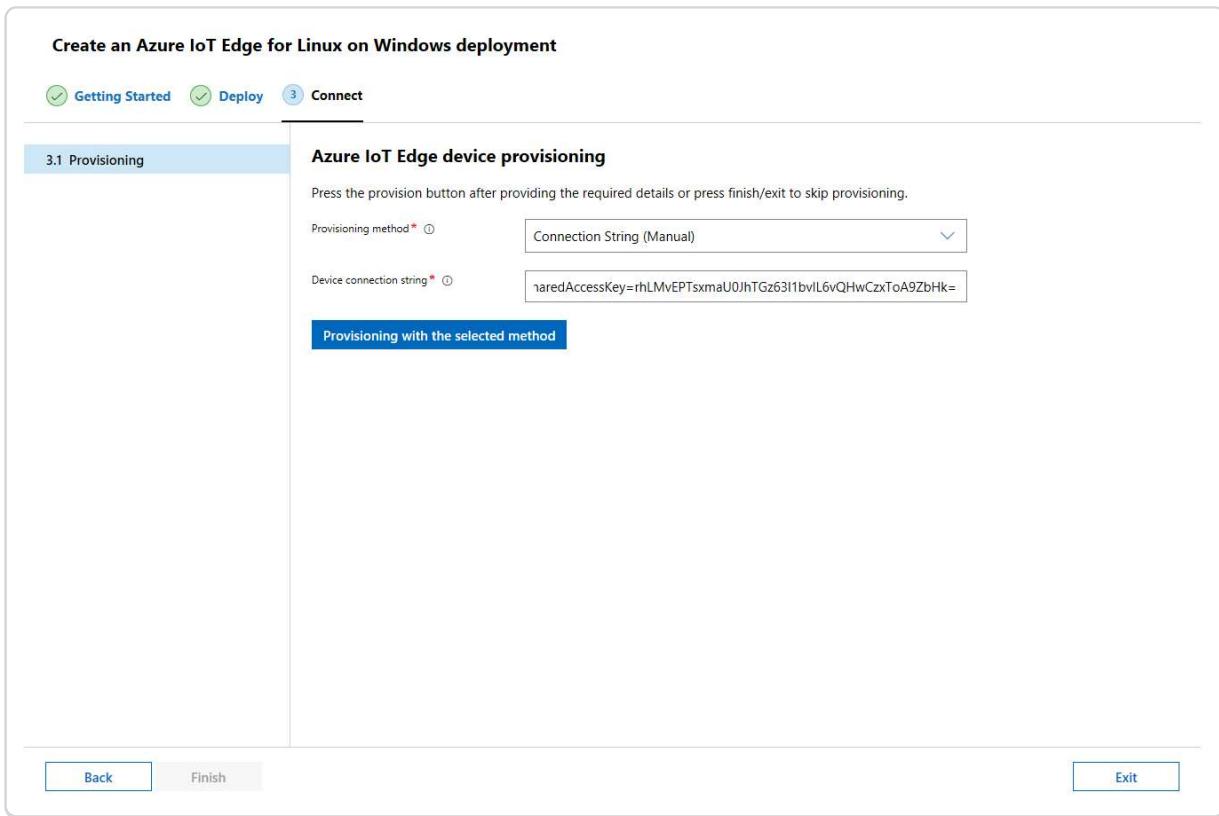
Press the provision button after providing the required details or press finish/exit to skip provisioning.

Provisioning method\* ⓘ Connection String (Manual)

Device connection string\* ⓘ `SharedAccessKey=rhLMvEPTsxmaU0JhTGz63l1bvL6vQHwCzxToA9ZbHk=`

Provisioning with the selected method

Back Finish Exit



14. Select **Finish** once Azure IoT device has successfully provisioned.

## Task 6: Verify successful configuration

1. Select your IoT Edge device from the list of connected devices in Windows Admin Center to connect to it.

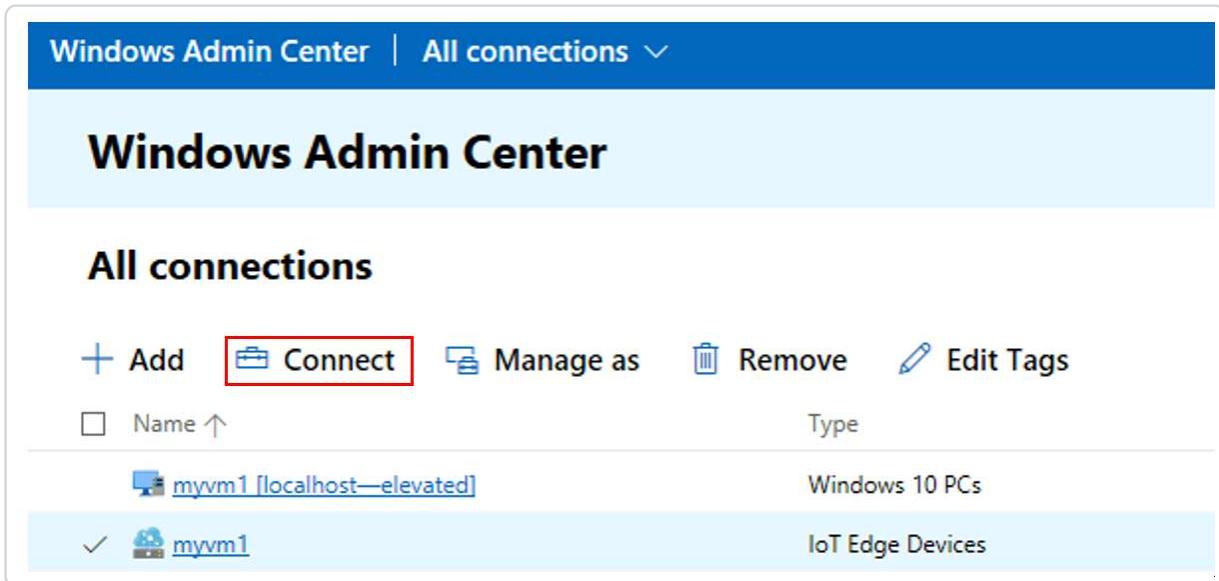
Windows Admin Center | All connections

## Windows Admin Center

### All connections

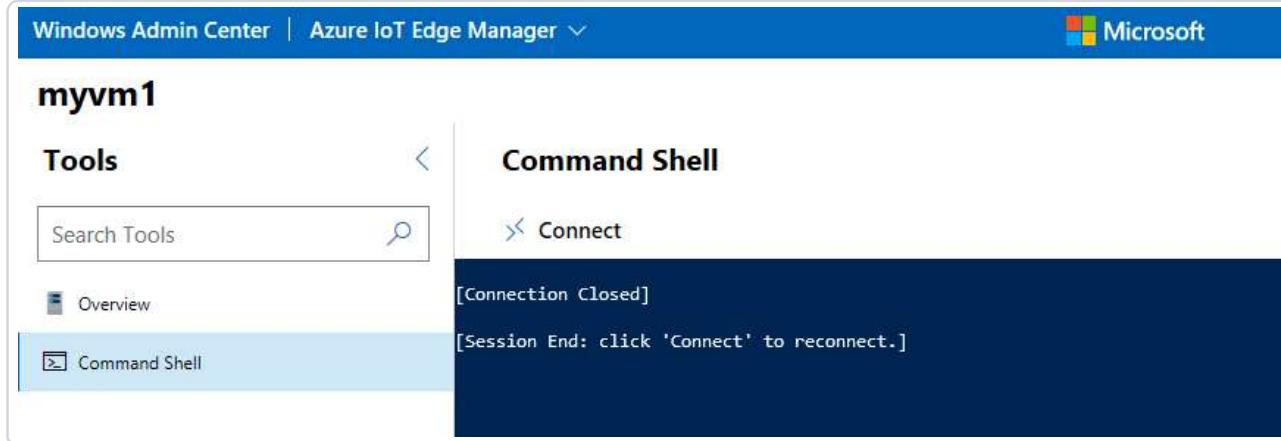
+ Add **Connect** Manage as Remove Edit Tags

Name ↑	Type
<a href="#">myvm1 [localhost—elevated]</a>	Windows 10 PCs
<a href="#">myvm1</a>	IoT Edge Devices



2. The device overview page displays some information about the device:

- The **IoT Edge Module List** section shows running modules on the device. When the IoT Edge service starts for the first time, you should only see the **edgeAgent** module running. The **edgeAgent** module runs by default and helps to install and start any additional modules that you deploy to your device.
  - The **IoT Edge Status** section shows the service status, and should be reporting **active** (running).
3. If you need to troubleshoot the IoT Edge service, use the **Command Shell** tool on the device page to ssh (secure shell) into the virtual machine and run the Linux commands.



\* If you need to troubleshoot the service, retrieve the service logs by inputting the following command:

```
```bash
journalctl -u iotedge
```
```

\* Use the `check` tool to verify configuration and connection status of the device by using the following command:

```
```bash
sudo iotedge check
```
```

[!TIP]

If your Azure IoT Edge VM is not running due to a break or interrupt in connection, please visit Hyper-V Manager:

1. Start the VM, if you are unable to start the VM, right click on the VM and clear "last saved state" and try again

2. Visit the networking tab at the bottom and wait until an IP address has been populated to ensure that your VM is running.

## Exercise 3: Set up Live Video Analytics

This exercise will be completed in the Azure Virtual Machine as well.

### Task 1: Download Tools and Resources

In your **Azure Virtual Machine** install the following tools:

- Visual Studio Code: <https://code.visualstudio.com/Download>
- .Net Core 3.1 SDK: <https://dotnet.microsoft.com/download/dotnet/thank-you/runtime-desktop-3.1.14-windows-x64-installer>
- Download VLC media player: <https://www.videolan.org/>

Unzip the following folders and **move to your Azure VM's Desktop**:

- Windows Application: <https://microsoft-my.sharepoint-df.com/:u:/p/fcabrera/EUvyooP-wZxMn4L1Hzjy8K4By7bDvrsLiV6EF-LZGGsOfw?e=SXZ3MD>
- Certificates: <https://microsoft-my.sharepoint-df.com/:u:/p/fcabrera/EdBZVhNFFpJLro5h7hJ8pjgBLI5qaXZ7fE8gn8GwsBbtTw?e=GEns2F>

### Task 2: Create Azure Resources

1. Go to [Azure portal](#) and select the Cloud Shell icon.



2. If you're using Cloud Shell for the first time, you'll be prompted to select a subscription to create a storage account and a Microsoft Azure Files share. Select **Create storage** to create a storage account for your Cloud Shell session information. This storage account is separate from the account that the script will create to use with your Azure Media Services account.

3. In the drop-down menu on the left side of the Cloud Shell window, select **Bash** as your environment. bash-environment



4. Run the following command.

```
bash -c "$(curl -sL https://aka.ms/1va-edge/setup-resources-for-samples)"
```

Upon successful completion of the script, you should see all of the required resources in your subscription. A total of 12 resources will be setup by the script:

- i. **Streaming Endpoint** - This will help in playing the recorded AMS asset.
- ii. **Virtual machine** - This is a virtual machine that will act as your edge device.
- iii. **Disk** - This is a storage disk that is attached to the virtual machine to store media and artifacts.
- iv. **Network security group** - This is used to filter network traffic to and from Azure resources in an Azure virtual network.
- v. **Network interface** - This enables an Azure Virtual Machine to communicate with internet, Azure, and other resources.
- vi. **Bastion connection** - This lets you connect to your virtual machine using your browser and the Azure portal.
- vii. **Public IP address** - This enables Azure resources to communicate to Internet and public-facing Azure services
- viii. **Virtual network** - This enables many types of Azure resources, such as your virtual machine, to securely communicate with each other, the internet, and on-premises networks.
- ix. **IoT Hub** - This acts as a central message hub for bi-directional communication between your IoT application, IoT Edge modules and the devices it manages.
- x. **Media service account** - This helps with managing and streaming media content in Azure.
- xi. **Storage account** - You must have one Primary storage account and you can have any number of Secondary storage accounts associated with your Media Services account.
- xii. **Container registry** - This helps in storing and managing your private Docker container images and related artifacts.

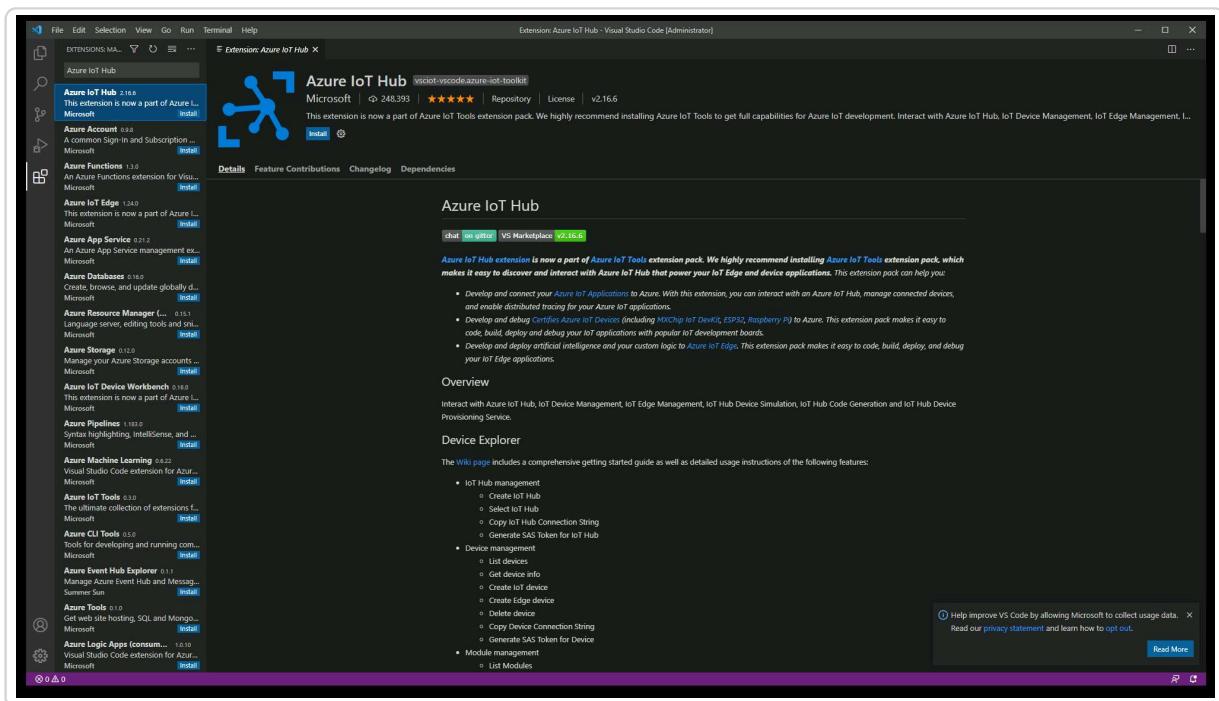
5. Follow the prompted steps in the script:

| Setting  | Value   |
|--|---|
| Subscription                                       | Confirm which subscription you would like to use                                |
| Region   | <b>eastus</b>   |
| Use your own edge device and as an IoT edge device | Yes - Y   |
| Device ID  | Navigate to Azure IoT Hub >> IoT Edge. The Device ID is the name of the device. |
| IoT Hub to Use                                     | Enter name of IoT Hub   |

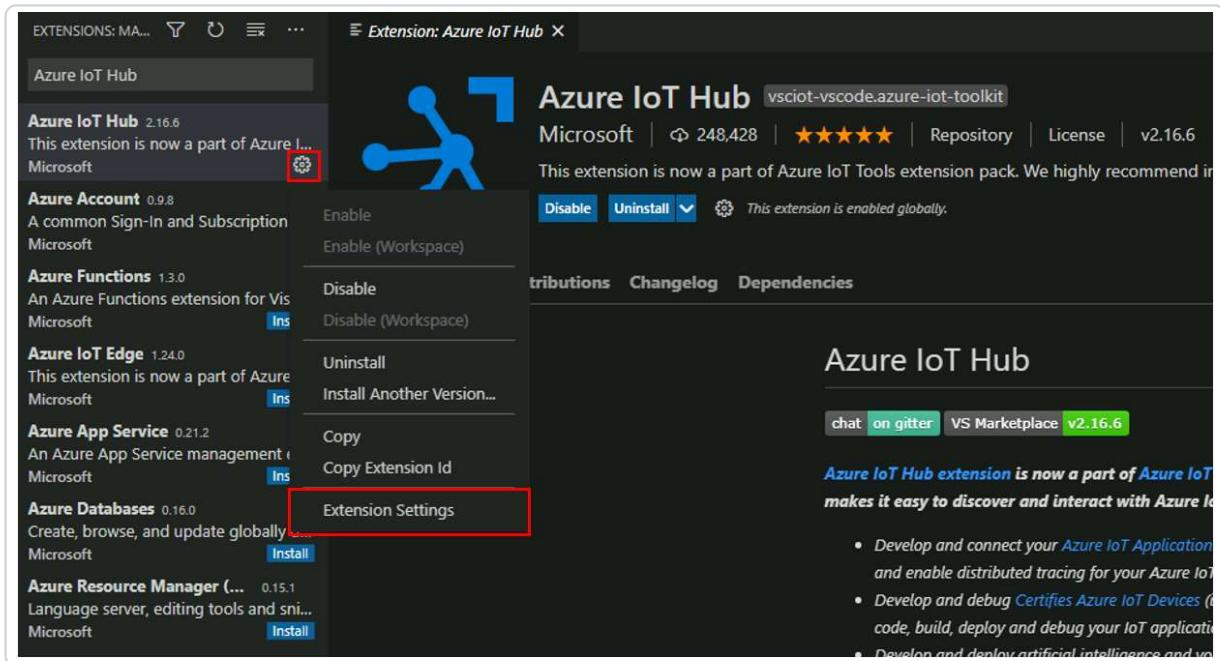
6. Wait a few minutes for the script to complete running - you should see a thumbs up emoticon at the end.

## Task 3: Configure the Azure IoT Tools extension

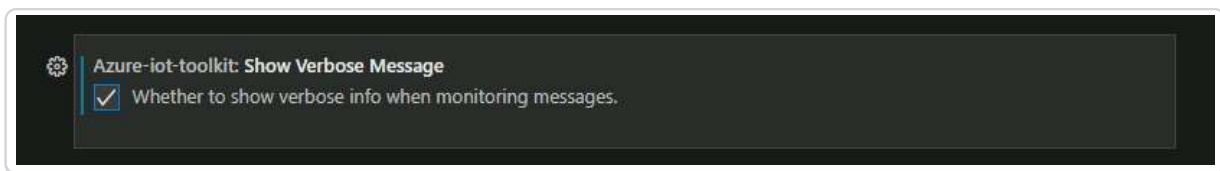
1. In your VM, open Visual Studio Code.
2. Open the Extensions tab (or press **Ctrl+Shift+X**) and search for Azure IoT Hub.



3. **Install Azure IoT Hub**
4. In the Extensions search, select the "gear icon" to open the menu and select **Extension Settings**.

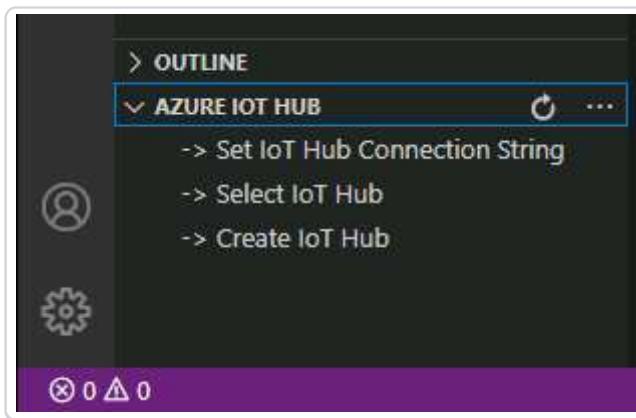


5. Search and enable "Show Verbose Message".



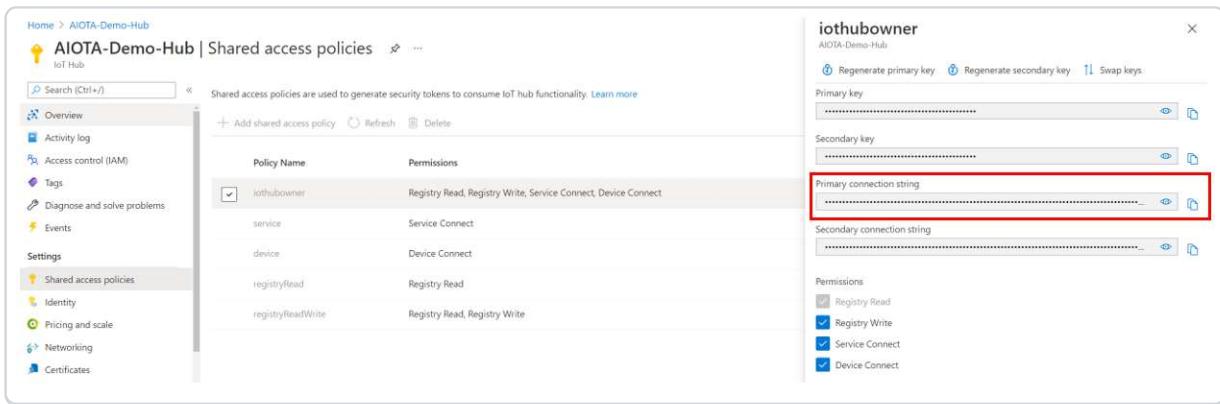
6. Select View > Explorer. Or, select Ctrl+Shift+E.

7. In the lower-left corner of the Explorer tab, select Azure IoT Hub.



8. Select the More Options icon to see the context menu. Then select Set IoT Hub Connection String.

9. In the Azure Portal, navigate to IoT Hub >> Settings >> Shared access policies >> Select the Policy for iothubowner and copy Connection string-primary key



Home > AIOTA-Demo-Hub  
IoT Hub  
AIOTA-Demo-Hub | Shared access policies

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Events Settings Shared access policies Identity Pricing and scale Networking Certificates

Shared access policies are used to generate security tokens to consume IoT hub functionality. Learn more

+ Add shared access policy Refresh Delete

| Policy Name       | Permissions  |
|-------------------|--|
| iothubowner       | Registry Read, Registry Write, Service Connect, Device Connect |
| service           | Service Connect  |
| device            | Device Connect   |
| registryRead      | Registry Read  |
| registryReadWrite | Registry Read, Registry Write                                  |

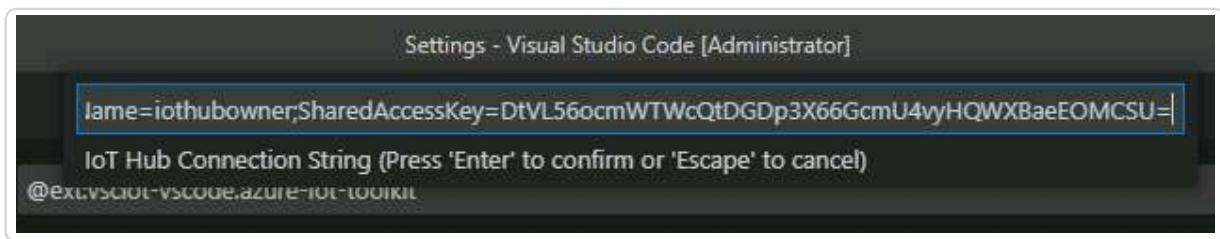
iothubowner  
AIOTA-Demo-Hub

Regenerate primary key Regenerate secondary key Swap keys

Primary key  
Secondary key  
Primary connection string  
Secondary connection string

Permissions  
 Registry Read  
 Registry Write  
 Service Connect  
 Device Connect

10. Paste in visual studio code.

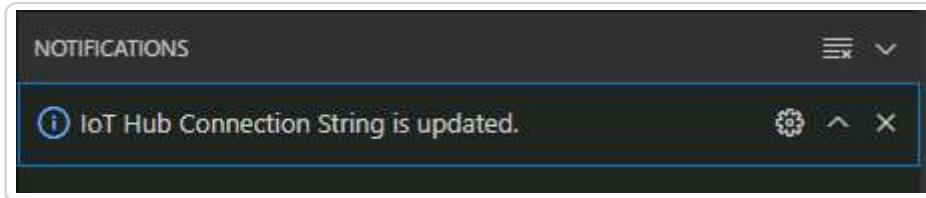


Settings - Visual Studio Code [Administrator]

IoT Hub Connection String (Press 'Enter' to confirm or 'Escape' to cancel)  
@ext:vscode-iot-toolkit

```
lame=iothubowner;SharedAccessKey=DtVL56ocmWTWcQtDGDp3X66GcmU4vyHQWXBaeEOMCSU=
```

11. Notification on the bottom right in Visual Studio should alert you that the IoT Hub Connection String has been updated.



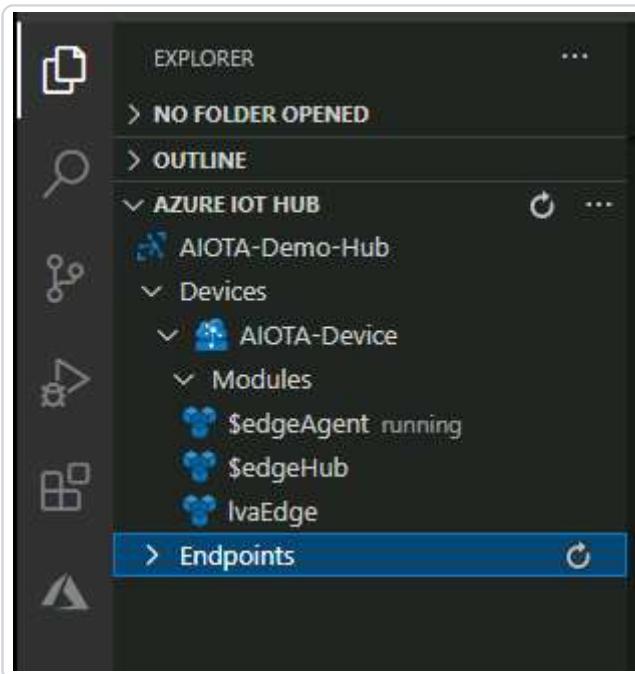
NOTIFICATIONS

IoT Hub Connection String is updated.

12. At this point you should sign into your Azure account in Visual Studio if you haven't done so already by selecting the **Azure** icon on the left hand panel.

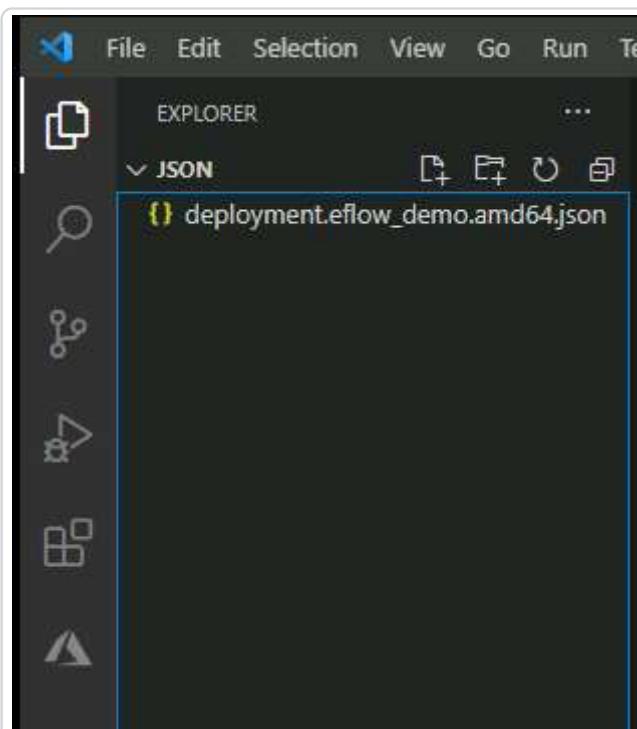


13. Select your IoT Hub under **EndPoints**.



## Task 4: Deploy Modules on Windows Host

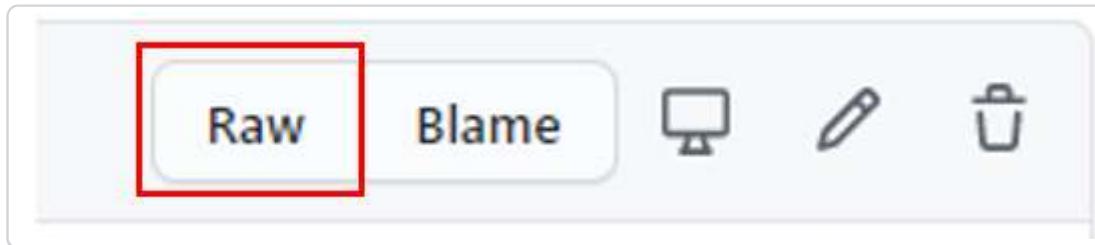
1. Create a new folder on your Desktop called **JSON**
2. In Visual Studio navigate to **File >> New File**.
3. Save the file as **deployment.eflow\_demo.amd64.json** in your **JSON** folder.
4. Open the **JSON** folder in Visual Studio in the explorer tab.



5. In a browser window, navigate to the deployment.JSON file:

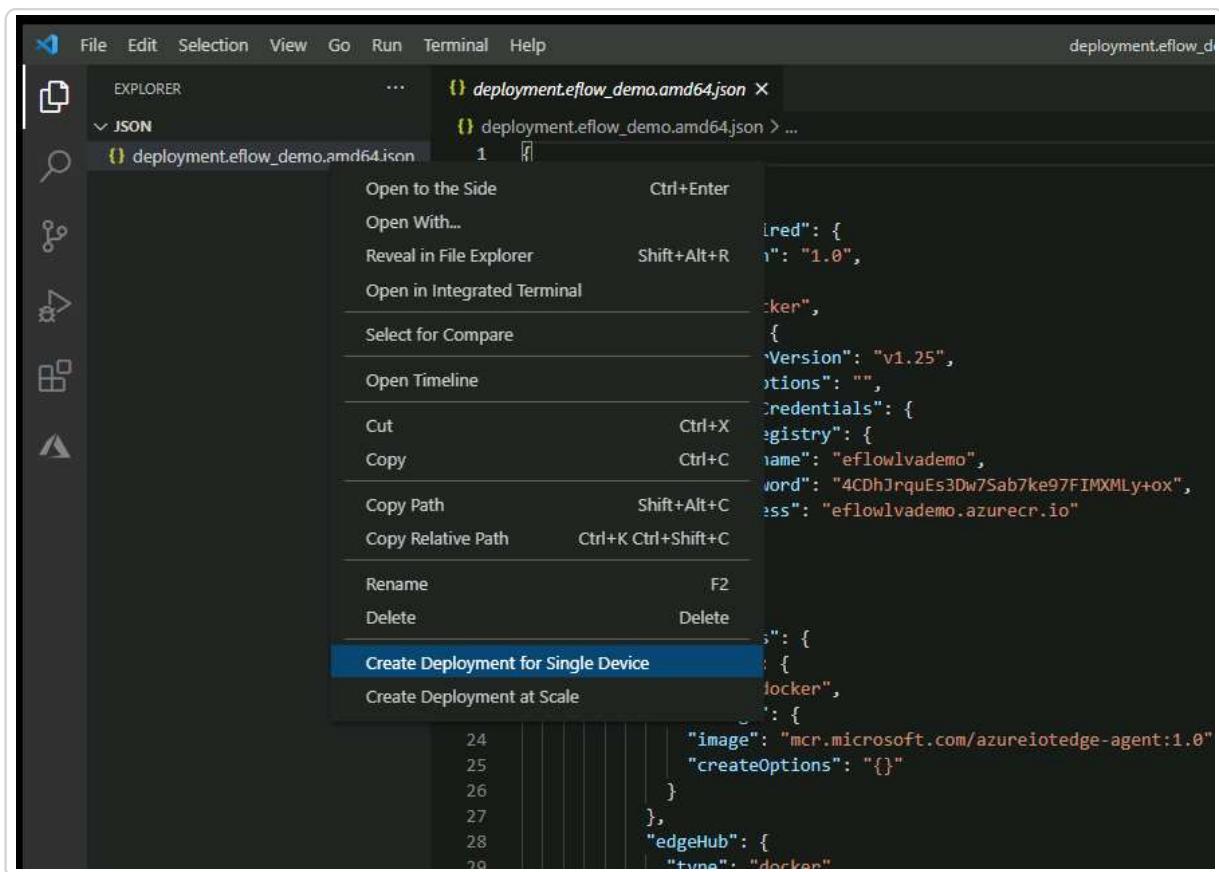
[https://github.com/fcabrera23/EFLOW\\_Demo/blob/main/deployment.eflow\\_demo.amd64.json](https://github.com/fcabrera23/EFLOW_Demo/blob/main/deployment.eflow_demo.amd64.json)

6. Select "Raw"

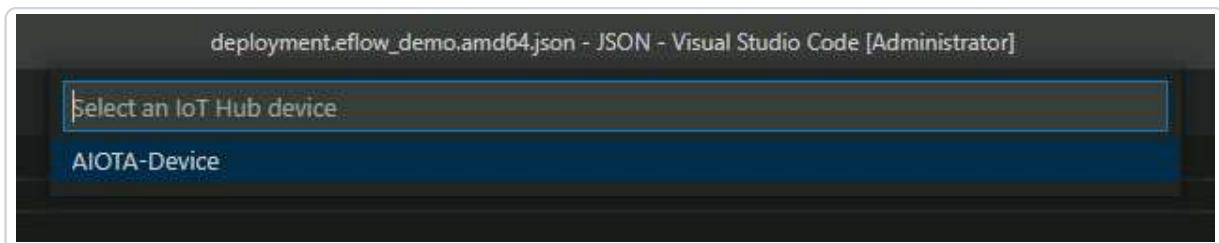


7. Copy the raw input and Paste into new file - and save.

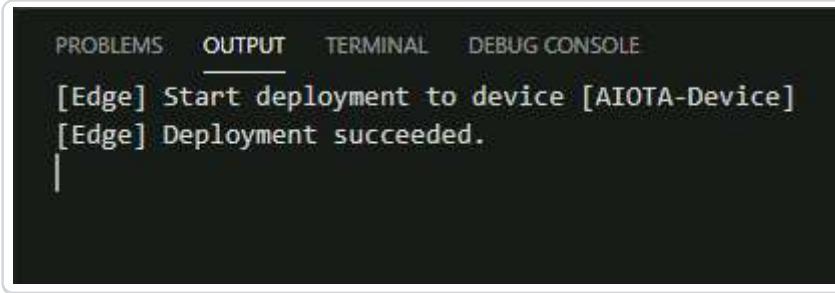
8. Right click on the file and select **Create Deployment for Single Device**



9. Select your IoT Hub Device



10. You should see that your deployment has succeeded.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
[Edge] Start deployment to device [AIOTA-Device]
[Edge] Deployment succeeded.
```

11. Now the modules are deployed, but no media graphs are active. So we must enable graphics.

12. Open a PowerShell Window and input the following commands:

Ssh into the EFLOW VM: Ssh-EflowVm

Run: sudo iptables -A INPUT -p udp --dport 554 -j ACCEPT

Run: sudo iptables -A INPUT -p tcp --dport 554 -j ACCEPT

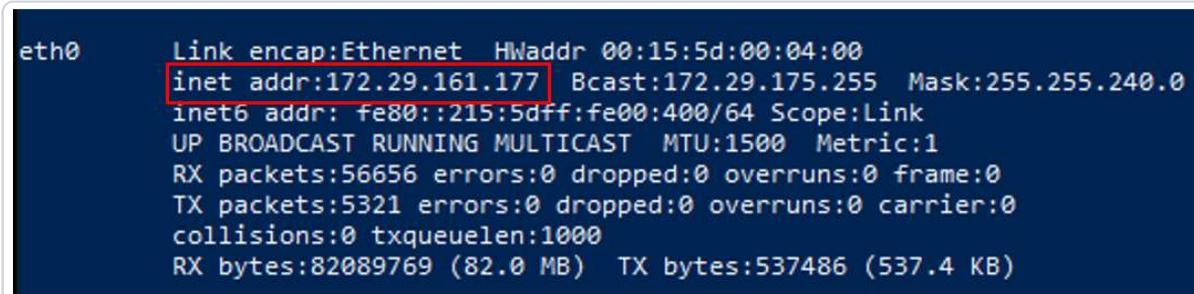
Run: sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

Run: sudo iptables -A INPUT -p tcp --dport 5671 -j ACCEPT

Run: sudo iptables-save | sudo tee /etc/systemd/scripts/ip4save > /dev/null

Run: mkdir ~/certs/

Identify the EFLOW VM IP sudo ifconfig



```
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:00
          inet addr:172.29.161.177  Bcast:172.29.175.255  Mask:255.255.240.0
                      inet6 addr: fe80::215:5dff:fe00:400/64 Scope:Link
                         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                         RX packets:56656 errors:0 dropped:0 overruns:0 frame:0
                         TX packets:5321 errors:0 dropped:0 overruns:0 carrier:0
                         collisions:0 txqueuelen:1000
                         RX bytes:82089769 (82.0 MB)  TX bytes:537486 (537.4 KB)
```

Run: wget https://raw.githubusercontent.com/Azure/live-video-analytics/master/edge/setup/prep\_device.sh

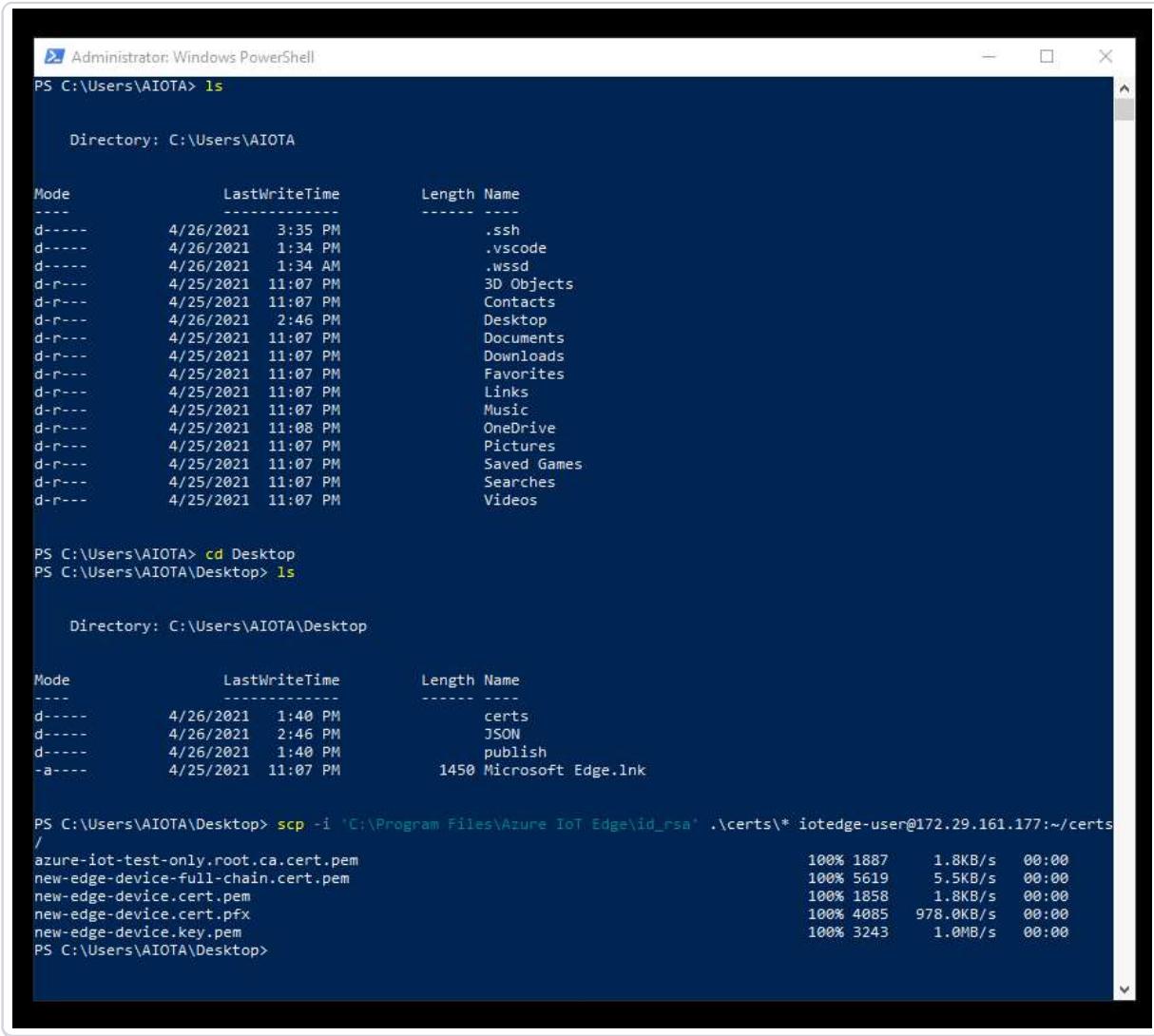
Run: sudo sh prep\_device.sh

13. In a new PowerShell Windows (representing the Windows host) input the following command to get the EFLOW VM IP address.

## Get-EflowVmAddr

[!NOTE] This should be identical to what you found in step 12 above.

14. Next, copy certificates to the EFLOW VM environment from the Windows host. Use SCP to copy the certificates downloaded. Make sure to navigate to your Desktop. See the picture below for commands on how to do that.



The screenshot shows a Windows PowerShell window with the following content:

```
Administrator: Windows PowerShell
PS C:\Users\AIOTA> ls

Directory: C:\Users\AIOTA

Mode                LastWriteTime       Length Name
----                -----        ----
d----
```

```
scp -i 'C:\Program Files\Azure IoT Edge\id_rsa' .\certs\* iotedge-user@eflowvm
```

[!NOTE] Don't forget to replace **eflowvm-ip** in the command above with your EFLOW VM's IP address found above in step 12 or 13.

## Task 5: Provision Azure IoT Edge for Linux Configuration

1. Back to the EFLOW PowerShell Window, run the following command to allow Azure IoT Edge to read the certificates.

```
sudo chown -R iotedge: ~/certs
```

2. Provision the Azure IoT Edge for Linux configuration To edit config.yaml run the following command:

```
sudo nano /etc/iotedge/config.yaml
```

3. Scroll down to **Certificates** section and update the file paths with what's below and don't forget to **UNCOMMENT** (remove hashes):

```
certificates:
  device_ca_cert: "/home/iotedge-user/certs/new-edge-device-full-chain.cert.pem"
  device_ca_pk: "/home/iotedge-user/certs/new-edge-device.key.pem"
  trusted_ca_certs: "/home/iotedge-user/certs/azure-iot-test-only.root.ca.cert"
```

```
certificates:
  device_ca_cert: "/home/iotedge-user/certs/new-edge-device-full-chain.cert.pem"
  device_ca_pk: "/home/iotedge-user/certs/new-edge-device.key.pem"
  trusted_ca_certs: "/home/iotedge-user/certs/azure-iot-test-only.root.ca.cert.pem"
#  auto_generated_ca_lifetime_days: 90
```

[!NOTE]

Make sure there are **no whitespaces** before certificates paths and two spaces indenting each sub part.

4. Scroll down a little further and update the **Host IP address** with the EFLOW VM's IP address.

```
#####
# Edge device hostname
#####
#
# Configures the environment variable 'IOTEDGE_GATEWAYHOSTNAME' injected into
# modules. Regardless of case the hostname is specified below, a lower case
# value is used to configure the Edge Hub server hostname as well as the
# environment variable specified above.
#
# It is important to note that when connecting downstream devices to the
# Edge Hub that the lower case value of this hostname be used in the
# 'GatewayHostName' field of the device's connection string URI.
#####
hostname: "172.31.237.63"
```

[!TIP]

To save the file and exit nano, press **CTRL**+**X**, confirm save and exit with **Y** and press **Enter**. This concludes the provisioning and configuration.

5. Restart IoT Edge by running the following command.

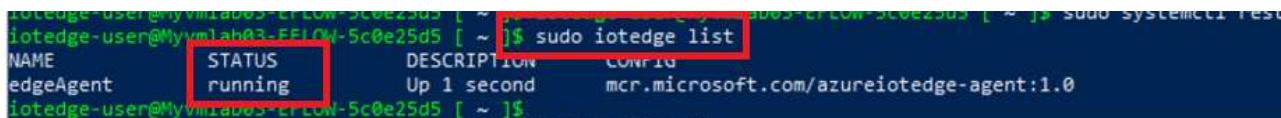
```
sudo systemctl restart iotedge
```

## Task 6: Test Video Stream

Make sure your iotedge is running before continue, run the following command inside your edge machine

```
sudo iotedge list
```

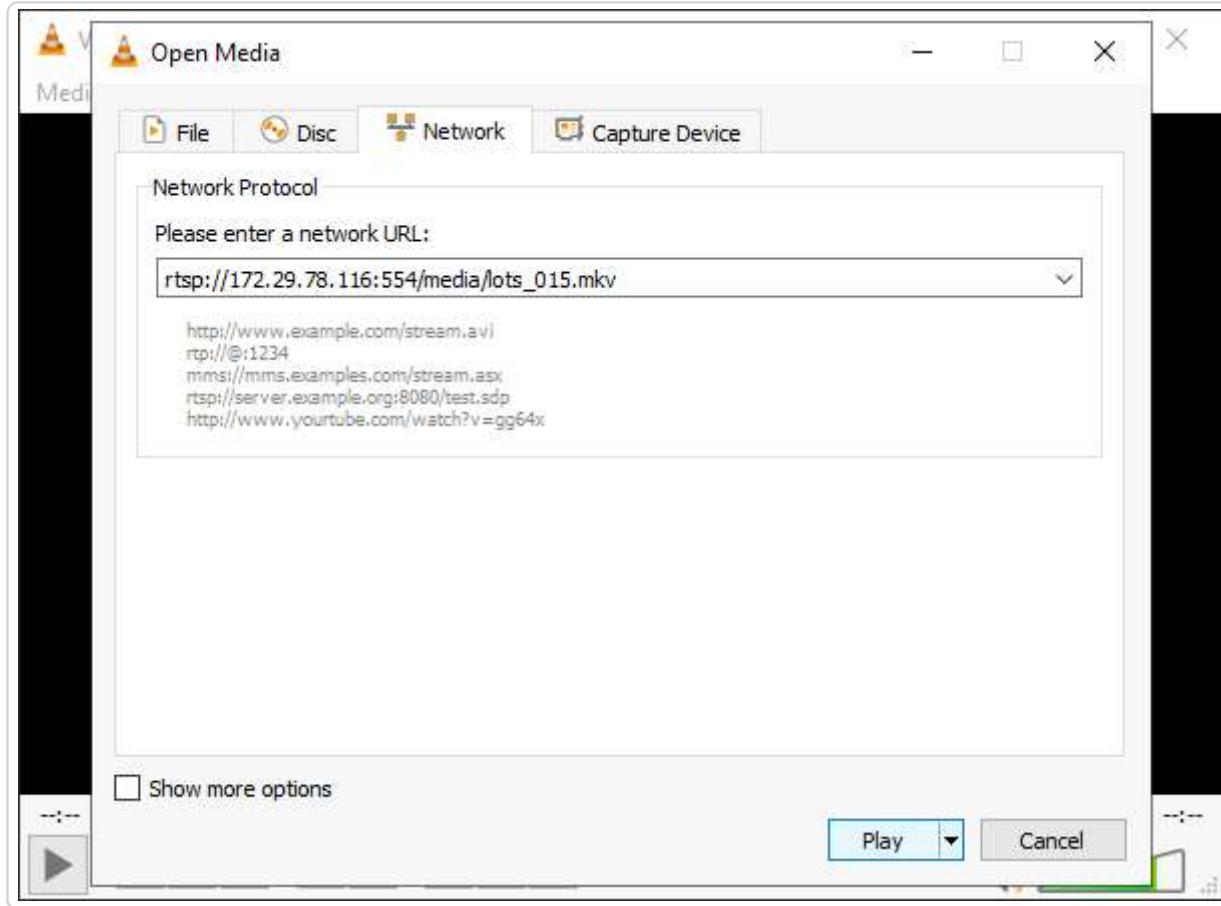
Your edge status should be running before continuing:



| NAME      | STATUS  | DESCRIPTION | CONFIG                                   |
|-----------|---------|-------------|--|
| edgeAgent | running | Up 1 second | mcr.microsoft.com/azureiotedge-agent:1.0 |

1. Open VLC media player
2. Navigate to **Media >> Open Network Stream**
3. Input the following as the **network URL**

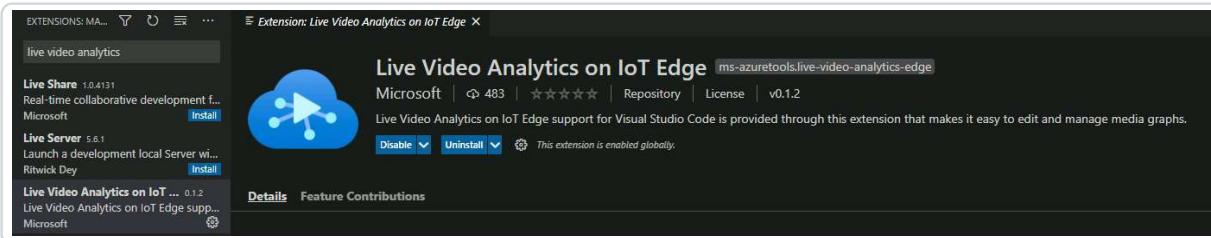
```
rtsp://<EFLOW VM IP>:554/media/lots_015.mkv
```



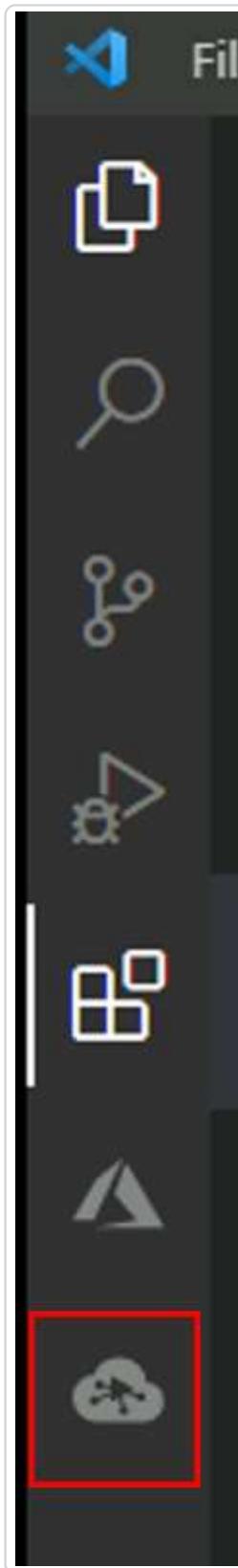
4. Select play to view the video stream

## Task 7: Enable Live Video Analytics: Inferencing

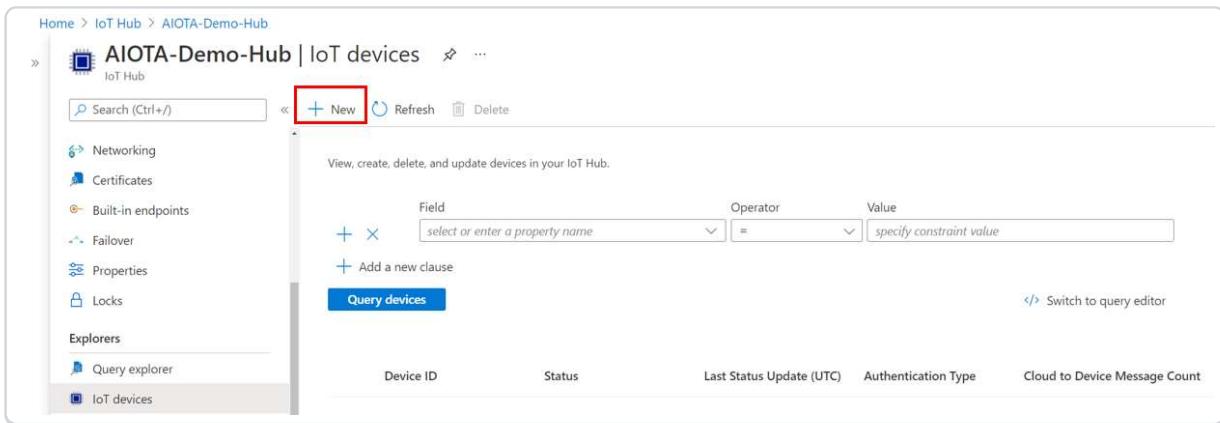
1. In Visual Studio Code, open the **Extensions** tab (or press **Ctrl+Shift+X**) and search for **Live Video Analytics on IoT Edge**.



2. **Install Live Video Analytics on IoT Edge** - once installed you will see the Live Video Analytics icon appear on the left bar LVA-Icon-Left



3. Create an Azure Child Device
4. Navigate to the **Azure Portal** >> **IoT Hub** >> **IoT Devices**
5. Select **+ New** to create a new device



The screenshot shows the Azure IoT Hub AIOTA-Demo-Hub IoT devices page. The 'New' button is highlighted with a red box. The page includes a search bar, a query editor, and a table with columns: Device ID, Status, Last Status Update (UTC), Authentication Type, and Cloud to Device Message Count. The 'IoT devices' tab is selected in the left sidebar.

6. Fill in the following parameters for your new device: | Setting | Value | |---| | Device ID | Give your device a name | | Authentication type | **Symmetric key** | | Autogenerate | Default - Keep Selected | | Connect this device to an IoT Hub | **Enable** | | IoT Hub to Use | Select your Parent IoT Edge Device |

## Create a device



Find Certified for Azure IoT devices in the Device Catalog



Device ID \* ⓘ

Demo123



Authentication type ⓘ

**Symmetric key** X.509 Self-Signed X.509 CA Signed

Primary key ⓘ

Enter your primary key

Secondary key ⓘ

Enter your secondary key

Auto-generate keys ⓘ



Connect this device to an IoT hub ⓘ

Enable

Disable

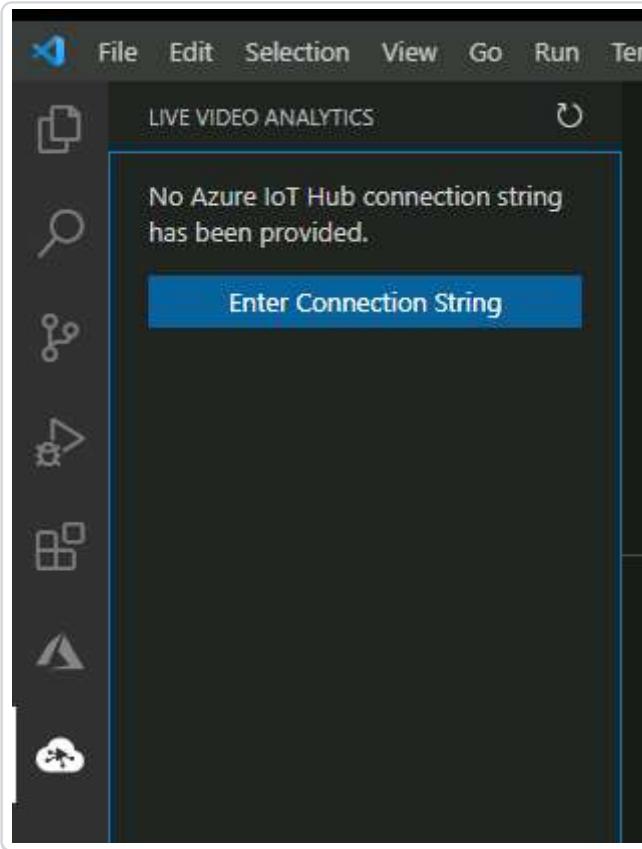
Parent device ⓘ

**AIOTA-Device** [Remove](#)

[Set a parent device](#)

**Save**

7. Add connection string in Live Video Analytics extension in Visual Studio



8. Azure Portal >> IoT Hub >> Share access policies >> select **iothubowner** >> copy Connection string-primary key

Home > IoT Hub > AIOTA-Demo-Hub

AIOTA-Demo-Hub | Shared access policies

Shared access policies are used to generate security tokens to consume IoT hub functionality. Learn more

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Events Settings Shared access policies Identity Pricing and scale Networking Certificates

Search (Ctrl+ /)

Policy Name Permissions

| Policy Name       | Permissions  |
|-------------------|--|
| iothubowner       | Registry Read, Registry Write, Service Connect, Device Connect |
| service           | Service Connect  |
| device            | Device Connect   |
| registryRead      | Registry Read  |
| registryReadWrite | Registry Read, Registry Write                                  |

iothubowner

AIOTA-Demo-Hub

Regenerate primary key Regenerate secondary key Swap keys

Primary key (red box)

Secondary key

Primary connection string (red box)

Secondary connection string

Permissions

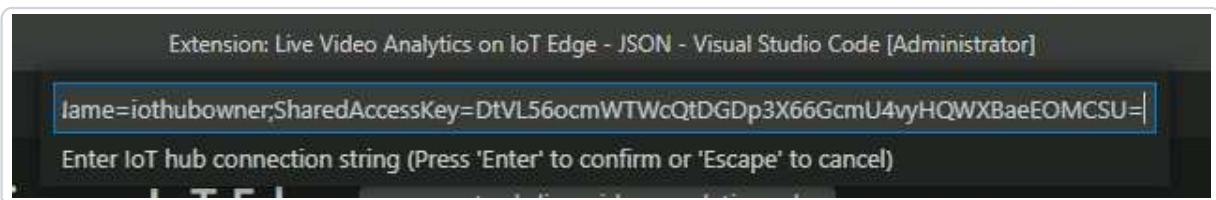
Registry Read (unchecked)

Registry Write (checked)

Service Connect (checked)

Device Connect (checked)

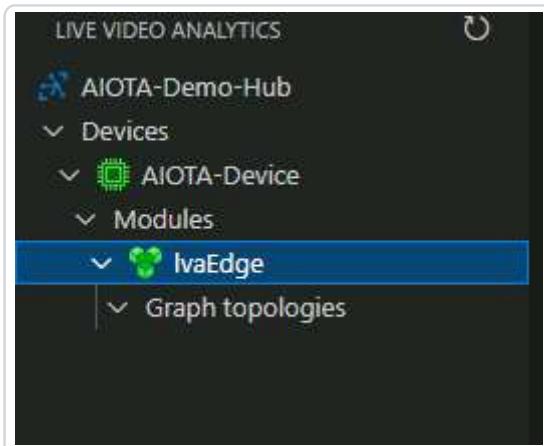
9. Enter IoT Hub connection string into Visual Studio



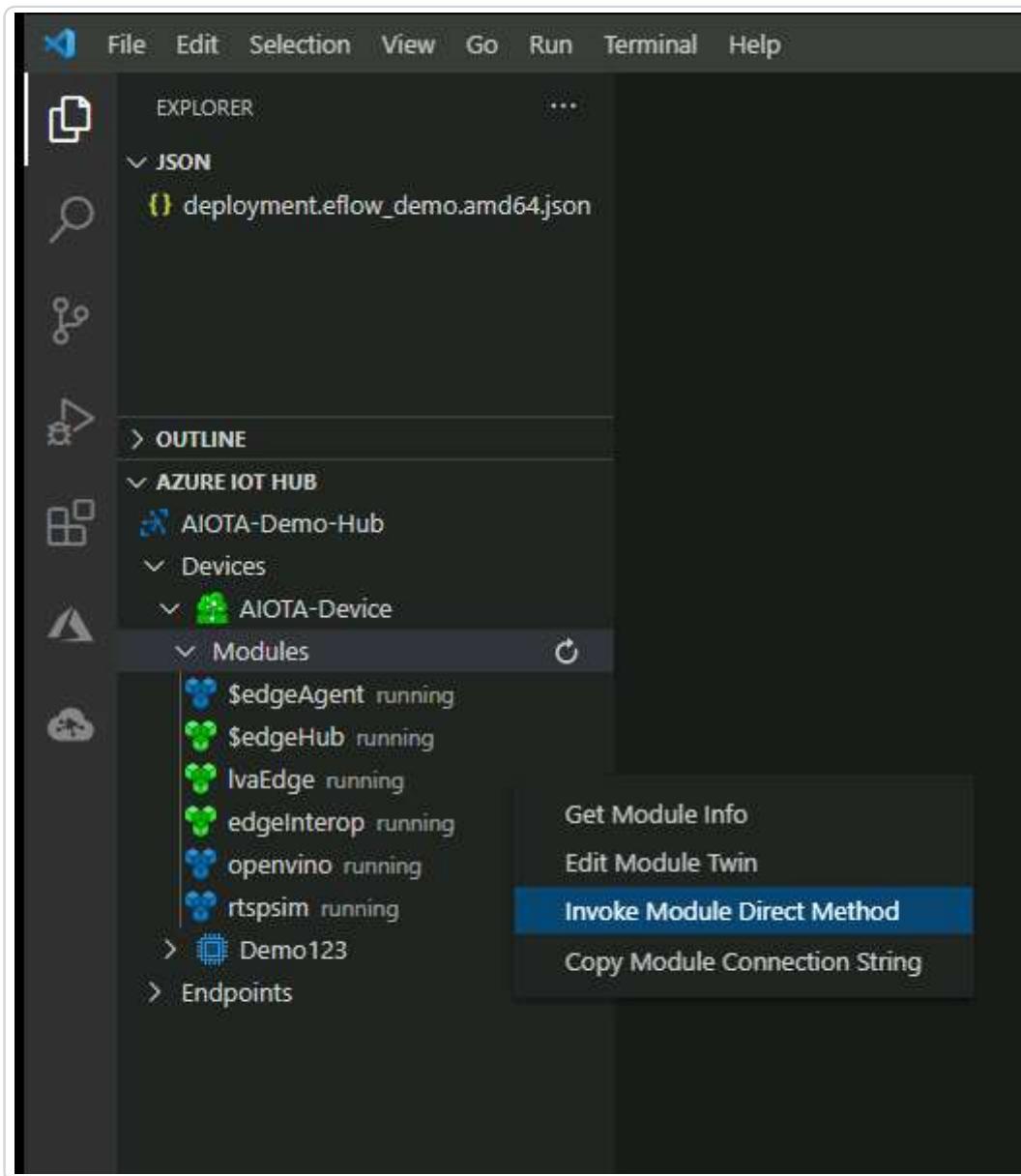
10. Select your IoT Edge PARENT Device

11. Select the Live Video Analytics module: **Ivaedge**

12. You should now see this in the Live Video Analytics pane.



13. Navigate to the Azure IoT Hub tab in Visual Studio >> Devices >> Modules >> IvaEdge right click and select Invoke Module Direct Method



14. Enter [Method name] as *GraphTopologySet*

JSON - Visual Studio Code [Administrator]

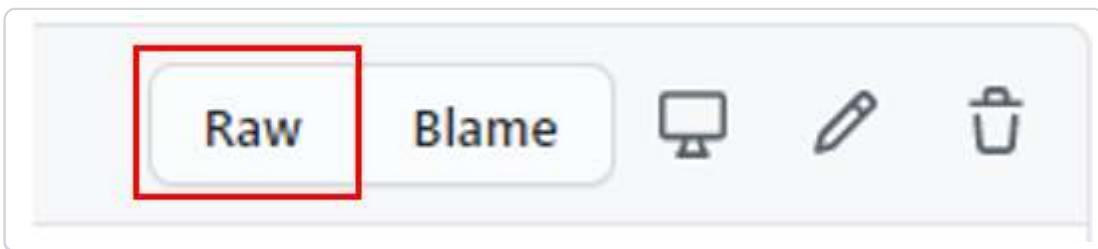
```
GraphTopologySet
```

Enter [Method Name] sent to [AIOTA-Device/lvaEdge] (Press 'Enter' to confirm or 'Escape' to cancel)

15. In a browser window, navigate to the lva\_graph\_topology\_track.JSON:

[https://github.com/fcabrera23/EFLOW\\_Demo/blob/main/lva\\_graph\\_topology\\_truck.json](https://github.com/fcabrera23/EFLOW_Demo/blob/main/lva_graph_topology_truck.json)

16. Select "Raw"



17. Copy the raw input

18. Paste the [Payload] (lva\_group\_topology\_track) in Visual Studio (black bar on the top) and hit enter.

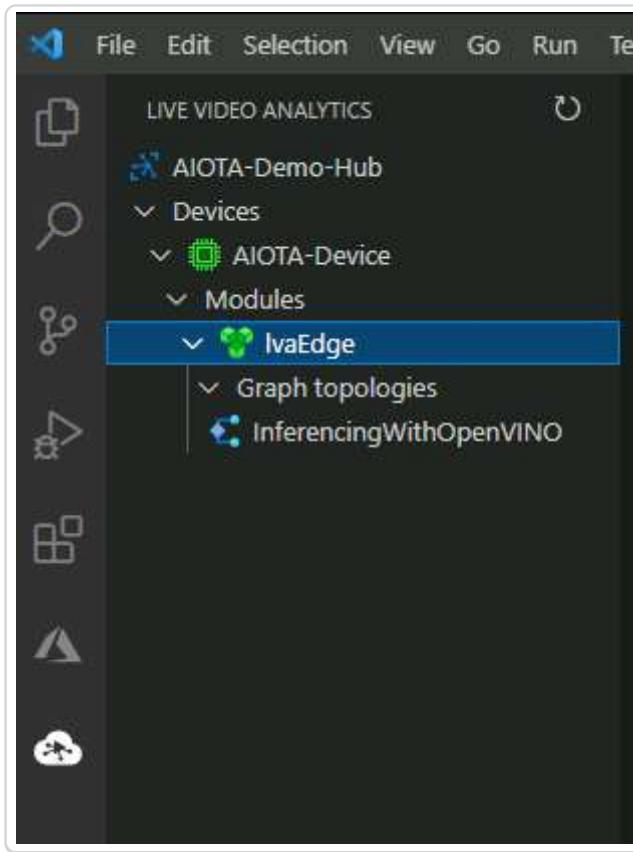
JSON - Visual Studio Code [Administrator]

```
    "enceOutput", "inputs": [ { "nodeName": "httpExtension" } ] } ] })
```

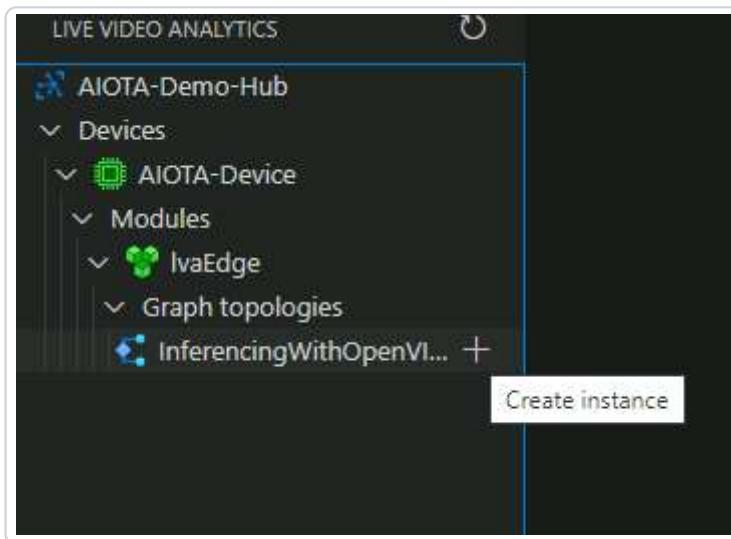
Enter [Payload] sent to [AIOTA-Device/lvaEdge] (Press 'Enter' to confirm or 'Escape' to cancel)

19. Navigate to the Live Video Analytics tab and you should see

**InferencingwithOpenVino** under Graph topologies



20. Select + next to **InferencingwithOpenVino** to create a new graph instance



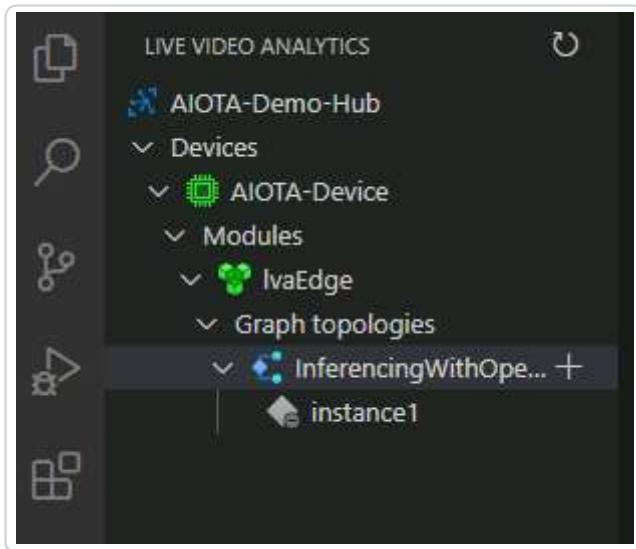
21. Input the following parameters:

| Setting       | Value                                 |
|---------------|---------------------------------------|
| Instance name | Give your device a name               |
| rtspUrl       | rtsp://rtspsim:554/media/lots_015.mkv |

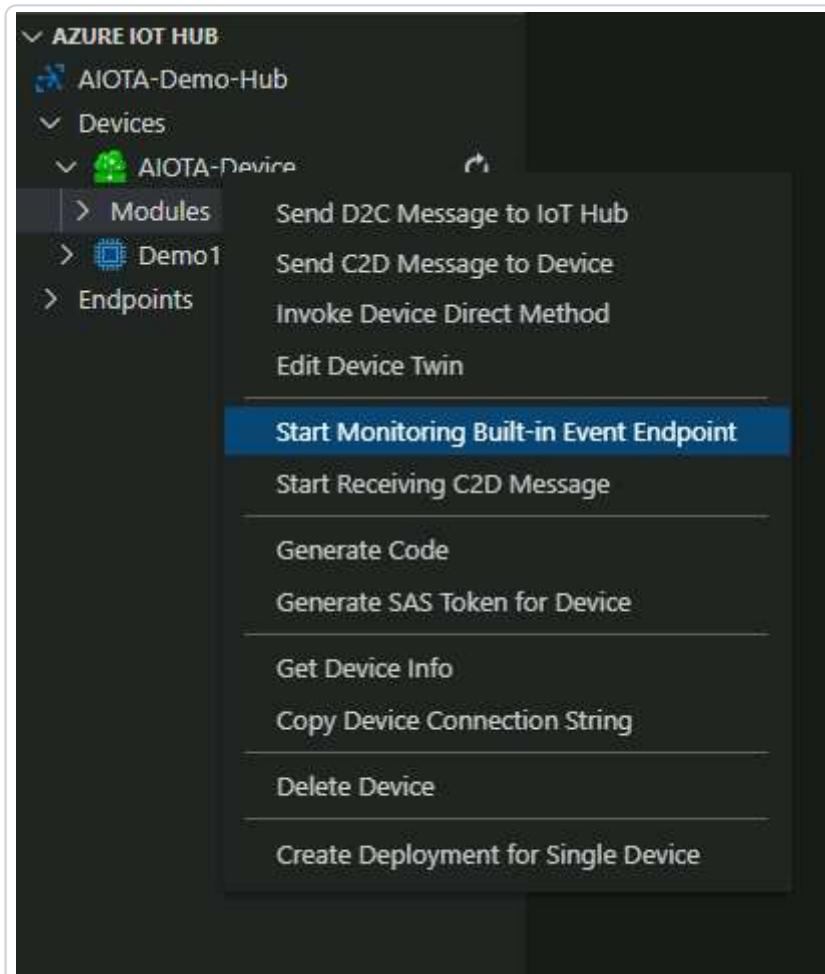
<kbd></kbd>

22. Select **Save** at the top right corner.

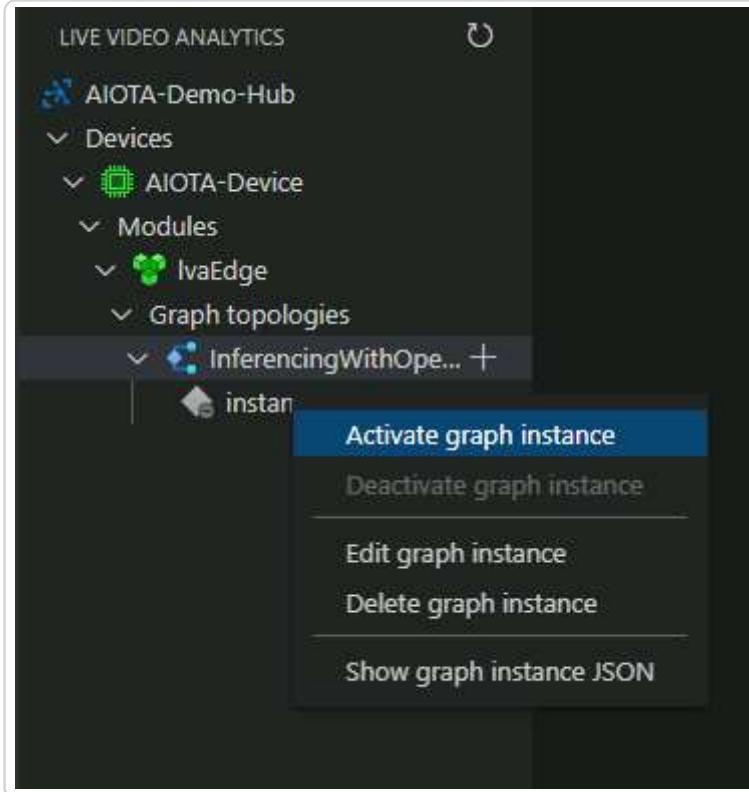
23. Under the graph topology, **InferecingwithOpenVino**, you should see your new instance has been created.



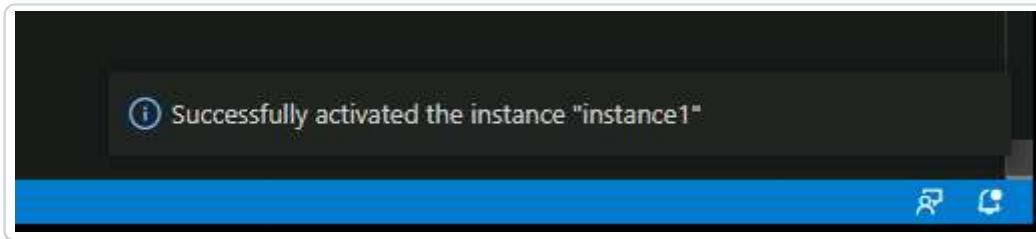
24. Navigate back to the **Azure IoT Hub** in Visual Studio >> **Device** > **IoT Edge Device** right click and select **Start Monitoring Built-in Event Endpoint**



25. Navigate back to the live video analytics tab in visual studio, and right click on your newly created instance and select **Activate Graph Instance**



26. You should receive a notification that you have successfully activated the instance success-message



27. In the output window you will now see the inferencing occurring for each bounded box of time.

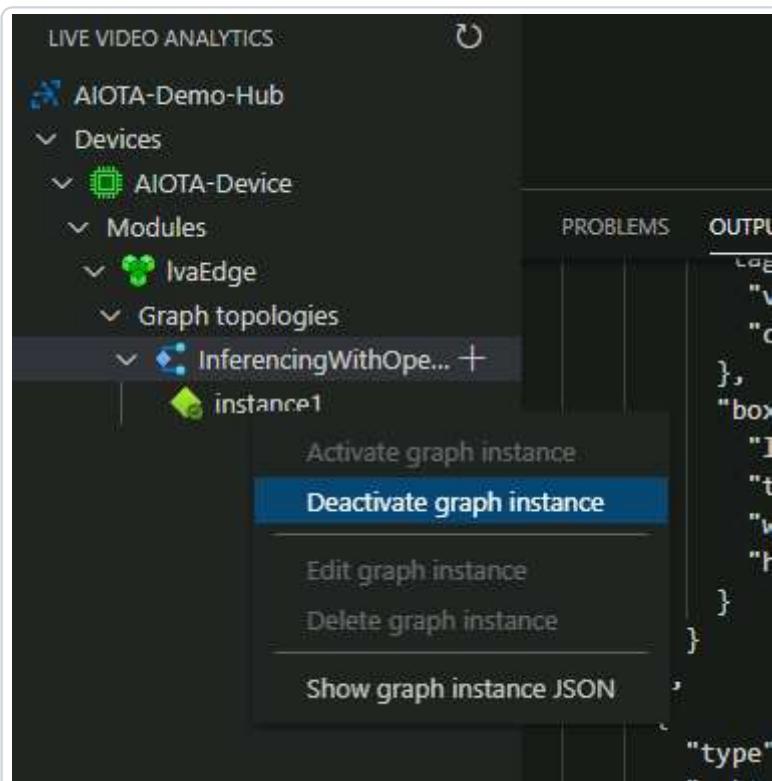
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

    "log": [
      {
        "value": "vehicle",
        "confidence": 0.78586924
      },
      "box": {
        "l": 0.24342363,
        "t": 0.08611458,
        "w": 0.05972965,
        "h": 0.06487141
      }
    ],
    {
      "type": "entity",
      "subtype": "vehicleDetection",
      "entity": {
        "tag": {
          "value": "vehicle",
          "confidence": 0.55225724
        },
        "box": {
          "l": 0.3998469,
          "t": 0.23444837,
          "w": 0.03865391,
          "h": 0.06640679
        }
      }
    }
  ],
  "properties": {
    "topic": "/subscriptions/db8411db-bdc3-47bc-b51e-9a687b462c43/resourceGroups/EFLOW/providers/microsoft.media/mediaservices/lvaeflowcowdymhbulpvu",
    "subject": "/graphInstances/instance1/processors/httpExtension",
    "eventType": "Microsoft.Media.Graph.Analytics.Inference",
    "eventtime": "2021-04-27T00:33:06.631Z",
    "dataVersion": "1.0"
  },
  "systemProperties": {
    "iothub-connection-device-id": "AIOTA-Device",
    "iothub-connection-module-id": "lvaEdge",
    "iothub-connection-auth-method": "{\"scope\":\"module\",\"type\":\"sas\",\"issuer\":\"iothub\",\"acceptingIpFilterRule\":null}",
    "iothub-connection-auth-generation-id": "637550401601814078",
    "iothub-queuedetime": 1619483586836,
    "iothub-message-source": "Telemetry",
    "messageId": "205b09a9-b91a-4e7c-b1a8-8fc889a8dc47",
    "contentType": "application/json",
    "contentEncoding": "utf-8"
  }
}

```

28. Right click on the instance and select **Deactivate Graph Instance**



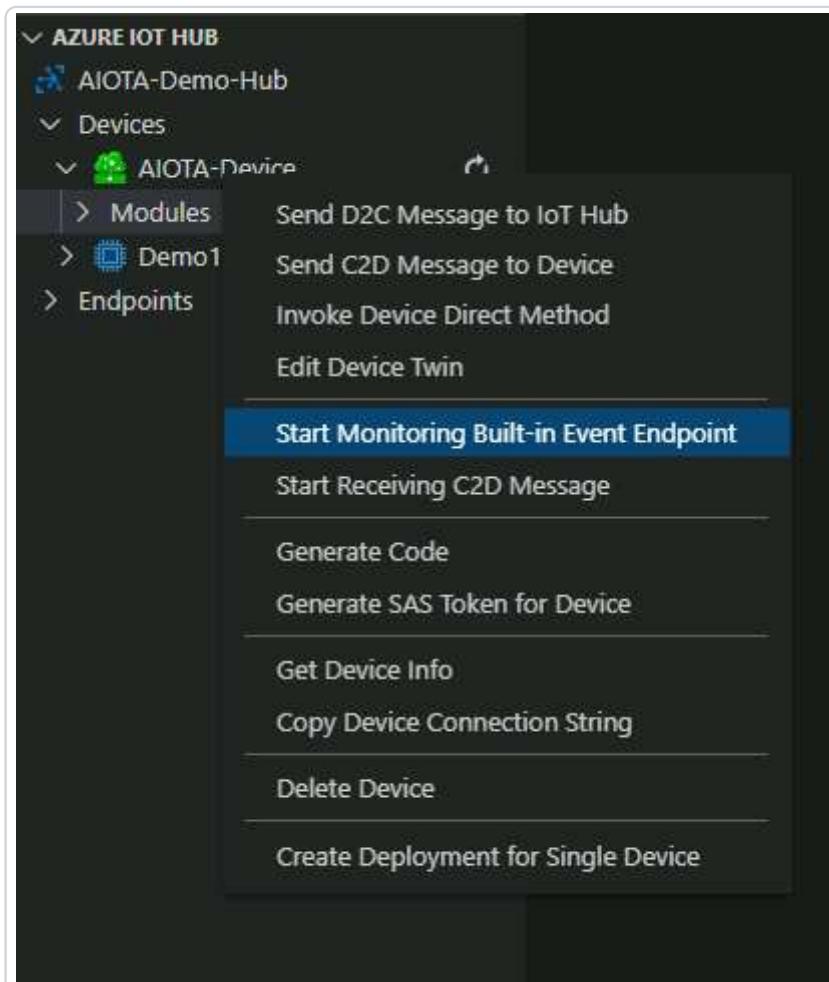
## Task 8: Connect Windows Video with Linux Inferencing

Recap:

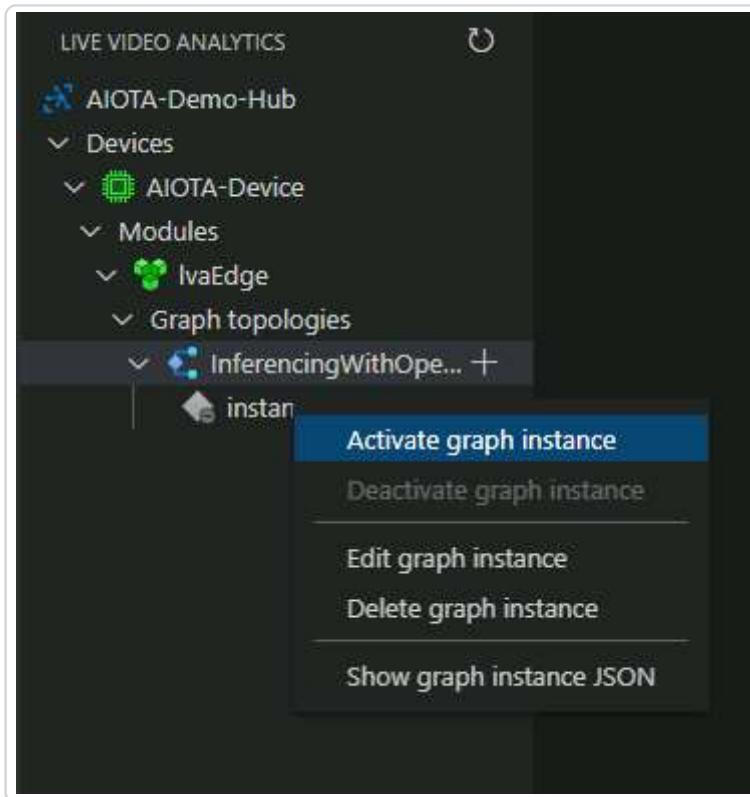
- We are able to see the video from windows side
- We were able to turn on/off inferencing from the Linux side

Next step: Connect the inferences with the Windows hub for a complete tool

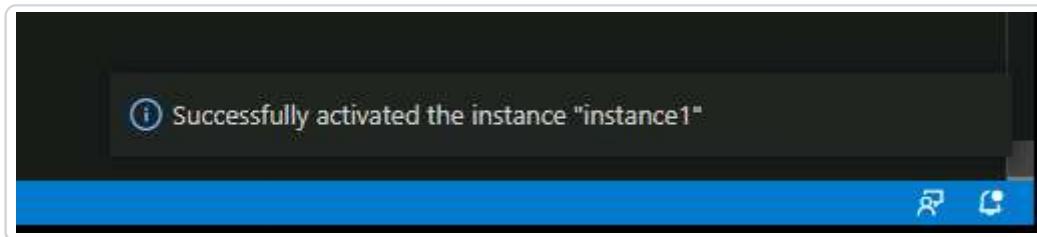
1. Navigate back to the **Azure IoT Hub** in Visual Studio >> **Device > IoT Edge Device** right click and select **Start Monitoring Built-in Event Endpoint**



2. Navigate back to the live video analytics tab in visual studio, and right click on your newly created instance and select **Activate Graph Instance**



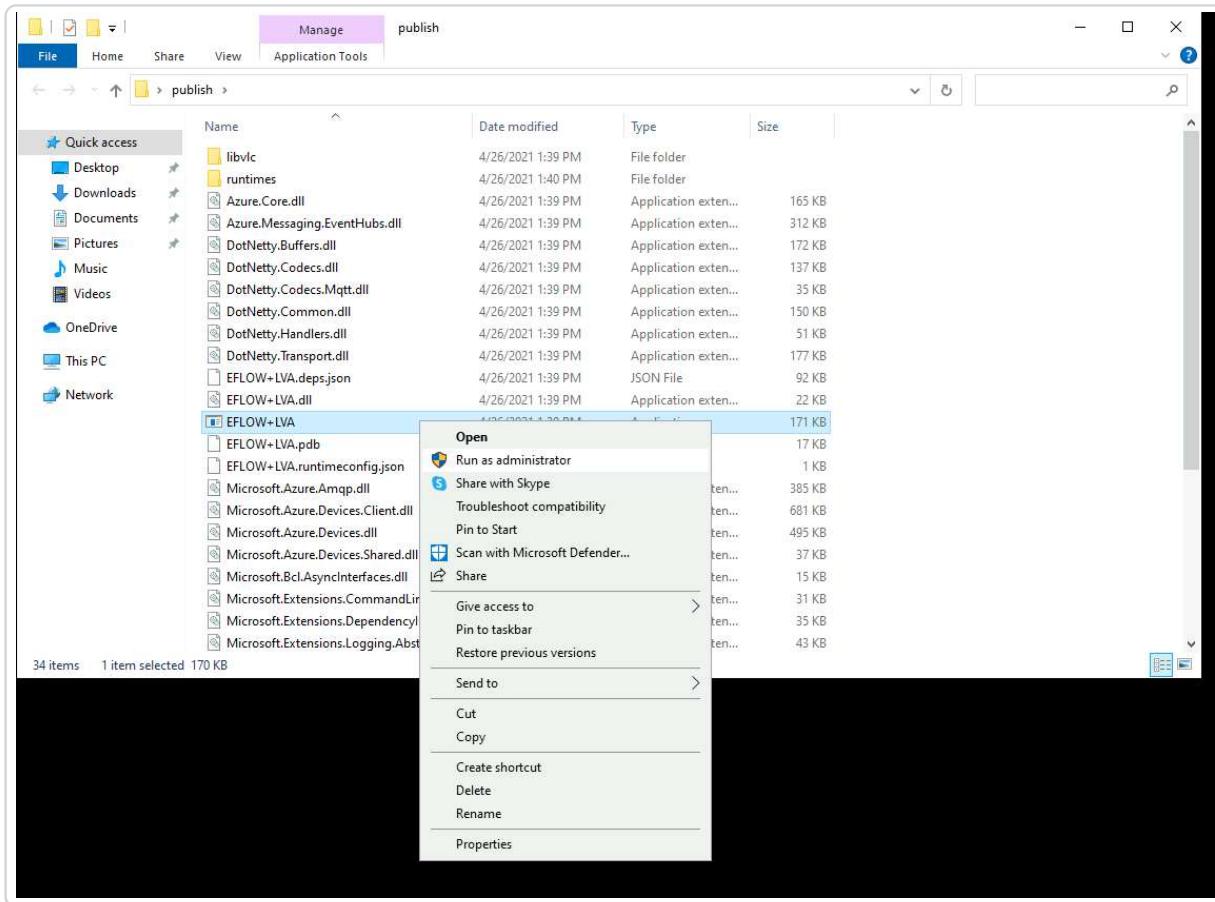
3. You should receive a notification that you have successfully activated the instance success-message



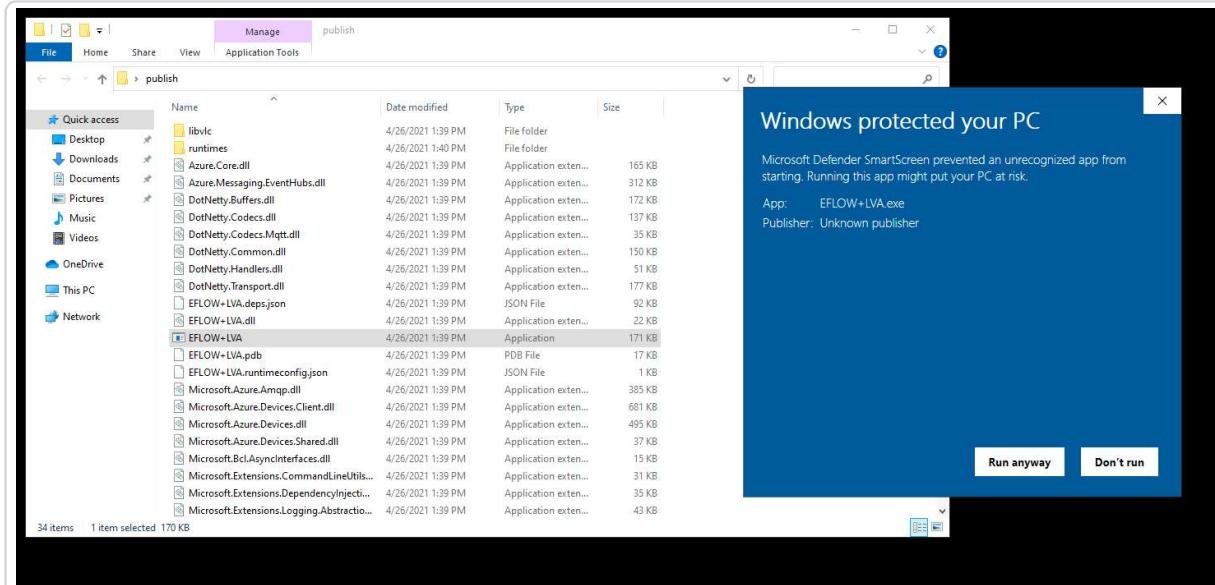
4. In the output window you will now see the inferencing occurring for each bounded box of time. output

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
log: [
  {
    "value": "vehicle",
    "confidence": 0.78586924
  },
  "box": {
    "l": 0.24342363,
    "t": 0.08611458,
    "w": 0.05972965,
    "h": 0.06487141
  }
},
{
  "type": "entity",
  "subtype": "vehicleDetection",
  "entity": {
    "tag": {
      "value": "vehicle",
      "confidence": 0.55225724
    },
    "box": {
      "l": 0.3998469,
      "t": 0.23444837,
      "w": 0.03865391,
      "h": 0.06640679
    }
  }
},
"properties": {
  "topic": "/subscriptions/db8411db-bdc3-47bc-b51e-9a687b462c43/resourceGroups/EFLOW/providers/microsoft.media/mediaservices/lvaeflowcowdymhbulpvu",
  "subject": "/graphInstances/instance1/processors/httpExtension",
  "eventType": "Microsoft.Media.Graph.Analytics.Inference",
  "eventtime": "2021-04-27T00:33:06.631Z",
  "dataVersion": "1.0"
},
"systemProperties": {
  "iothub-connection-device-id": "AIOTA-Device",
  "iothub-connection-module-id": "lvaEdge",
  "iothub-connection-auth-method": "{\"scope\":\"module\",\"type\":\"sas\",\"issuer\":\"iothub\",\"acceptingIpFilterRule\":null}",
  "iothub-connection-auth-generation-id": "637550401601814078",
  "iothub-queuedetime": 1619483586836,
  "iothub-message-source": "Telemetry",
  "messageId": "205b09a9-b91a-4e7c-b1a8-8fc889a8dc47",
  "contentType": "application/json",
  "contentEncoding": "utf-8"
}
}
```

5. Navigate to the **publish** folder > select the **EFLOW+LVA** application and run it as an administrator



6. Select More info > Run anyway in case Microsoft Defender SmartScreen prevents the app from running.



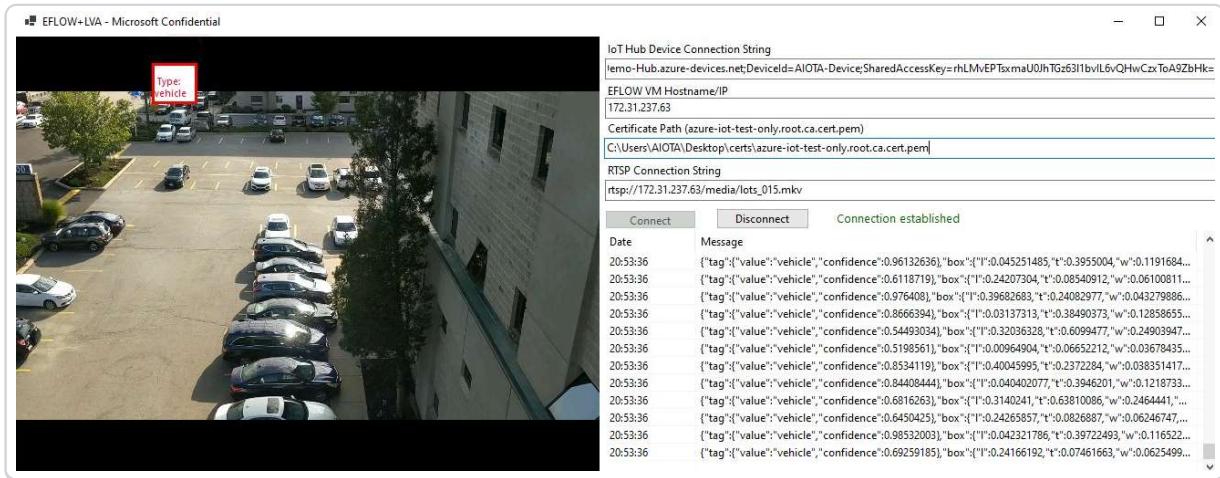
7. Fill in the remaining parameters:

**Setting**

**Value**

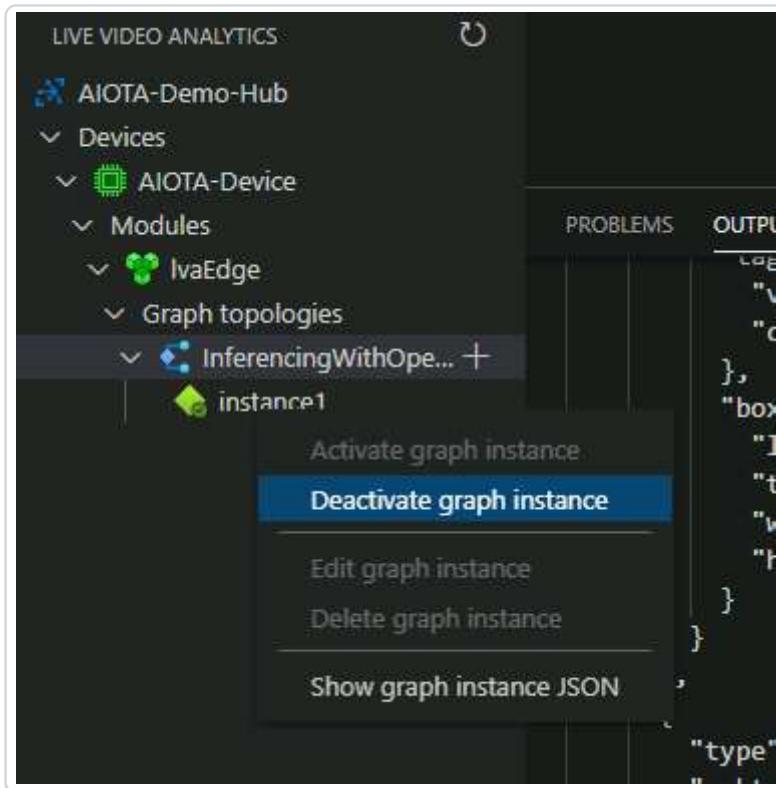
| Setting   | Value   |
|---|---|
| IoT Hub Device Connection String                        | Azure Portal >> IoT Hub >> IoT Device >> Select Device ID >> Copy Primary Connection String |
| EFLOW VM Hostname/IP                                    | Insert EFLOW VM IP  |
| Certificate Path (azure-iot-test-only.root.ca.cert.pem) | [Enter the path to the certificates folder]\azure-iot-test-only.root.ca.cert.pem            |
| RTSP Connection String                                  | rtsp://[EFLOW VM IP]/media/lot_015.mkv  |

8. Select *connect* - if the connection did not work - please see error message.



9. When you are done, select *disconnect* on the EFLOW + LVA Application.

10. Navigate back to the live video analytics tab in visual studio, and right click on the instance and select **Deactivate Graph Instance**.



## Exercise 4: Clean Up

### Task 1: Delete resources

When you're done using the virtual network and VM, delete the resource group and all of the resources it contains:

1. Search for and select **myResourceGroup**.
2. Select **Delete resource group**.
3. Enter **myResourceGroup** for **TYPE THE RESOURCE GROUP NAME** and select **Delete**.