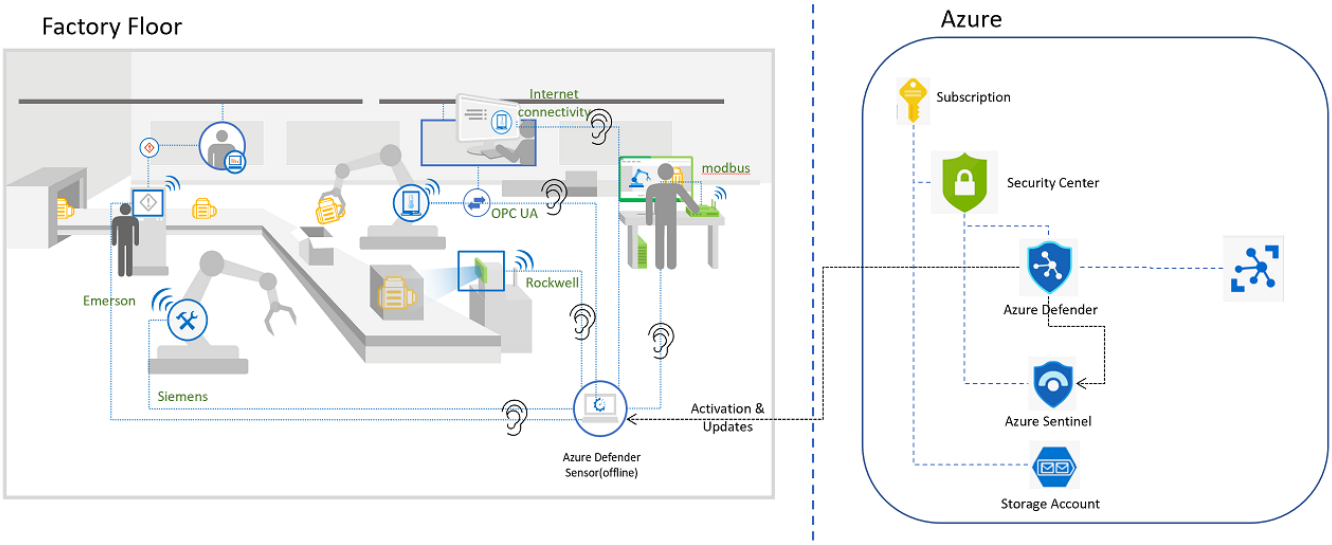
Internet of Things - Microsoft Defender for IoT HOL

Before starting this Lab make sure you completed the steps specified in the prerequisites file in this repository.

Architecture Diagram

During this workshop we will be focusing on setting up our Microsoft Defender for IoT sensors, for online alerts and also for offline scenarios. You will learn how to configure your environment, assess the results, and integrate with SIEM systems like Microsoft Sentinel. This Hands-on-Lab (HOL) will be focus on securing your facilities. It will cover brownfield and greenfield devices (currently not part of the HOL). The scenario below is one of many you would apply these lessons to, other scenarios are Oil, Gas, Utility, and Energy companies.



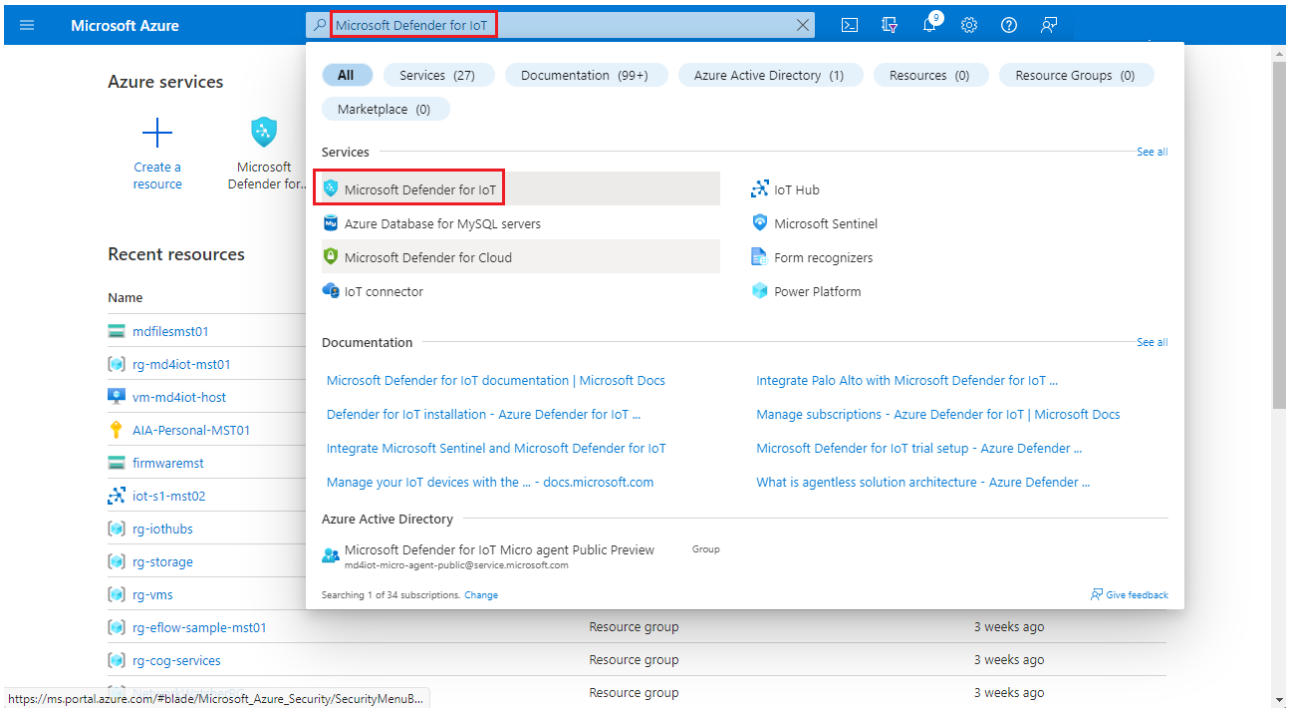
Content:

Exercise #1: Enabling Defender

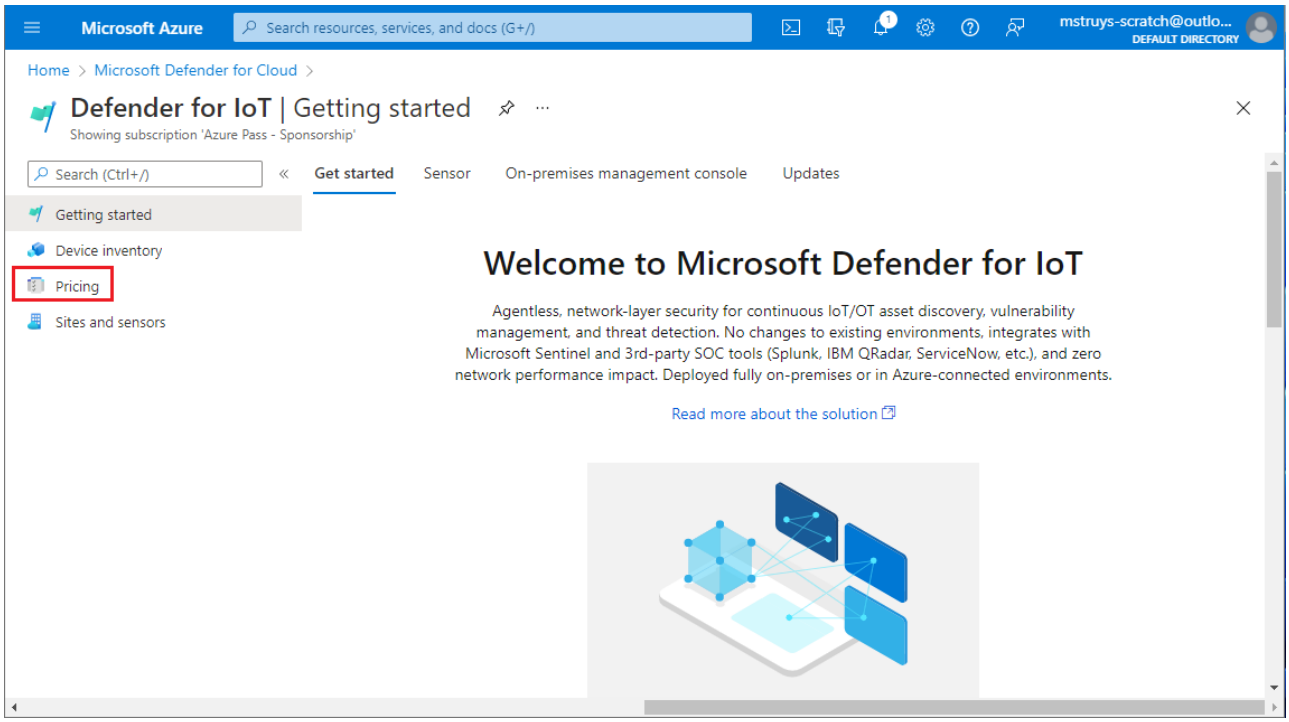
Task 1: Enabling Microsoft Defender for IoT

You will execute this task on your physical machine, not on the Virtual Machine that you will use later in this HOL to host your Microsoft Defender for IoT sensors.

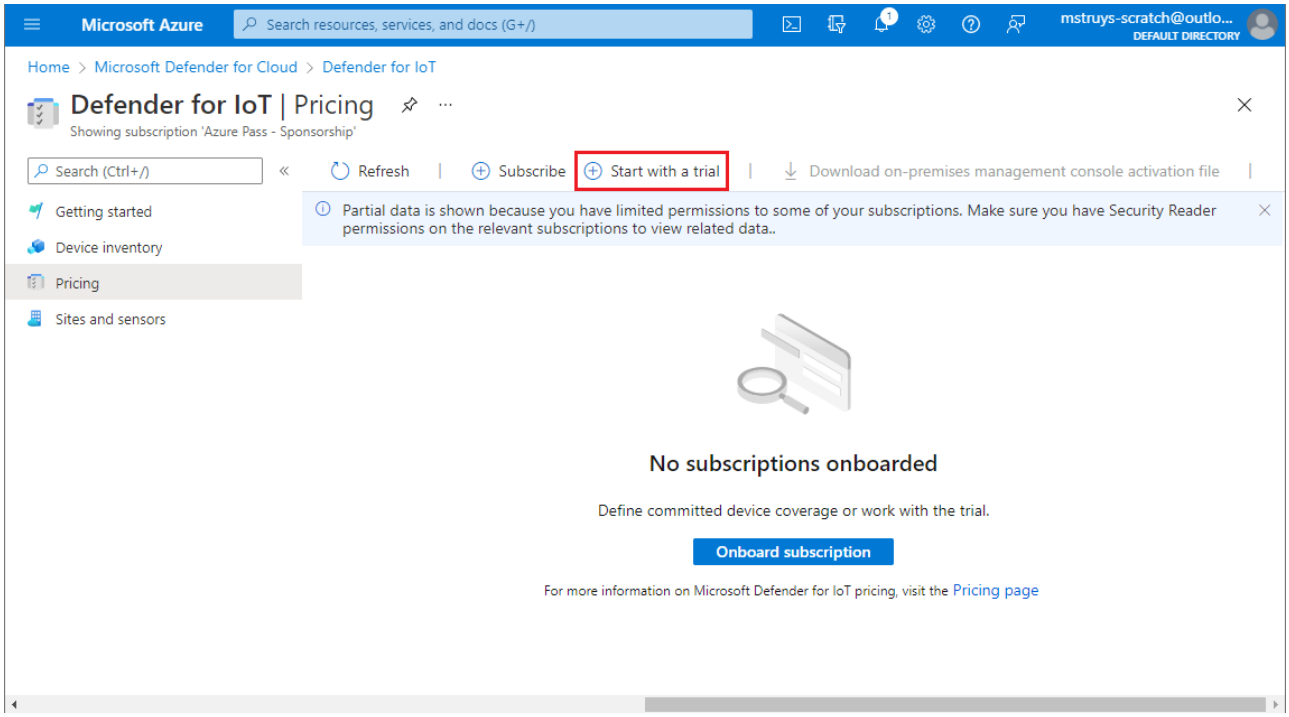
1. In the Azure Portal, search for **Microsoft Defender for IoT**. Select **Microsoft Defender for IoT** in the popup window, to open the Microsoft Defender for IoT Page.



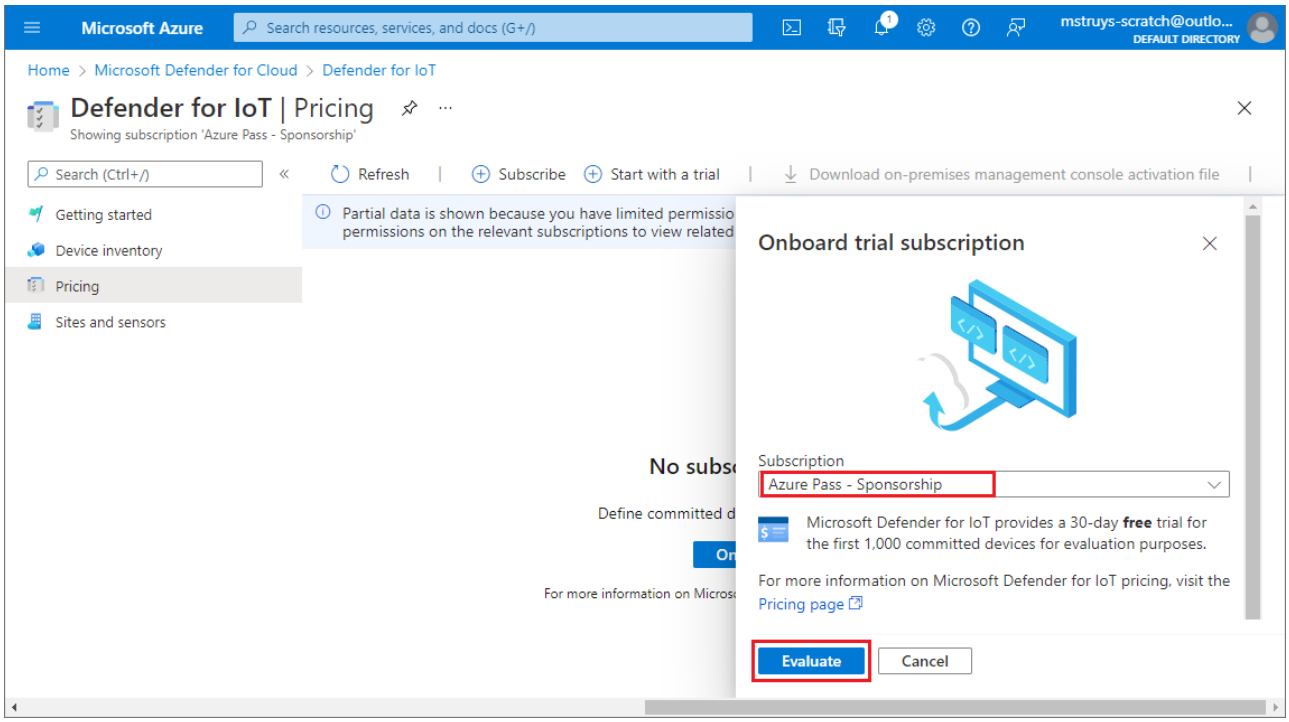
2. On the Defender for IoT page, in the **Getting Started** section, select **pricing**.



3. On the **Pricing** page, select **Start with a trial**.



4. In the popup screen leave all defaults (make sure you are using the same subscription you have been using for this lab) and click **Evaluate**, followed by **Confirm**.



You now have a valid Microsoft Defender for IoT Trial with 1000 committed devices. These devices represent all those equipments/sensors connected to your network in the facility you are analyzing. This configuration allows you for a 30 days trial for free.

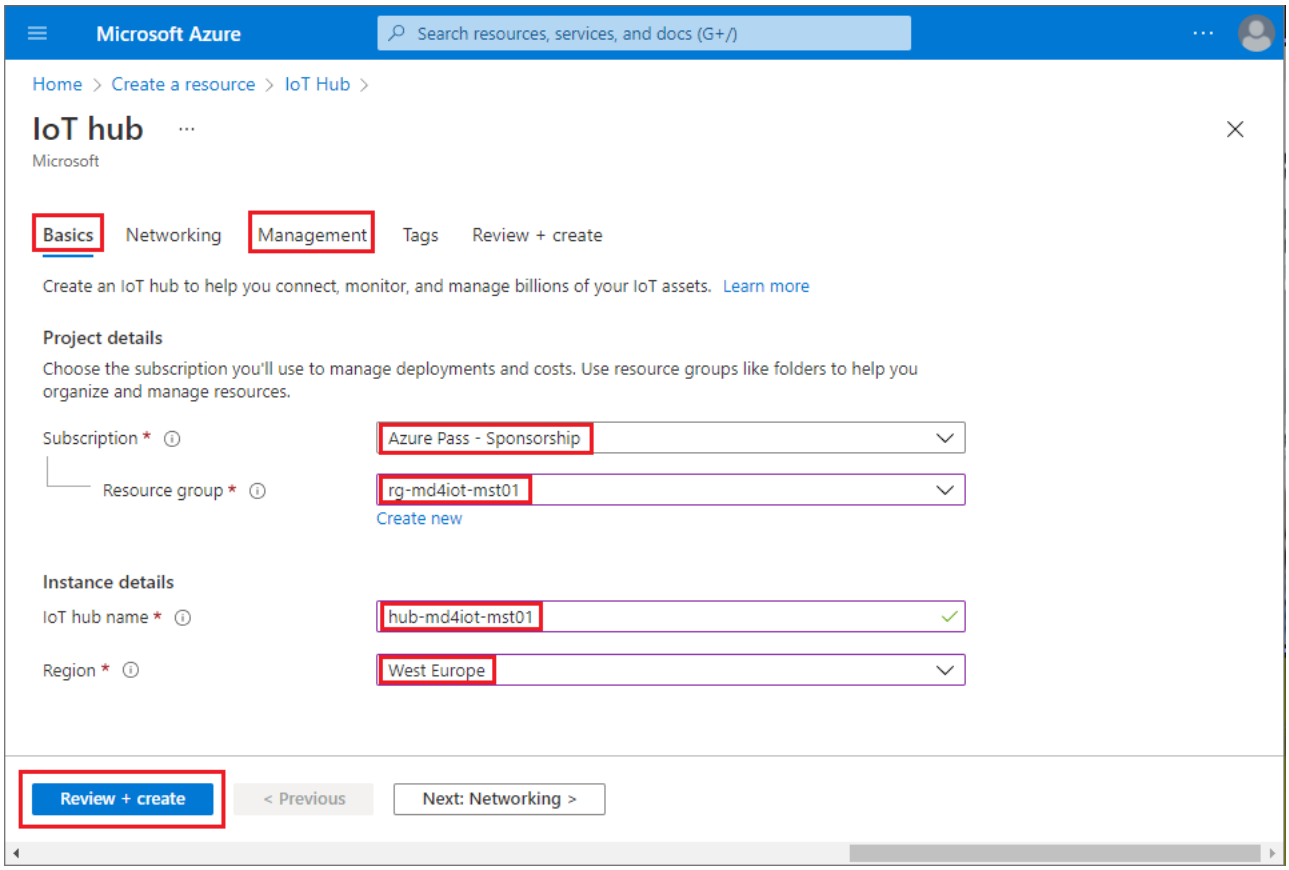
Task 2: Create an IoT Hub:

During this HOL you will work both with an online sensor and an offline sensor. The offline sensor can operate completely disconnected, but the online sensor needs to be connected to an Azure IoT Hub. Before onboarding your sensors you will create an IoT Hub for your online sensor to connect to. You will execute this task on your physical machine, not in the Virtual Machine that we will use later in this HOL to host your Microsoft Defender for IoT sensors.

1. Go to the resource group you created for this training. In the Overview panel, click on **+ Create** and type **IoT Hub** in the search box, then click **Create**.

2. In the next screen you will ask to fill the **Basics** tab:

    - **Subscription**: Select the Subscription you are woking on.
    - **Resource Group**: Should be the resource group created in previous step.
    - **IoT Hub Name**: hub-md4iot+SUFFIX
    - **Region**: A region close to your physical location (e.g. West Europe).



3. Next, click on **Management** tab and make sure that **S1:Standard Tier** is selected in the **Pricing and scale tier** section.

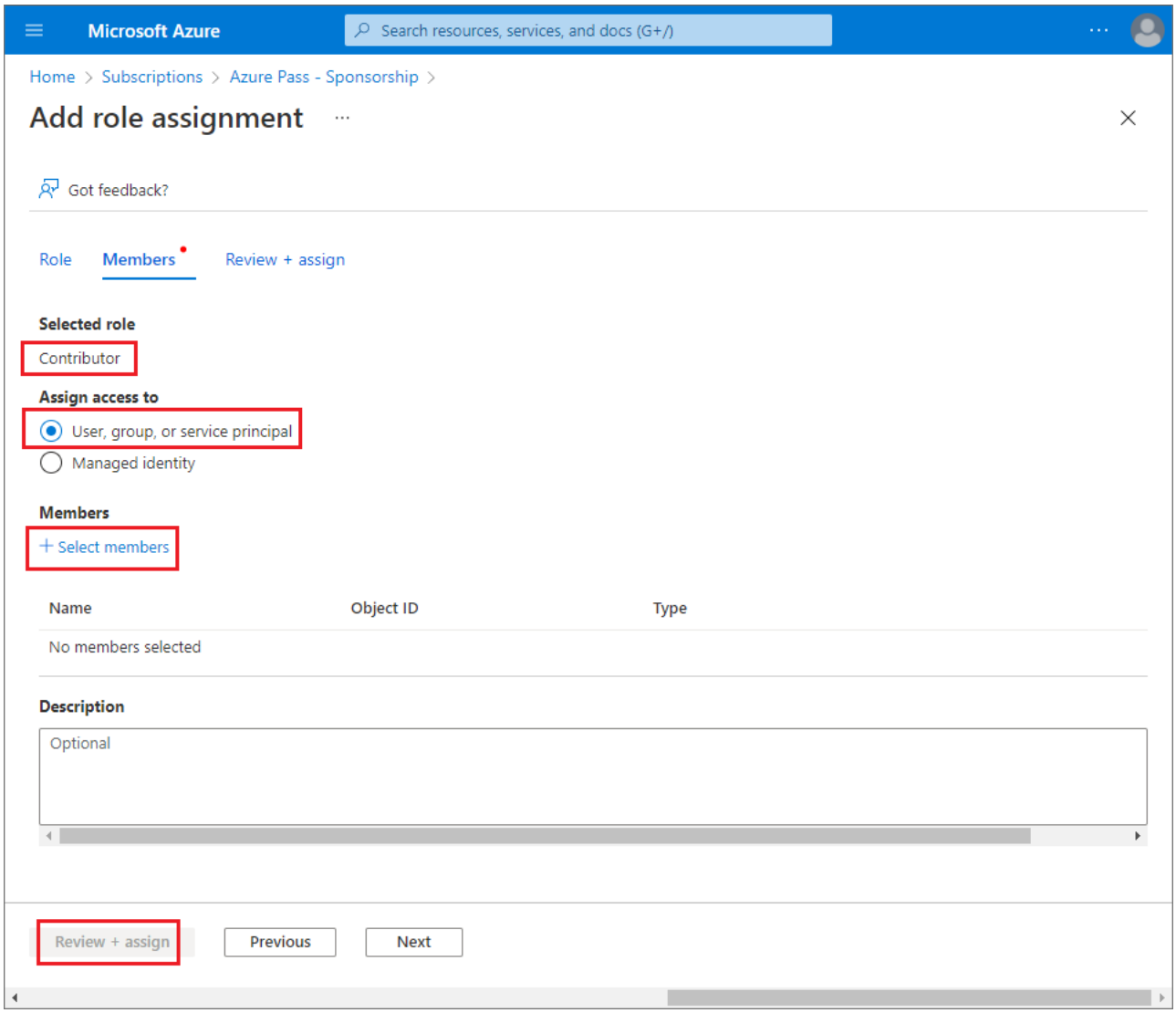
4. Finally, click **Review + create**, once validation is completed, click **Create**.

5. While the IoT Hub is creating , in the Azure Portal look for the Subscription, click on **Access Control(IAM)**, then select **+ Add**. A new window will open on your right, select the following:

    - **Role**: Contributor
    - **Assign access to**: User, group or service principal
    - **Select members**: search for the email you are using in this subscription. Select that email and click **Select**.

6. Click **Review + assign** and again **Review + assign**.

    Microsoft Sentinel will need this access to collect the alerts in further exercises when your sensor is online.

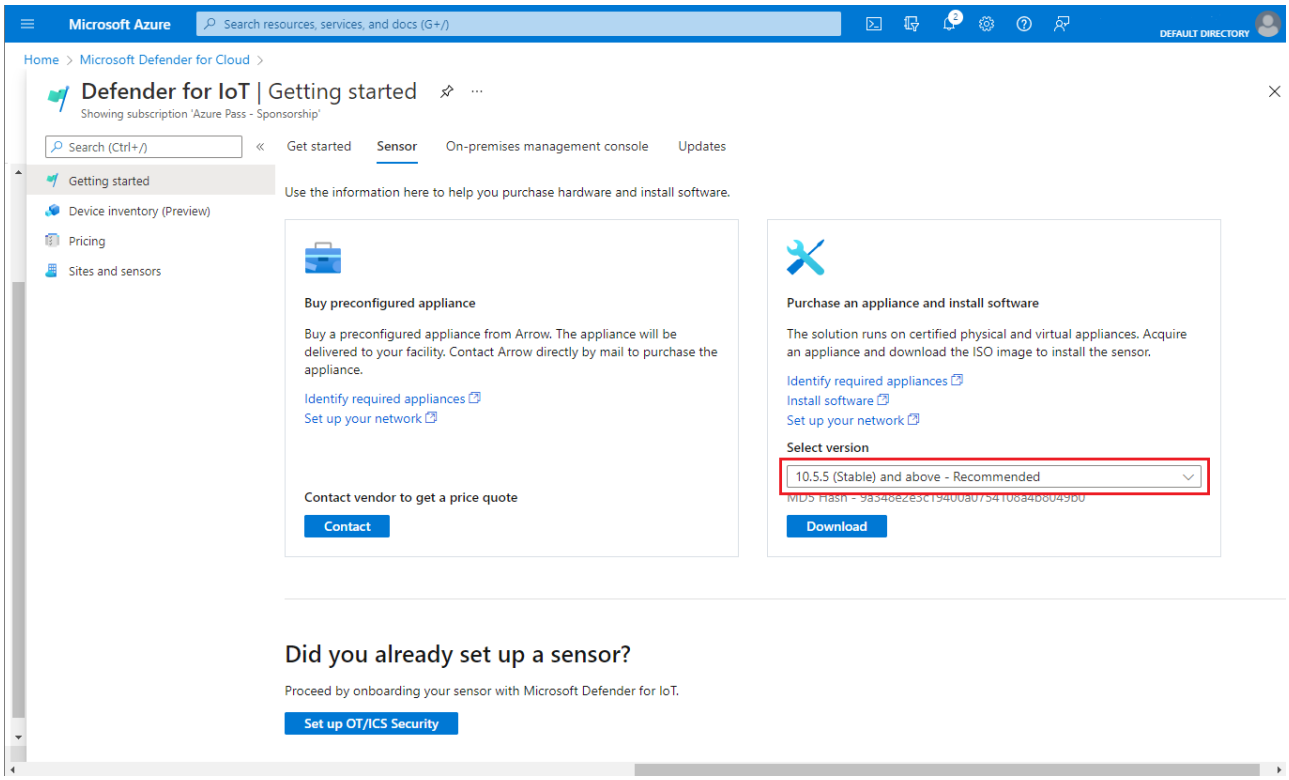


Task 3: Onboarding sensors

For the hands-on lab we will work with two type of sensors, an offline sensor that does not need to be connected to the public Internet and an online sensor that is connected to Azure. In the next steps we will begin by onboarding the offline sensor. You will execute most of this task on your physical machine, not in the Virtual Machine that we will use later in this HOL to host your Microsoft Defender for IoT sensors.

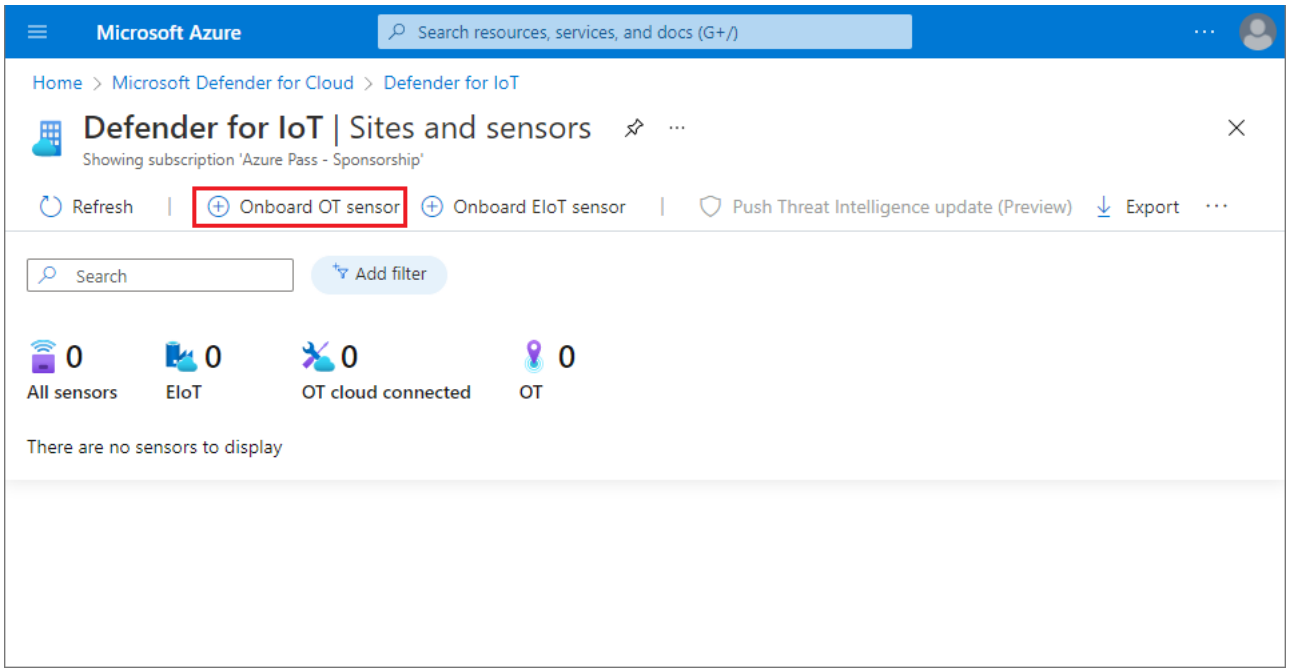
1. Go back to Microsoft Defender for IoT to create the sensors. You can find it by searching for **Microsoft Defender for IoT** in the Azure Portal.

2. You can download the latest sensor iso image here (from the **Sensor** section). You ***already did this step*** as a prerequisite in the **Before HOL Section**. The ISO file is already available in your VM so you don't have to download it to your VM right now. However, you need to know where to find the ISO file. In the **Getting Started** section, select **Sensor**, then pick the **10.5.5 (Stable) and above - Recommended** version. To download it, you would click **Download**. This results in the ISO file being downloaded to your physical device.

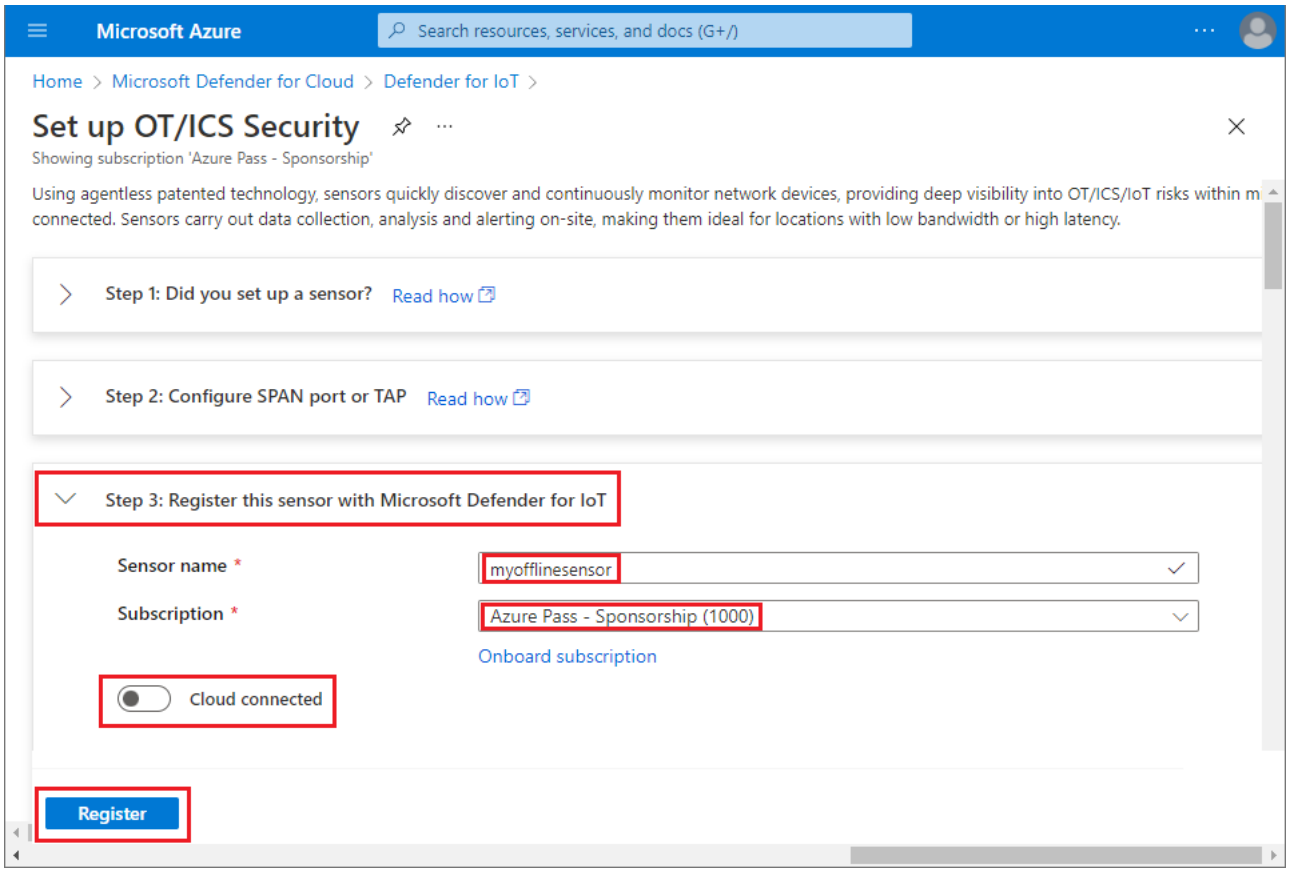


> **NOTE:** At this moment, you might see a Window asking for contact details. You don't have to provide your contact details. Just go to the bottom of the windows and click on **Continue without submitting**.

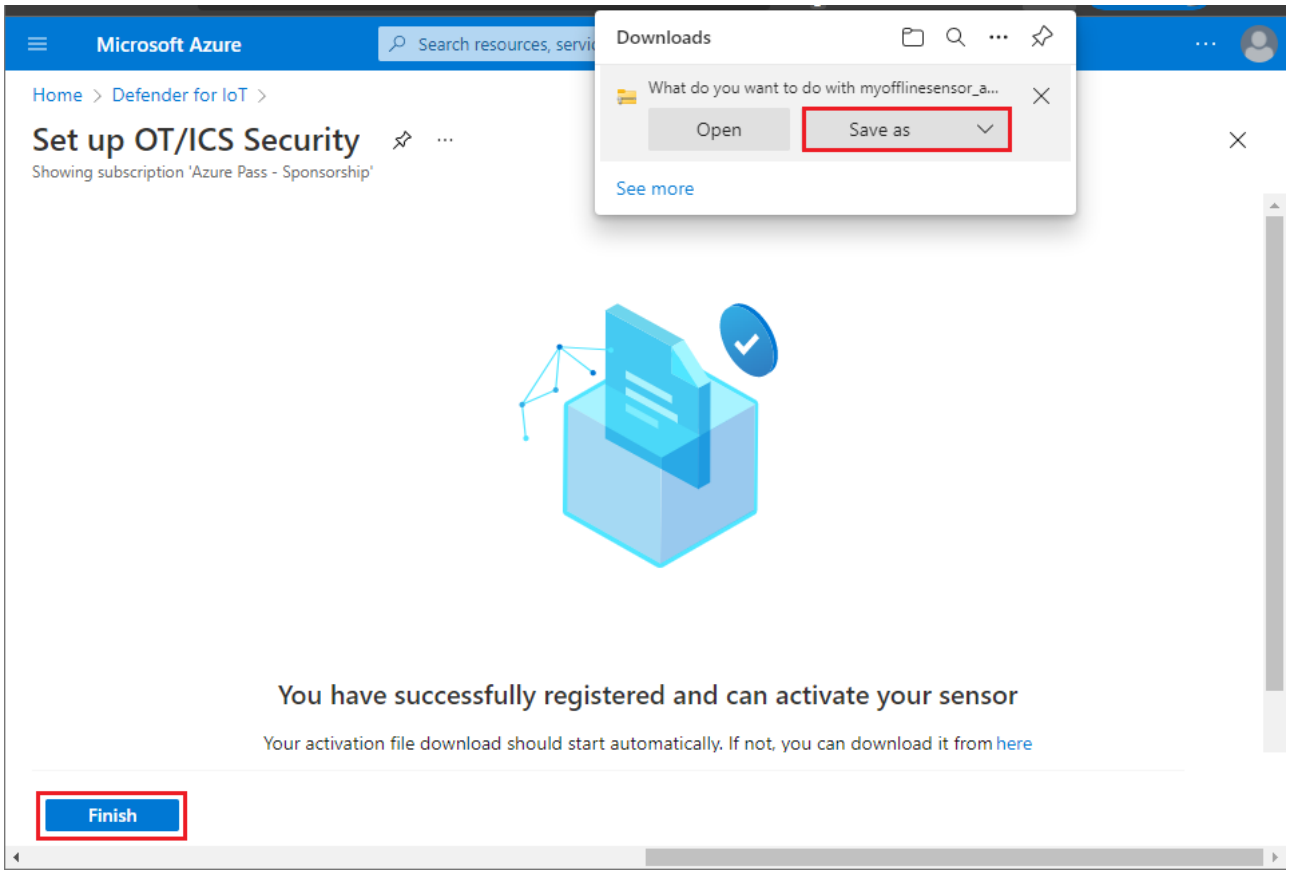
3. Next go to **Sites and Sensors** and click on **+ Onboard OT sensor**.



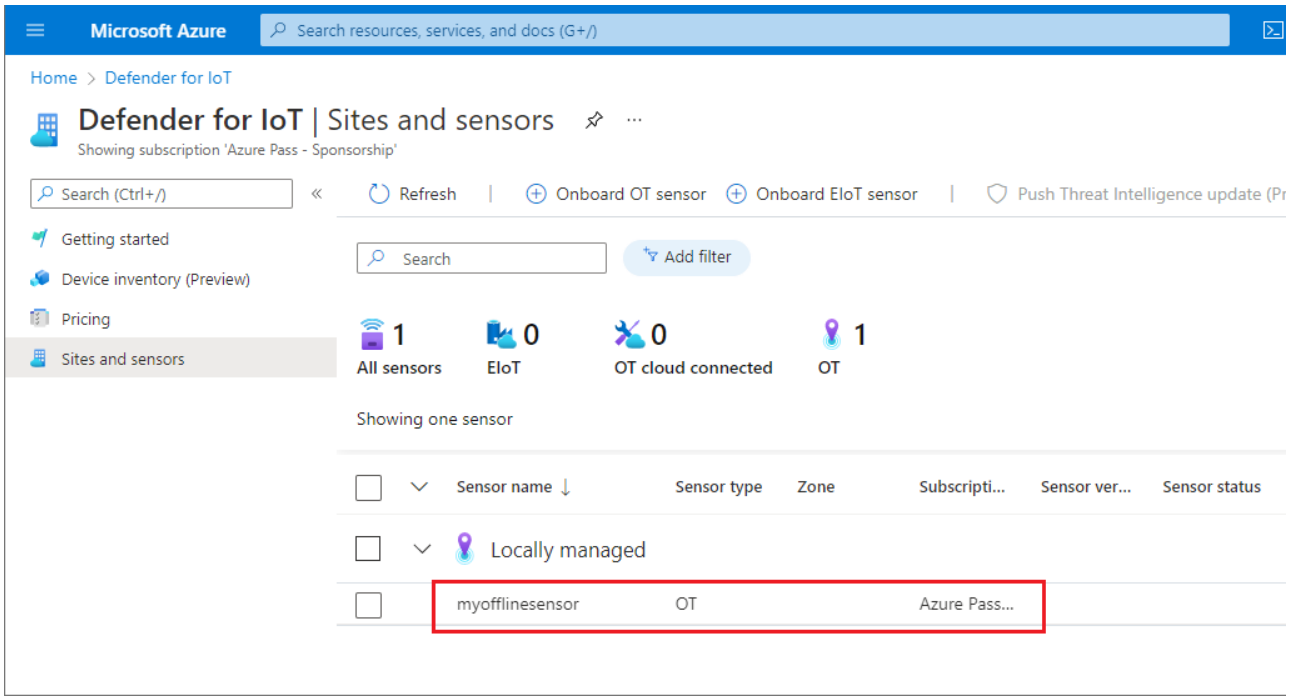
4. In the Setup OT/ICS Security screen, expand step 3 and set the following values: Sensor name = **myofflinesensor**, select your subscription and disable **Cloud Connected**. Click **Register**.



5. In the next step, you will be prompted to save the sensor activation file. Save it with the default filename and click **Finish**.



6. You should see your new sensor onboarded, locally managed, in your list of sensors as shown below.

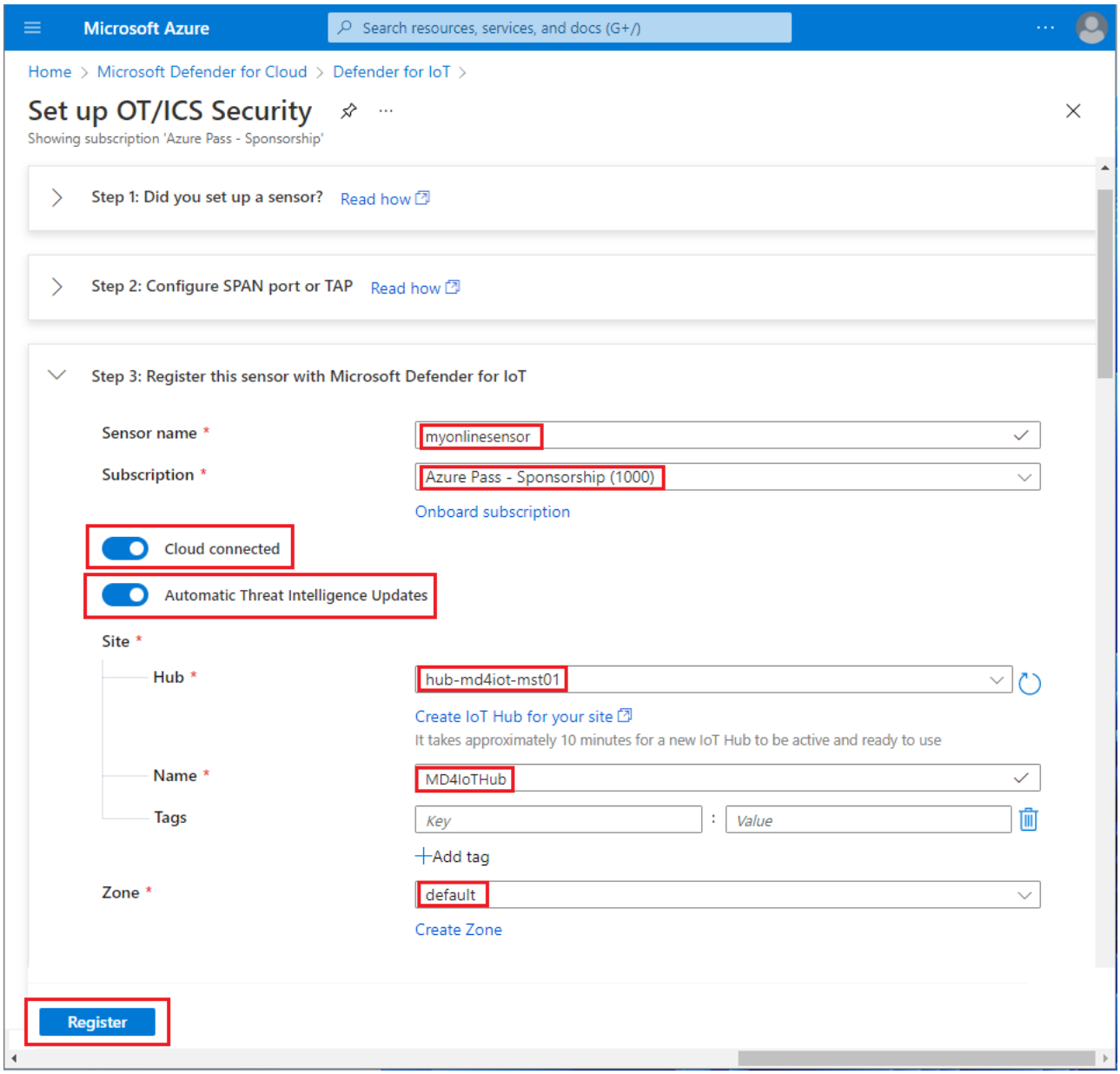


7. Now, we will create another sensor. This will be an online sensor. Click on **+ Onboard OT sensor**, in the next screen input the following information:

    - **Sensor name**: myonlinesensor
    - **Subscription**: Select the subscription you are using for this lab.
    - **Cloud Connected**: Enabled (= default).
    - **Automatic Threat Intelligence Updates (Preview)**: Enabled (= default).

    Site Section

    - **Hub**: Select the IoT Hub you created in the previous step.
    - **Name**: MD4IoTHub. Usually this name will represent the site you will be analyzing, such as *Plant I*.
    - **Zone**: Default.



8. Click **Register**.

9. In the next step, save the activation file and click **Finish**.

10. Check again your **Sites and sensors** section. You should now see both sensors onboarded.

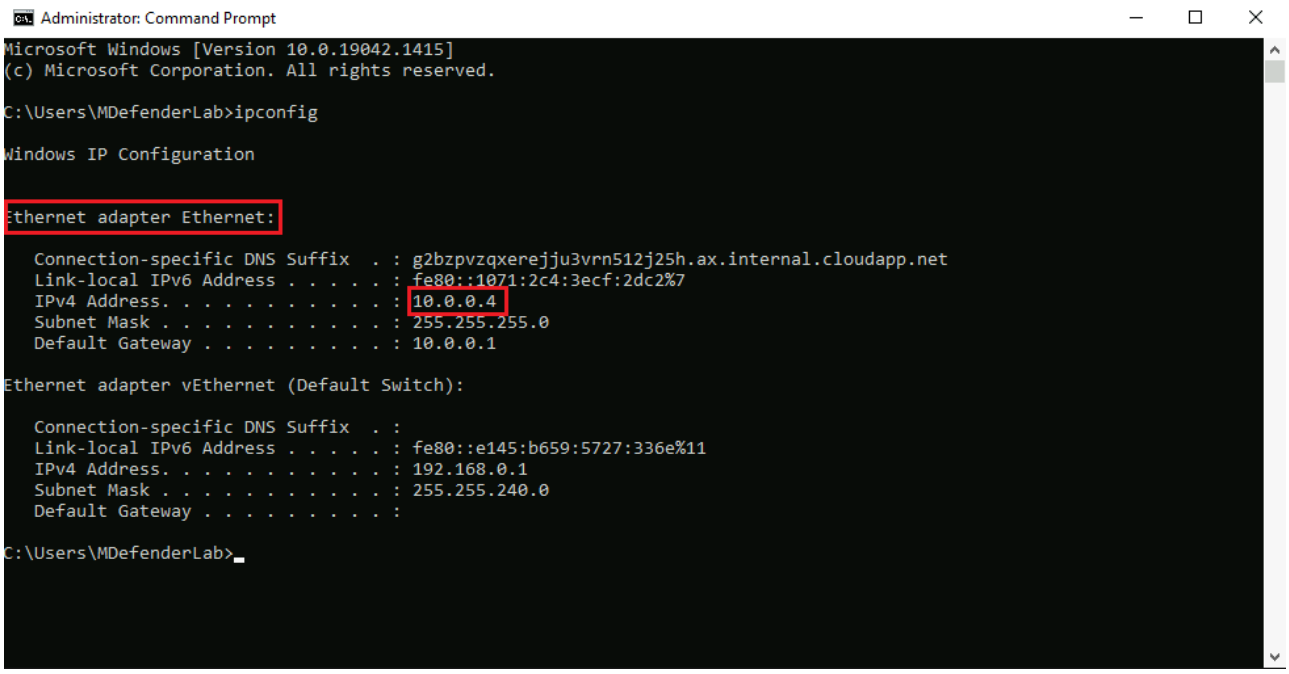
11. At this point you have 2 files downloaded locally (the activation files for your sensors). Since you are using RDP to connect to the Virtual Machine that will host your Microsoft Defender for IoT Sensor, you can simply copy the activation files and paste them in your VM using copy (ctrl-c) and paste (crtl-v).

Exercise #2: Setting up your offline sensor

During this exercise you will create a new nested Virtual Machine inside the Virtual Machine that you created as part of the prerequisites.

Task 1: Set up your nested Virtual Machine

1. On the Windows 10 Virtual machine created previously, login with with RDP if you have not done so before. Open a command prompt and run the command "ipconfig".



2. Take note of the IP address used on your Windows 10 Host's Ethernet Adapter. **NOTE: Ignore the (Default Switch)**

> **NOTE:** In this example, the Win10 host Ethernet Adapter is assigned an IP of 10.0.0.4, therefore we will use 192.168.0.0/24 as the network scope of the “NATSwitch”. If your primary adapter is already using 192.168.x.x, then use 172.27.0.0/24 for your “NATSwitch”.

3. Open a PowerShell prompt as an Administrator by searching for PowerShell and right-clicking to "Run as administrator".

4. Run the next two commands in the PowerShell window.

```
New-VMSwitch -SwitchName "NATSwitch" -SwitchType Internal
```

```
New-VMSwitch -SwitchName "MySwitch" -SwitchType Internal
```

5. Run the following command to store the network adapter information to a local variable.

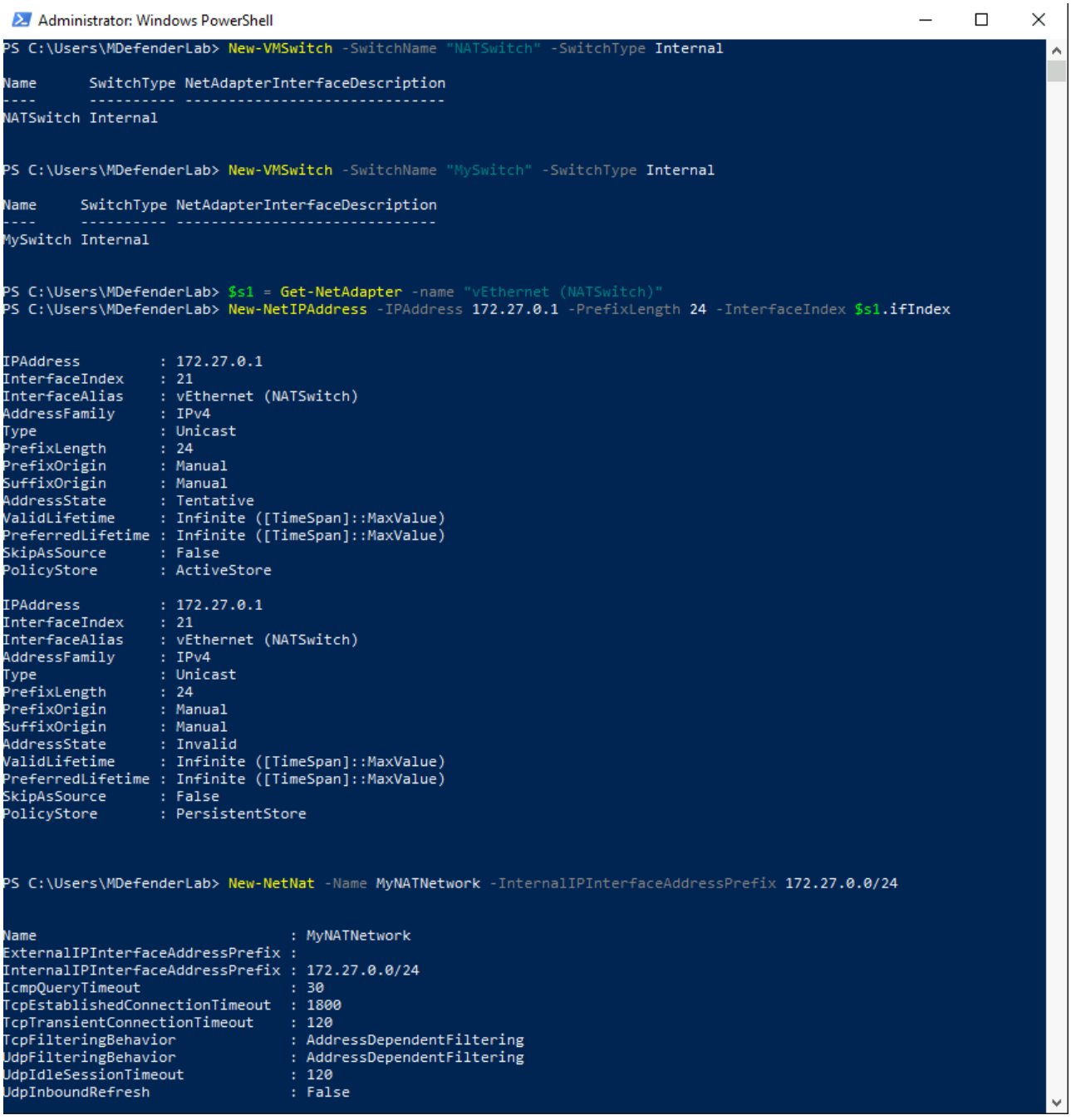
```
$s1 = Get-NetAdapter -name "vEthernet (NATSwitch)"
```

6. Assign an IP address to the NATSwitch (either 192.168.0.1 or 172.27.0.1) depending on your network address based on step 1.

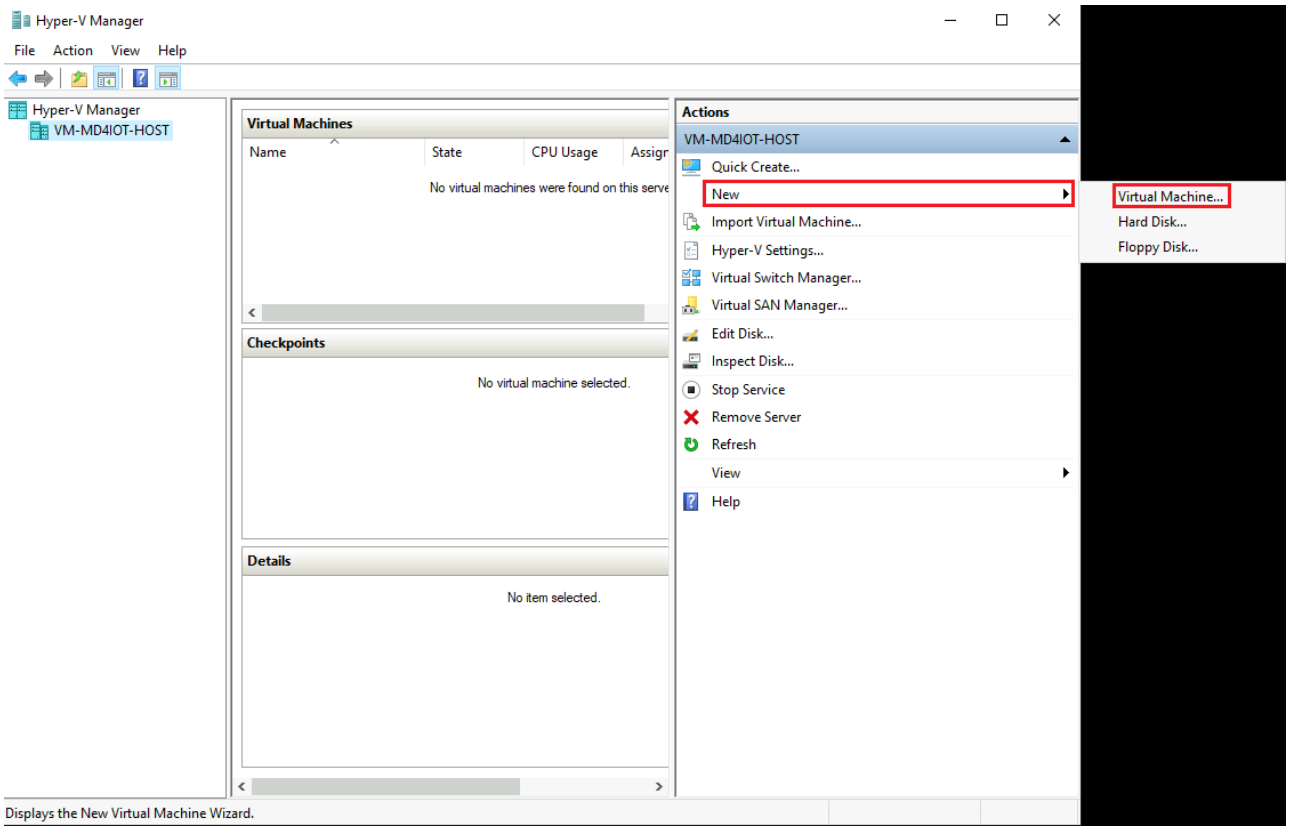
```
New-NetIPAddress -IPAddress 192.168.0.1 -PrefixLength 24 -InterfaceIndex
$s1.ifIndex
```

7. Create the new NAT network. Again, your IP address space will either be 192.168.0.0/24 or 172.27.0.0/24 depending on step 1.

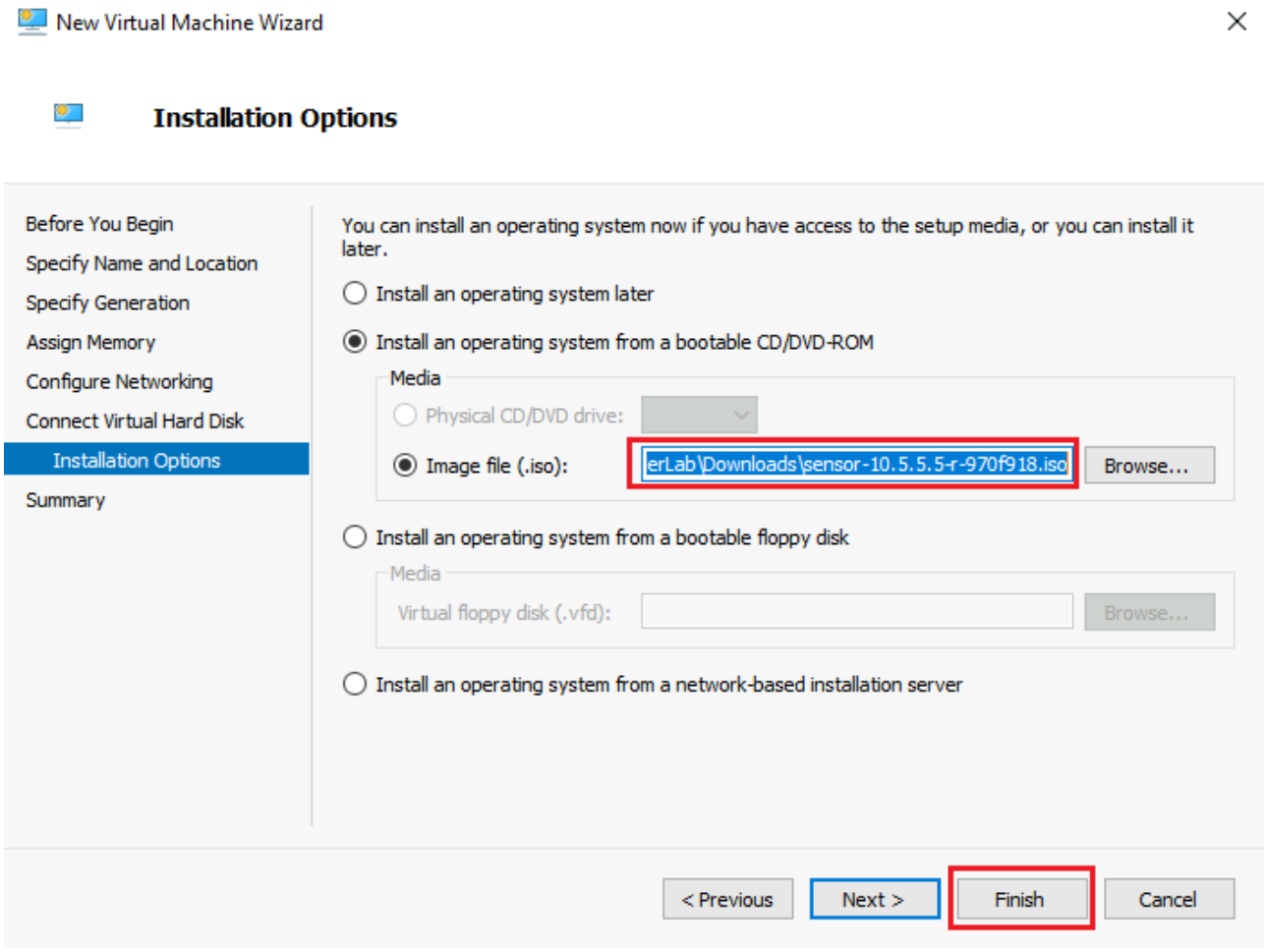
```
New-NetNat -Name MyNATnetwork -InternalIPInterfaceAddressPrefix
192.168.0.0/24
```



8. Inside the VM, in the windows search box, type **Hyper-V** and enter. This should open a new window with the Hyper-V console. Select **New** on the left side. This will show multiple options, select **Virtual Machine**.



- In the first tab, assign the name **md4iotsensoroffline** to your VM, then click **Next**
- **Specify Generation**, select **Generation 1**, click **Next** again.
- Change the memory to **8196MB**, click **Next**.
- **Configure Network** tab, select in **Connection**, **NATSwitch**, click **Next**.
- **Connect Virtual Hard Disk** tab, **Create a virtual hard disk** click **Next**.
- **Installation Options**, select **Install an operating system from a bootable CD/DVD-ROM** then select **Image file (.iso)** and browse to the Azure defender .iso file that you downloaded in the prerequisites. Click **Finish**



9. Right click on the Virtual machine that you just created, select **Settings** in the **Add Hardware** section and select **Network Adapter**, followed by clicking on **Add**. Now select the virtual switch created previously with the name **My Switch**, and click **Apply**. Increase the Processor number from **1** to **4** Virtual Processors, click **Apply** and click **Ok**.

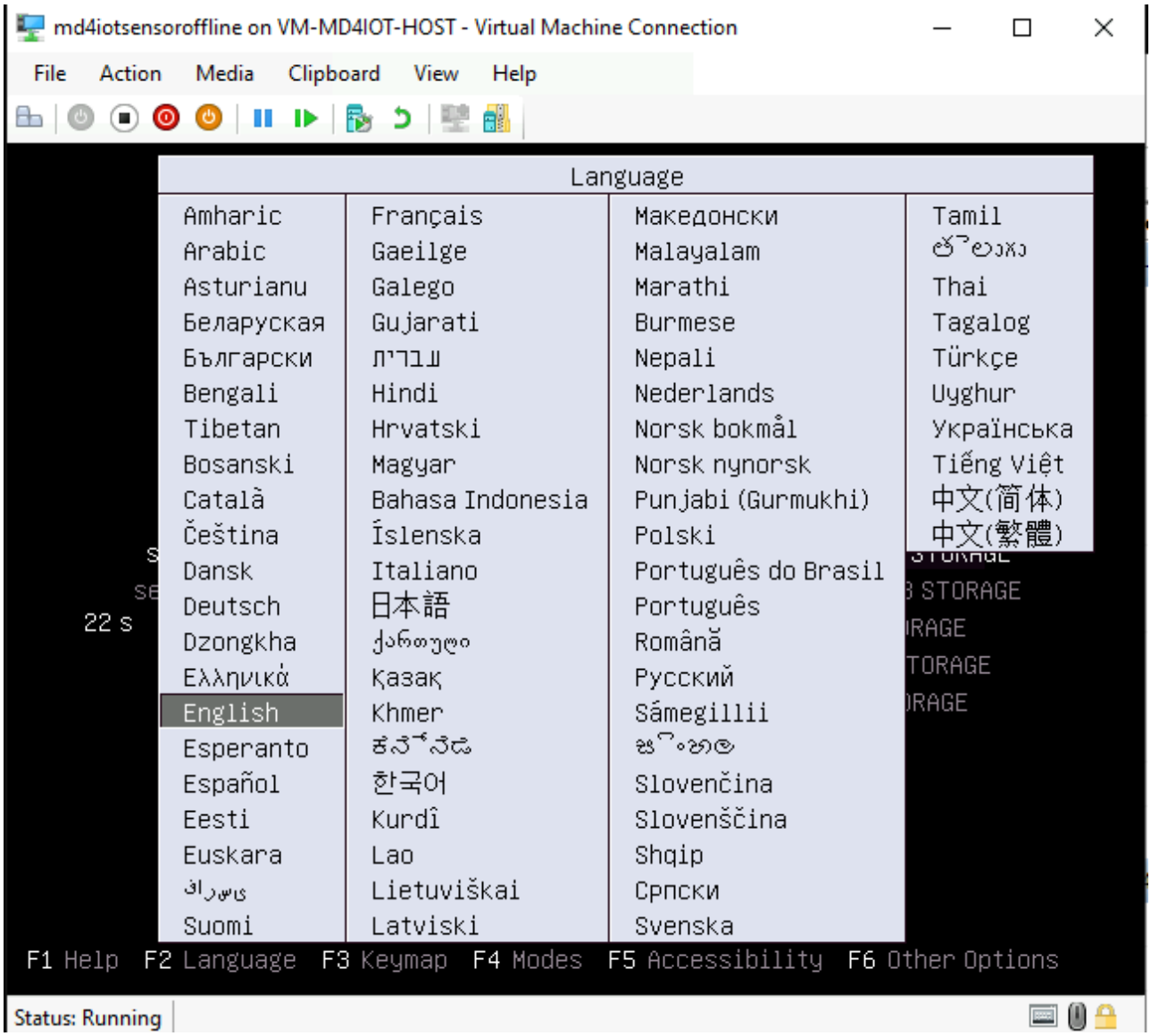
Task 2: Configure a Microsoft Defender for IoT offline sensor

During this task we will configure Azure Defender based on the IPs highlighted before, this first configuration will be based on an offline sensor.

1. In the Hyper-V Manager, find the **Connect...** in the lower right hand of the screen and click on it, and in the newly opened VM connection window click **Start**.

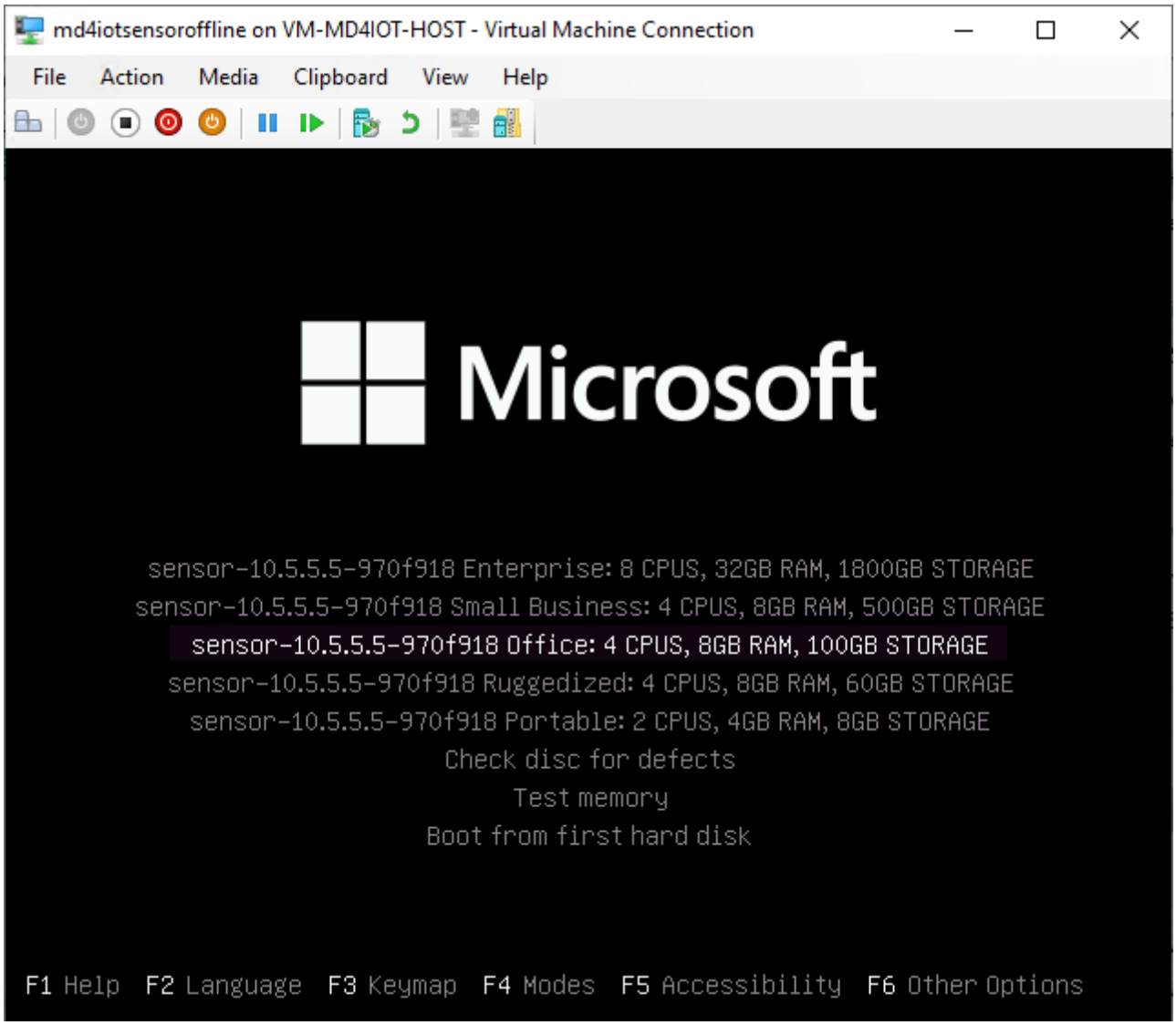
2. When you connect to the Ubuntu VM you should see the following screen to start the configuration process.

> **Note!**: If you don't see the screen below, your installation timed out or you pressed enter, selecting a different configuration by mistake, delete the virtual machine and start this task over. The timeout period is relatively short so make sure you connect immediately to the nested VM and select the language and the sensor type (in Task 2).



3. Press **Enter** for English.

4. Select the third option *(Office 4CPUs)* and press **Enter**.

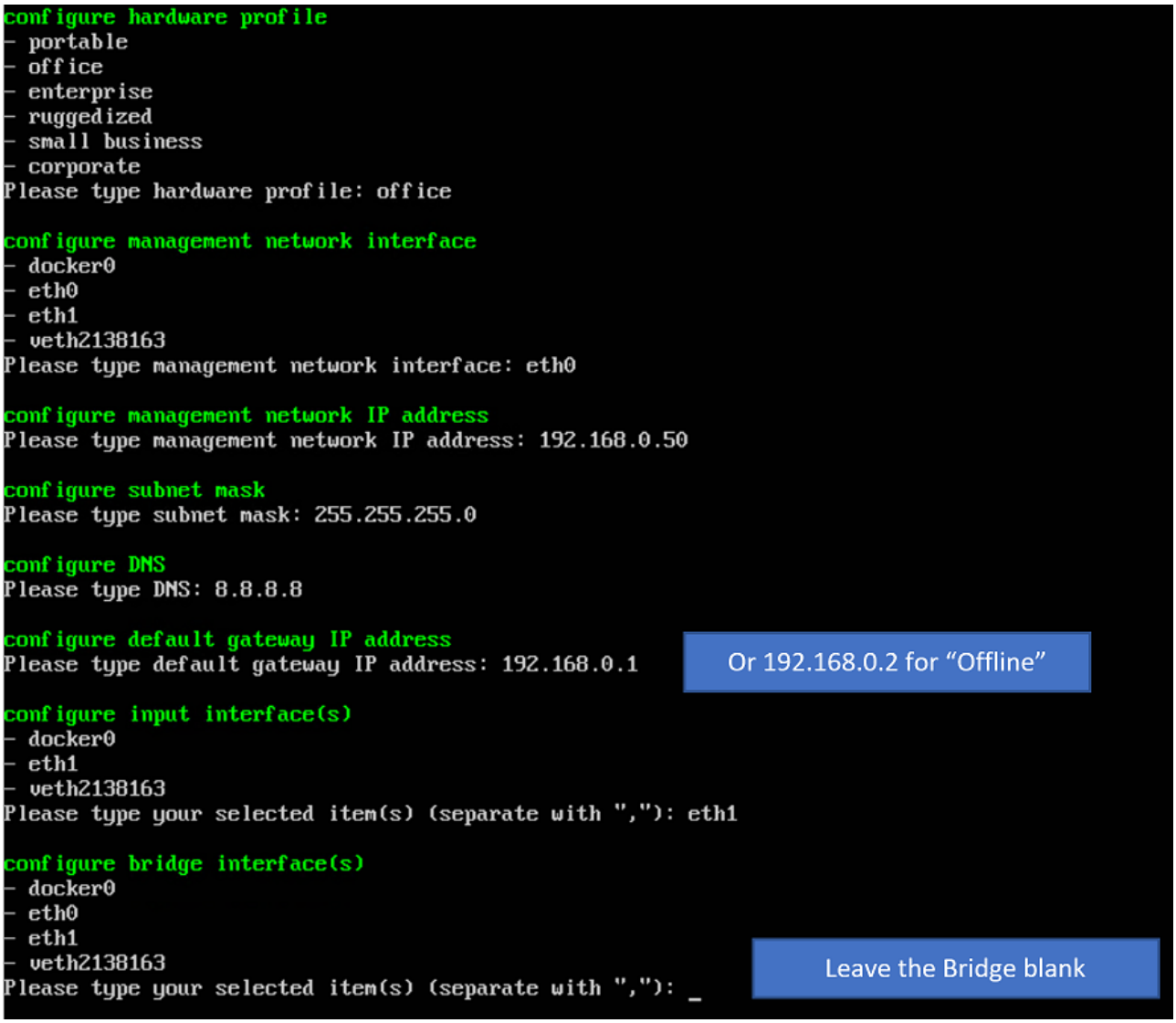


At this moment, the offline sensor will be installed (including its operating system). This installation takes some time, expect it to run for approximately 15 minutes.

5. As part of the installation process, you will be asked to provide some parameters, it is ***VERY IMPORTANT*** you paid attention to the previous task because you will use the network information you captured before. This information is unique to each Virtual Machine. So the following is an **EXAMPLE**.

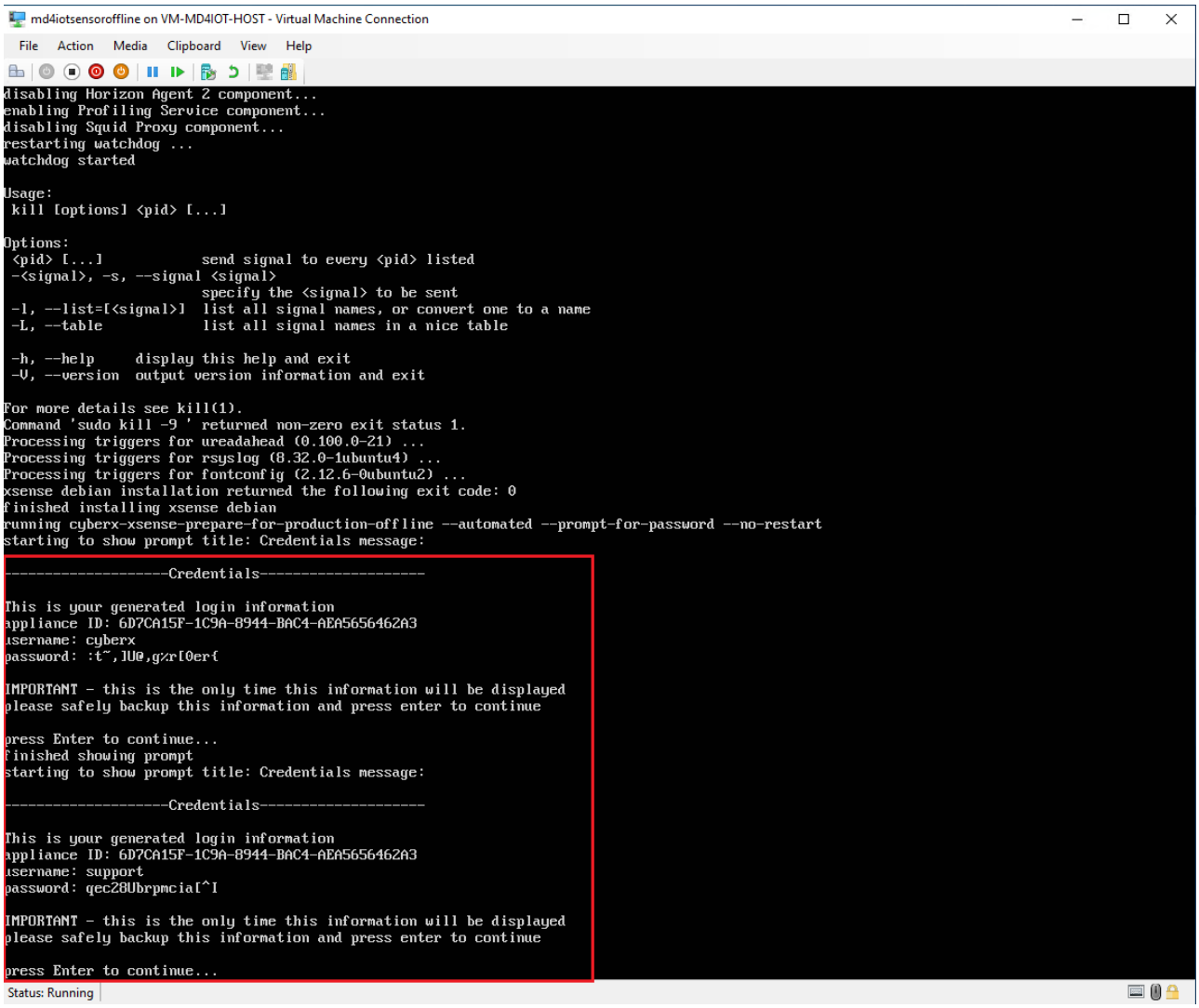
    - **configure hardware profile**: **office**, then press enter.
    - **Configure network interface**, type **eth0**
    - **Configure management network interface**: in this example we're using **192.168.0.50**, you will use one of the **Ipv4 Addresses** depending on your network scope from the previous task, either **192.168.0.50 or 172.27.0.50**. Click Enter to continue. ***Take a note of this IP you will need it later on***.
    - **Subnets mask**: **255.255.255.0** this will be the SAME for everyone.
    - **Configure DNS**: **8.8.8.8**
    - **Configure default gateway IP Address**: We are intentionaly mis-configuring this value to force the sensor in **offline** mode. Use either 192.168.0.**2** or 172.27.0.**2**.
    - **Configure input interface(s)**: **eth1**
    - **Configure bridge interface**: Just press Enter
    - Then type **Y** to apply the changes and click **Enter**.

Below, a ***sample*** screen, your parameters might be different.



Now the installation will continue running for another 10-15 minutes.

6. ***IMPORTANT STEP!!!*** Once the installation is complete, you will have the login information availabe in the screen **TAKE A SCREENSHOT!!** before continuing, press **Enter**. Now you will have the support account login information, again **TAKE THE SCREENSHOT!!** press **Enter** to continue. If you fail to capture the credentials, you will need to start over.



7. Once the installation finished you will ask to login, enter the credentials from previous step. In this screen you can also validate the IP, you will use that IP in your browser.

   Note: *At this stage your IPs should look similar to the example below. If you can't reach the portal validate the IPs. If you restarted your VM there is a chance your IPs changed so you will need to go back and reconfigure them, if that is the case follow the troubleshooting guidance below.*

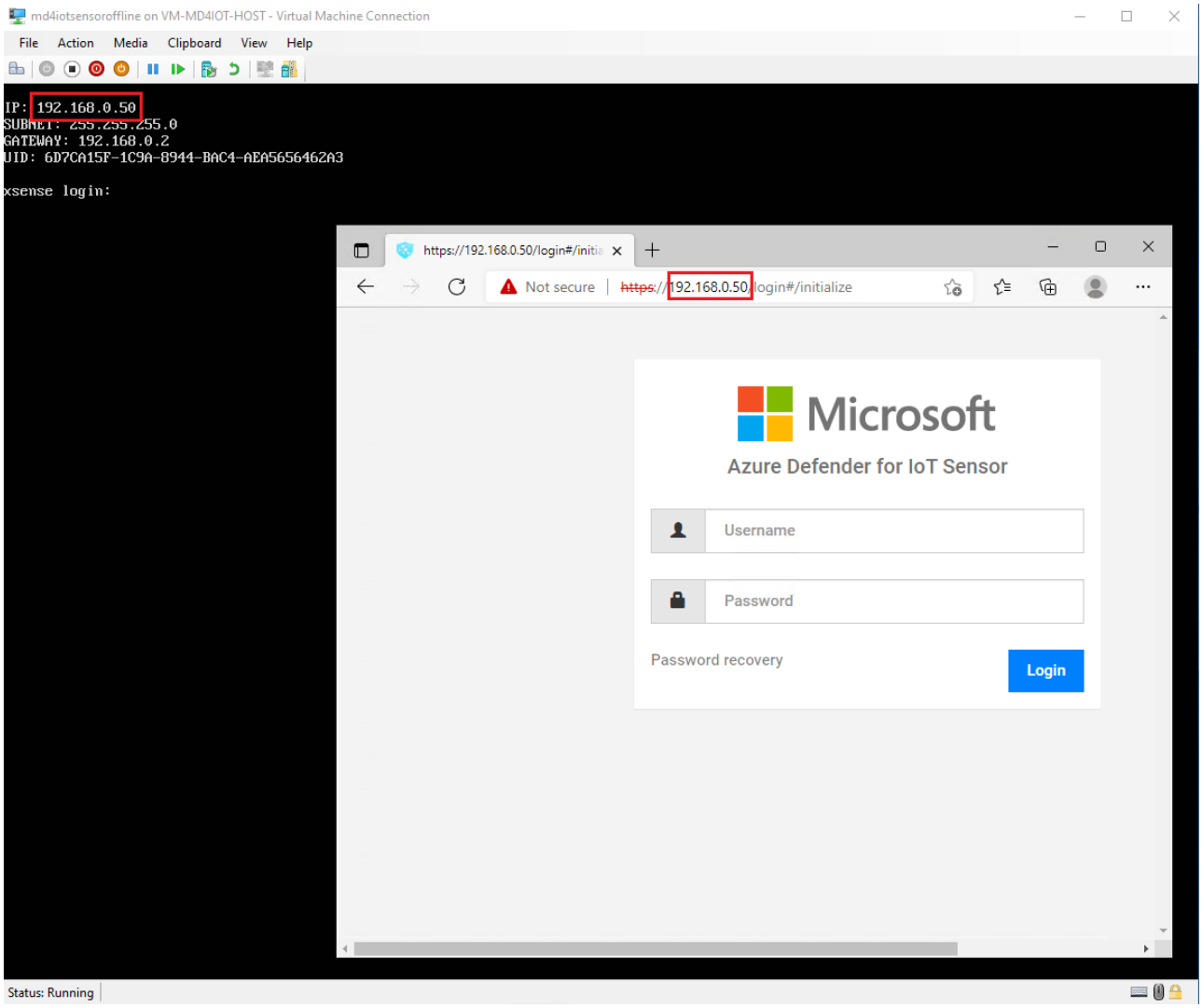
   Troubleshooting Note: *Once the installation is complete, you will be able to access Azure Defender Console. Check if you can open a cmd window, ping the IP Address you entered in the step 'Configure management network interface'. If the request times out, you will need to reconfigure this step again, for that review the IPs one more time and use the command below to start over:*

   ```
   sudo cyberx-xsense-network-reconfigure
   ```

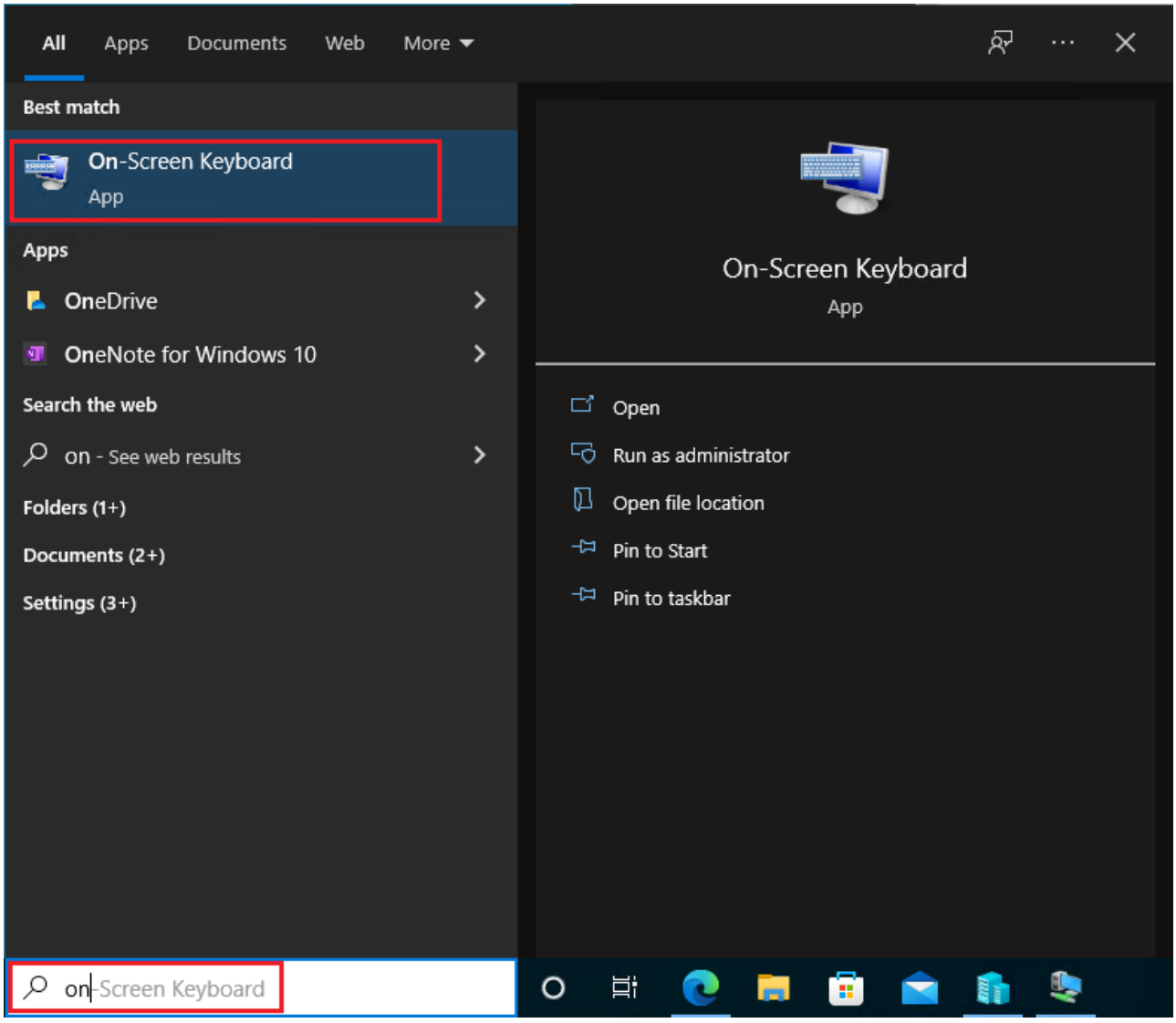
   In the next steps you will be prompt to enter the password capture above, some characteres look alike but they are not, this image will help you to identify some of them.

0	O	o	1	l	I	`	'	|
Zero	Capital “Oh” (O)	Lowercase “oh” (o)	Number 1	Lowercase ‘el’ (l)	Capital “Eye” (I)	Backtick (Same key as ~)	Apostrophe	Pipe (Shift +\)

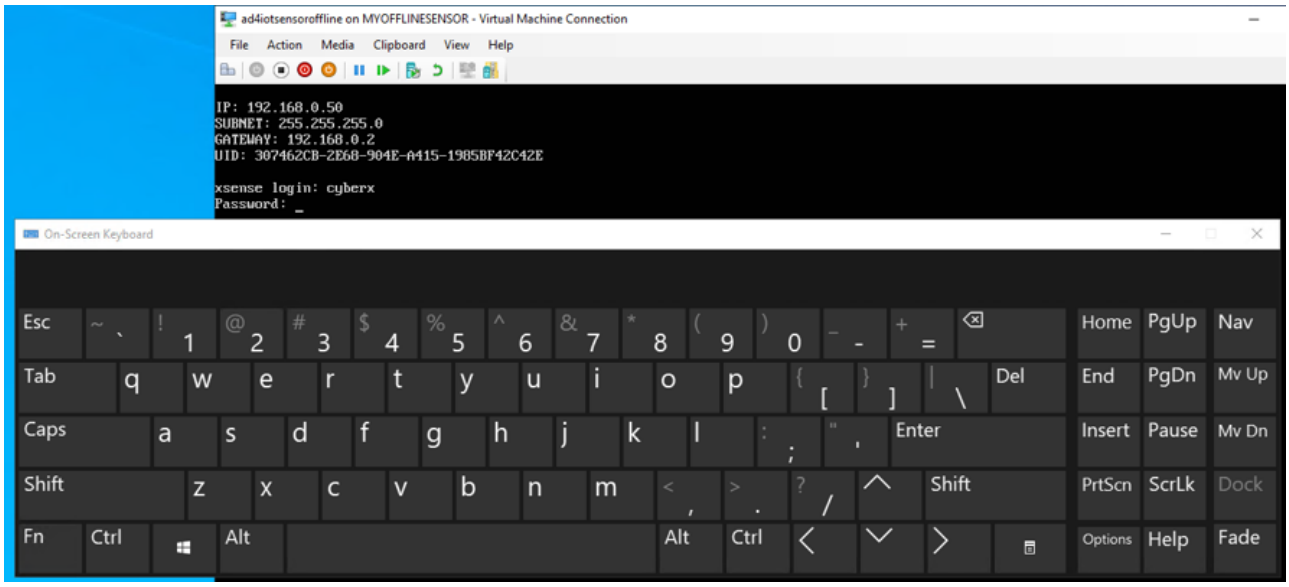
8. Login with the credentials provided in step **4**.



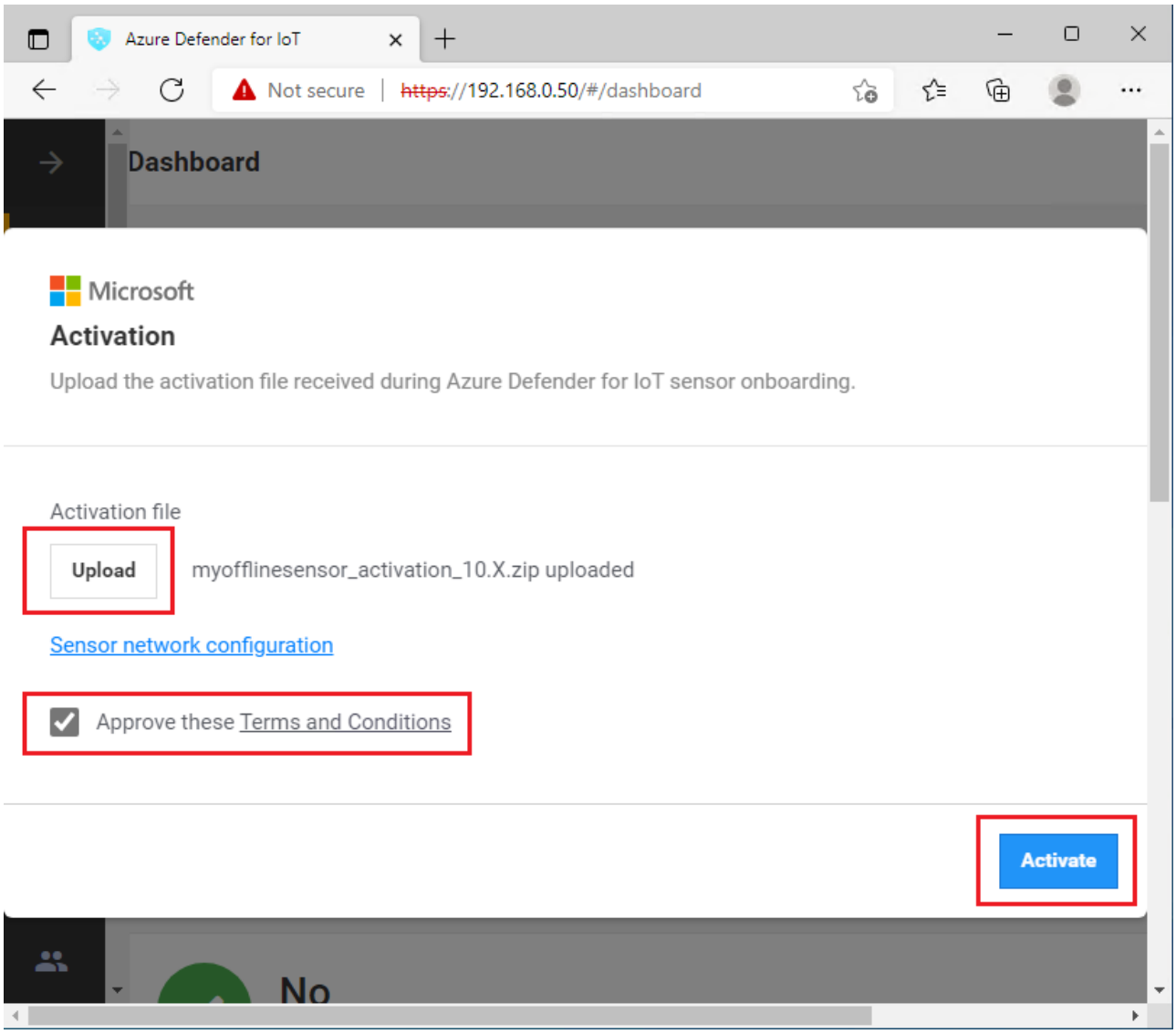
> **NOTE:** the "md4iotsensoroffline" VM's keyboard layout is US by default, and it may not match the layout of your physical keyboard. To avoid issues when entering the password, you may use the windows 10 on-screen keyboard. To run it, type "osk" in the search box and click on "On-Screen Keyboard"...



> ...and use it to enter the credentials:

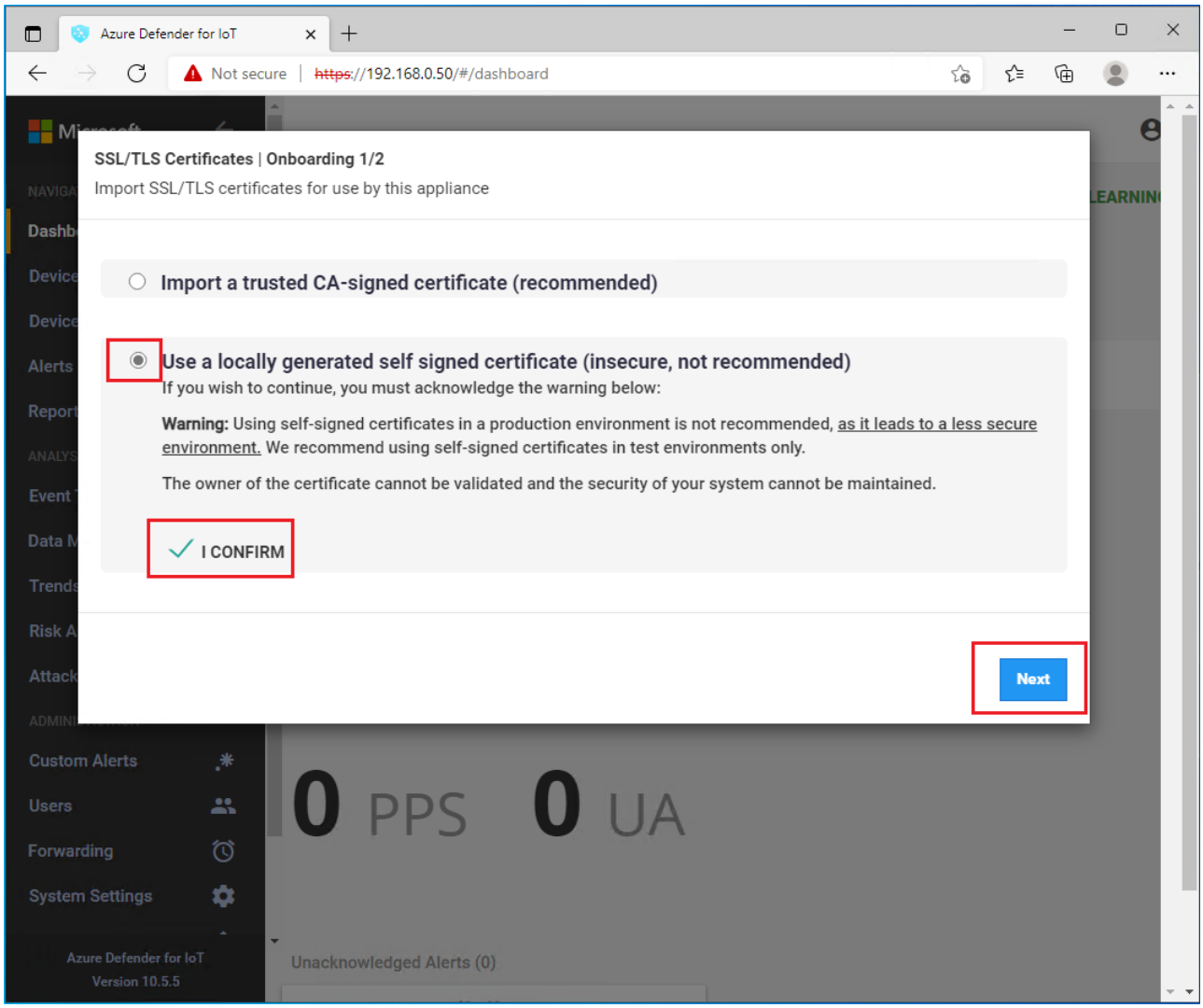


9. Next, you will be ask to activate the product, click **Upload**, then **Browse Files**, in your downloads folder select the file you downloaded from the Storage Explorer, in this example **myofflinesensor.zip**.



10. Click **Approve these terms and Conditions**, then **Activate**.

11. You will be prompted to select **SSL/TLS Certificates | Onboarding 1/2** for this lab will use the second option **Use a locally generated self signed certificate(..)**. Then click **I CONFIRM**, **Next**.



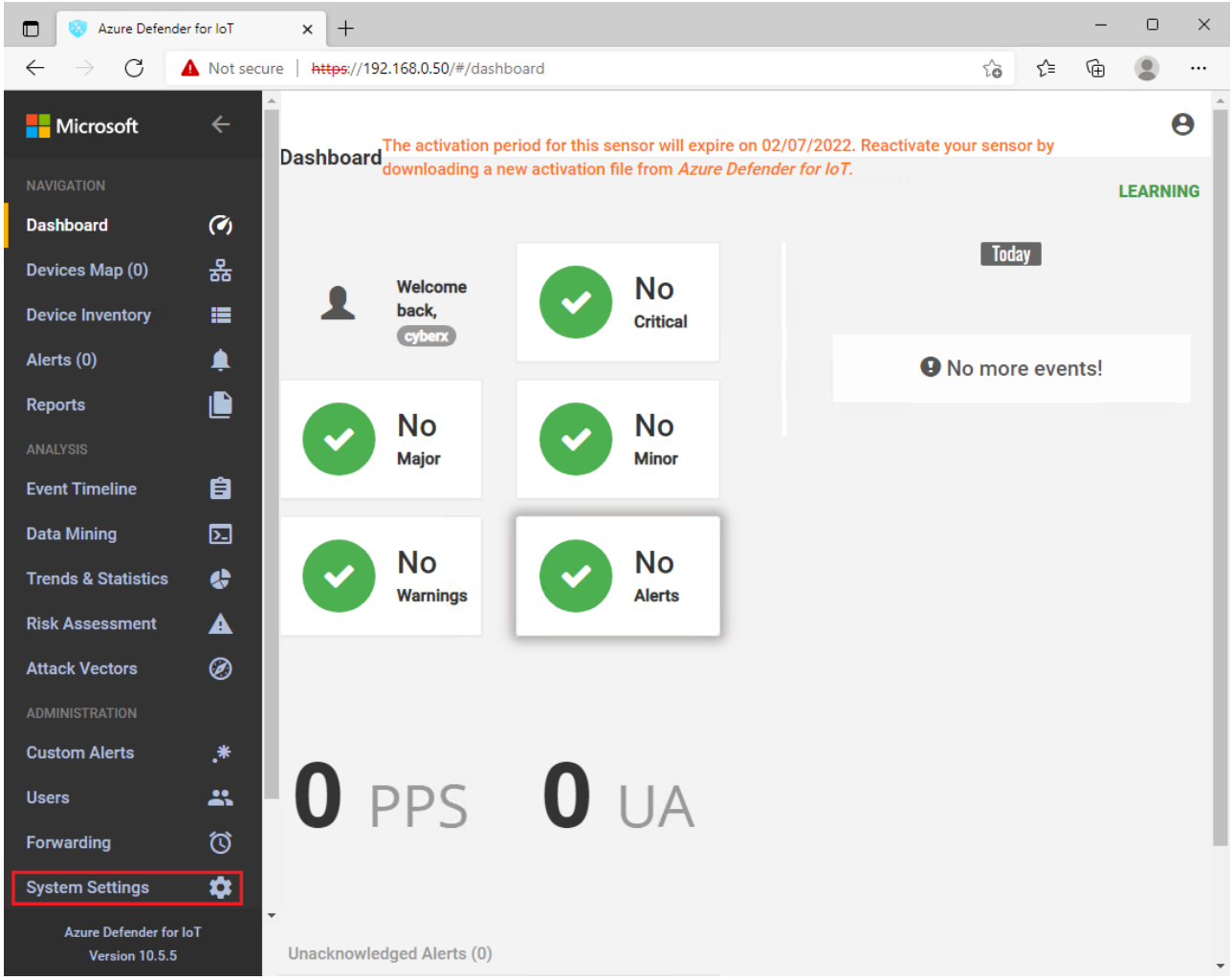
12. For this lab in the next step we will **Disable** the system wide validation. **Finish**.

13. Let's analyze together what information we already have available before moving forward.

Exercise 3: Enabling system settings

Task 1: System Properties

1. In your offline sensor you will find **System Settings** on the left side of the Azure Defender portal, click there as shown below.

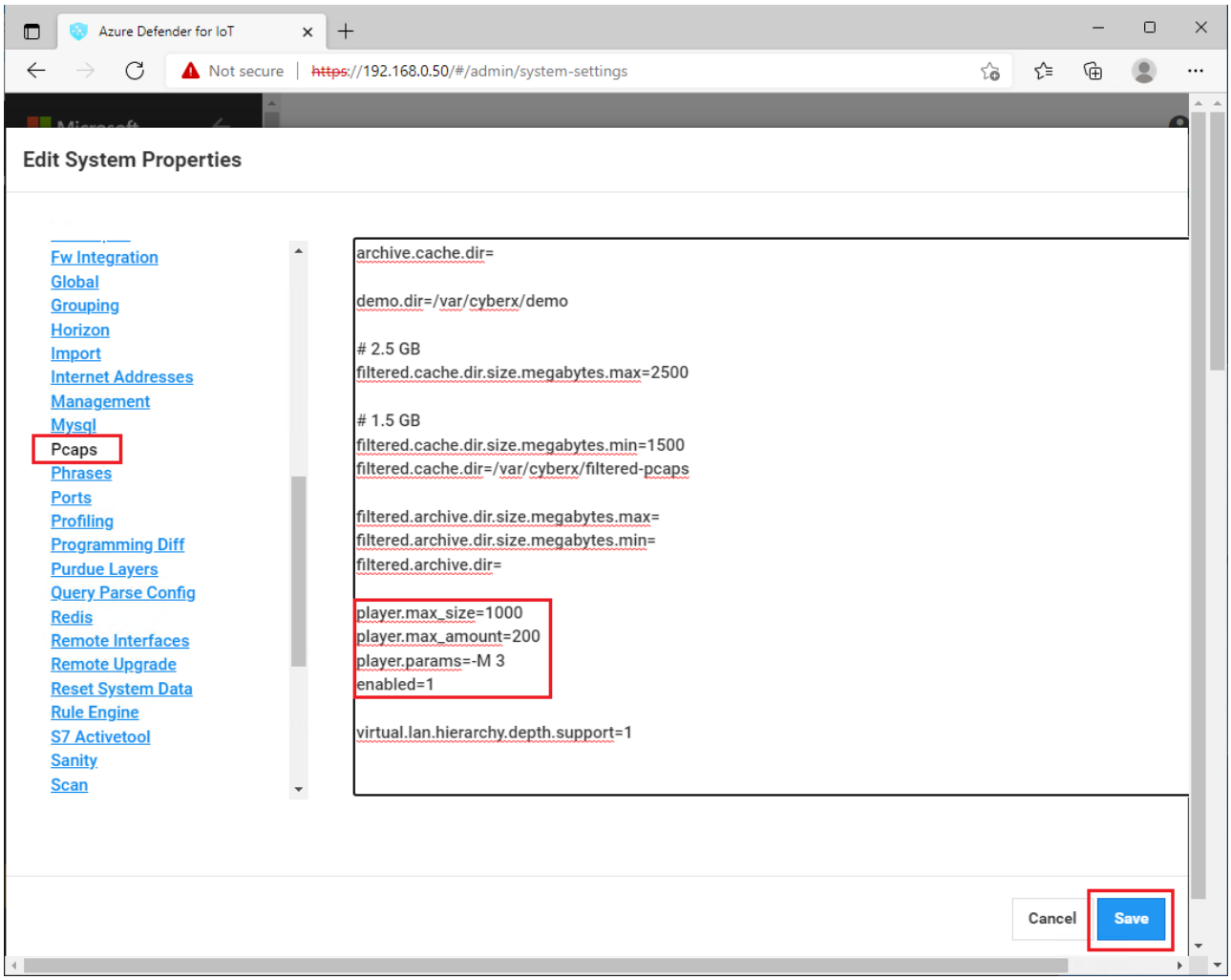


2. Next, look for the icon **System Properties** on the right side. Click on the icon. You you will see a pop up warning, click **Ok**.

3. In the new window on the left side, scroll down until you see **Pcaps**, click there. Now on the right side scroll all the way down and we will modify three parameters as shown below:

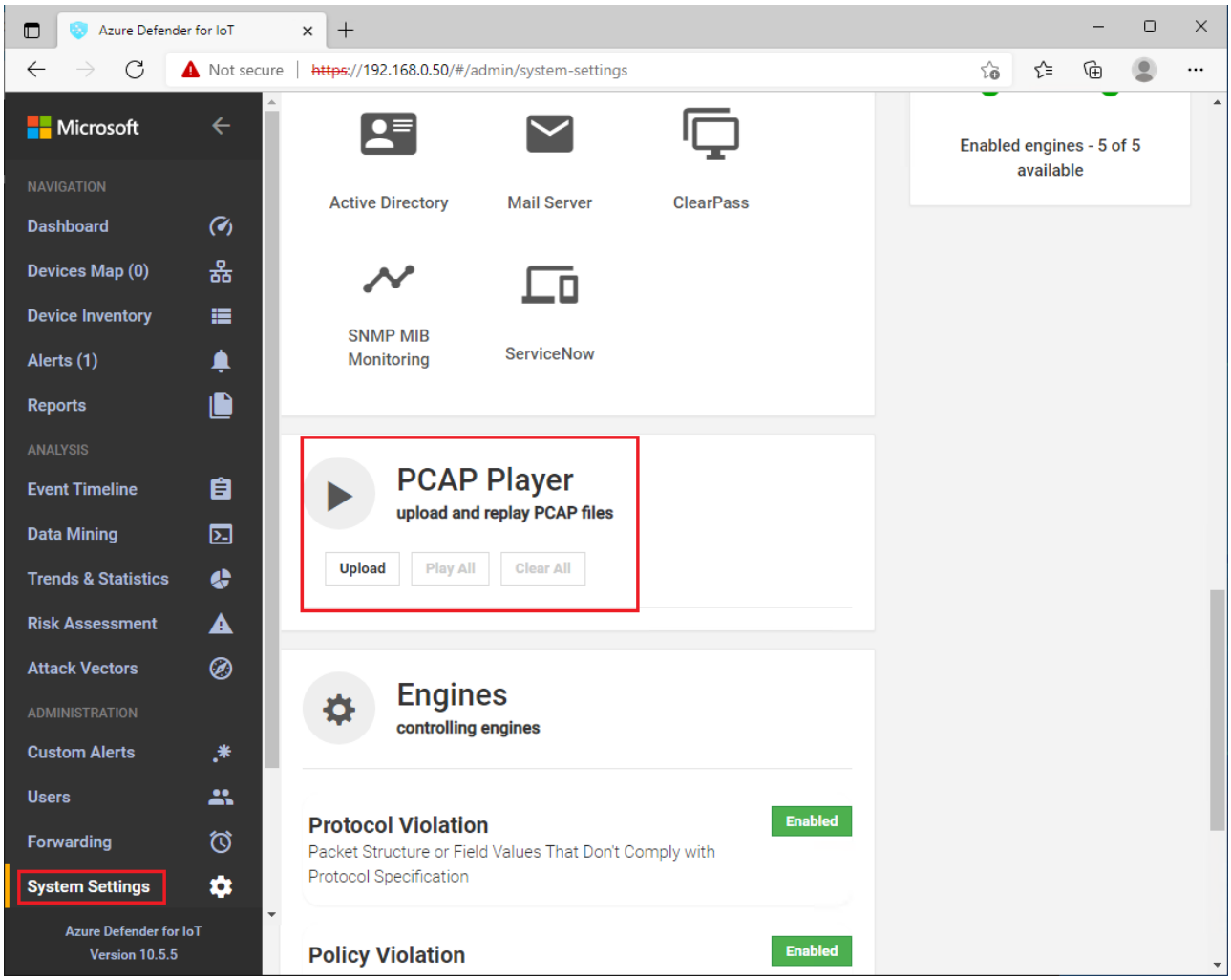
    - **player_max_amount=200**
    - **enabled=1**
    - **player.params=-M 3**

    Amongst others, these settings enable the PCAP player and allow it to playback faster than real-time.



4. Click **Save** and then **Ok**.

5. At this point you should see the Pcap Player available (you can close the **Edit System Properties** screen now by clicking the **Cancel** button):



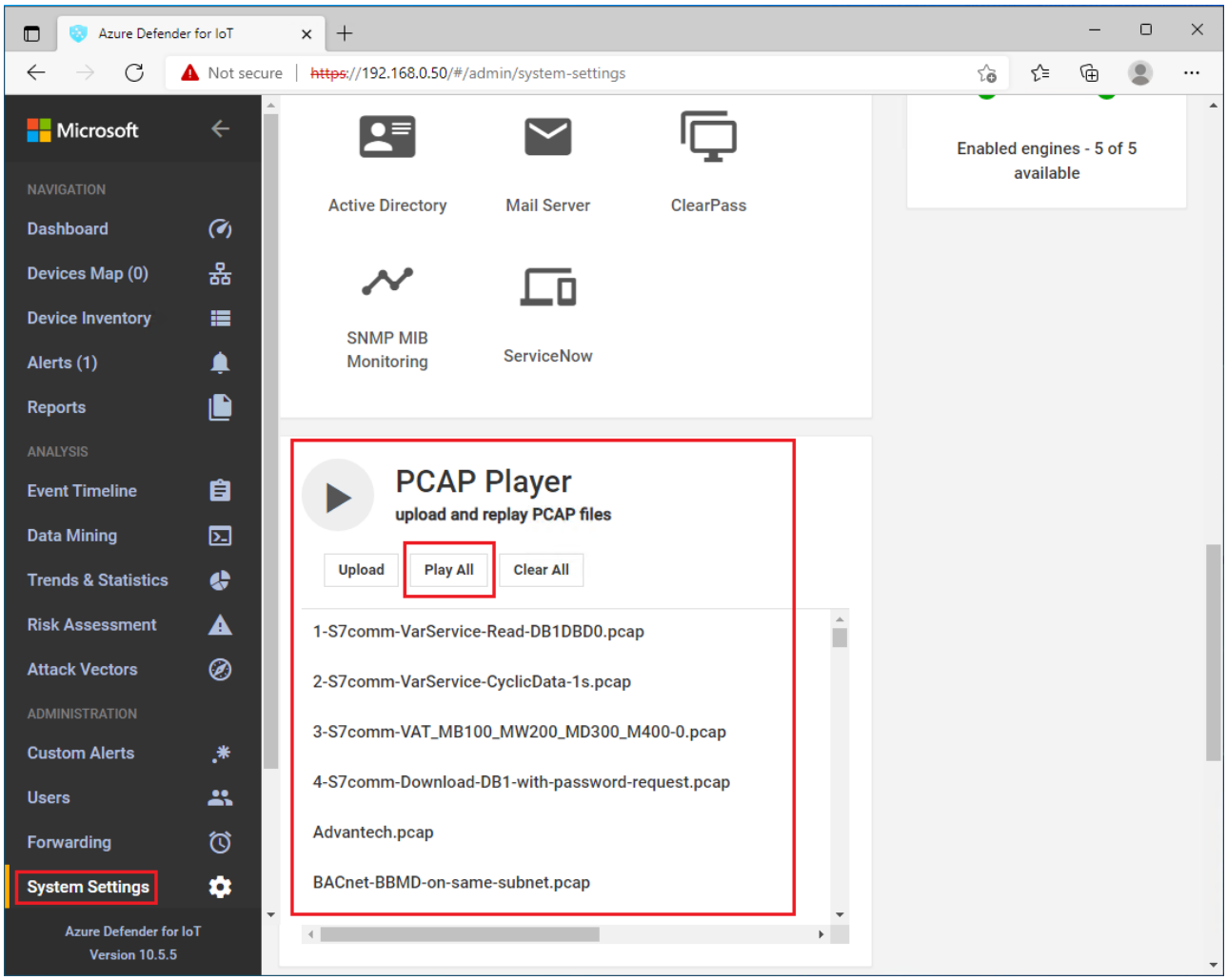
Task 2: Pcap Files

1. In a previous step you already downloaded a **holpcaps.zip** file from the Storage account. It should be in your Azure Virtual Machine's **Downloads** folder.

2. Unzip **holpcaps.zip**

3. Go back to Azure Defender, Click on **System Settings**, then **PCAP Player** now select **Upload**, **Browse Files**, browse to the folder where you download the files in the previous step, select all the files and click **Open**. This operation will take a few minutes to upload all the files.

4. At this point you should see all the files uploaded.



5. Click on **Play All**, in a few minutes you will receive a message saying all the files has been played.

Exercise 4: Analyzing the Data

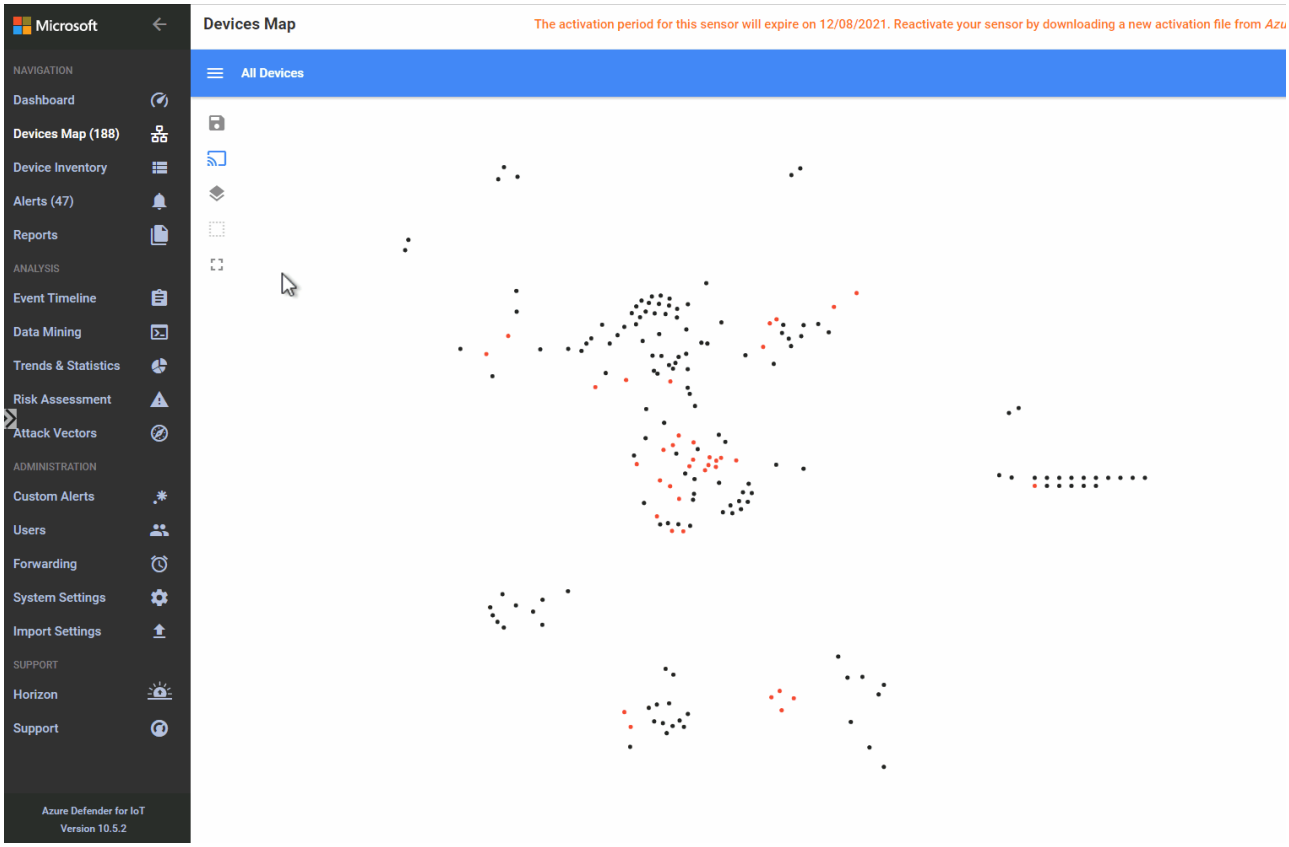
After Defender for Cloud learnt about your environment it will be able to share insights pretty fast.

Task 1: Devices Map

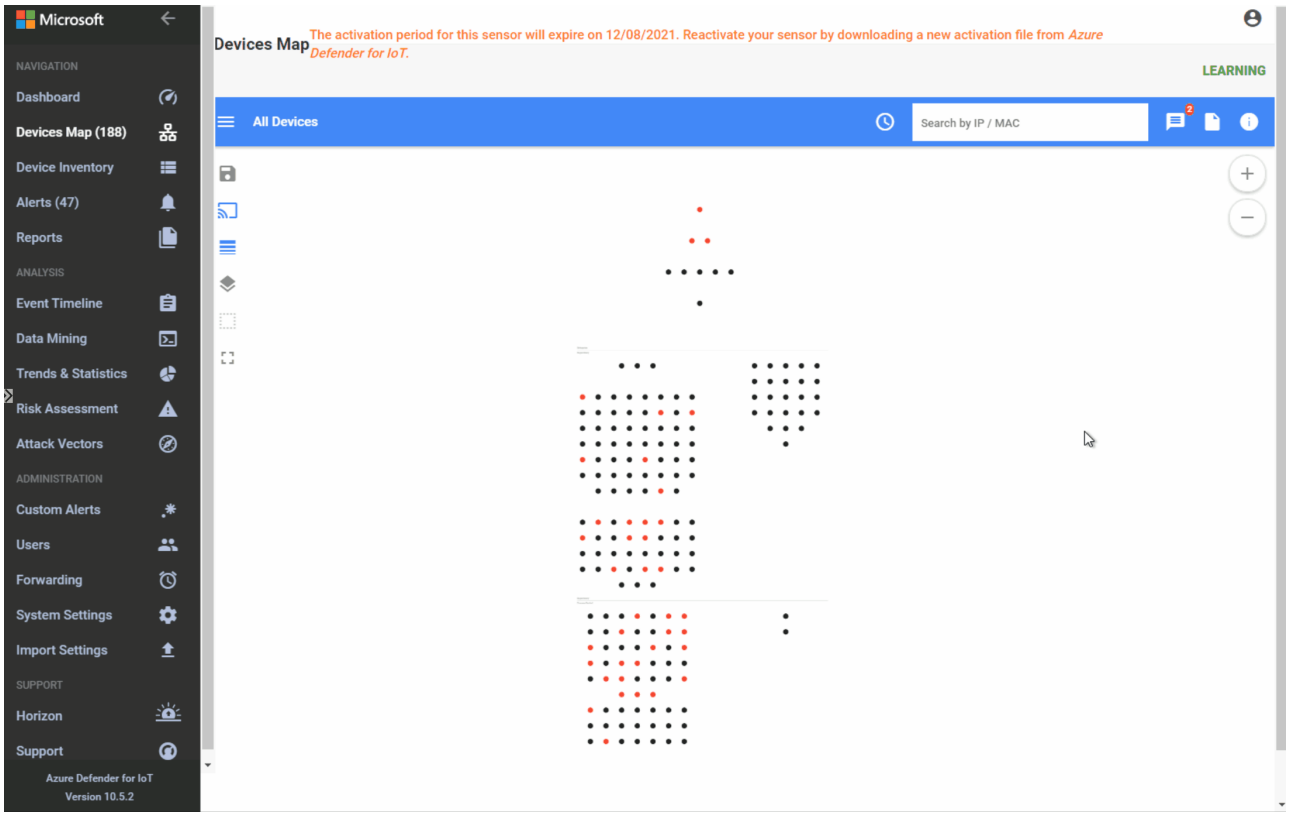
Your first interaction with Devices map you will see a similar map like the one below (details of what you actually see may vary):

Pcap Files Uploaded

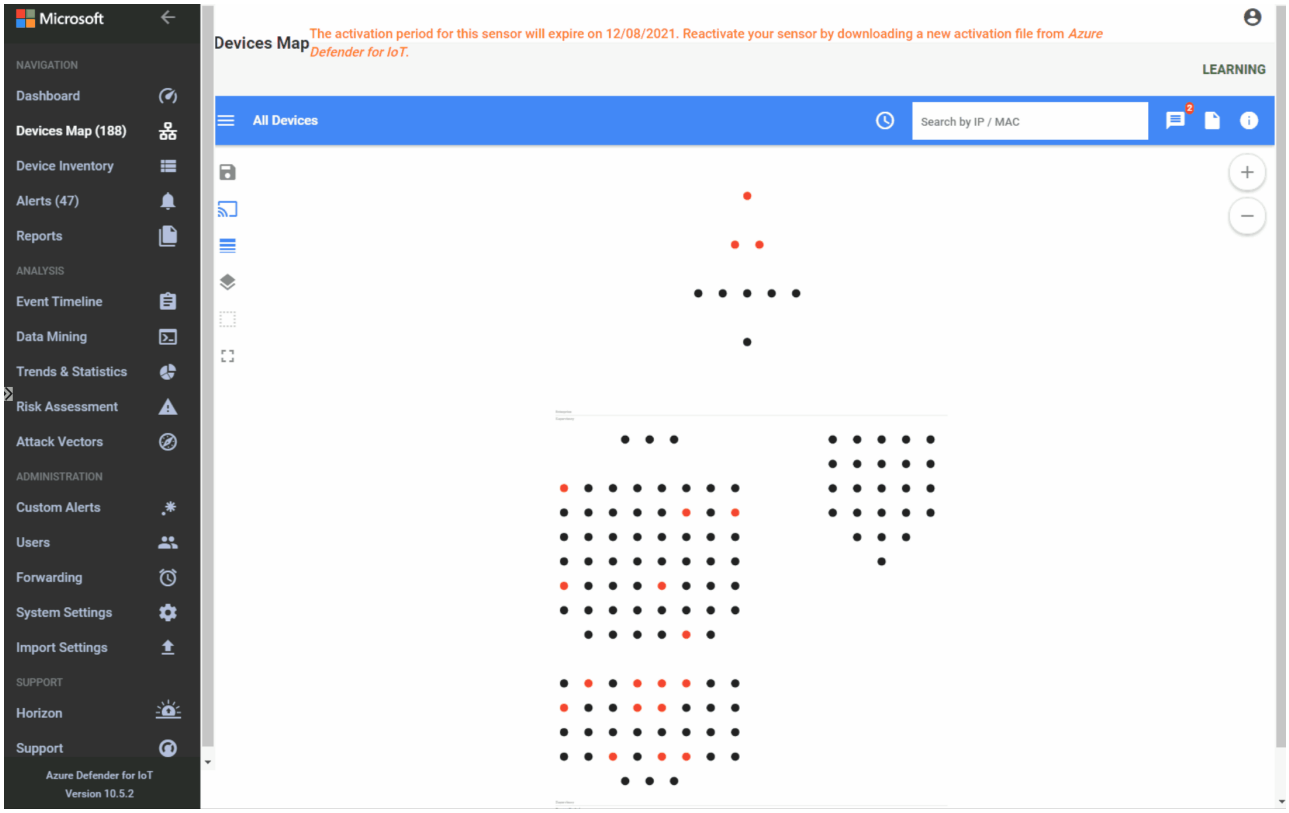
1. Use the four icon bar on the left to select **Layout by Purdue**. In this model you will see the different layers between Corporate IT and site operations.



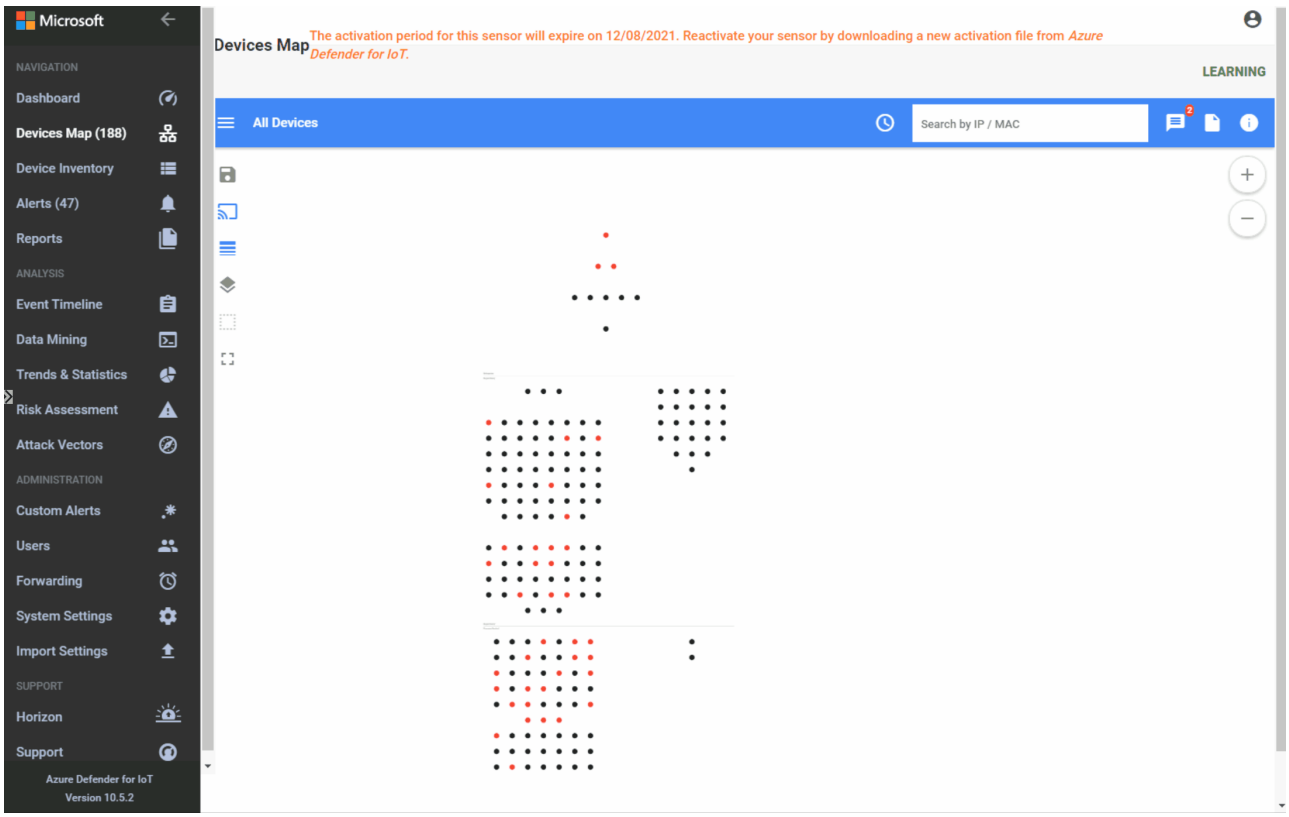
2. Check your notifications available and you can take action at this point.



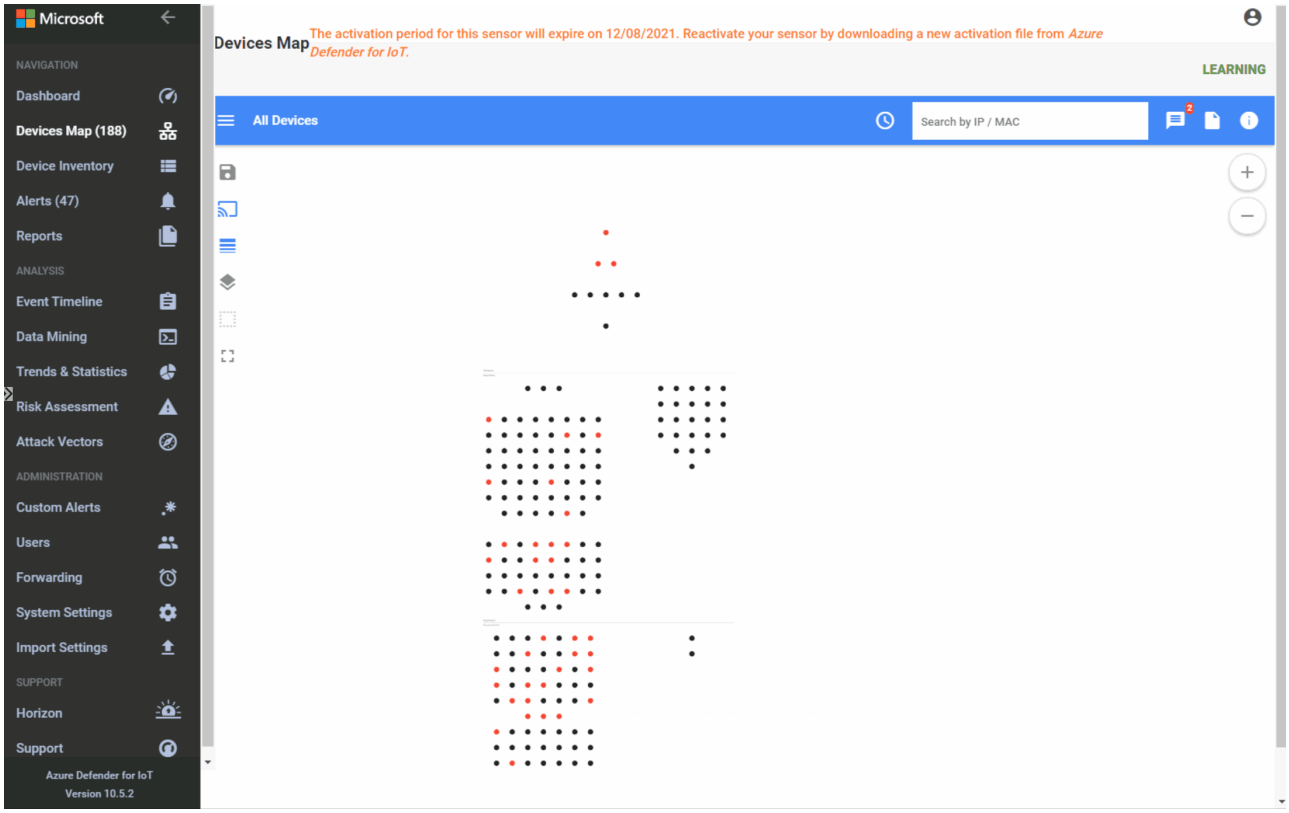
3. For each device right click to analyze properties, show events, reports and simulate attack vectors.



4. In the hamburger menu on the left, click the highlights and select one of the **OT Protocols** i.e. **MODBUS** and click on **Filter**. Now your map will show those devices only



5. Then filter your devices by **CIP** OT Protocol, at the bottom of your map you will see a PLC, where the Vendor is Rockwell Automation, has already 3 alerts activated. Right click on the device, **View Properties**. In this view you will be able to analyze the Backbone of your PLCs, take actions and analyze the Alerts.

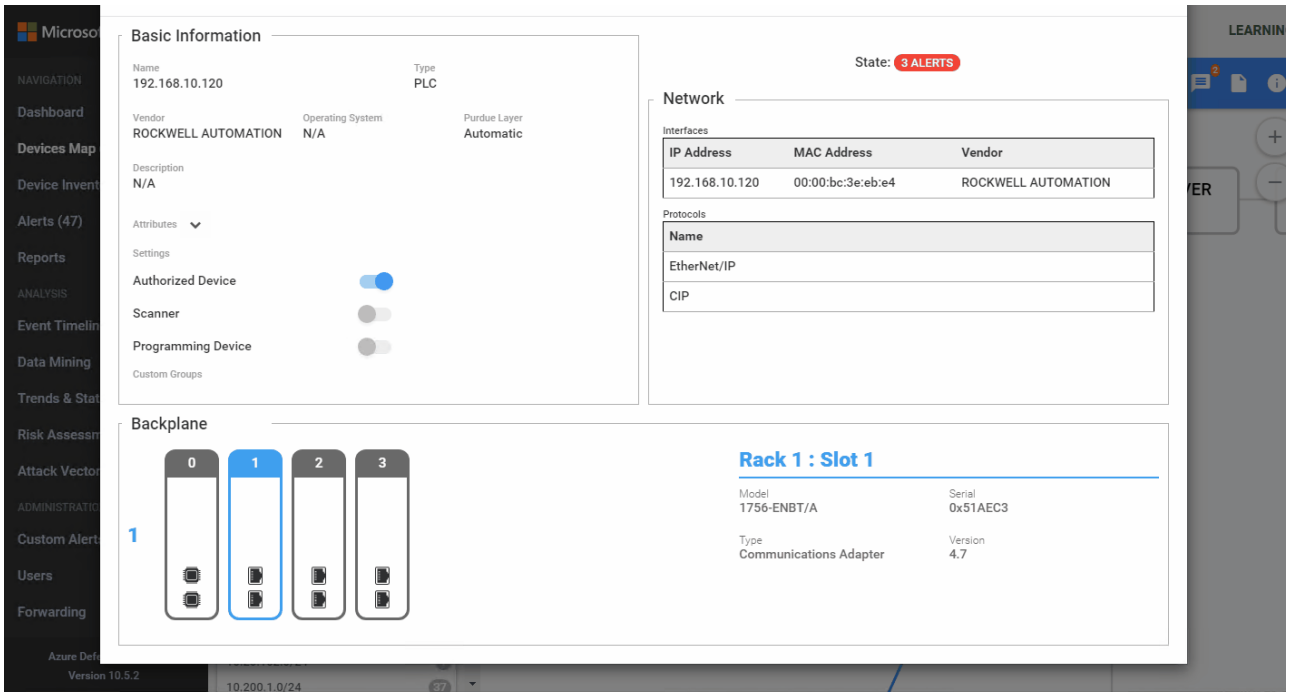


Task 2: Alerts

1. Once you click Alerts in your PLC you will see a new window pop up showing three different types of alerts.

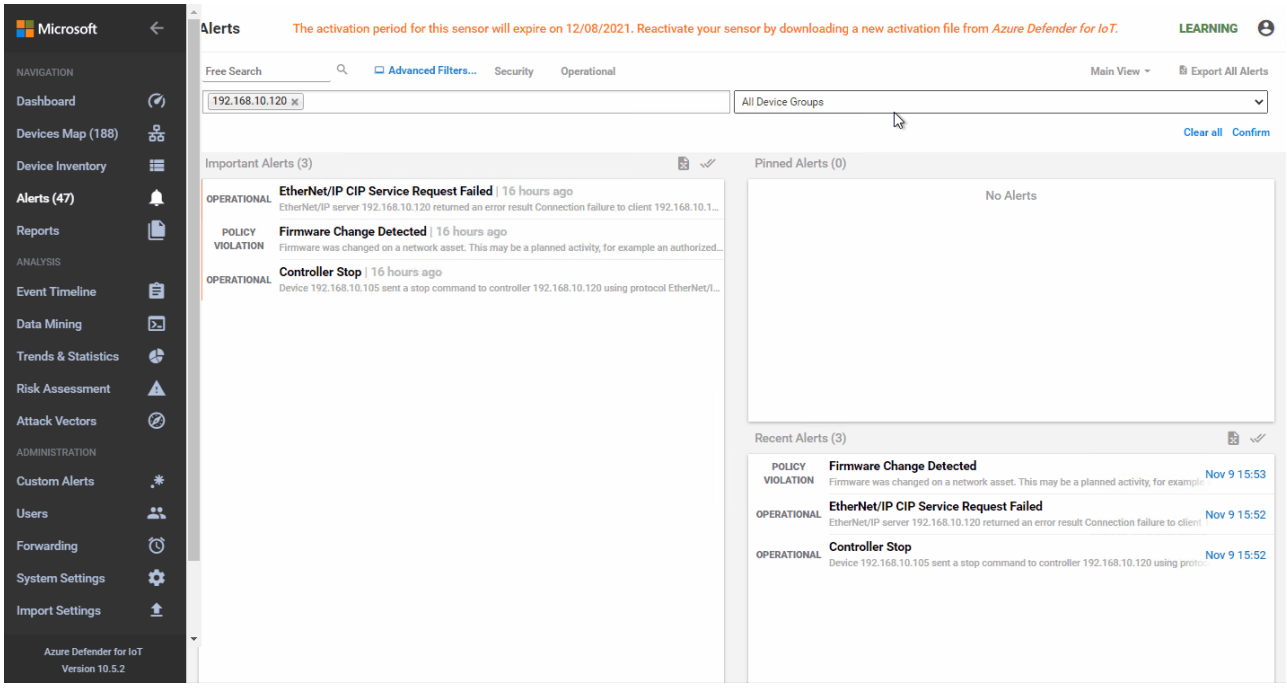
    - Operational(high Alert and lower alert)
    - Policy Violation

    For each of these alerts you will be able to analyze the pcap file, export a report, analyze the timeline or mute the alert.



2. If we remove the device filter from the top of the screen, then click **Confirm** you will see 20 Alerts in process.

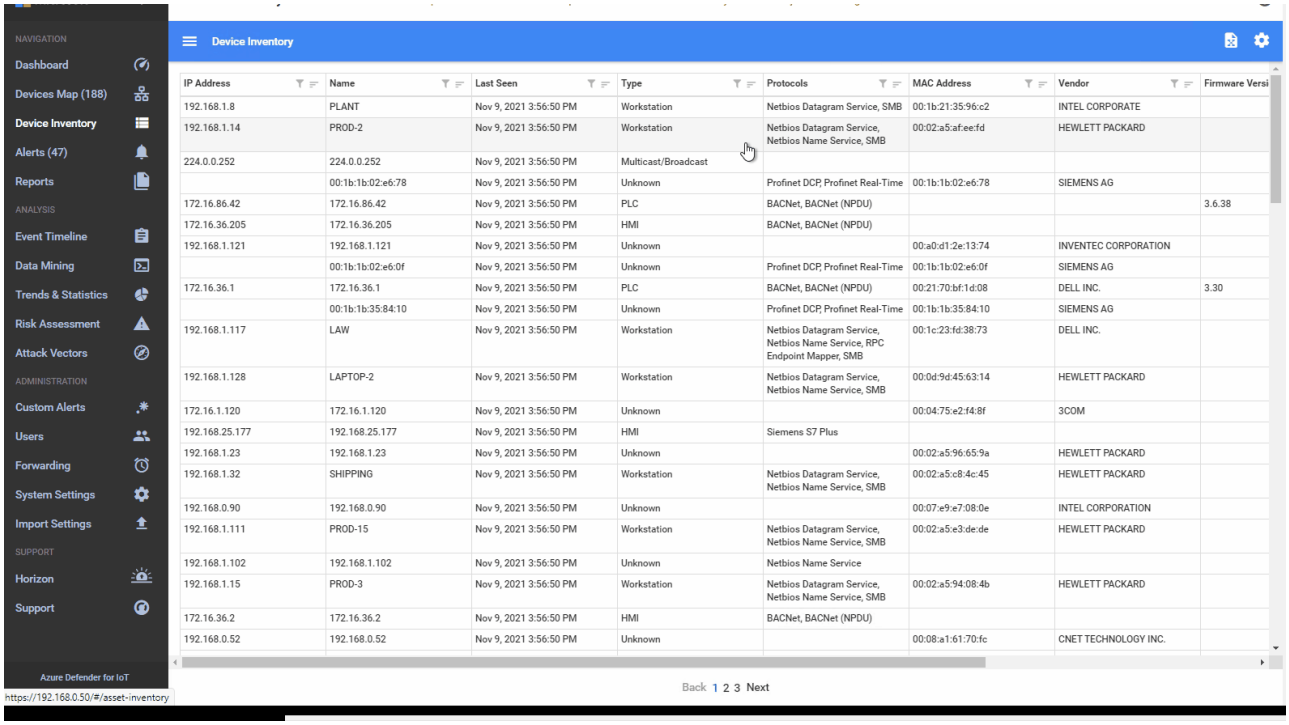
3. Apply **Custom Groups** to filter different scenarios, such as **Unclassified subnets** then **Confirm**



Task 3: Device Inventory

1. In this view, filter all your devices by **Is Authorized**, True or False are possible values.

> **NOTE:** if you don't see the column "Is Authorized", click on the "Device Inventory Settings" gear icon (upper-right corner) and add it to the view.



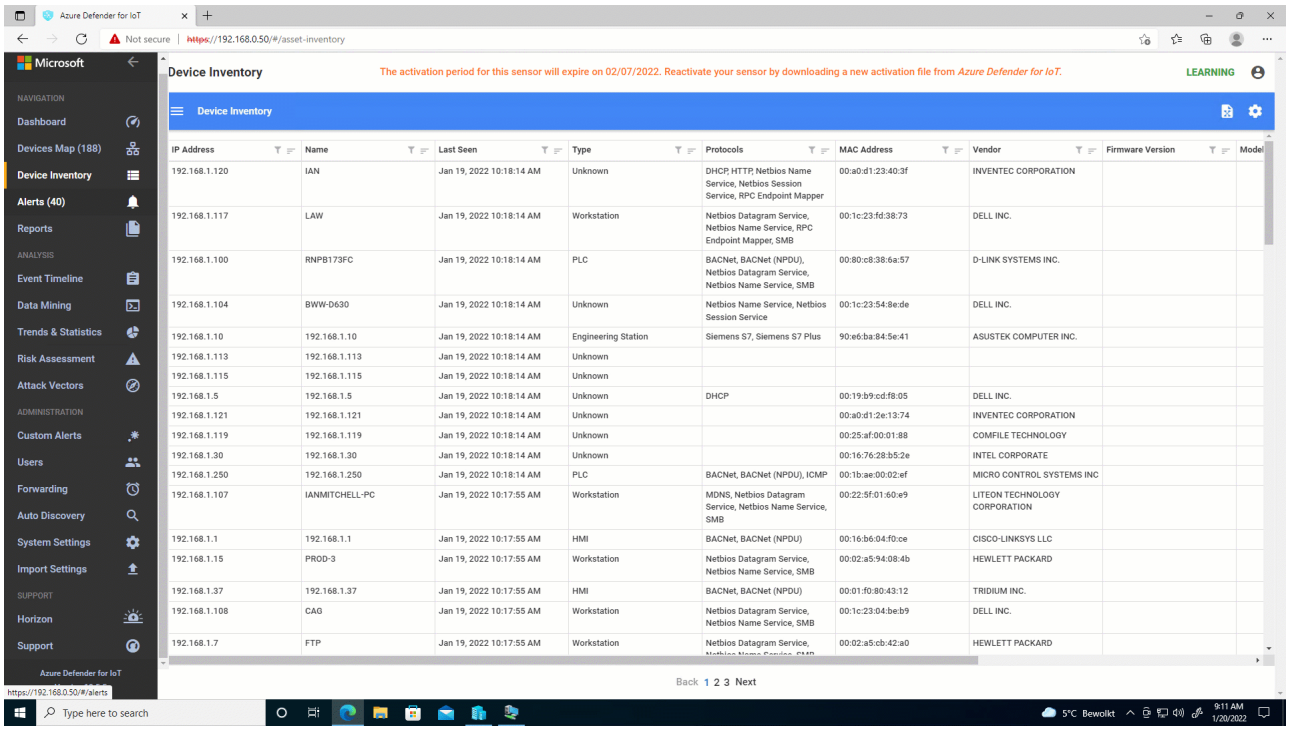
2. Organize your devices based on filters.

3. Export the list to a csv files.

Task 4: Event Timeline

This view will allow you a Forensic analysis of your alerts.

1. Choose **Advanced Filters**, filter the timeline by **CIP**, let's analyze the alert timeline.

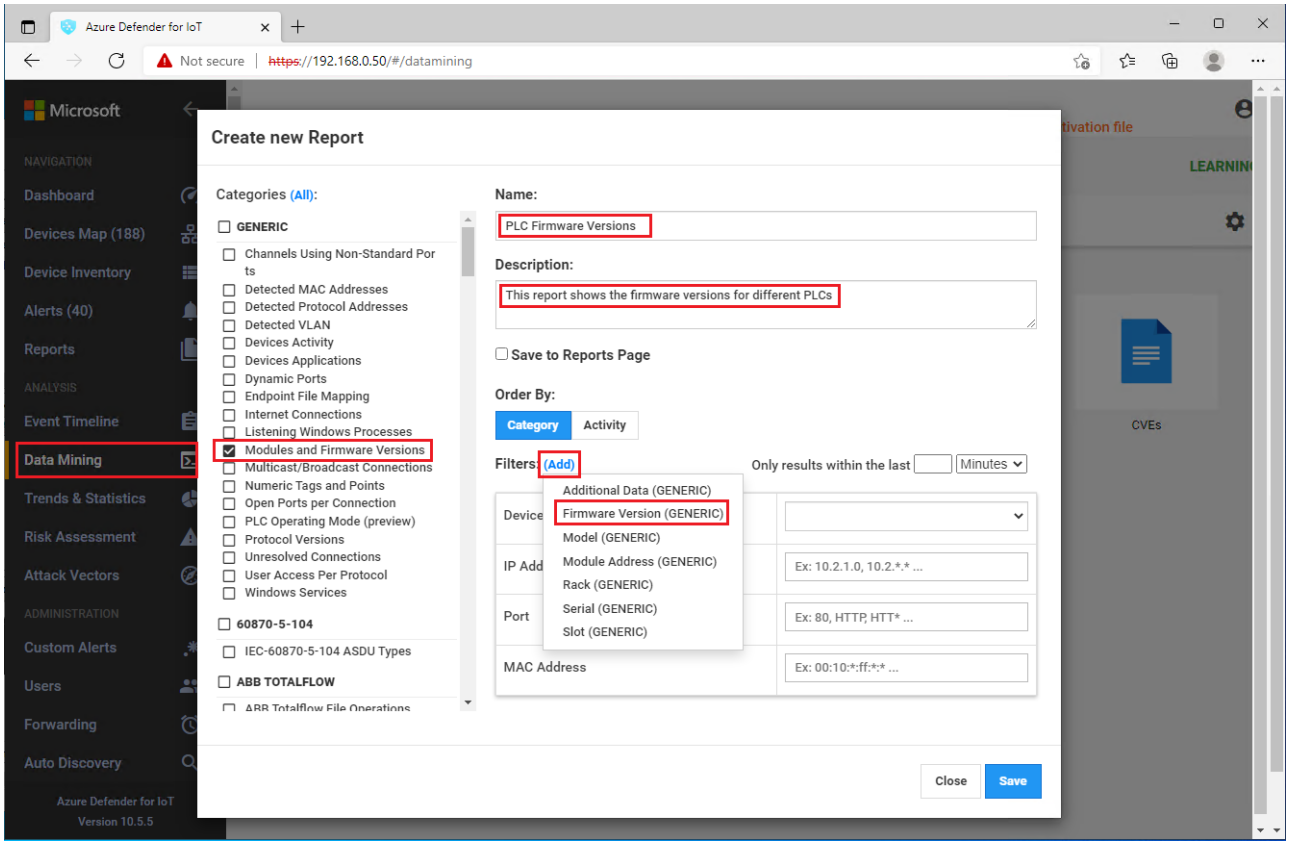


Task 5: Data Mining

In this section you can create multiple custom reports. As an example we will create a Report based on firmware updates versions.

1. Go To **+**, **New report**, in the categories section select **Modules and Firmware update versions**

2. Assign a name to your report. Then go to Filters, **add** and select **Firmware version(generic)**



3. In the new field added **Firmware Version(GENERIC)** add **0.4.1**, then **Save**.

4. You can remove the filter to list all the firmware updates version in your list also.

5. Export you report(pdf, csv) for further actions.

Task 6: Risk Assessment

1. Go to the Risk assessment, run the assessment. During this task we will show you how to analyze the assessment.

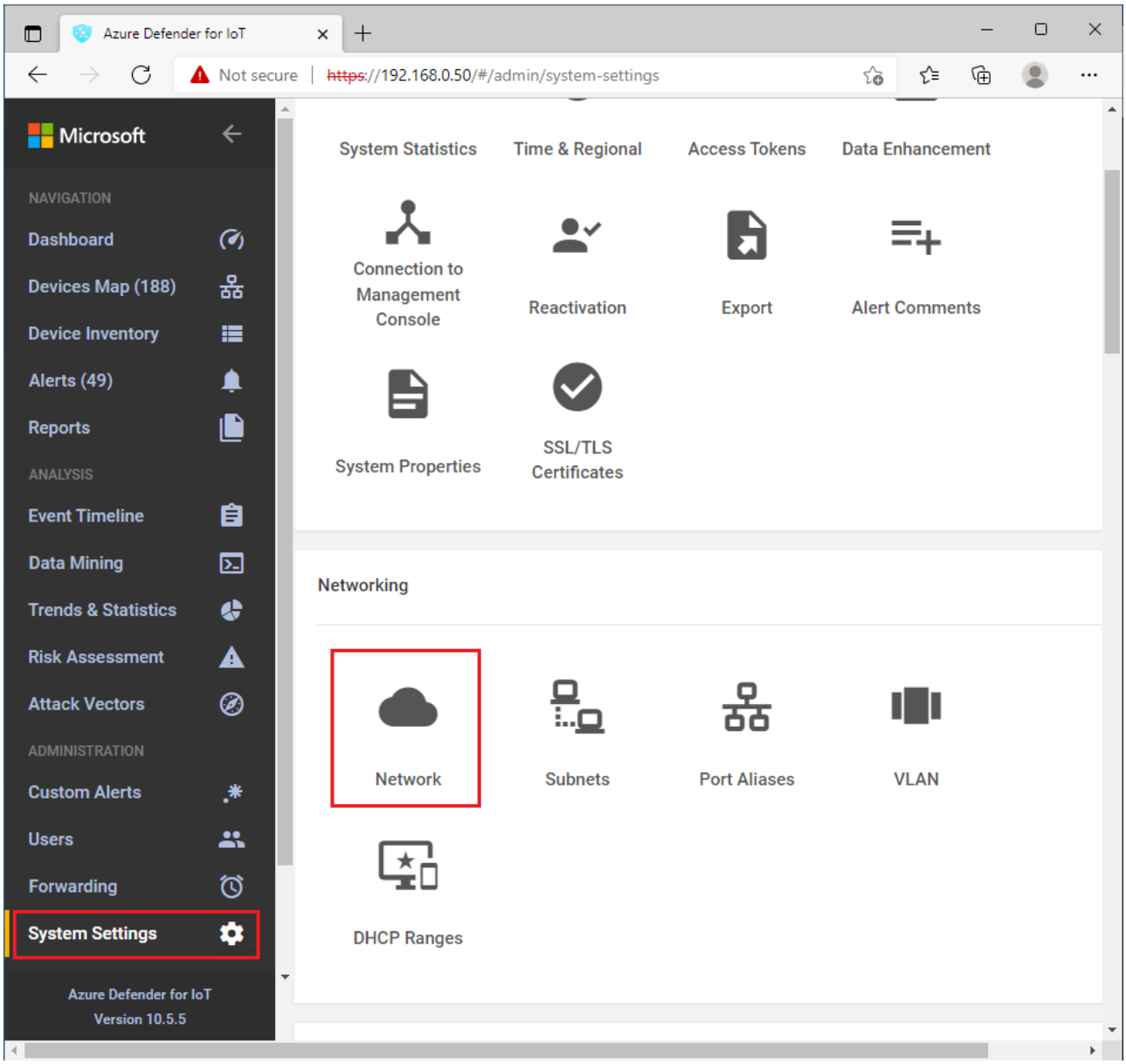
Exercise 5: Online Sensor

To modify our sensor to become an online sensor, we will use the same virtual machine that we used for the offline sensor, but we will reactivate the sensor using **System settings**. In a real scenario you probably would create a new sensor, running in its own virtual machine or physical appliance.

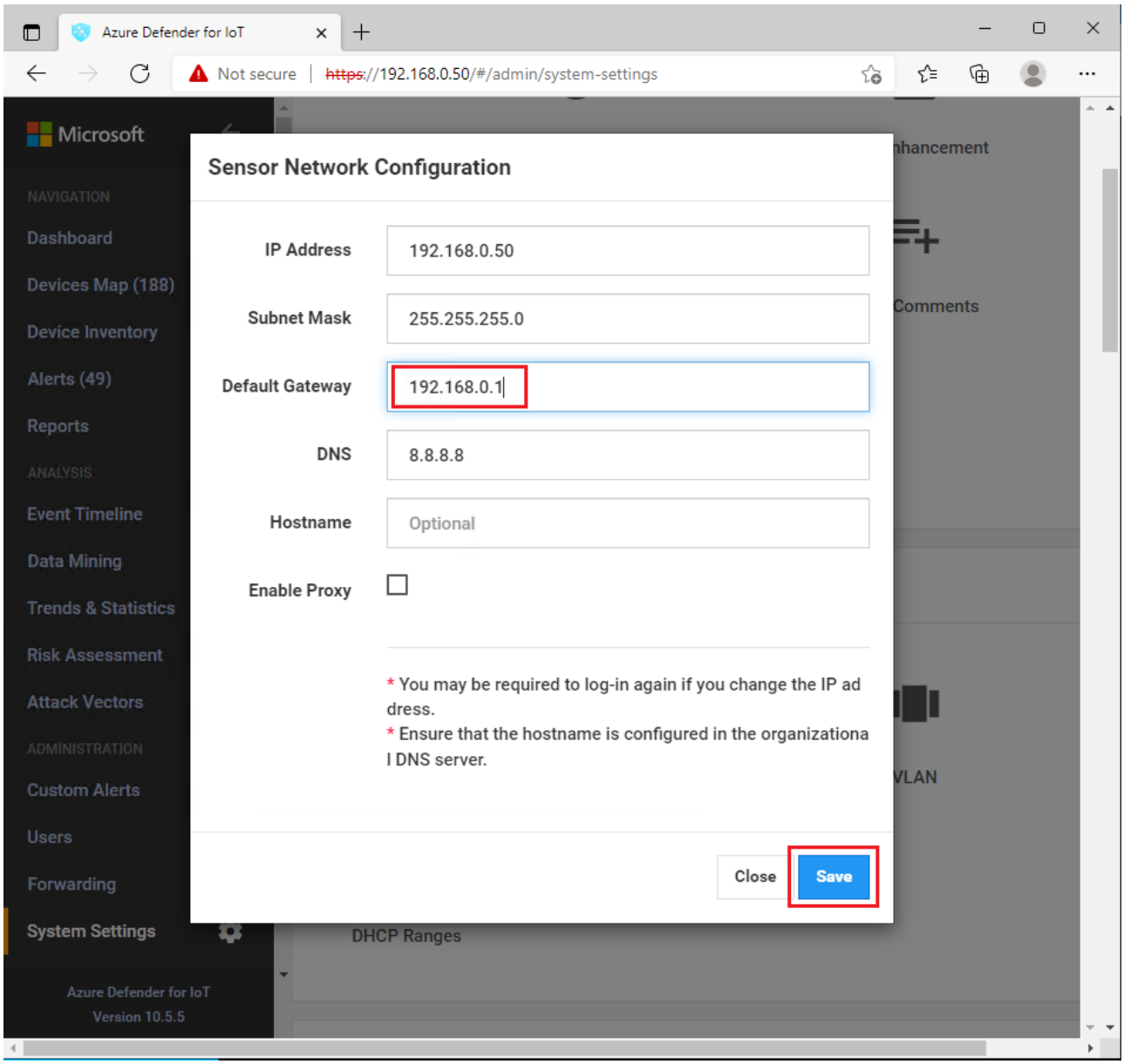
Task 1: Reconfiguring sensor

To modify your sensor to be connected with Azure, you will need to modify the network configuration.

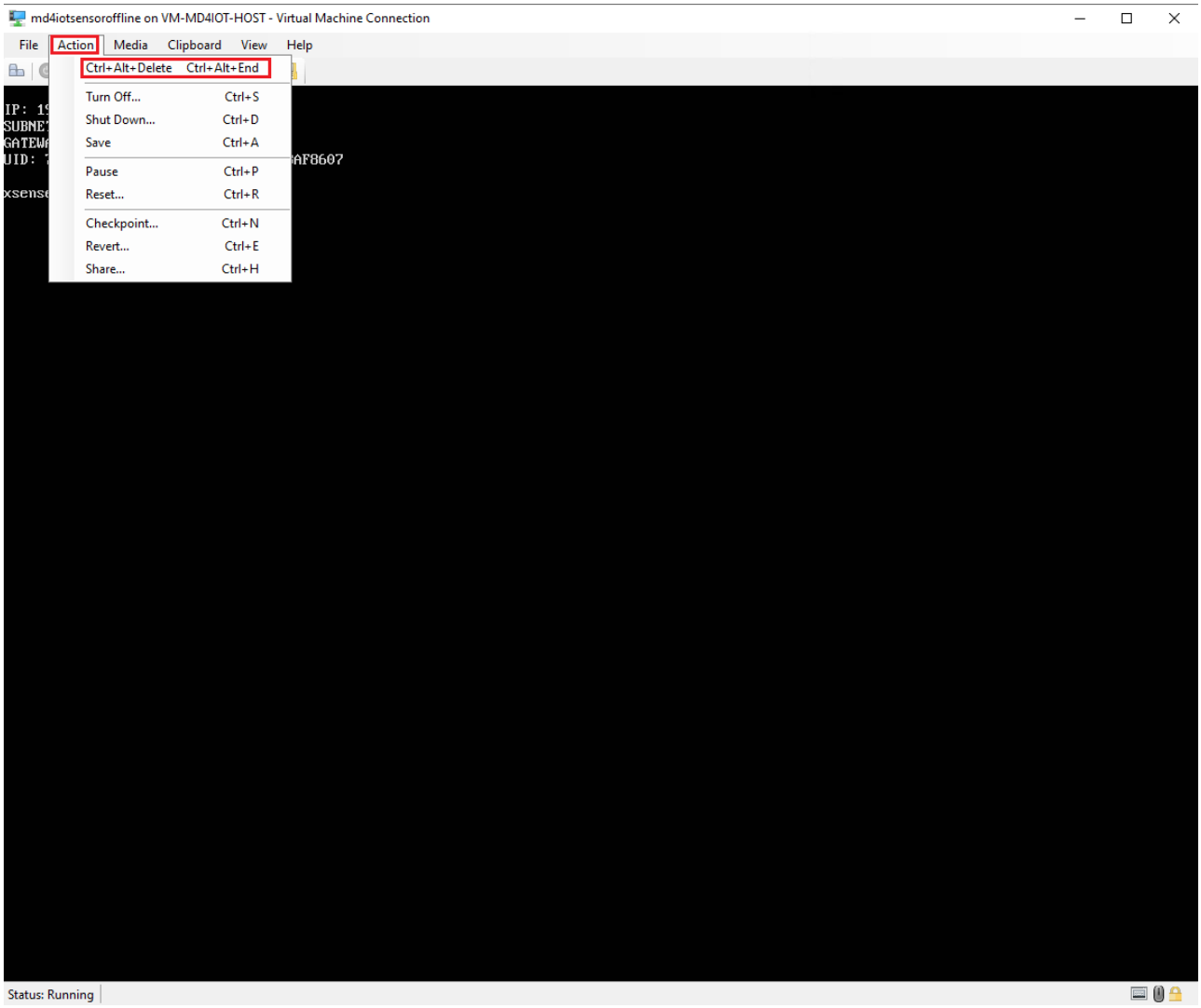
1. In your sensor's Azure Defender for IoT Portal (in the Virtual Machine), select **System Settings** and **Network**.



2. Change the IP Address of the Default Gateway to 192.168.0.1 or 172.27.0.1, depending on the settings you used earlier in the HOL.

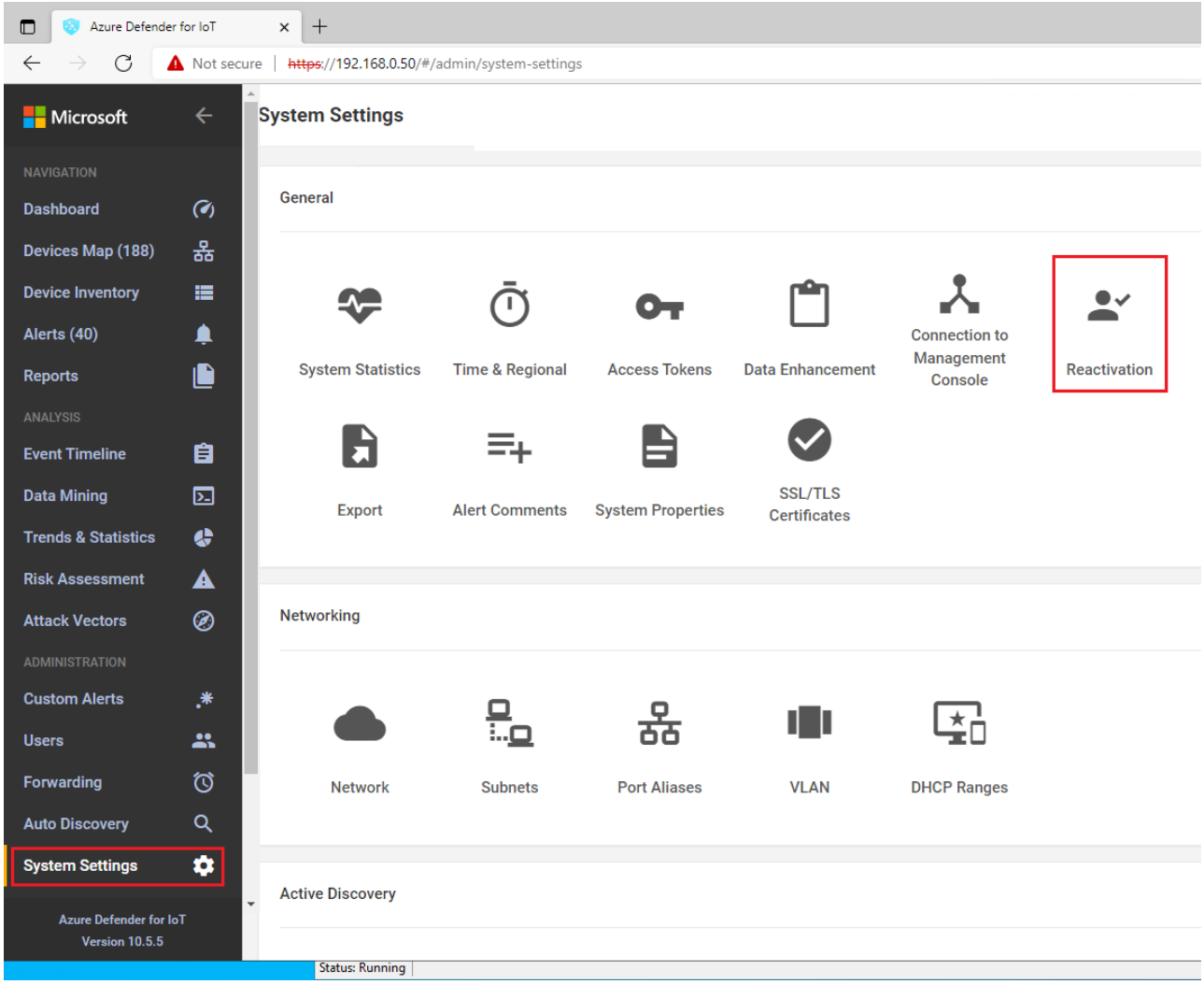


3. On the "md4iotsensoroffline" Virtual Machine Connection, select **Action** and **Ctrl+Alt+Delete** to reboot the sensor.



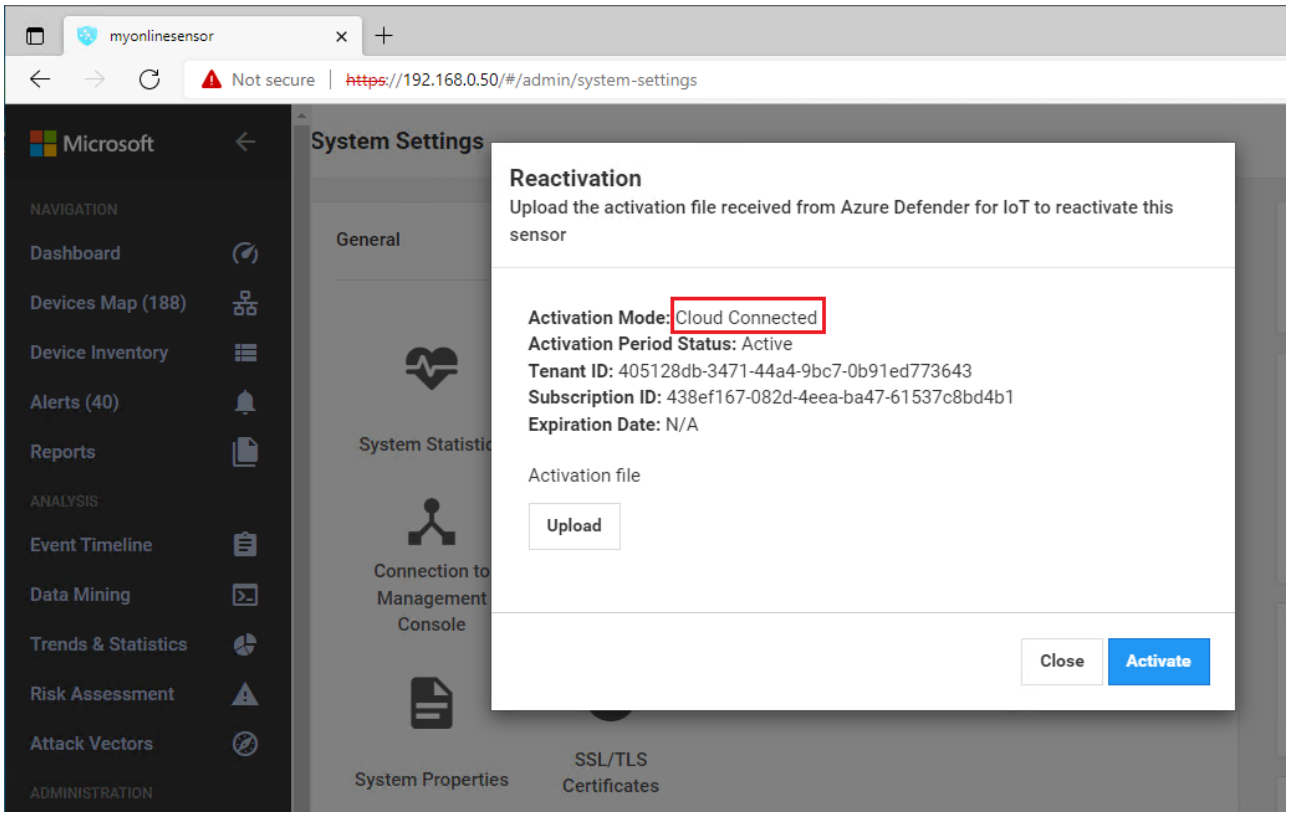
4. Login to your sensor's Azure Defender for IoT Portal (in the Virtual Machine) again, select **System Settings** and then, **Reactivation**.

5. In the new window, select **Upload**, **Browse File**, select the zip file you downloaded from the storage account in previous steps **myonlinesensor.zip**, then **Open** and **Activate**, **Ok** to the instructions

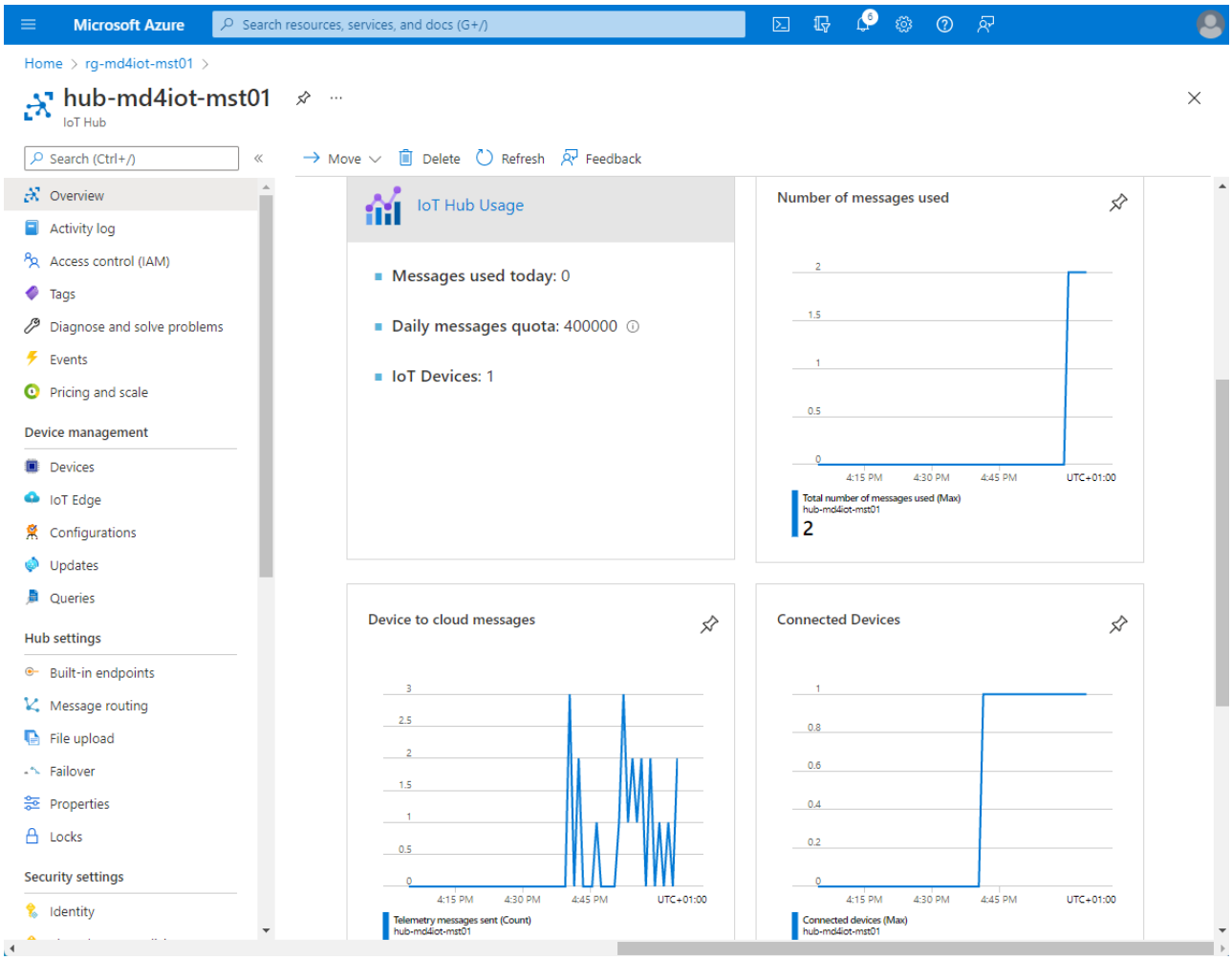


6. Last, you should receive a message showing your sensor modified to **Connected**.

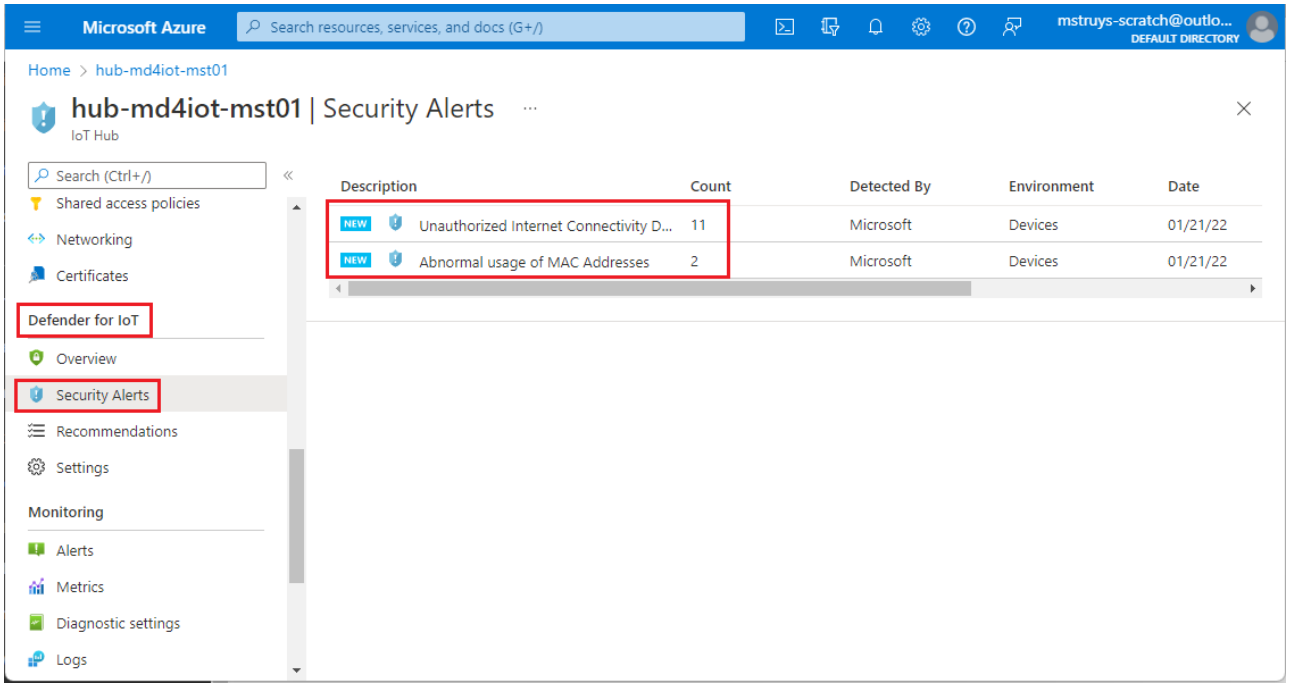
7. Close the screen, open again the **Reactivation** window and double check if your sensor is **Cloud Connected** as shown below:



8. Run the Pcap files again in your console. In a few minutes you can verify if IoT Hub in Azure Portal on your physical machine is receiving messages from your sensor:



9. In the same IoT Hub now you should see the alerts generated by Defender for IoT. Scroll down to **Defender for IoT**, select **Security Alerts**, on the right side you will see some alerts already available.



Exercise 6: Integrate with Sentinel

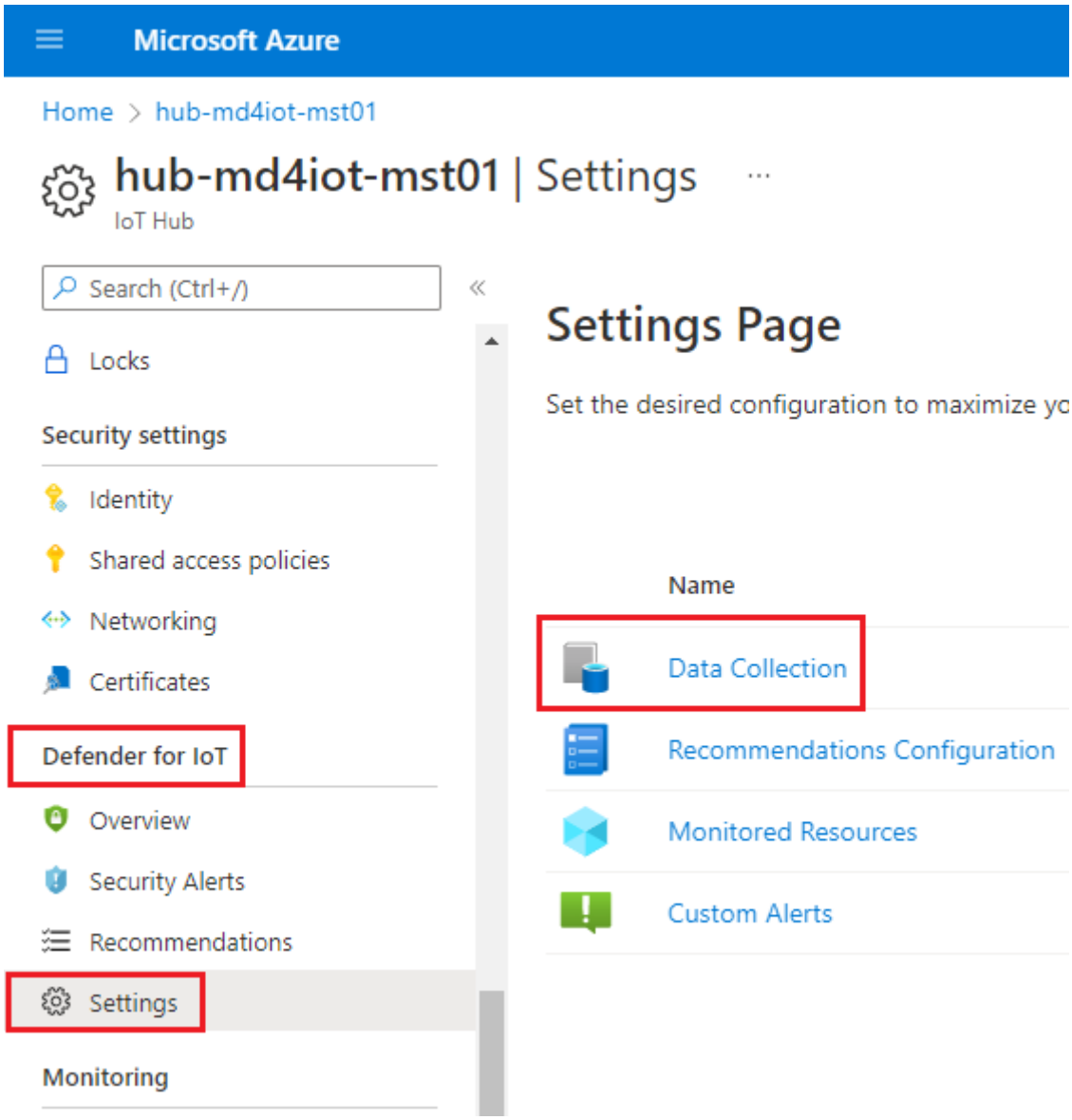
You will execute most of this task on your physical machine, not in the Virtual Machine that hosts your your Microsoft Defender for IoT sensor.

> **Note**: Please ensure you have completed Task 6 in the 'Before HOL' instructions prior to working through the following tasks.

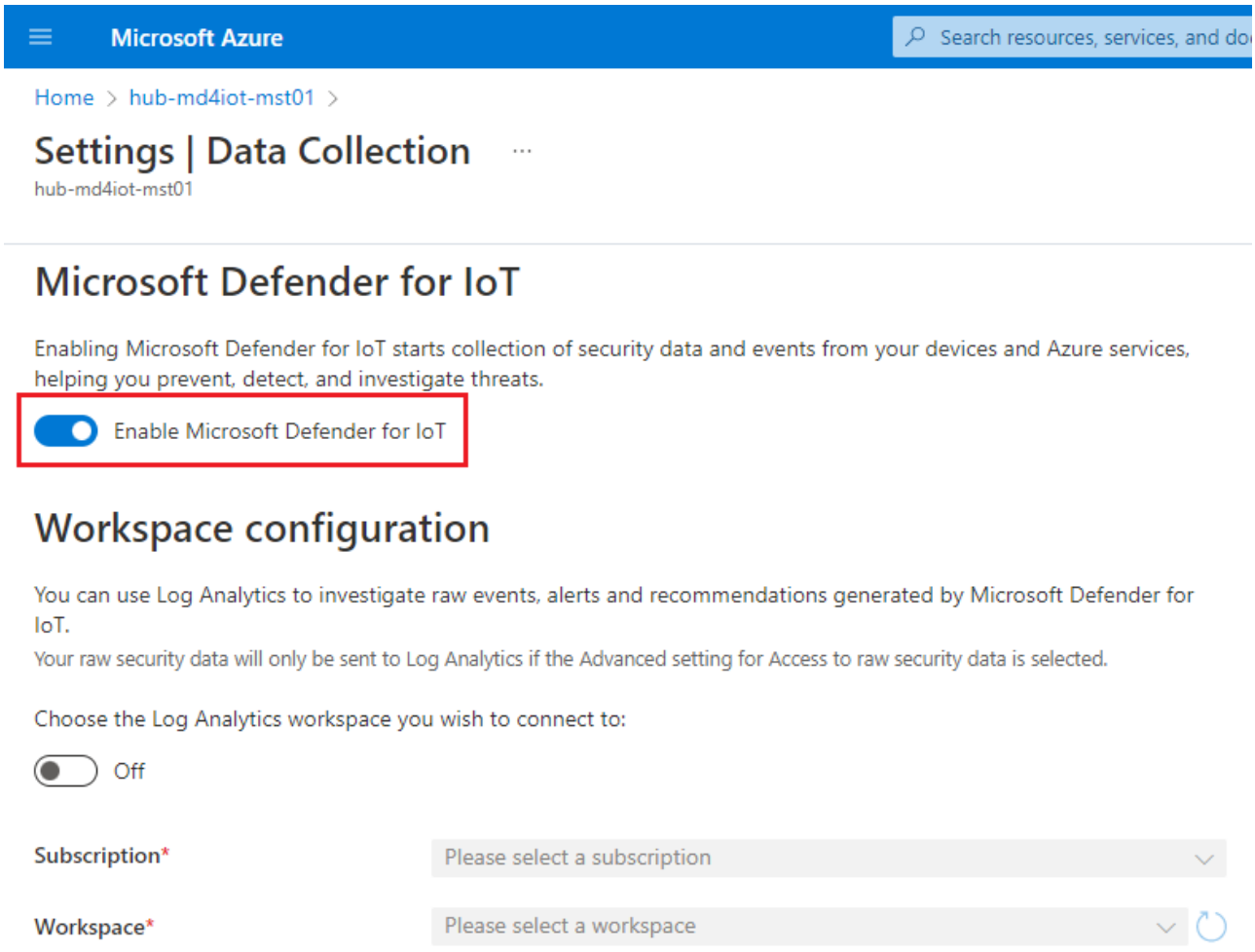
Task 1: Enabling IoT to Integrate with Sentinel

1. Ensure your IoT Hub is configured to send Security Alerts to Sentinel.

2. Navigate to your IoT Hub > Defender for IoT > Settings > Data Collection

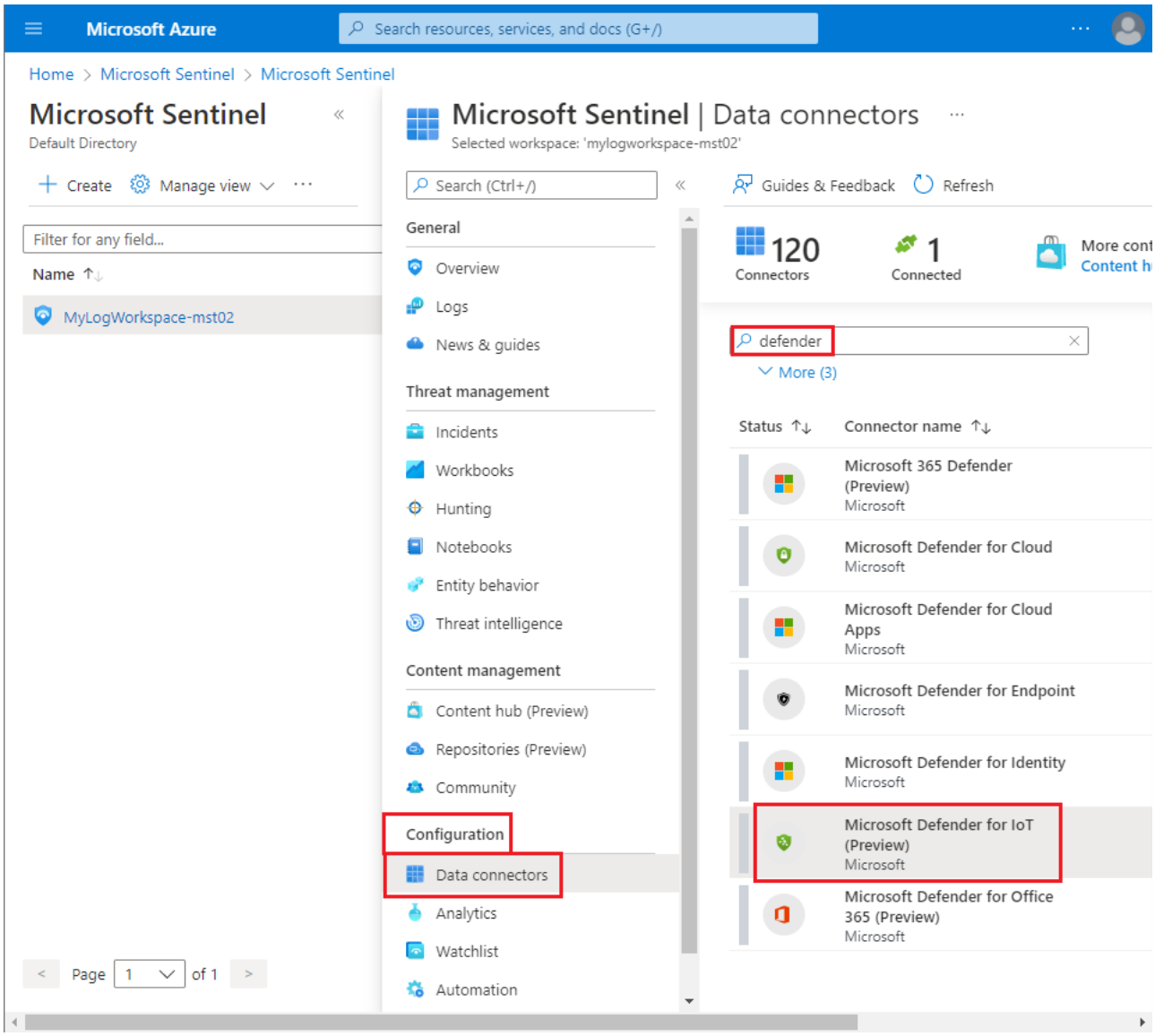


3. Double check that Data Collection blade, is enabled for **Enable Microsoft Defender for IoT**

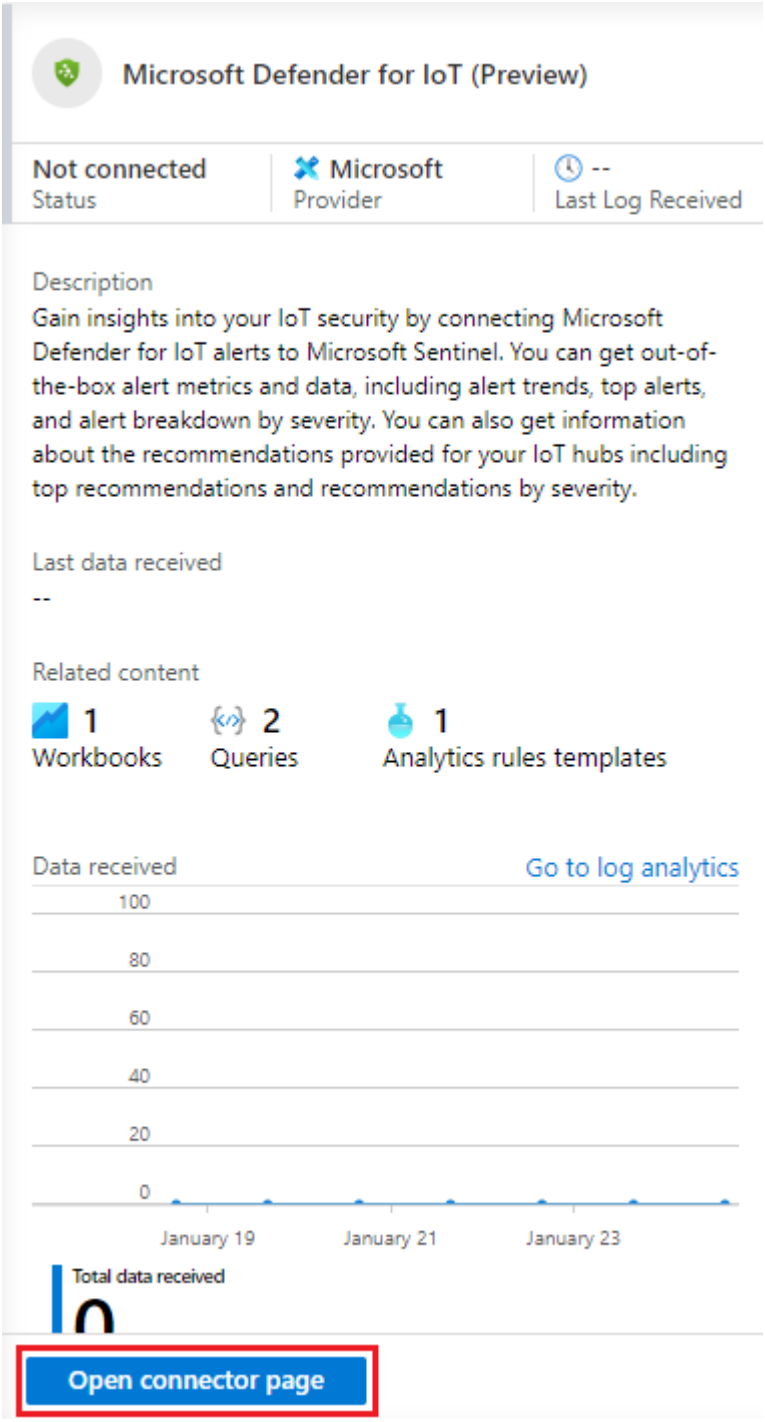


Task 2: Connecting Data Connectors

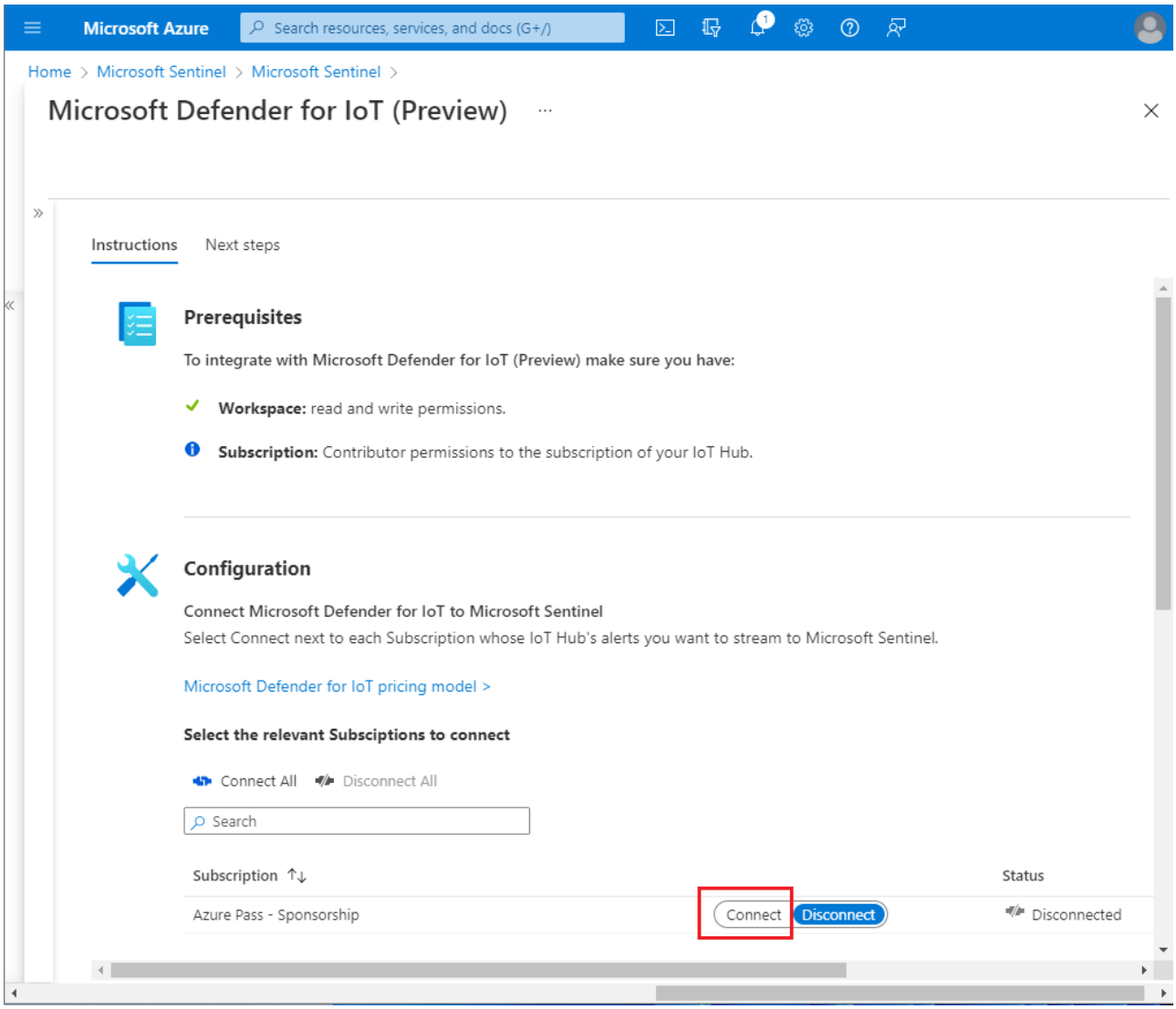
1. With the *Microsoft Defender for IoT* switch enabled, go to **Microsoft Sentinel** > Configuration > Data Connectors > Search **Microsoft Defender for IoT** to connect Microsoft Defender for IoT to Microsoft Sentinel.



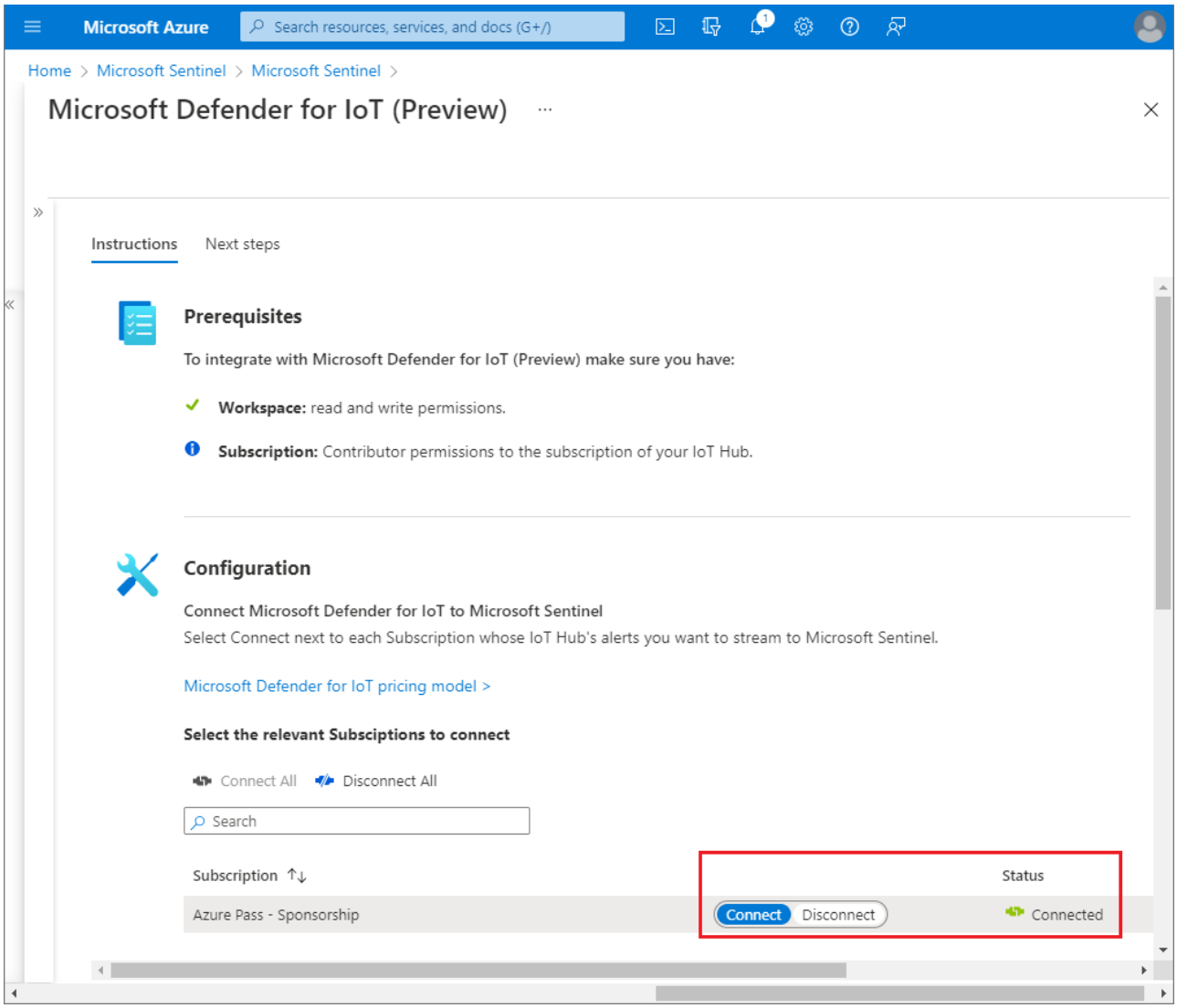
2. Click the **Open Connector Page**



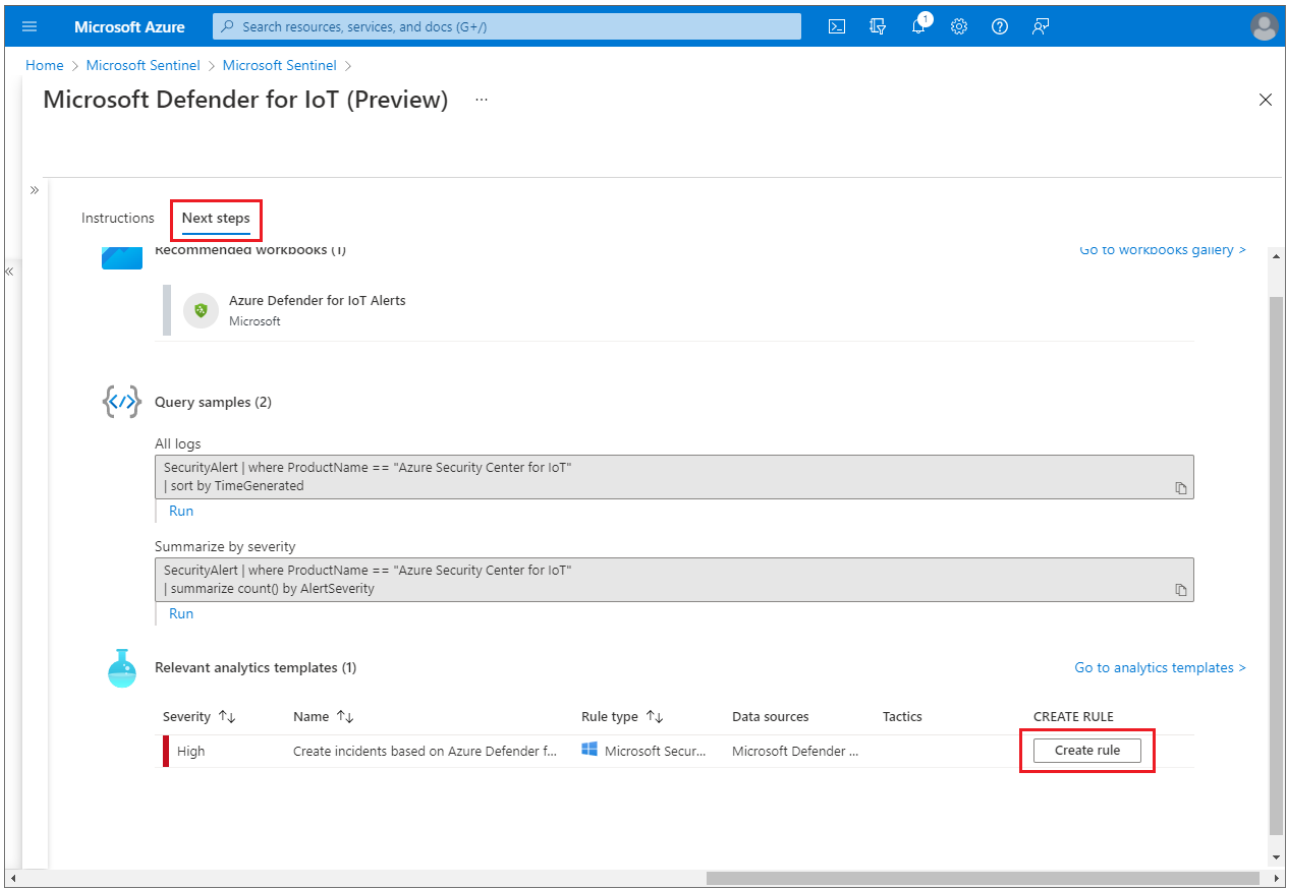
3. Review the instructions and click the “Connect” button to connect Microsoft Defender for IoT to Sentinel. If the connection continues to fail, this will most likely be due to the user not having the "Contributor" permissions and you may have missed the access step in the prerequisites.



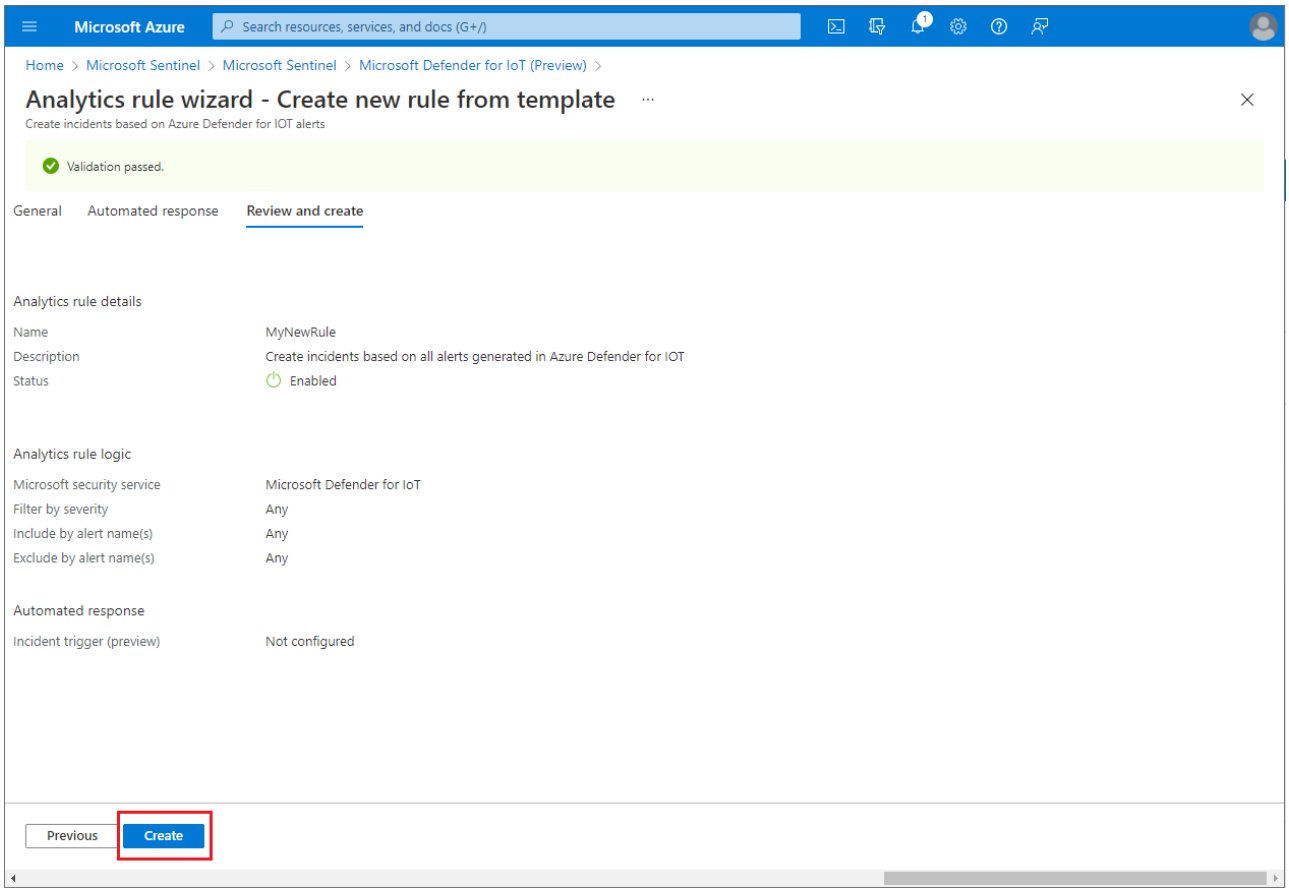
4. If connected correctly you should expect to see the Status change to “Connected” and the link light up green.



5. Use the next steps tab to enable Out of the Box alerts. For example, click the create rule and follow the instructions to turn on the rule.

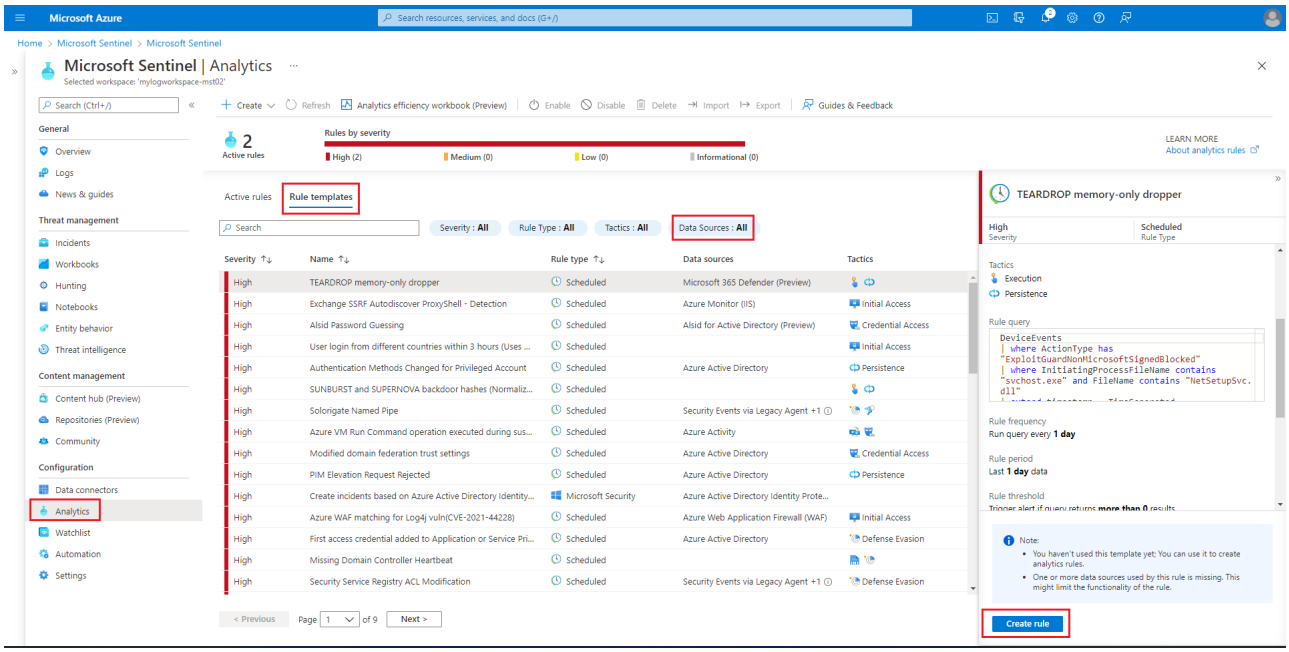


6. Fill in the “Name” and click **Review and Create**, followed by **Create**. This is enabling incidents to be created based on the Azure Defender IoT alerts that are ingested into Sentinel.



7. Additionally, you can create the rule not only on the data connectors page but also on the Microsoft Sentinel “Analytics” blade. See an example below when you go to the “Rule Templates” tab and filter

data sources by “Microsoft Defender for IoT (Preview)”.

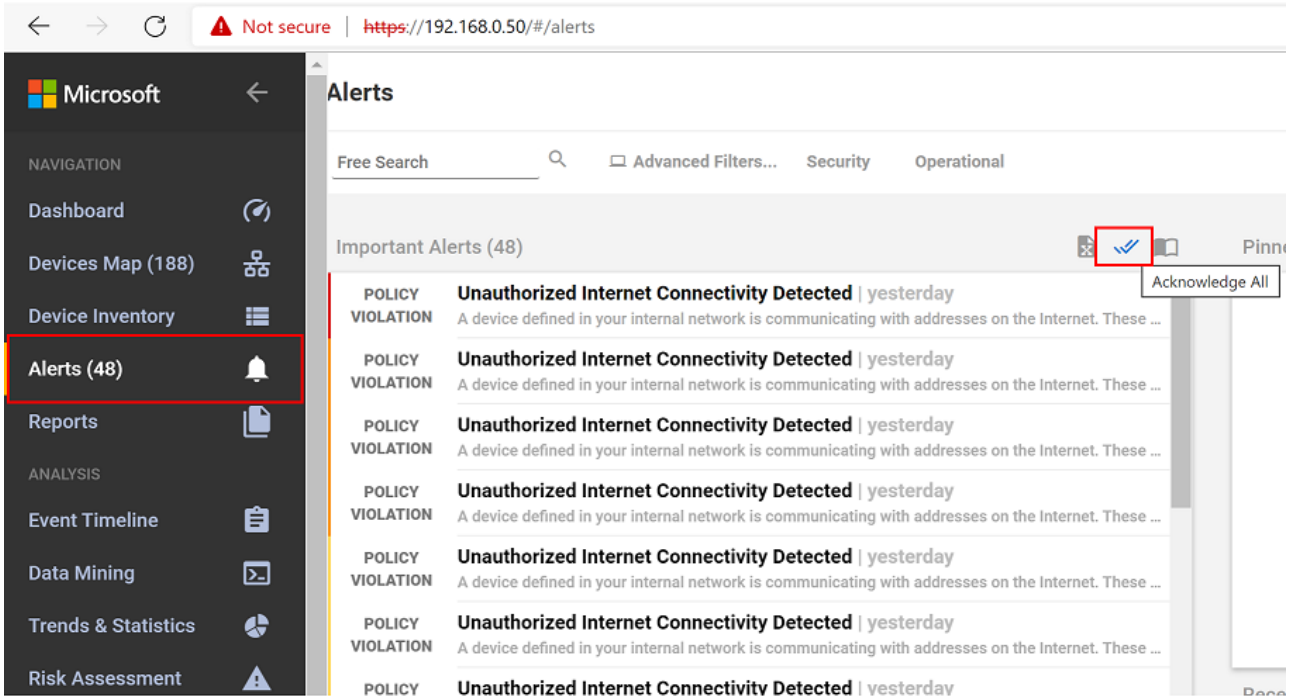


Task 3: Acknowledge Alerts and Re-run PCAPs

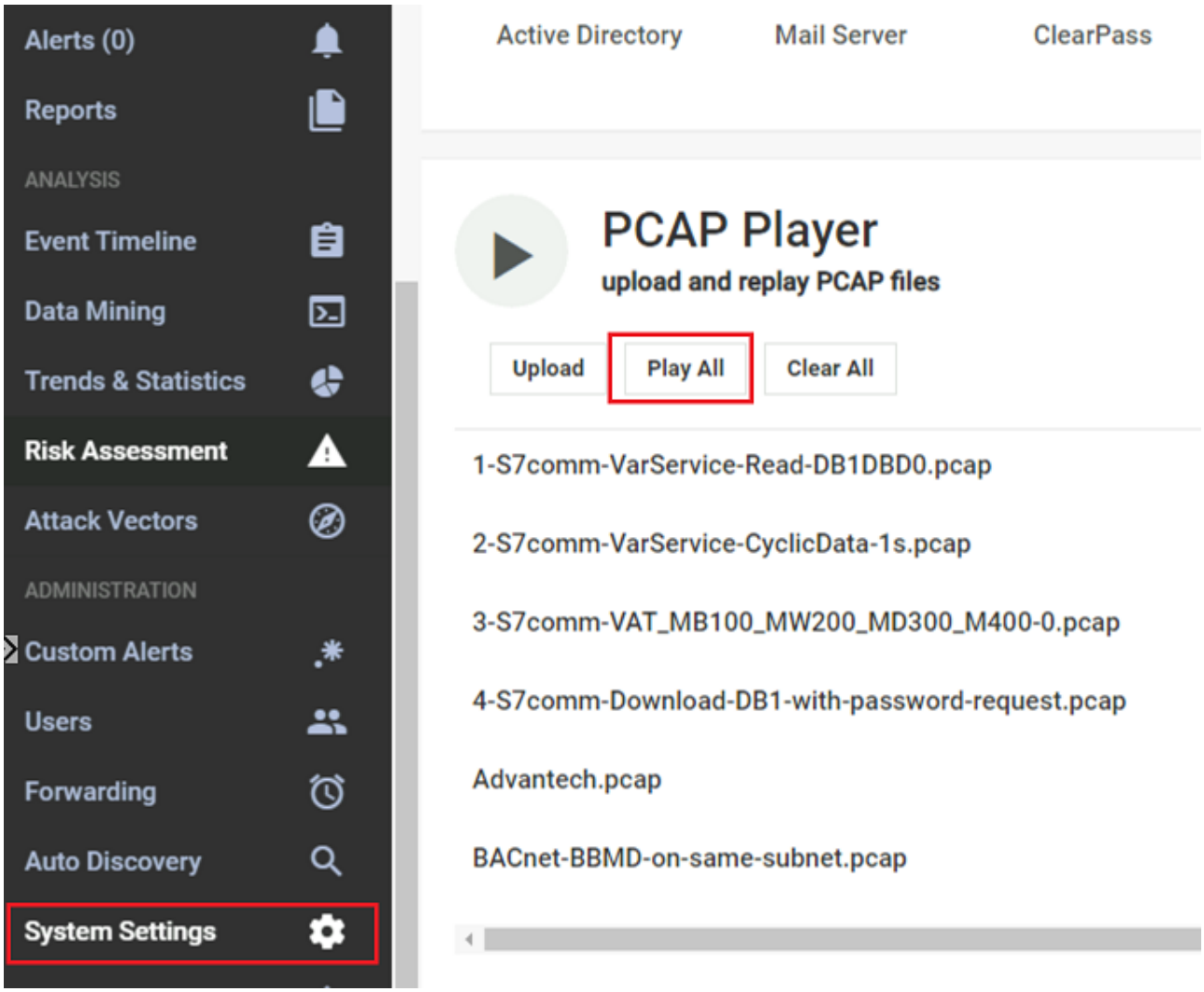
You will execute most of this task on the Virtual Machine that hosts your your Microsoft Defender for IoT sensor.

1. Go back to your browser interface and acknowledge all of the alerts. The reason we are doing this is so we can re-run the alerts to show how they are sent and analyzed by Sentinel.

    1. Navigate to the Alerts Page
    2. Click the double check box
    3. Click **Ok** to acknowledge the alerts



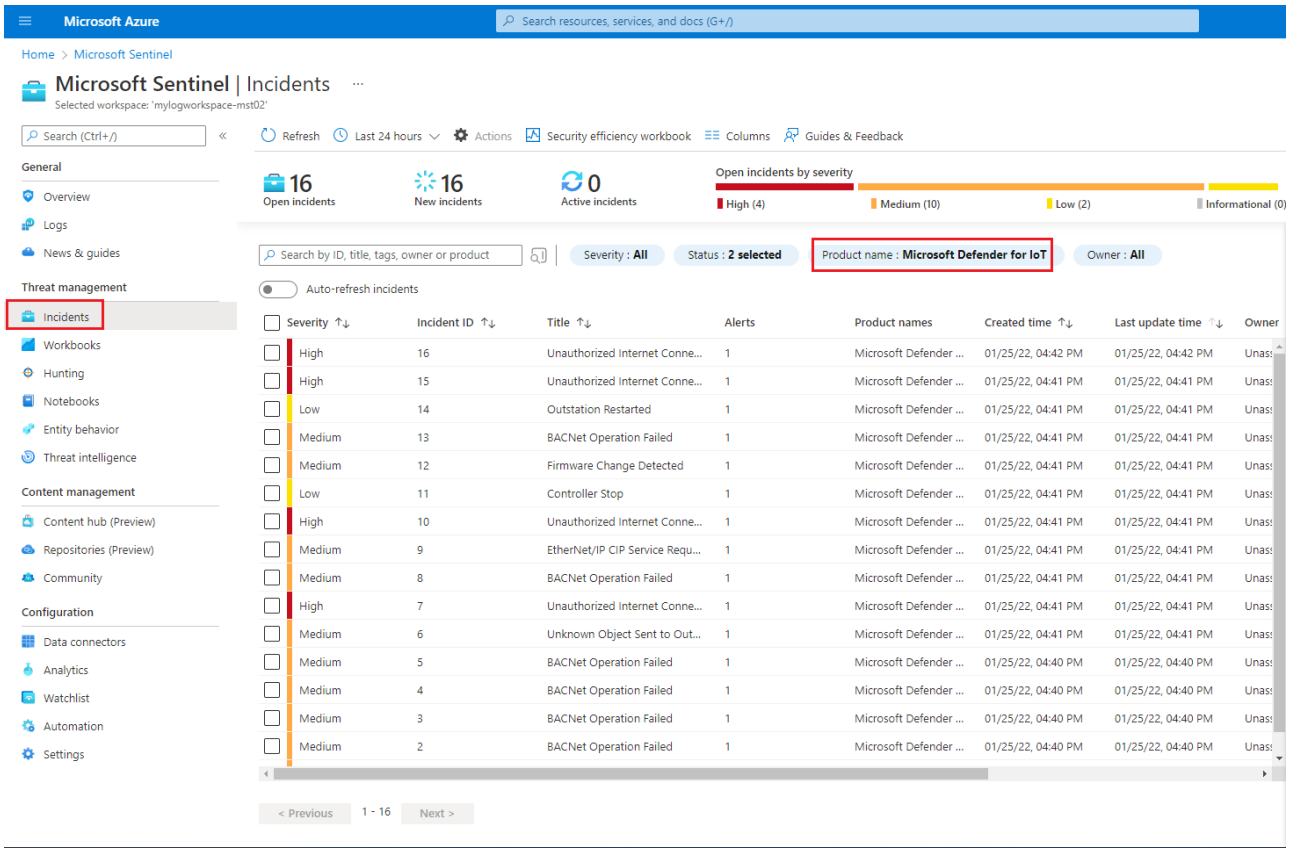
    1. Now go to the System Setting tab.
    2. Click the **Play All** on the PCAP Files to replay simulating the alerts.



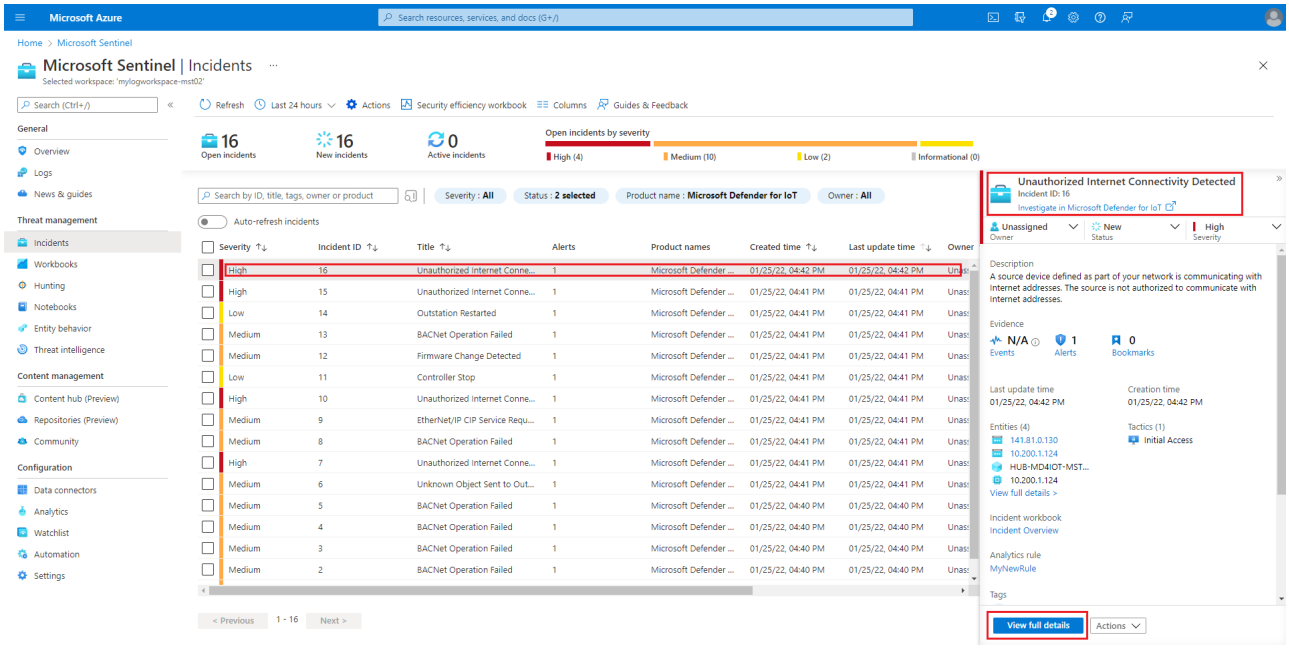
Task 4: Sentinel interaction with IoT Incidents

You will execute most of this task on your physical machine, not in the Virtual Machine that hosts your your Microsoft Defender for IoT sensor.

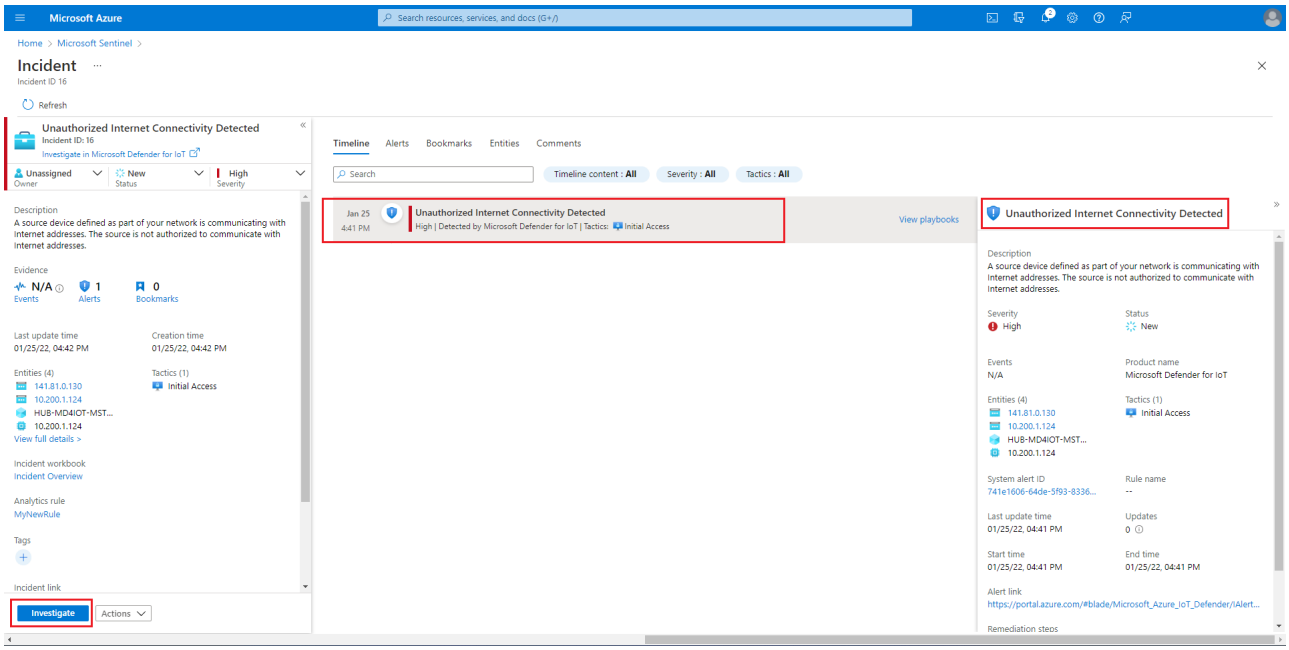
1. Go back to the Sentinel console and under the **Threat Management** section, select the **Incidents** tab. Filter by Product Name **Azure Defender for IoT**.



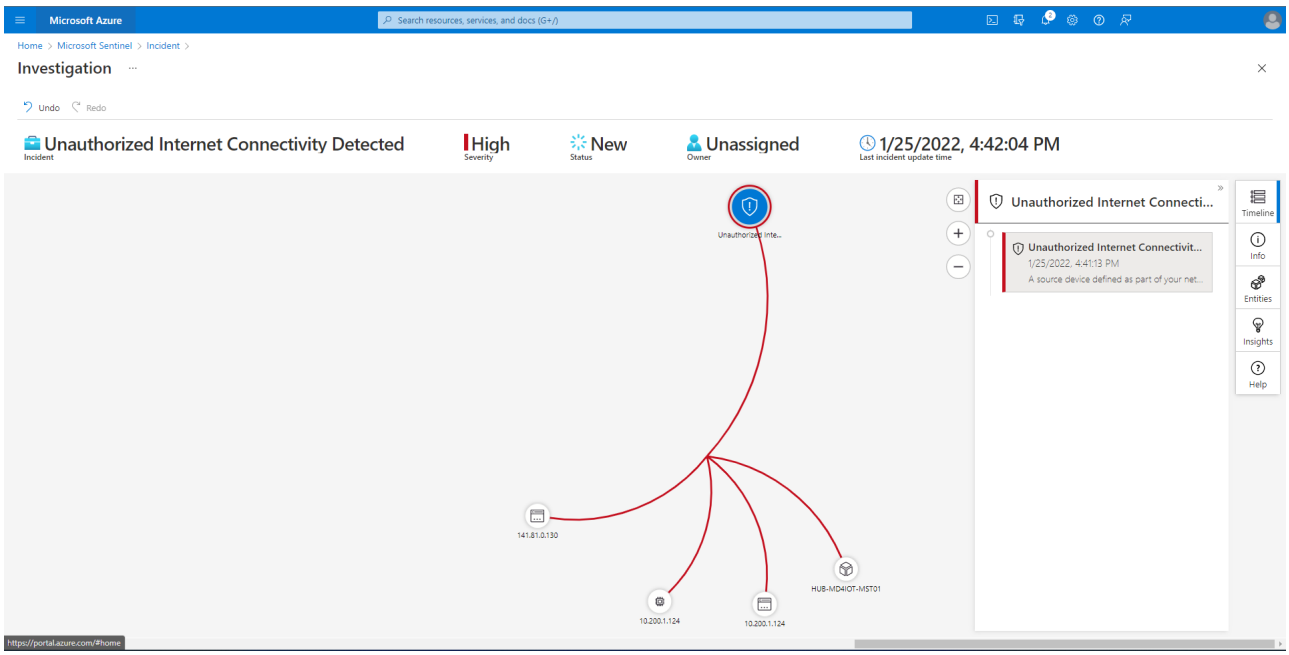
2. Select one of the alerts and click **View full details**



3. It will take you to this screen to get all the information relative to the incident. This allows analyst to get more details on the entity including what other alerts made up the incident, playbooks to enrich the context of the alert, and comments section to leave details on what the analyst discovered during review or how they came to the determination to dismiss the incident.



4. By clicking the **Investigate** button, you can dig deeper in the cause of the incident and the relation to other incidents.

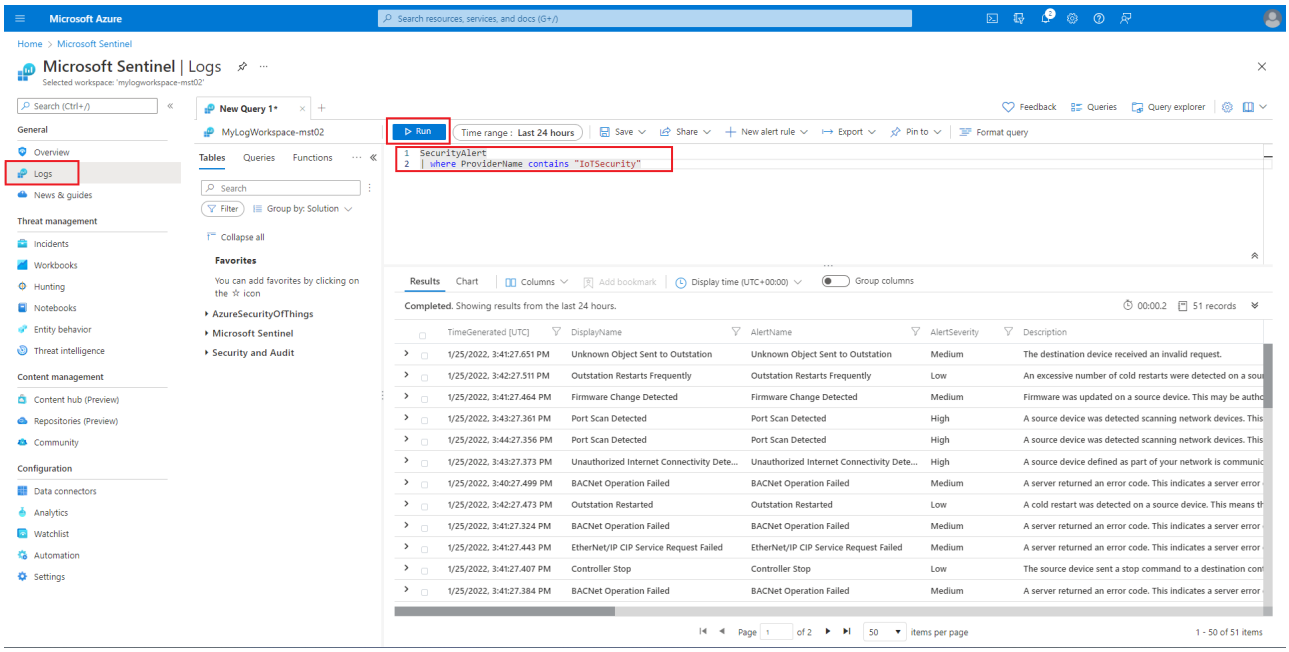


Task 5: Kusto Query Language to Find Alert Details

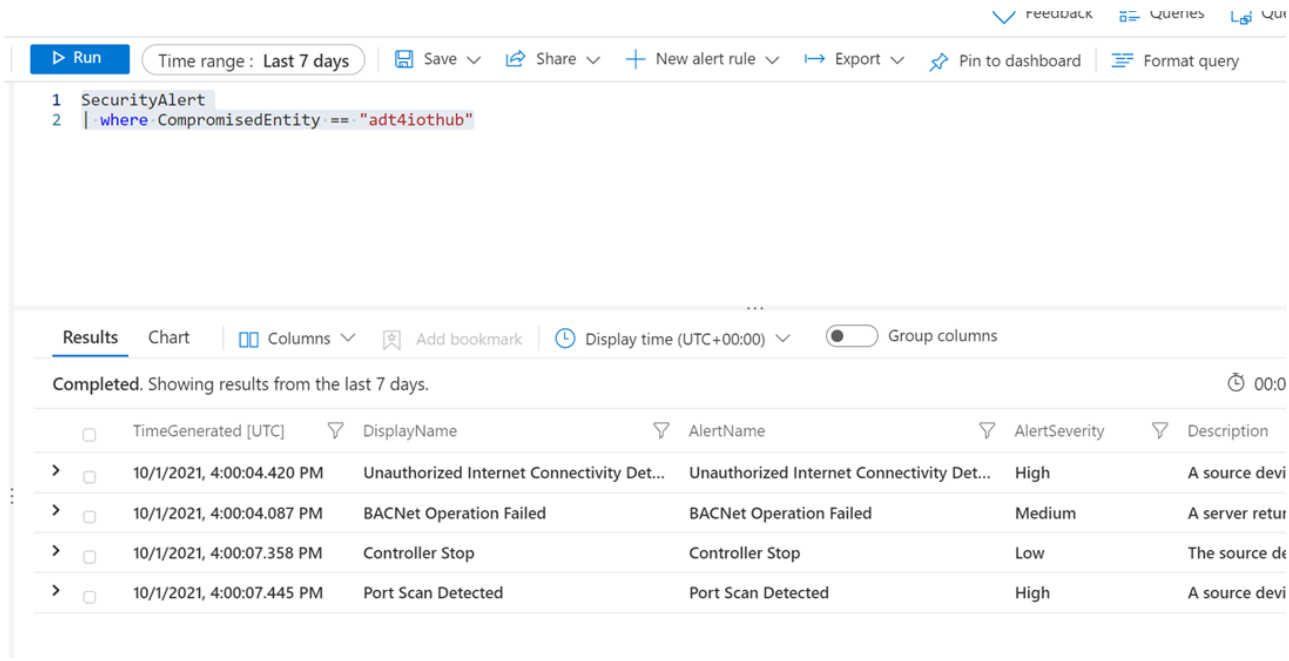
1. Navigate to the “Logs” tab and run this query. Querying the data will provide the ability to join tables and datasets to curate data from multiple sources. KQL is a similar language to SQL but will take some research and some dedicated time to become familiar with.

   Here are two basic examples:

```
SecurityAlert | where ProviderName contains "IoTSecurity"
```




```
SecurityAlert | where CompromisedEntity == "hub-md4iot-mst01"
```



Exercise 7: Clean Up

Task 1: Delete resources

The Azure Passes will allow you to run the services for 90 days for training purposes. Although it is a best practice to delete all your resources after the training.

Search for the Resource Group created for this training.

Select Delete resource group on the top right side.

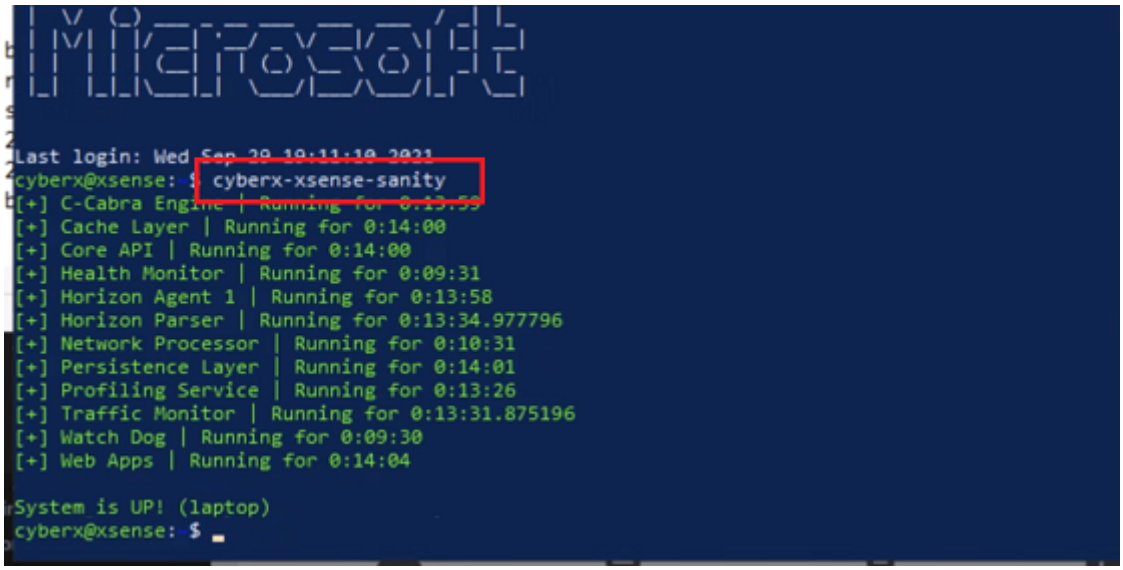
Enter your-resource-group-name for **TYPE THE RESOURCE GROUP NAME** and select Delete. This operation will take a few minutes.

After that is done go to Microsoft Defender for IoT and deactivate the subscription.

Appendix 1: Troubleshooting

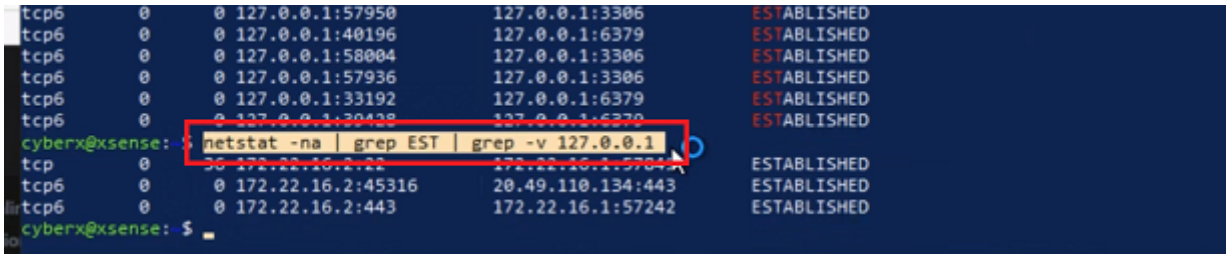
1. If your Defender portal is not working properly run the following command to validate if the components are running properly

```
cyberx-xsense-sanity
```

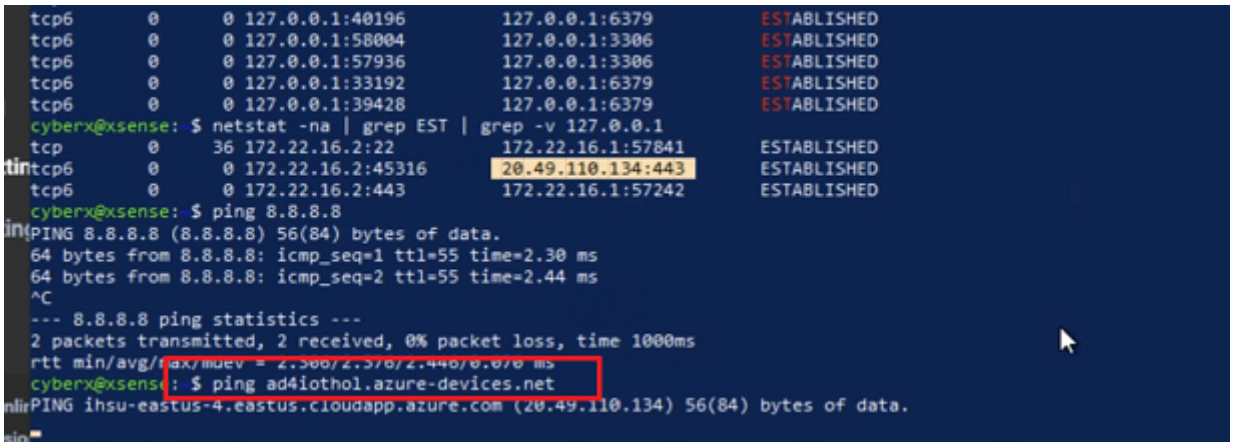


2. If your IoT hub is not receiving messages, check if ubuntu machine can reach IoT Hub, first run the following command to identify the IP of your IoT Hub:

```
netstat -na | grep EST | grep -v 127.0.0.1
```



Then, ping the IoT Hub using the connection string from the overview blade in Azure Portal.


tcp6 0 0 127.0.0.1:40196 127.0.0.1:6379 ESTABLISHED
tcp6 0 0 127.0.0.1:58004 127.0.0.1:3306 ESTABLISHED
tcp6 0 0 127.0.0.1:57936 127.0.0.1:3306 ESTABLISHED
tcp6 0 0 127.0.0.1:33192 127.0.0.1:6379 ESTABLISHED
tcp6 0 0 127.0.0.1:39428 127.0.0.1:6379 ESTABLISHED
cyberx@xsense: $ netstat -na | grep EST | grep -v 127.0.0.1
tcp 0 36 172.22.16.2:22 172.22.16.1:57841 ESTABLISHED
tcp6 0 0 172.22.16.2:45316 20.49.110.134:443 ESTABLISHED
tcp6 0 0 172.22.16.2:443 172.22.16.1:57242 ESTABLISHED
cyberx@xsense: $ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=2.30 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=2.44 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
$ ping ad4iothol.azure-devices.net
PING ihsu-eastus-4.eastus.cloudapp.azure.com (20.49.110.134) 56(84) bytes of data.