Set up Azure IoT Edge for Azure Sphere

- Article
- 04/13/2023
- 10 contributors

Azure IoT Edge provides a filtering and data processing layer between a downstream device, like Azure Sphere and Azure IoT Hub. Consider using Azure IoT Edge if your Azure Sphere device produces a considerable amount of data or data that requires post-processing.

This topic documents the steps to set up an Azure IoT Edge device with Azure Sphere. Azure Sphere supports both versions 1.1 and 1.2 of Azure IoT Edge; significant differences are noted in the instructions. The main differences are:

- The package name changed from `iotedge` to `aziot-edge`.
- The default config file has a new name and location. In Azure IoT Edge 1.1, the config file was */etc/iotedge/config.yaml*. In Azure IoT Edge 1.2, the config file is */etc/aziot/config.toml*.
- Azure IoT Edge 1.2 requires a fully qualified domain name (FQDN) accessible via a DNS server on the network.

After you have completed the tasks in this topic, your Azure Sphere device will be configured to send data to an Azure IoT Hub through an Azure IoT Edge device that acts as a transparent gateway. You can add additional data filtering and processing with a module on the Azure IoT Edge device by following the Azure IoT Edge Module guide.

Before you begin

The steps in this topic assume that:

- Your Azure Sphere device is connected to your PC by USB.
- You have an Azure subscription.
- You have created an Azure IoT Hub instance and manually provisioned a device. Note that you'll need to use the tenant CA certificate for the IoT Hub later in the procedure (Step 6).
- You have created a resource group, and all resources related to the Azure IoT Hub and IoT Edge services must belong to this resource group.

Important

Although you can create an Azure subscription for no charge, the sign-up process requires you to enter a credit card number. Azure provides several levels of subscription service. By default, the Standard tier, which requires a monthly service charge, is selected when you create an Azure IoT Hub instance. To avoid a monthly charge, select the Free tier. The Free tier includes the services required to use your device with an Azure IoT Hub instance, including the **Device Twin**. If you don't have an Azure subscription, **create a free account** before you begin.

Overview

Setting up an Azure IoT Edge device and configuring it to work with an Azure Sphere device requires a multi-step process and you should plan about 8 hours to work through the steps, especially if you are not familiar with Azure IoT Edge. If this is your first time working with Azure IoT Edge, follow along with the Quickstart instructions in each section to set up and configure an IoT Edge device that runs on a Linux virtual machine.

These steps must be completed only once per Azure Sphere tenant and Azure IoT Edge device; however, each Azure Sphere device must be manually configured in Azure IoT Hub, and the Azure IoT Edge device must be set as the parent of the Azure Sphere device.

Setup steps can be broken into three logical groupings:

- **Create and configure the IoT Edge device as a transparent gateway**
    - Step 1. Create an Azure IoT Edge device.
    - Step 2. Configure the Azure IoT Edge gateway device as a server.
    - Step 3. Open Azure IoT Edge gateway device ports for communication.
    - Step 4. Update the gateway hostname in config.toml (Azure IoT Edge version 1.2) or config.yaml (Azure IoT Edge version 1.1).
- **Configure the Azure Sphere device in Azure IoT Hub**
    - Step 5. Set the Azure IoT Edge device as parent of the Azure Sphere device.
- **Establish a trusted connection between the Azure Sphere device and the IoT Edge device**
    - Step 6. Provide the tenant CA certificate of the Azure Sphere device to the Azure IoT Edge device.
    - Step 7. Provide the root CA certificate of Azure IoT Edge device to the Azure Sphere device.

Step 1. Create an Azure IoT Edge device

You must set up an Azure IoT Edge device and register it with Azure IoT Hub, if you have not already done so.

You can use the Device Provisioning Service (DPS) to provision the Azure IoT Edge device. However, you cannot use DPS to provision the Azure Sphere device. Automatic provisioning of devices downstream of the Azure IoT Edge device with the Azure IoT Hub DPS is not supported.

You can follow the steps in the IoT Edge Quickstart to set up an Azure IoT Edge device that runs on a Linux VM or Windows device and register it with Azure IoT Hub.

Use the steps in the Quickstart to:

- Register an Azure IoT Edge device to your Azure IoT Hub instance. Do not create a new IoT hub in this step, just register your Azure IoT Edge device to the IoT hub you have already created.
- Install and start the Azure IoT Edge runtime on your Azure IoT Edge device.

Note

In the IoT Edge Quickstart, stop before deploying a module.

Step 2. Configure the IoT Edge gateway device as a server

Follow the instructions to Configure an Azure IoT Edge device to act as a transparent gateway (Azure IoT Edge 1.2) or Azure IoT Edge 1.1, but stop when you reach the section "Open ports on gateway device." Although these instructions tell you to copy the certificate files to your IoT Edge device, do not copy the files to your VM until Step 8 of this procedure.

As part of the steps you completed to configure the device gateway as a server, you will have already:

- Configured the Azure IoT Edge device certificates.
- Deployed the Azure IoT Edge hub module.
- Configured routing of messages through the Azure IoT Edge.

Also as part of these steps, you will have created these certificates:

- Root CA certificate: `certs/azure-iot-test-only.root.ca.cert.pem`
- Device CA certificate and private key (used to register the IoT Edge device to the IoT Hub):
    - `certs/iot-edge-device-identity-<cert-name>-full-chain.cert.pem`
    - `private/iot-edge-device-identity-<cert-name>.key.pem`
- IoT Edge CA certificate and private key (to be copied to an IoT Edge device and referenced in its config file):
    - `certs/iot-edge-device-ca-<cert-name>-full-chain.cert.pem`
    - `private/iot-edge-device-ca-<cert-name>.key.pem`

If you are following the Quickstart, use the Linux instructions for configuring the demo certificates if the machine on which you are generating the certificates is a Linux-based computer. Use the Windows instructions to generate the demo certificates if the machine on which you are generating the certificates is a Windows-based computer. See the section to copy root certificates to a Linux VM or to a local Windows computer.

The Azure IoT Edge root CA certificate will be modified in Step 7, but you will need the original certificate in Step 8. Keep a copy of the original certificate so you can reuse it.

Note

When updating the certificates section of the config.toml (Azure IoT Edge 1.2) or config.yaml (Azure IoT Edge 1.1) file, make sure that the `certificates:` line in the config.toml (Azure IoT Edge 1.2) or config.yaml (Azure IoT Edge 1.1) has no preceding whitespace and that each of the nested certificates is indented by two spaces.

Once you have updated the file, verify that the **edgehub** module is running on your Azure IoT Edge device:

```
sudo iotedge list
```

For more information, see How an IoT Edge device can be used as a gateway (Azure IoT Edge 1.2) or Azure IoT Edge 1.1.

If you are using test certificates, stop before generating the downstream device certificate, as documented in Create downstream device certificates (Azure IoT Edge 1.2) or Azure IoT Edge 1.1.

Step 3. Open Azure IoT Edge gateway device ports for communication

Gateway devices must be able to receive messages from their downstream devices. For a gateway scenario to work, at least one of the IoT hub's supported protocols must be open for inbound traffic from downstream devices.

Azure Sphere uses protocol MQTT. This protocol uses port 8883.

For more information see Open ports on gateway device (Azure IoT Edge 1.2) or Azure IoT Edge 1.1.

Follow these steps to open port 8883 on a Windows VM after setting up an Edge device in the Quickstart:

1. If necessary, log in to your Azure portal using your Azure account.
2. Navigate to the virtual machine you created in the section Step 1: Create an Azure IoT Edge device.
3. In the **Settings** section at left, select **Networking**, then **Add inbound port rule**.
4. In the **Add inbound port rule** blade, under **Destination port ranges**, change the value to **8883**.
5. Under **Protocol**, select **TCP**.
6. Under **Name**, change the value to **MQTT**.
7. Leave all other settings as the default, and then select **Add**.

Follow these steps to open port 8883 on a Linux VM after setting up an Azure IoT Edge device in the Quickstart:

1. Run the following command to check whether the port for MQTT is open:

   ```
   sudo netstat -lptu
   ```

2. If necessary, use the following command to open the port:

   ```
   sudo ufw allow 8883
   ```

This will define an inbound security rule to allow communication for the MQTT protocol to the Azure IoT Edge Gateway.

Step 4. Update the gateway hostname

This step uses different procedures depending on the version of Azure IoT Edge you are using: Azure IoT Edge 1.2 or Azure IoT Edge 1.1.

Azure IoT Edge 1.2: Update the gateway hostname in config.toml

The Azure IoT Edge runtime supports hostnames that are less than 64 characters. Physical machines usually do not have long hostnames, but if you are using a virtual machine as the Azure IoT Edge device, like the example in the Quickstart, you must manually configure the hostname.

To troubleshoot an invalid hostname, see common error resolutions.

Follow these steps to configure the Azure IoT Edge device hostname after configuring the MQTT port in the Quickstart:

1. Find the fully qualified domain name (FQDN) for your IoT Edge gateway by navigating to your IoT Edge device (Linux VM) in the Azure Portal and copying the value for **DNS name** from the overview page.
2. If necessary, log in to the Azure IoT Edge device.
3. Open the config.toml file in a text editor.

   ```
   /etc/aziot/config.toml
   ```

4. Paste the FQDN into the hostname section of *config.toml*. Make sure that the name is all lowercase.

   tomlCopy
   ```
   hostname: "<iotedge_machinename>.<mydomain>"
   ```
   Note

   By design with Azure IoT Edge v1.2 and above, *hostname* MUST be an FQDN name (an IP address is not allowed anymore, as in v1.1); therefore, a DNS server with the appropriate A record on the same network is mandatory.

5. Restart the `iotedge` daemon.

   ```
   sudo systemctl restart iotedge
   ```

6. If you see errors (colored text prefixed with "[ERROR]") in the status, examine daemon logs for detailed error information.

   ```
   sudo journalctl -u iotedge --no-pager --no-full
   ```

7. To avoid warnings, set up the DNS configuration for modules on the Azure IoT Edge device to include a configuration file at */etc/docker/daemon.json*, for example:

   JSONCopy
   ```
   {
        "dns": ["<IP address of your DNS server>"]
   }
   ```

Azure IoT Edge 1.1: Update the gateway hostname in config.yaml

The Azure IoT Edge runtime supports hostnames that are less than 64 characters. Physical machines usually do not have long hostnames, but if you are using a virtual machine as the Azure IoT Edge device, like the example in the Quickstart, you must manually configure the hostname.

To troubleshoot an invalid hostname, see common error resolutions.

Follow these steps to configure the Azure IoT Edge device hostname after configuring the MQTT port in the Quickstart:

1. In the Azure portal, navigate to your virtual machine. Copy the value for **DNS name** (FQDN of the machine) from the **Overview** section.
2. If necessary, log in to the Azure IoT Edge device.
3. Open the *config.yaml* file in a text editor.

   ```
   /etc/iotedge/config.yaml
   ```

4. Paste the FQDN into the hostname section of *config.yaml*. Make sure that the name is all lowercase.

   YAMLCopy
   ```
   `hostname: "<machinename>.<region>.cloudapp.azure.com"`
   ```
   Note

   You might have to use the bare machine name for the hostname (either the IP address or actual hostname) if a DNS resolver is on the network, since Azure Sphere does not support netbios.

5. Restart the `iotedge` daemon.

   ```
   sudo systemctl restart iotedge
   ```

6. If you see errors (colored text prefixed with "[ERROR]") in the status, examine daemon logs for detailed error information.

   ```
   sudo journalctl -u iotedge --no-pager --no-full
   ```

7. To avoid warnings, set up the DNS configuration for modules on the Azure IoT Edge device to include a configuration file at */etc/docker/daemon.json*, for example:

   JSONCopy

   ```
   {
       "dns": ["<IP address of your DNS server>"]
   }
   ```

Step 5. Set the Azure IoT Edge device as parent of the Azure Sphere device

Follow these steps to set the Azure IoT Edge device as the parent of the Azure Sphere device:

1. Navigate to the Azure Sphere device that was manually provisioned above.
2. Select **Device ID**.
3. Select the gear icon under **No parent device**.
4. Select the Azure IoT Edge device that you want to set as the parent.
5. Select **OK**, then **Save**.

The Azure IoT Edge device now becomes the parent of the Azure Sphere device.

Step 6. Provide the tenant CA certificate of the Azure Sphere device to the IoT Edge device

To verify Azure Sphere device certificates, the Azure IoT Edge device must have its own copy of the tenant CA.

1. Download the tenant CA certificate, if you have not already done so. Note: You should have already completed this as part of configuring your Azure IoT Hub.
    - From the command prompt, sign in with your Azure Sphere login:

      Azure Sphere CLICopy
      ```
      azsphere login
      ```

    - Download the CA certificate for your Azure Sphere tenant. This command downloads the certificate to a file named *CAcertificate.cer* in the current working directory. Ensure that you download the file to a directory in which you have write permission, or the download operation will fail. The output file must have a .cer extension.
        - Azure Sphere CLI
        - Azure Sphere classic CLI

      Azure Sphere CLICopy
      ```
      azsphere ca-certificate download --destination
      CAcertificate.cer
      ```

2. Convert the tenant CA certificate to PEM format. Example instructions for converting the format on a Windows machine are as follows:
    - Locate the path of the certificate on your computer and double-click the certificate to open it.
    - Open the **Details** tab and select **Copy to File**.
    - In the **Certificate Export** wizard, select **Next**.
    - Select the Base-64 encoded X.509 (CER) format, then select **Next**.
    - Enter the file name for the certificate to export, then select **Next**.
    - Select **Finish** to complete the wizard.
    - Rename the downloaded certificate file to have the .pem file extension.
3. Append the tenant certificate to the Azure IoT Edge root certificate. Remember that you created the Azure IoT Edge certificates in Step 2.
    - If necessary, log in to the Azure IoT Edge device.
    - Locate the path of the root CA certificate of IoT Edge device and open it in a text editor. If you followed the Quickstart then used the scripts provided in the Azure IoT Edge git repository to

create demo certificates, the root CA certificate is named *azure-iot-test-only.root.ca.cert.pem*.
- Open the Azure Sphere Tenant CA certificate (in PEM format) in a text editor. Copy the content and paste it at the end of the IoT Edge root CA certificate.
- Save the changes made to the Azure IoT Edge root CA certificate, then close the file.
- Restart the Azure IoT Edge device. For a Linux Azure IoT Edge device, run: `sudo systemctl restart iotedge`.
- Verify that the modules are running on your Azure IoT Edge device. For a Linux Azure IoT Edge device, run: `sudo iotedge list`.

Step 7. Provide the root CA certificate of Azure IoT Edge device to the Azure Sphere device

To verify the Azure IoT Edge device certificates, the Azure Sphere device (the downstream device) must have its own copy of the Azure IoT Edge device root CA certificate, which you configured in Step 2.

For more information see Connect a downstream device to an Azure IoT Edge gateway.

1. Locate the original Azure IoT Edge device root certificate.
    - Remember that the original Azure IoT Edge device root certificate is the one you created in Step 2.
    - If you followed the Quickstart and then used the scripts provided in the IoT Edge git repository to create demo certificates, the root CA certificate is called *azure-iot-test-only.root.ca.cert.pem*.
2. Copy the Azure IoT Edge root certificate to the Azure Sphere device by adding it to the application package.
    - For detailed steps, see Image package creation.
    - See the section to copy root certificates to a Linux VM or to a local Windows computer.

Step 8. Copy the IoT root certificate to a remote machine

Follow these steps to copy the Azure IoT Edge root certificate to a remote Linux VM or to a local Windows machine:

- You can install the WinSCP tool to transfer the files. The tool provides a GUI so it's easier to use than a command-line approach.
- To use the command line, use the SCP (Secure Copy) command-line tool.
- To transfer a file from a local Windows machine to a remote Linux VM, run the following command in Windows PowerShell:

  PowerShellCopy
  ```
  powershell -Command scp -r <path-to-file-name> <userName@remote-ip>:<path-to-destination-directory>
  ```

  Sample syntax:

  ```
  scp C:\Documents\cert.pem AzureUser@edgevmname-west.westus22.cloudapp.azure.com:/home/azureUser/test/
  ```

- To transfer a file from a remote Linux VM to a local Windows machine, run the following command in Windows PowerShell:

  PowerShellCopy
  ```
  powershell -Command scp -r <userName@remote-ip>:<path-to-file-name> <path-to-destination-dir>
  ```

  Sample syntax:

  ```
  scp azureUser@edgevmname-west.westus22.cloudapp.azure.com:/home/azureUser/test/cert.pem C:\Documents\
  ```

Troubleshooting

If you experience problems running Azure IoT Edge in your environment, use these articles for troubleshooting and diagnostics:

- Troubleshoot your IoT Edge device (Azure IoT Edge 1.2) or Azure IoT Edge 1.1
- Common problems and resolutions for Azure IoT Edge (Azure IoT Edge 1.2) or Azure IoT Edge 1.1

Next steps

- After you complete the steps in this topic, you can run the Azure IoT sample, following the instructions for connecting using Azure IoT Edge.