

Quickstarts: Deploy code to a Linux device

[Quickstart: Deploy your first IoT Edge module to a virtual Linux device](#)

1. Install the Azure CLI

[Install the Azure CLI](#)

- macOS
 - Azure CLI

```
$ brew update && brew install azure-cli
```

- Azure IoT CLI Extension

```
$ az extension add --name azure-iot
```

Login

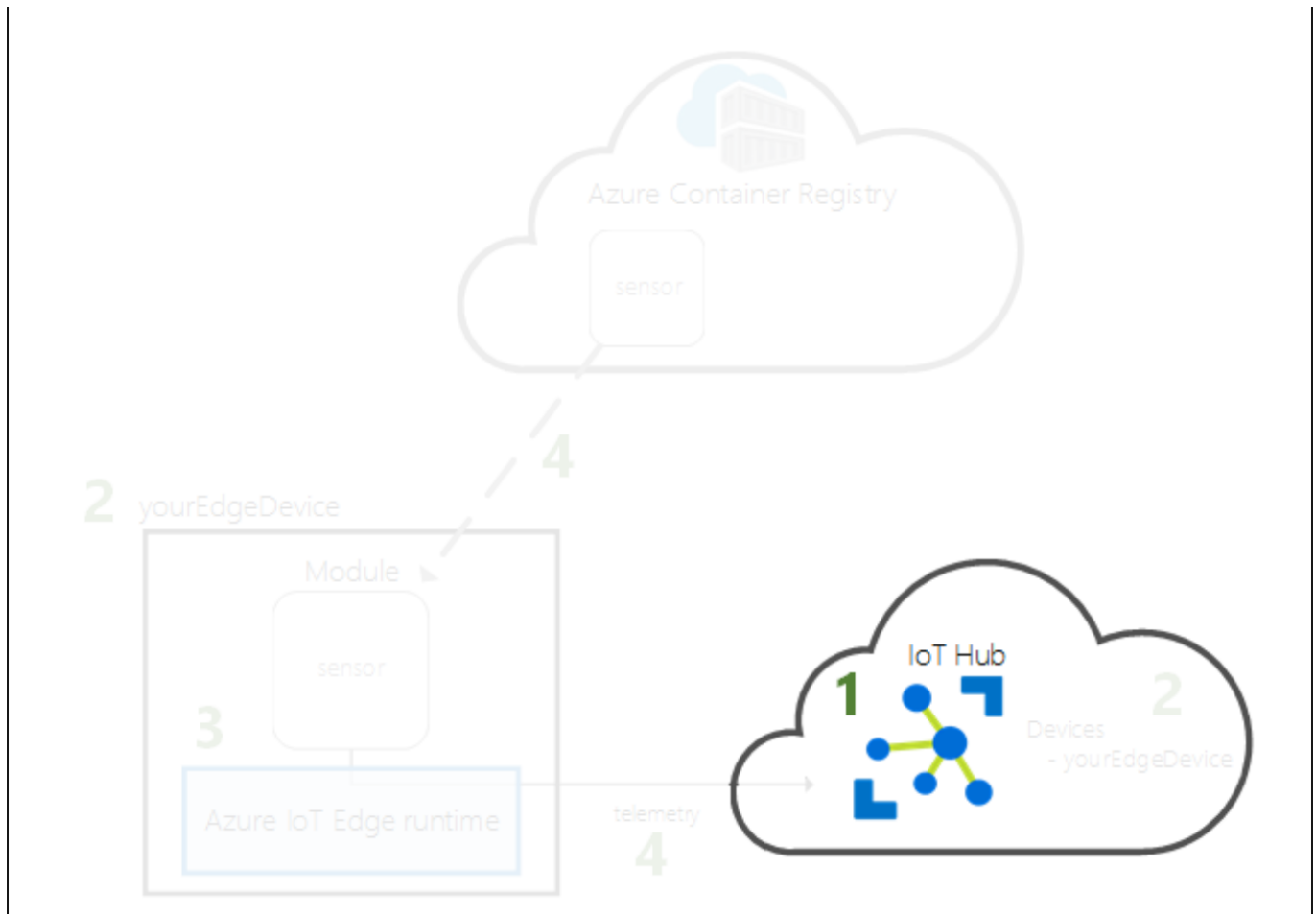
```
$ az login
```

2. Create Cloud resources

A resource group to manage all the resources you use in this section

```
$ az group create --name IoTEdgeResources --location westus2
```

3. Create an IoT hub



The following code creates a **free F1 hub** in the resource group **IoTEdgeResources**. Replace {hub_name} with a unique name for your IoT hub. It might take a few minutes to create an IoT Hub.

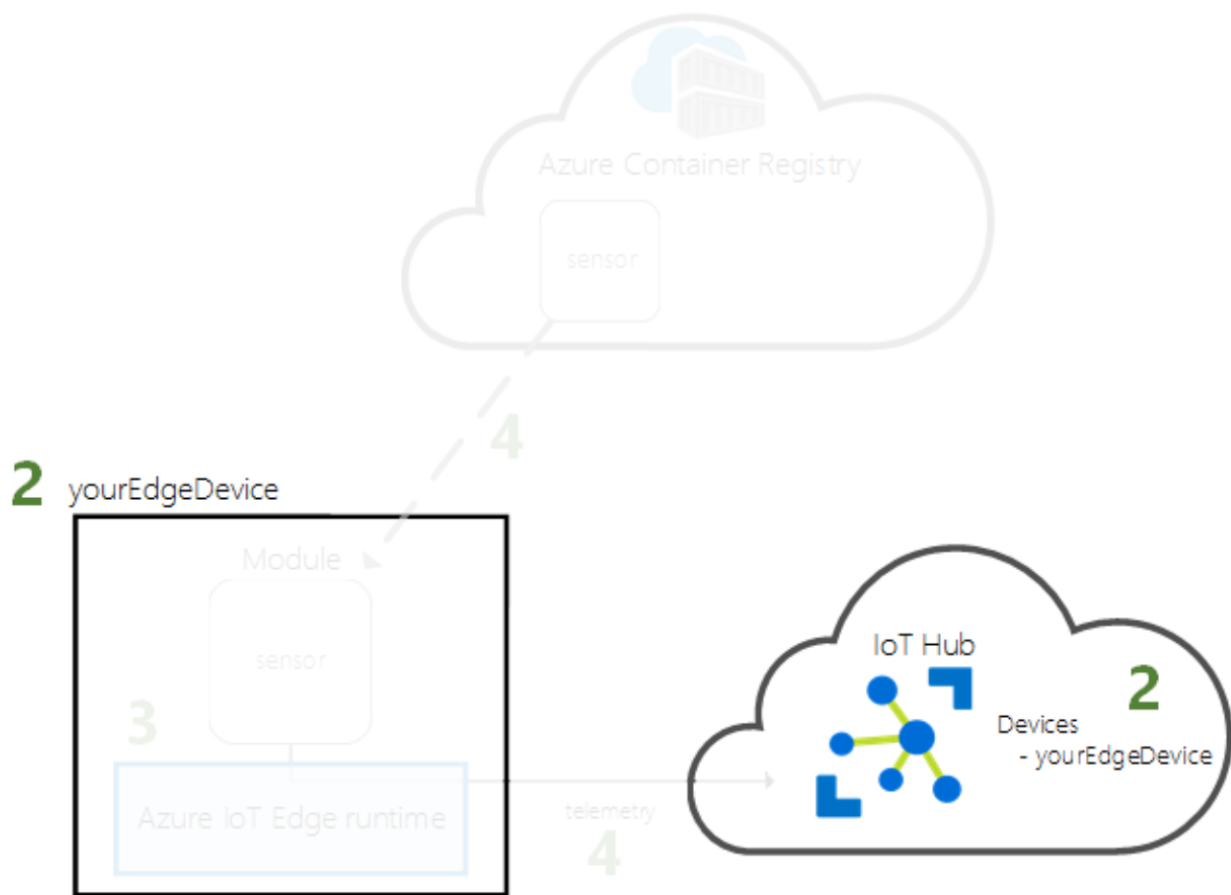
```
$ az iot hub create --resource-group IoTEdgeResources --name {hub_name} --
sku F1 --partition-count 2
```

Exmaple

```
az iot hub create --resource-group IoTEdgeResources --name kakaiothub --
sku S1 --partition-count 2
```

- Note: If you get an error because there's already one free hub in your subscription, change the SKU to S1.

4. Register an IoT Edge device



Create a **device identity** for your IoT Edge device so that it can communicate with your IoT hub. The device identity lives in the cloud, and you use a **unique device connection string** to associate a physical device to a device identity.

4-1. Create and declare device identity to be for an IoT Edge device

Since IoT Edge devices behave and can be managed differently than typical IoT devices, *declare this identity to be for an IoT Edge device* with the **--edge-enabled** flag.

```
$ az iot hub device-identity create --device-id {myEdgeDevice} --edge-enabled --hub-name {hub_name}
```

Example

```
$ az iot hub device-identity create --device-id kakaEdgeDevice --edge-enabled --hub-name kakaiothub
```

4-2. View the connection string

View the **connection string** for your device, which links your physical device with its identity in IoT Hub.

connection string contains:

- the name of your IoT hub
- the name of your device

- a shared key that authenticates connections between the two

```
$ az iot hub device-identity connection-string show --device-id  
{myEdgeDevice} --hub-name {hub_name}
```

Example

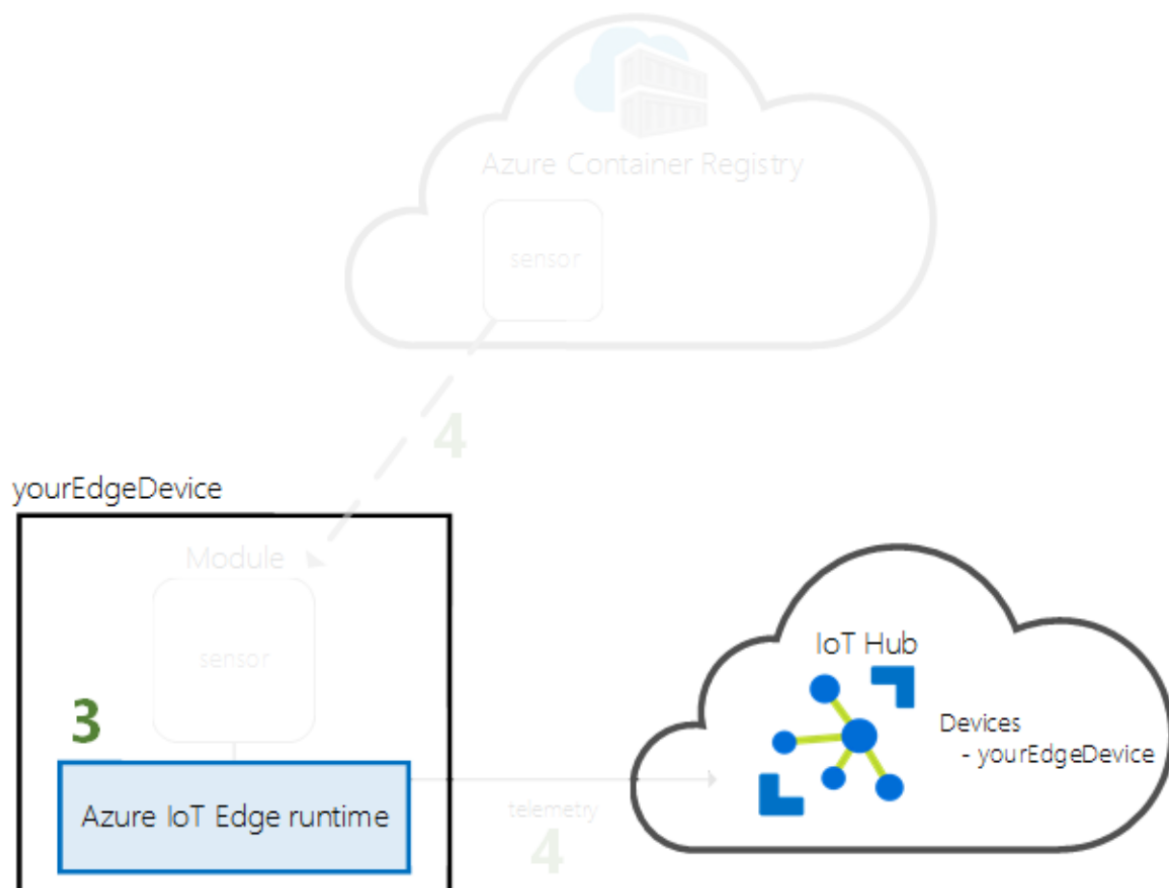
```
$ az iot hub device-identity connection-string show --device-id  
kakaEdgeDevice --hub-name kakaiothub
```

Output:

```
{  
  "connectionString": "HostName={hub_name}.azure-devices.net;DeviceId=  
{myEdgeDevice};SharedAccessKey={key}"  
}
```

5. Configure your IoT Edge device

Create a virtual machine with the Azure IoT Edge runtime on it.



The **IoT Edge runtime** is deployed on all IoT Edge devices.

It has three components:

- **The IoT Edge security daemon:** starts each time an IoT Edge device boots and bootstraps the device by starting the IoT Edge agent.
- **The IoT Edge agent:** facilitates deployment and monitoring of modules on the IoT Edge device, including the IoT Edge hub.
- **The IoT Edge hub:** manages communications between modules on the IoT Edge device, and between the device and IoT Hub.

During the runtime configuration, you provide a **device connection string**. This string associates your physical device with the IoT Edge device identity in Azure.

5-1. Deploy the IoT Edge device (Create VM and install IoT Edge runtime)

This section uses an **Azure Resource Manager template** to *create a new virtual machine and install the IoT Edge runtime* on it.

Use the following CLI command to create your IoT Edge device based on the prebuilt [iotedge-vm-deploy](#) template.

template: [iotedge-vm-deploy/edgeDeploy.json](#)

```
$ az deployment group create \
--resource-group IoTEdgeResources \
--template-uri "https://aka.ms/iotedge-vm-deploy" \
--parameters dnsLabelPrefix='<REPLACE_WITH_VM_NAME>' \
--parameters adminUsername='azureUser' \
--parameters deviceConnectionString=$(az iot hub device-identity
connection-string show --device-id myEdgeDevice --hub-name
<REPLACE_WITH_HUB_NAME> -o tsv) \
--parameters authenticationType='password' \
--parameters adminPasswordOrKey="<REPLACE_WITH_PASSWORD>"

# Example
$ az deployment group create \
--resource-group IoTEdgeResources \
--template-uri "https://aka.ms/iotedge-vm-deploy" \
--parameters dnsLabelPrefix='kaka-edge-vm1' \
--parameters adminUsername='azureUser' \
--parameters deviceConnectionString=$(az iot hub device-identity
connection-string show --device-id kakaEdgeDevice --hub-name kakaiothub -o
tsv) \
--parameters authenticationType='password' \
--parameters adminPasswordOrKey="password"
```

Once the deployment is complete, you should receive JSON-formatted output in the CLI that contains the SSH information to connect to the virtual machine. Copy the value of the public SSH entry of the outputs section:

```
Bash
}
],
"outputs": {
  "public SSH": {
    "type": "String",
    "value": "ssh azureUser@my-edge-vm1.westus2.cloudapp.azure.com"
  }
},
"parameters": {
  "adminPasswordOrKey": {
    "type": "SecureString"
  },
  "adminUsername": {
    "type": "String",
    "value": "azureUser"
  },
}
```

5-2. View the IoT Edge runtime status

The rest of the commands in this quickstart take place on your IoT Edge device itself, so that you can see what's happening on the device.

```
$ ssh {admin username}@{DNS name}
```

Once connected to your virtual machine, verify that the runtime was successfully installed and configured on your IoT Edge device.

1. Check to see that the IoT Edge security daemon is running as a system service.

```
$ sudo systemctl status iotedge
```

```
~$ sudo systemctl status iotedge
● iotedge.service - Azure IoT Edge daemon
   Loaded: loaded (/lib/systemd/system/iotedge.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-03-23 15:35:19 UTC; 13min ago
     Docs: man:iotedged(8)
    Main PID: 2203 (iotedged)
      Tasks: 22 (limit: 4915)
    CGroup: /system.slice/iotedge.service
            └─2203 /usr/bin/iotedged -c /etc/iotedge/config.yaml
```

2. If you need to troubleshoot the service, retrieve the service logs.

```
$ journalctl -u iotedge
```

3. View all the modules running on your IoT Edge device. *The edgeAgent module runs by default.*

```
$ sudo iotedge list
```

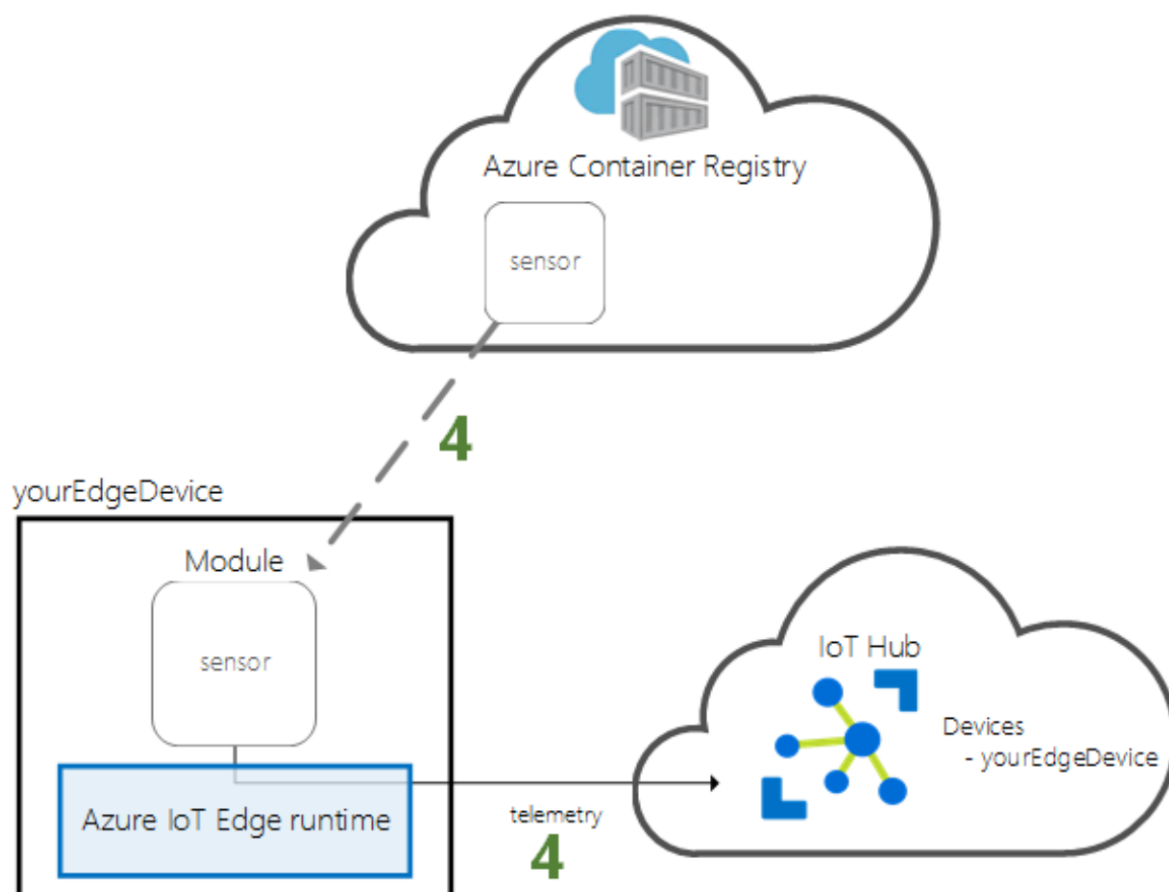
```
~$ sudo iotedge list
NAME                STATUS      DESCRIPTION        CONFIG
SimulatedTemperatureSensor running    Up 10 minutes     mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0
edgeAgent           running    Up 11 minutes     mcr.microsoft.com/azureiotedge-agent:1.1
edgeHub             running    Up 10 minutes     mcr.microsoft.com/azureiotedge-hub:1.1
```

Note:

You should only have **edgeAgent** and **edgeHub** before you deploy **SimulatedTemperatureSensor** module

6. Deploy a module

Manage your Azure IoT Edge device from the cloud to deploy a module that will send telemetry data to IoT Hub.



One of the **key capabilities of Azure IoT Edge** is deploying code to your IoT Edge devices from the cloud. **IoT Edge modules** are executable packages implemented as containers.

Tutorials: [Deploy a module from the cloud](#)

7. View generated data

Use the SSH connection from Azure CLI. Confirm that the module deployed from the cloud is running on your IoT Edge device

```
$ sudo iotedge list
```

```
~$ sudo iotedge list
NAME                                STATUS      DESCRIPTION      CONFIG
SimulatedTemperatureSensor          running     Up 10 minutes    mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0
edgeAgent                           running     Up 11 minutes    mcr.microsoft.com/azureiotedge-agent:1.1
edgeHub                             running     Up 10 minutes    mcr.microsoft.com/azureiotedge-hub:1.1
```

View the messages being sent from the temperature sensor module:

```
$ sudo iotedge logs SimulatedTemperatureSensor -f
```

```
[Information]: Trying to initialize module client using transport type [Amqp.Tcp.Only].
[Information]: Successfully initialized module client of transport type [Amqp.Tcp.Only].
03/23/2021 17:27:05> Sending message: 1, Body: [{"machine":{"temperature":21.136530438152388,"pressure":1.0155541005490063},"ambient":{"temperature":20.996108681660196,"humidity":24},"timeCreated":"2021-03-23T17:27:05.951341Z"}]
03/23/2021 17:27:11> Sending message: 2, Body: [{"machine":{"temperature":21.573214758687286,"pressure":1.0653029471922224},"ambient":{"temperature":20.94204497404492,"humidity":25},"timeCreated":"2021-03-23T17:27:11.095008Z"}]
03/23/2021 17:27:16> Sending message: 3, Body: [{"machine":{"temperature":22.341203587405012,"pressure":1.152795345400571},"ambient":{"temperature":21.14924632834701,"humidity":26},"timeCreated":"2021-03-23T17:27:16.1235206Z"}]
03/23/2021 17:27:21> Sending message: 4, Body: [{"machine":{"temperature":22.278802931904234,"pressure":1.1456864099637736},"ambient":{"temperature":21.291615768704386,"humidity":26},"timeCreated":"2021-03-23T17:27:21.1541164Z"}]
03/23/2021 17:27:26> Sending message: 5, Body: [{"machine":{"temperature":22.038818969525778,"pressure":1.1183464648826835},"ambient":{"temperature":20.54242264062279,"humidity":24},"timeCreated":"2021-03-23T17:27:26.1829278Z"}]
03/23/2021 17:27:31> Sending message: 6, Body: [{"machine":{"temperature":21.960637563122734,"pressure":1.109439722381071},"ambient":{"temperature":21.26116383995915,"humidity":25},"timeCreated":"2021-03-23T17:27:31.2114908Z"}]
03/23/2021 17:27:36> Sending message: 7, Body: [{"machine":{"temperature":22.547110388193797,"pressure":1.1762530821992934},"ambient":{"temperature":20.568248050319145,"humidity":26},"timeCreated":"2021-03-23T17:27:36.2392343Z"}]
03/23/2021 17:27:41> Sending message: 8, Body: [{"machine":{"temperature":22.96495063484877,"pressure":1.2238551356156826},"ambient":{"temperature":20.624004575481642,"humidity":24},"timeCreated":"2021-03-23T17:27:41.2698295Z"}]
03/23/2021 17:27:46> Sending message: 9, Body: [{"machine":{"temperature":23.17063204940438,"pressure":1.2472871955017646},"ambient":{"temperature":20.574608843342684,"humidity":25},"timeCreated":"2021-03-23T17:27:46.2966919Z"}]
03/23/2021 17:27:51> Sending message: 10, Body: [{"machine":{"temperature":24.106133951389292,"pressure":1.3538633615506788},"ambient":{"temperature":21.483243646558023,"humidity":25},"timeCreated":"2021-03-23T17:27:51.3233247Z"}]
03/23/2021 17:27:56> Sending message: 11, Body: [{"machine":{"temperature":24.97638527617156,"pressure":1.453005917538532},"ambient":{"temperature":20.965538168077142,"humidity":26},"timeCreated":"2021-03-23T17:27:56.3493917Z"}]
03/23/2021 17:28:01> Sending message: 12, Body: [{"machine":{"temperature":25.635644862957133,"pressure":1.528111440083724},"ambient":{"temperature":20.72373539918276,"humidity":25},"timeCreated":"2021-03-23T17:28:01.3757616Z"}]
03/23/2021 17:28:06> Sending message: 13, Body: [{"machine":{"temperature":26.218625317778727,"pressure":1.594526934936817},"ambient":{"temperature":21.480966033405142,"humidity":26},"timeCreated":"2021-03-23T17:28:06.3988021Z"}]
03/23/2021 17:28:11> Sending message: 14, Body: [{"machine":{"temperature":26.195856935203008,"pressure":1.5919330685674313},"ambient":{"temperature":21.18257124194995,"humidity":24},"timeCreated":"2021-03-23T17:28:11.4240806Z"}]
```

8. Clean up resources

```
$ az group delete --name IoTEdgeResources --yes
```

You can confirm the resource group is removed by viewing the list of resource groups.

```
$ az group list
```