Deploy Azure IoT Edge modules from Visual Studio Code

This section shows how to create a JSON deployment manifest, then use that file to push the deployment to an IoT Edge device.

Prerequisites

- An `IoT hub` in your Azure subscription.
- An `IoT Edge device`
- Visual Studio Code
- Azure IoT Tools for Visual Studio Code

1. Configure a deployment manifest

A *`deployment manifest`* is a `JSON document` that describes *which modules to deploy*, *how data flows between the modules*, and *desired properties of the module twins*.

To deploy modules using Visual Studio Code, save the deployment manifest locally as a .JSON file.

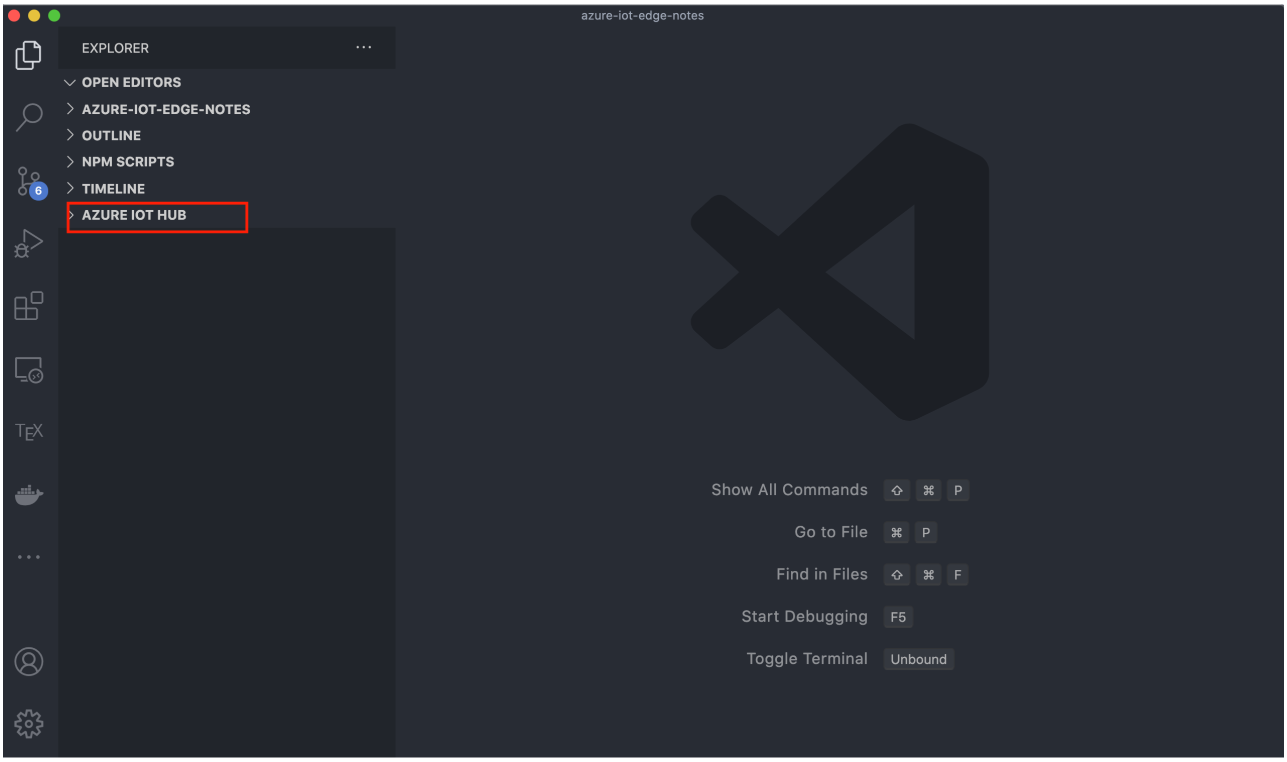
Here's a basic deployment manifest with one module as an example:

```
{
  "modulesContent": {
    "$edgeAgent": {
      "properties.desired": {
        "schemaVersion": "1.1",
        "runtime": {
          "type": "docker",
          "settings": {
            "minDockerVersion": "v1.25",
            "loggingOptions": "",
            "registryCredentials": {}
          }
        },
        "systemModules": {
          "edgeAgent": {
            "type": "docker",
            "settings": {
              "image": "mcr.microsoft.com/azureiotedge-agent:1.1",
              "createOptions": "{}"
            }
          },
          "edgeHub": {
            "type": "docker",
            "status": "running",
            "restartPolicy": "always",
            "settings": {
              "image": "mcr.microsoft.com/azureiotedge-hub:1.1",
              "createOptions": "{\"HostConfig\":{\"PortBindings\":{\"443/tcp\":[{\"HostPort\":\"443\"}],\"5671/tcp\":[{\"HostPort\":\"5671\"}],\"8883/tcp\":[{\"HostPort\":\"8883\"}]}}}"
            }
          }
        },
        "modules": {
          "SimulatedTemperatureSensor": {
            "version": "1.0",
            "type": "docker",
            "status": "running",
            "restartPolicy": "always",
            "settings": {
              "image": "mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0",
              "createOptions": "{}"
            }
          }
        }
      }
    },
    "$edgeHub": {
      "properties.desired": {
        "schemaVersion": "1.1",
        "routes": {
            "route": "FROM /messages/* INTO $upstream"
        },
        "storeAndForwardConfiguration": {
          "timeToLiveSecs": 7200
        }
      }
    },
    "SimulatedTemperatureSensor": {
      "properties.desired": {}
    }
  }
}
```

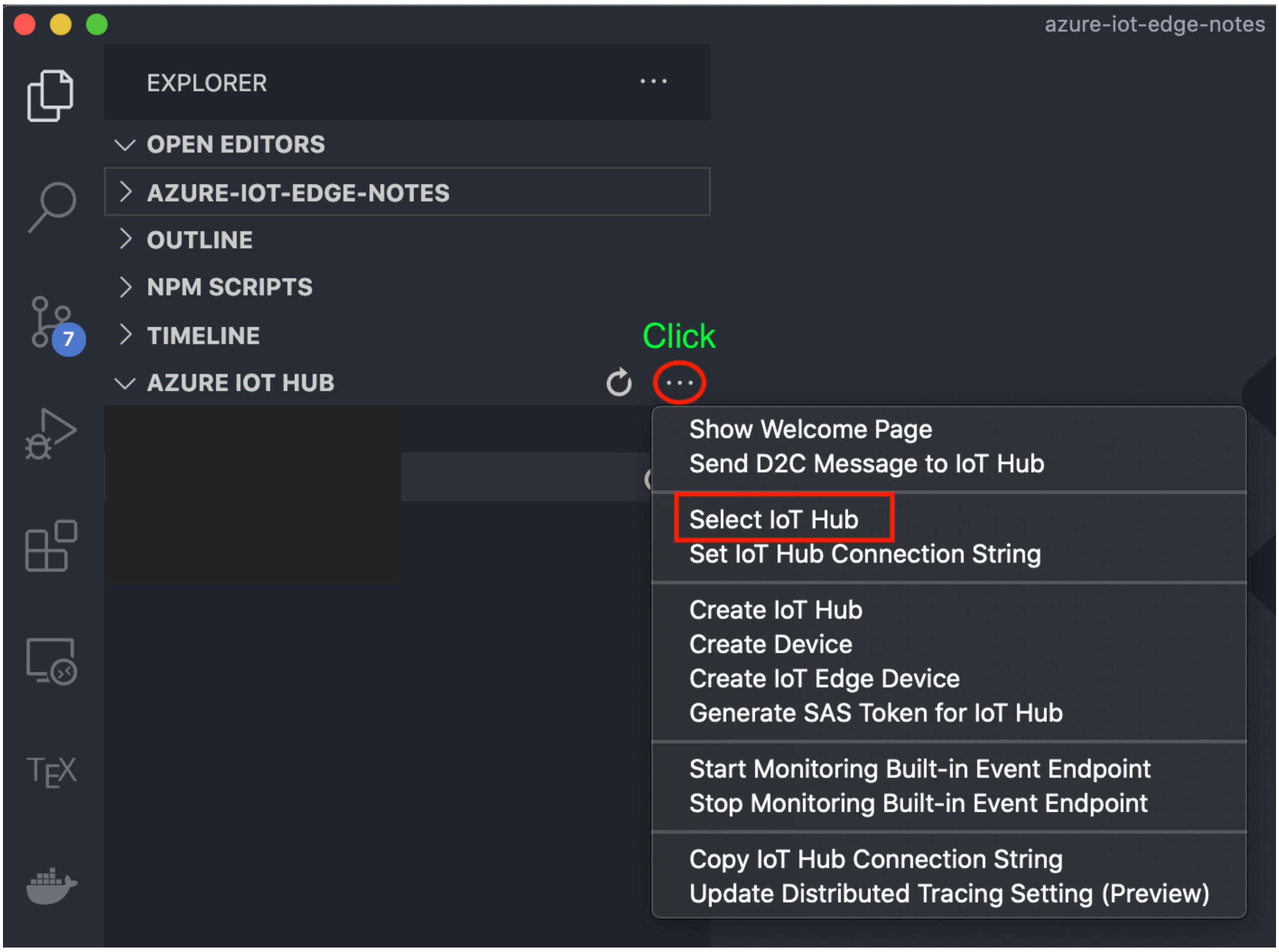
2. Sign in to access your IoT hub

You can use the Azure IoT extensions for Visual Studio Code to perform operations with your IoT hub

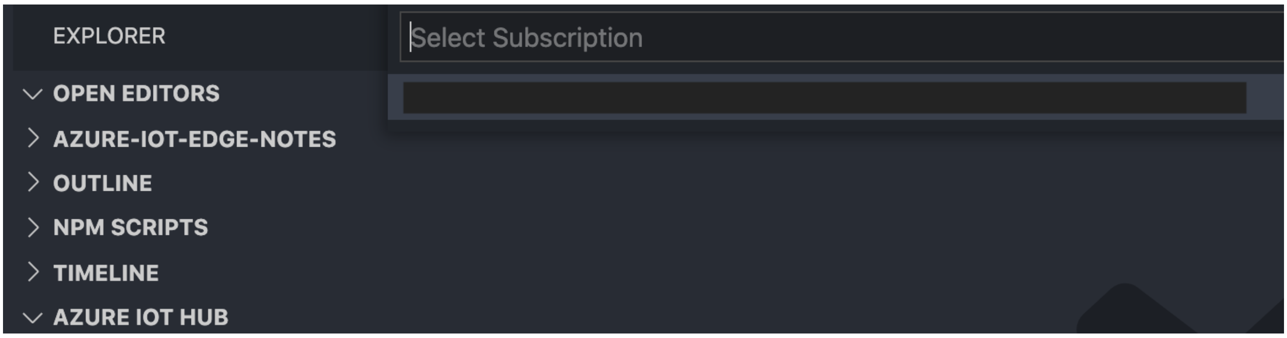
1. Click the `Azure IoT Hub` section



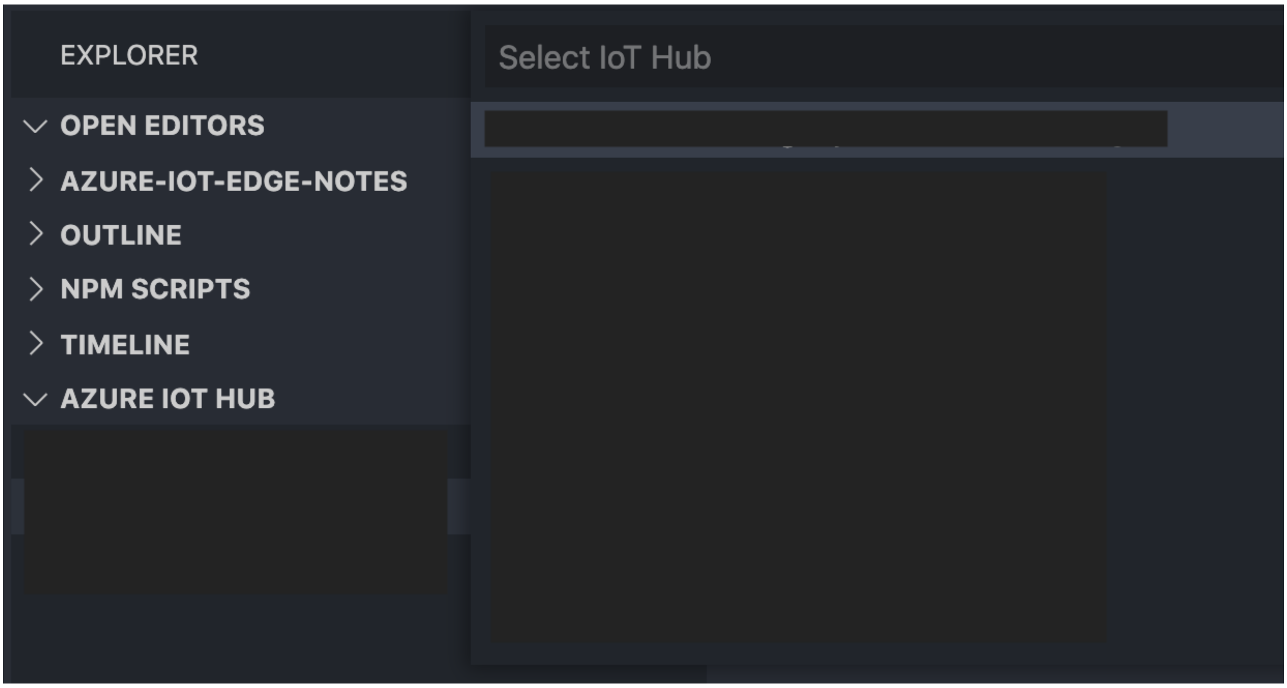
2. Choose `Select IoT Hub`


azure-iot-edge-notes
EXPLORER
OPEN EDITORS
AZURE-IOT-EDGE-NOTES
OUTLINE
NPM SCRIPTS
TIMELINE
AZURE IOT HUB
Click
Show Welcome Page
Send D2C Message to IoT Hub
Select IoT Hub
Set IoT Hub Connection String
Create IoT Hub
Create Device
Create IoT Edge Device
Generate SAS Token for IoT Hub
Start Monitoring Built-in Event Endpoint
Stop Monitoring Built-in Event Endpoint
Copy IoT Hub Connection String
Update Distributed Tracing Setting (Preview)

3. Select your `Azure subscription`



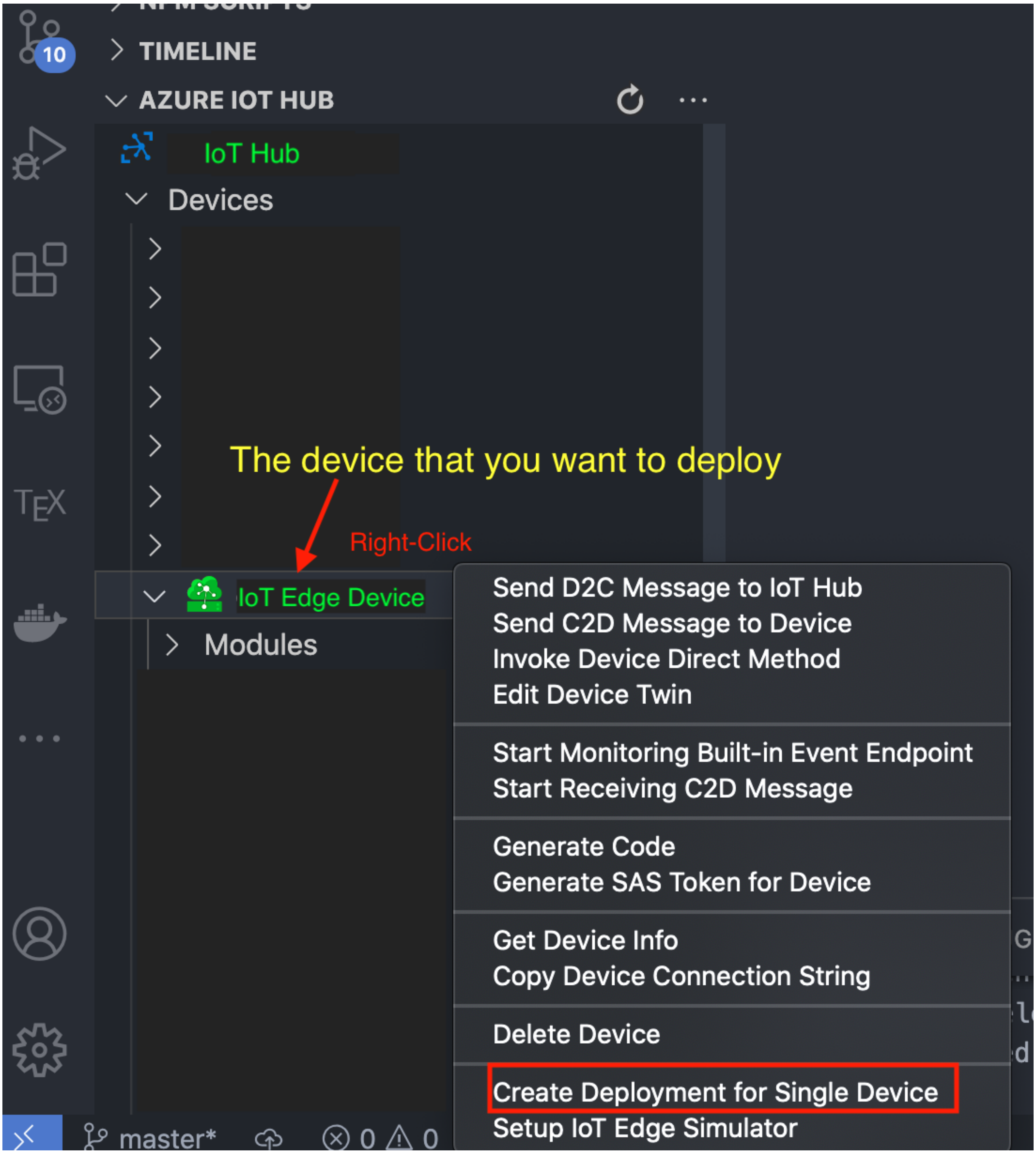
4. Select your `IoT hub`



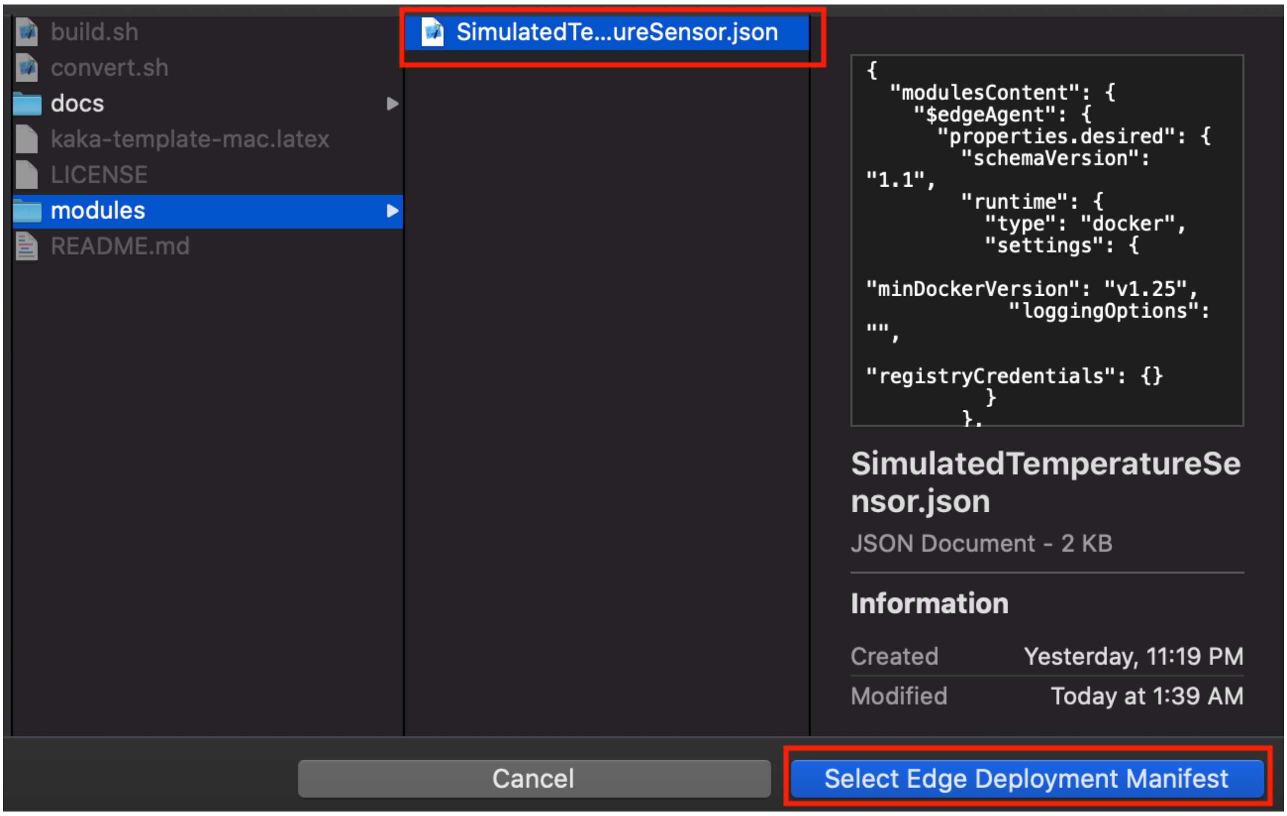
3. Deploy to your device

You deploy modules to your device by applying the deployment manifest that you configured with the module information

1. In the Visual Studio Code explorer view, expand the `Azure IoT Hub` section, and then expand the `Devices` node
2. *Right-click* on the IoT Edge device that you want to configure with the deployment manifest
3. Select `Create Deployment for Single Device`


NPM SCRIPTS
TIMELINE
AZURE IOT HUB
IoT Hub
Devices
The device that you want to deploy
Right-Click
IoT Edge Device
Modules
Send D2C Message to IoT Hub
Send C2D Message to Device
Invoke Device Direct Method
Edit Device Twin
Start Monitoring Built-in Event Endpoint
Start Receiving C2D Message
Generate Code
Generate SAS Token for Device
Get Device Info
Copy Device Connection String
Delete Device
Create Deployment for Single Device
Setup IoT Edge Simulator
master*
0 0

4. Navigate to the deployment `manifest JSON file that you want to use`, and click `Select Edge Deployment Manifest`



```
{
  "modulesContent": {
    "$edgeAgent": {
      "properties.desired": {
        "schemaVersion":
"1.1",
        "runtime": {
          "type": "docker",
          "settings": {
"minDockerVersion": "v1.25",
            "loggingOptions":
"",
"registryCredentials": {}
          }
        },
```

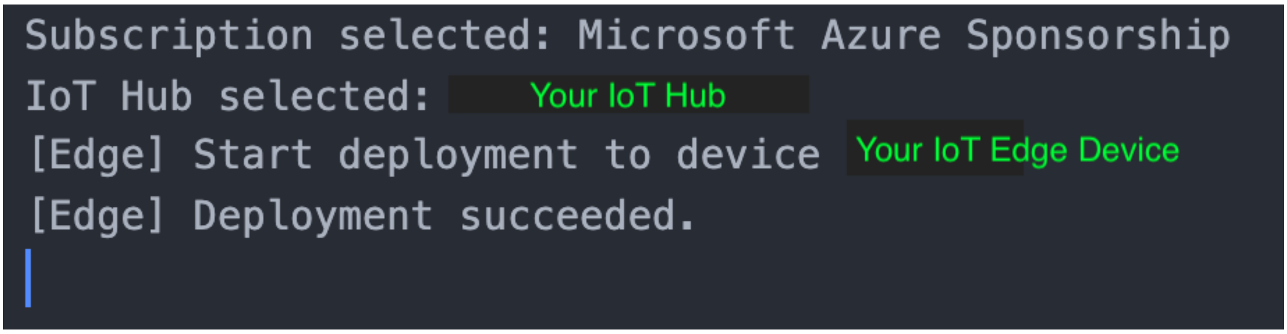
SimulatedTemperatureSensor.json

JSON Document - 2 KB

Information

Created	Yesterday, 11:19 PM
Modified	Today at 1:39 AM

5. The results of your deployment are printed in the VS Code output



```
Subscription selected: Microsoft Azure Sponsorship
IoT Hub selected: Your IoT Hub
[Edge] Start deployment to device Your IoT Edge Device
[Edge] Deployment succeeded.
```

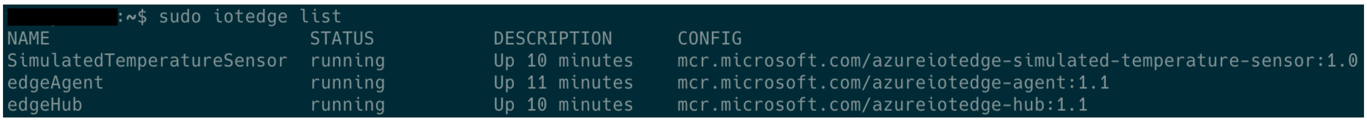
4. View modules on your device

Once you've deployed modules to your device, you can view all of them in the Azure IoT Hub section. Select the arrow next to your IoT Edge device to expand it. All the currently running modules are displayed.

If you recently deployed new modules to a device, hover over the Azure IoT Hub Devices section header and select the refresh icon to update the view.

Right-click the name of a module to view and edit the module twin.

```
$ sudo iotedge list
```



```
[redacted]:~$ 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
```