

Data to Azure IoT Hub Example

- [概述](#)
- [先决条件](#)
- [环境准备](#)
- [开始测试](#)

概述

映翰通提供[azure_iot_device](#)示例以便于客户基于InGateway二次开发实现上传数据至Azure IoT Hub并接收Azure IoT Hub下发的数据。示例包含以下内容：

- [azure_iot_device/iothub_client_example](#)：主要基于Azure IoT Device SDK实现上传数据至Azure IoT Hub以及接收Azure IoT Hub下发的数据，Azure IoT Device SDK的详细使用方法请访问[azure-iot-sdk-python](#)。
- [azure_iot_device/iothub_client_cert](#)：连接Azure IoT所需的证书脚本，直接使用即可，无需修改。
- [azure_iot_device/sftp.json](#)：用于连接InGateway的SFTP配置文件。

先决条件

在进行开发和测试前，你需要具备以下条件：

- InGateway
 - 固件版本：[2.0.0.r12191](#)及以上
 - SDK版本：[1.3.4](#)及以上
- VS Code软件
- Azure IoT账号

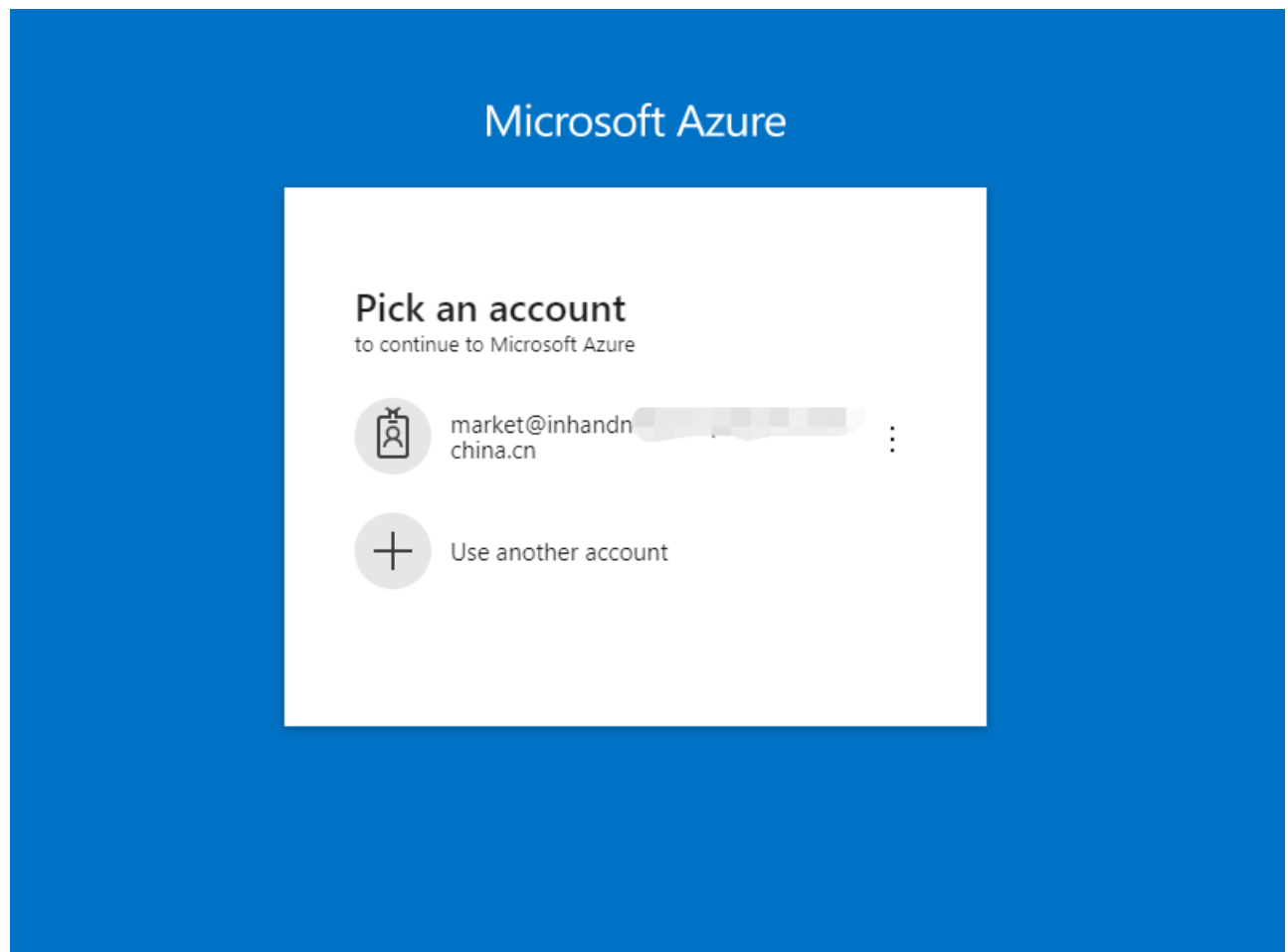
环境准备

- [配置Azure IoT](#)
- [配置开发环境](#)

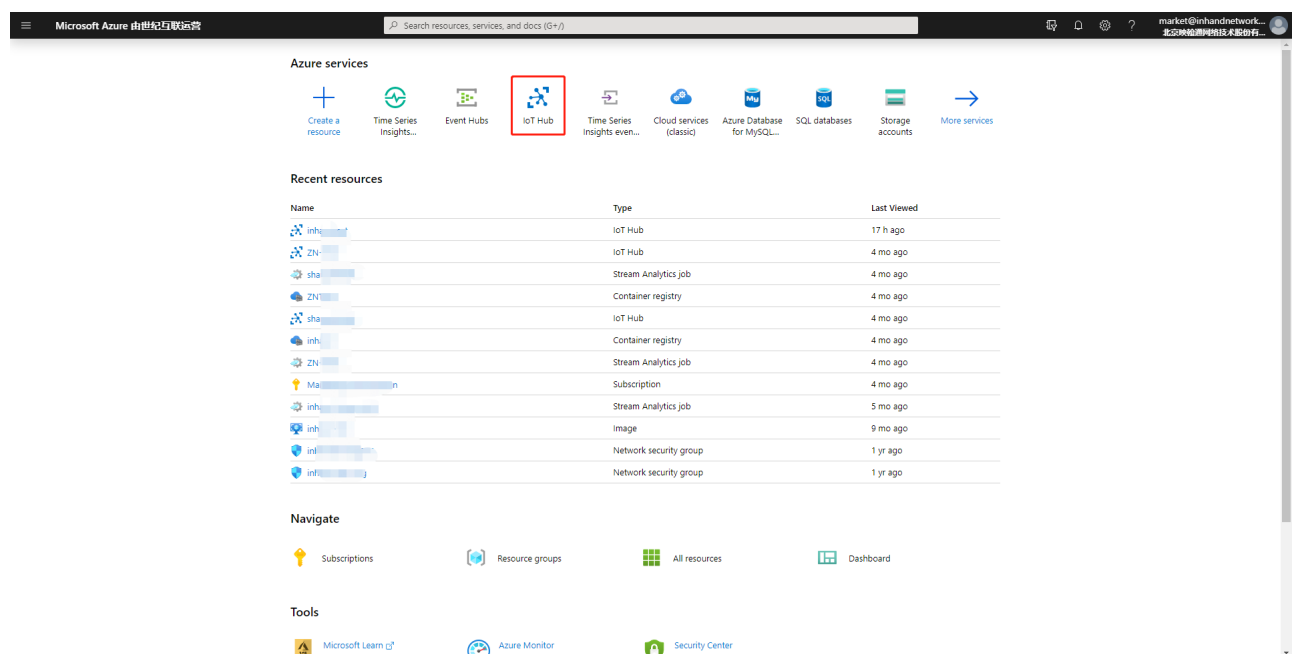
配置Azure IoT

如果你已经在Azure IoT上配置了相应的IoT Hub和IoT device，可以跳过这一小节。

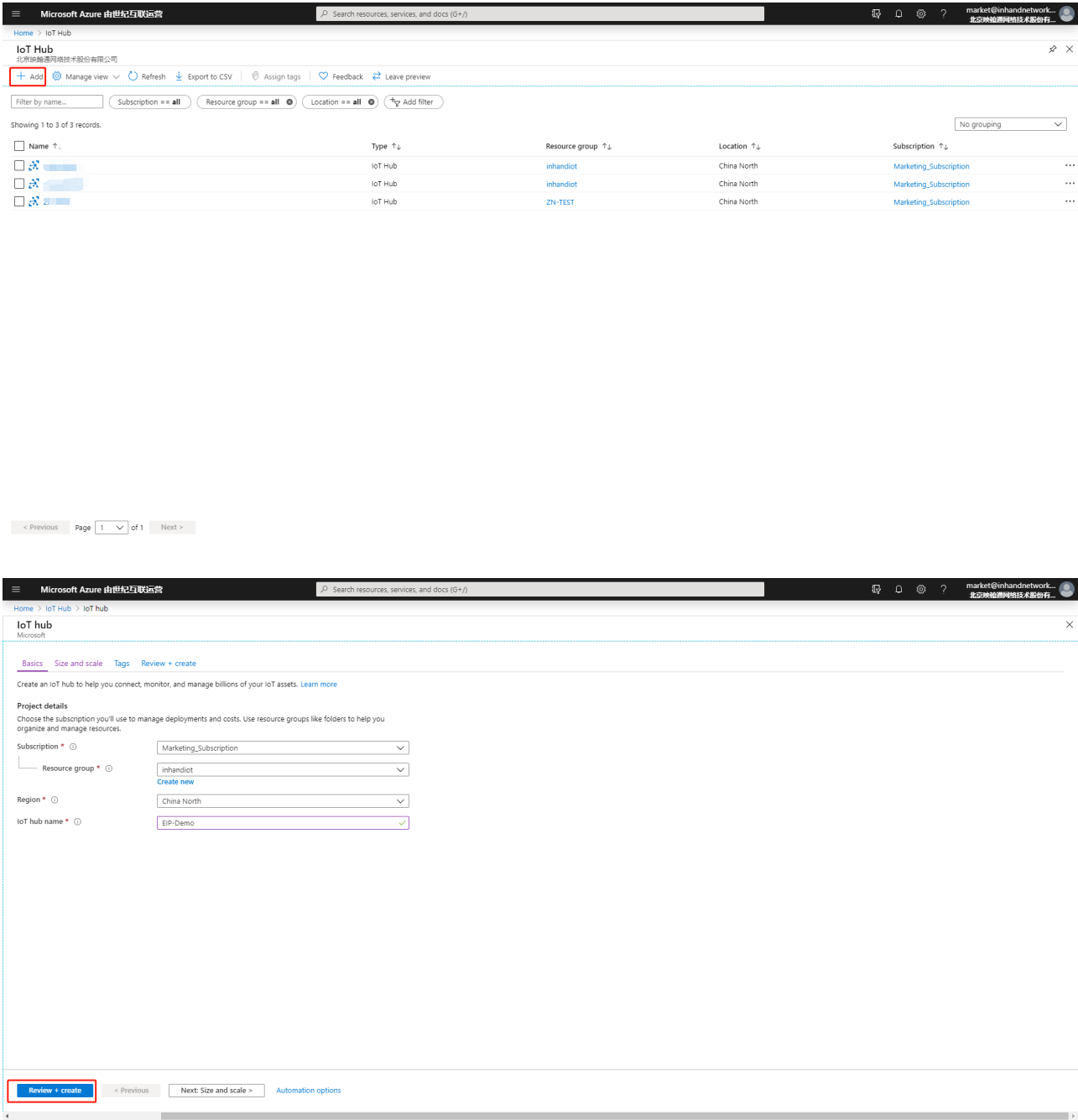
- 步骤1：登录Azure IoT
访问<https://portal.azure.cn>登录Azure。



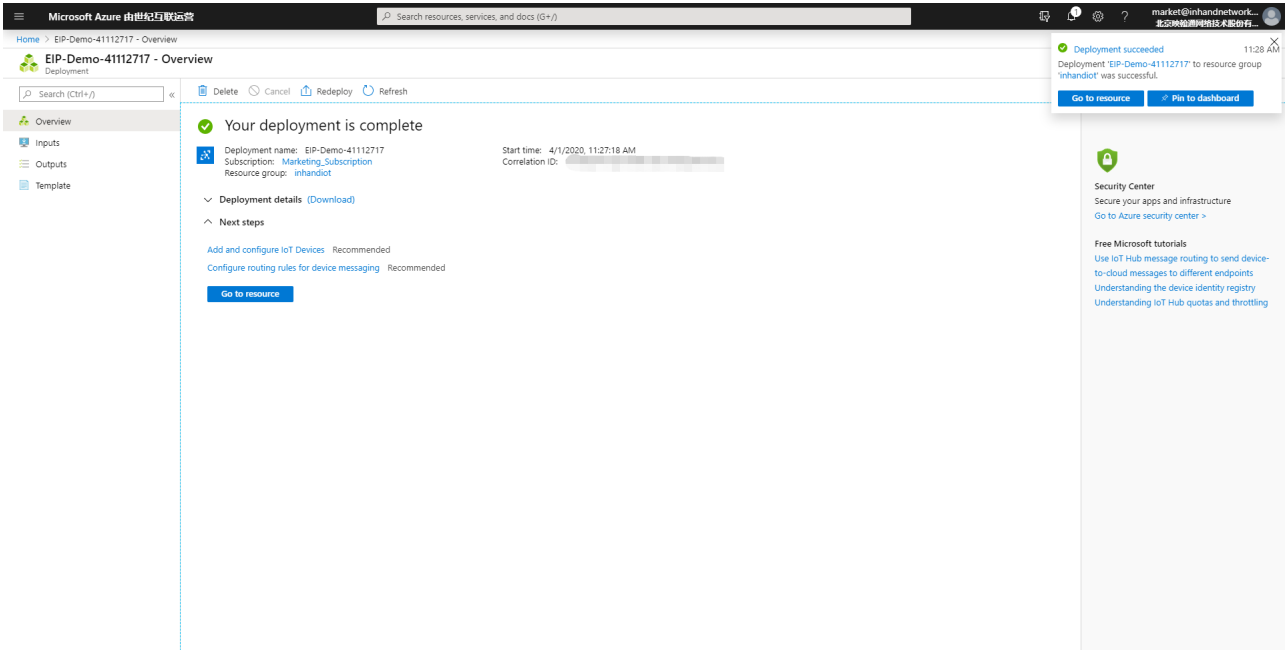
- 步骤2：添加IoT Hub
登录成功后如下图所示，选择“IoT Hub”。



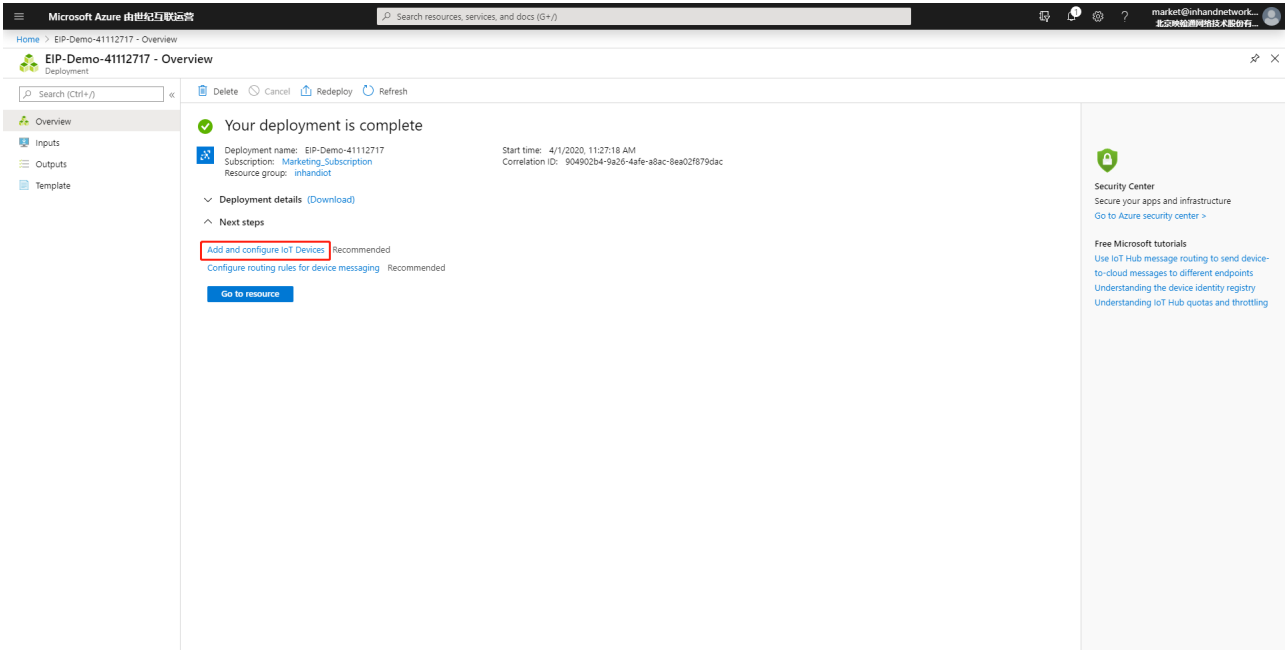
点击“Add”创建一个IoT Hub。

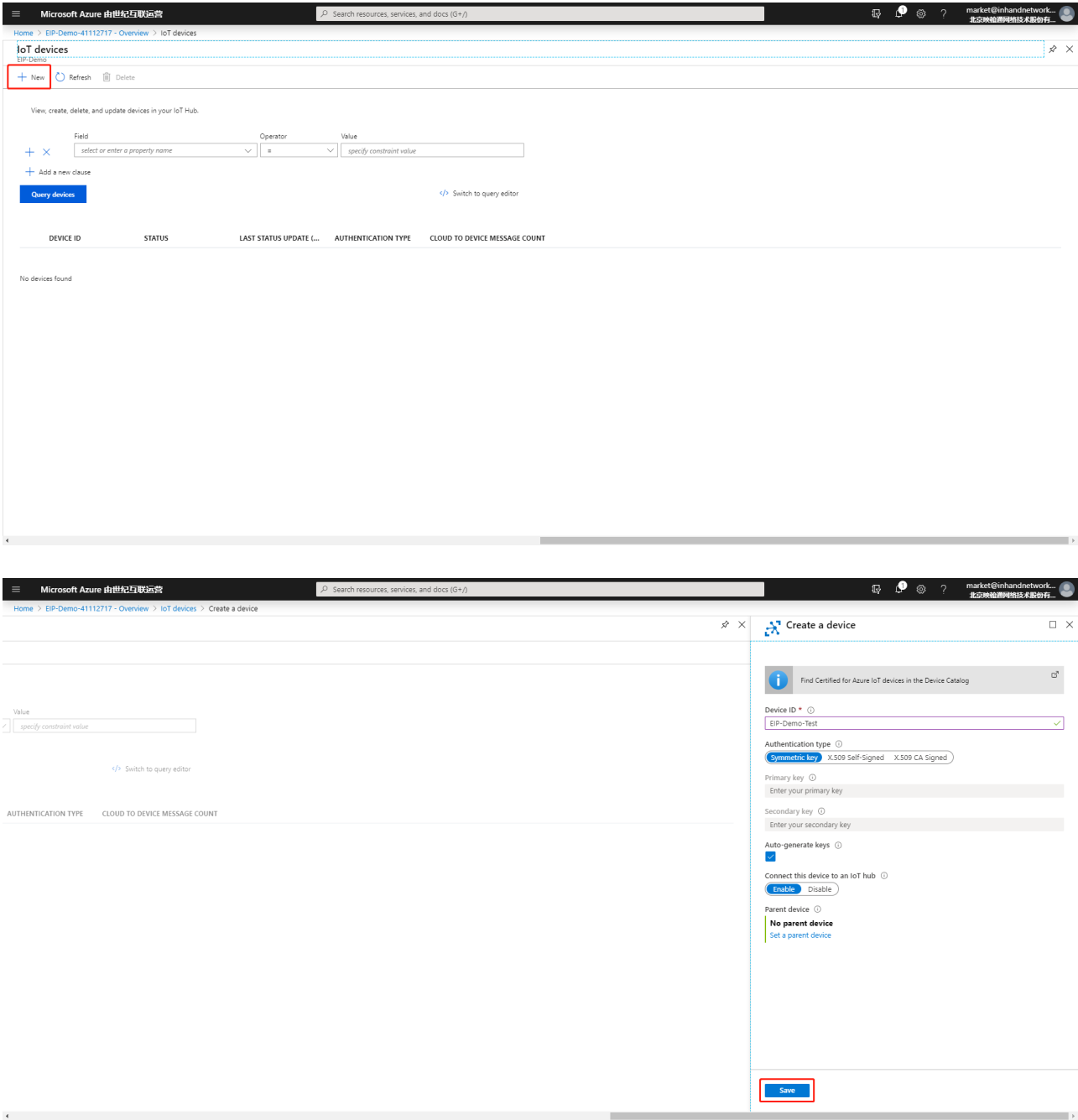


创建成功后如下图所示：

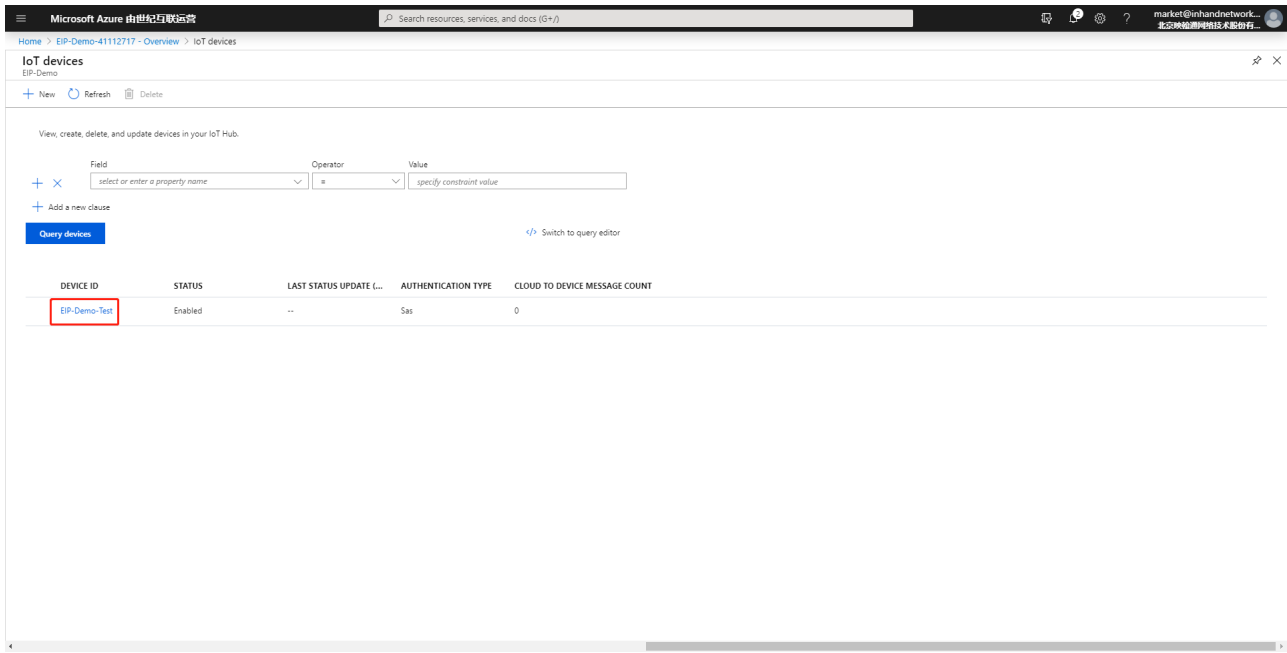


- 步骤3: 添加IoT Device
在IoT Hub中创建一个IoT Device。



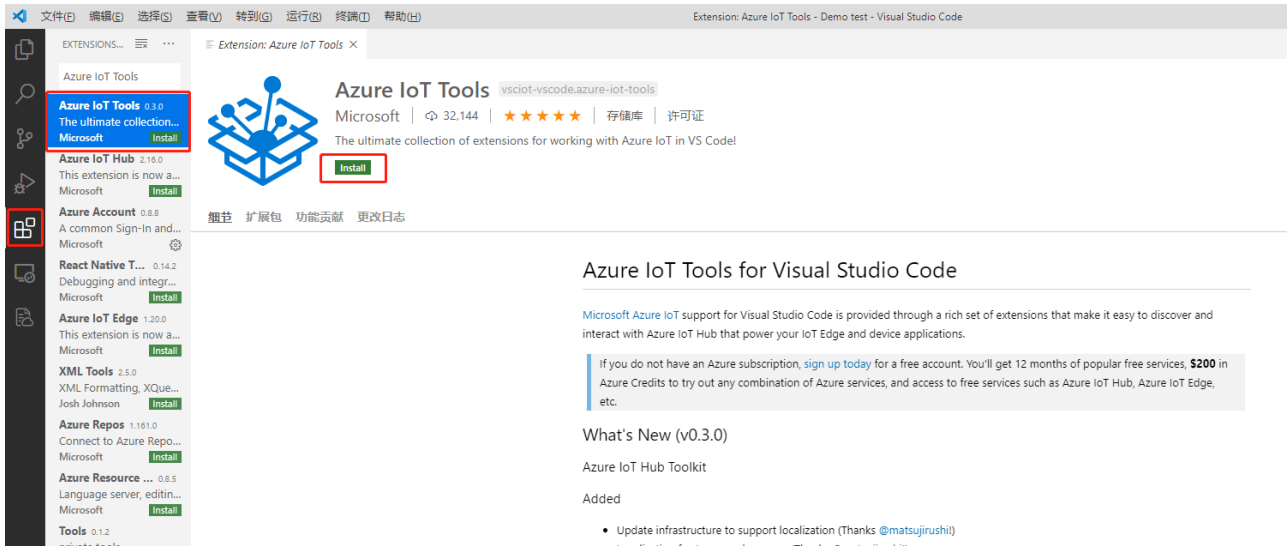


创建成功后如下图所示：

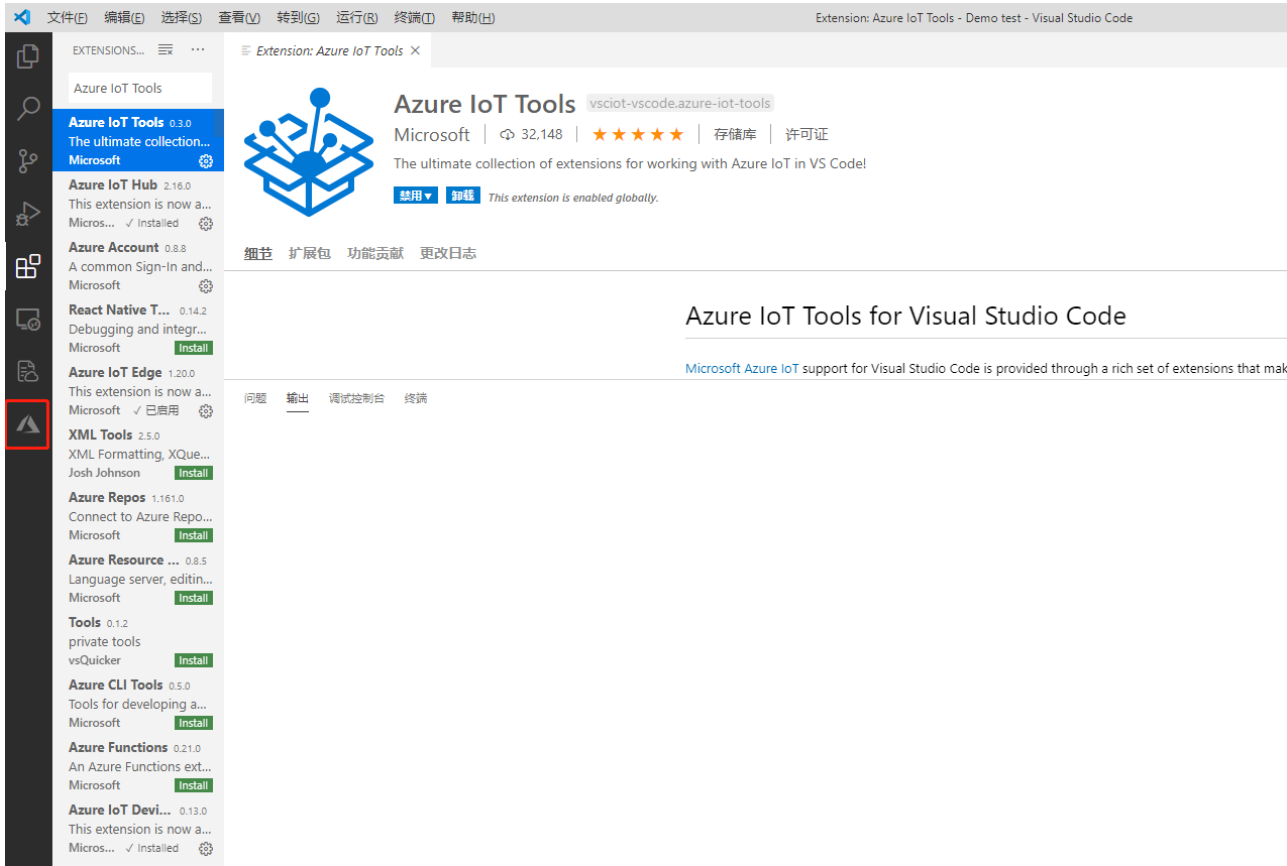


配置开发环境

- InGateway配置
- 建立项目文件夹
- 在VS Code中安装Azure IoT Tools插件
- 安装Azure IoT SDK
- InGateway配置
设备联网、软件更新、IDE软件获取等基础的配置操作请查看[MobiusPi Python Development Quick Start](#)。以下操作我们将假设你已经完成了InGateway的软件更新、设备联网、开启调试模式等配置。
- 建立项目文件夹
建立一个“Demo test”文件夹作为项目文件夹，将从[InGateway-Python-Examples](#)下载的 [iothub_client_example.py](#)、[iothub_client_cert.py](#)和[sftp.json](#)放入项目文件夹中。
- 在VS Code中安装Azure IoT Tools插件
在VS Code中点击“Extensions”，在搜索框中输入[Azure IoT Tools](#)并安装[Azure IoT Tools](#)插件。



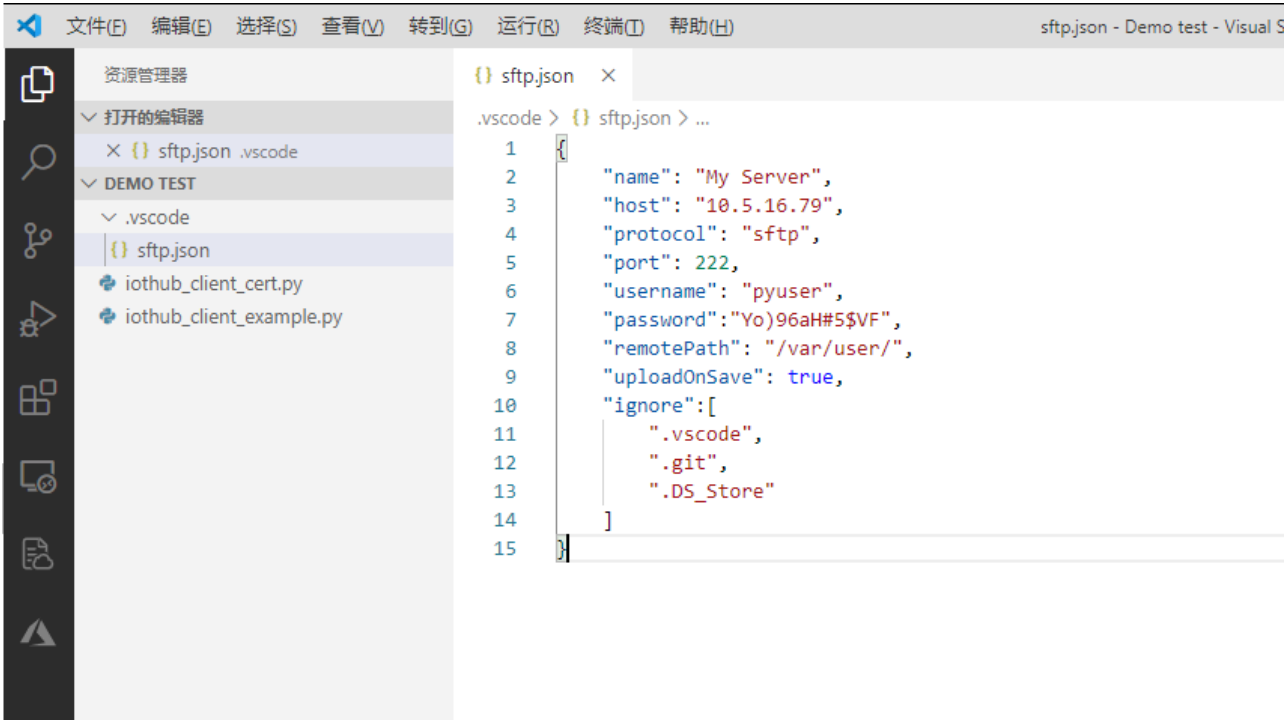
安装成功后在左侧可以看到Azure模块。



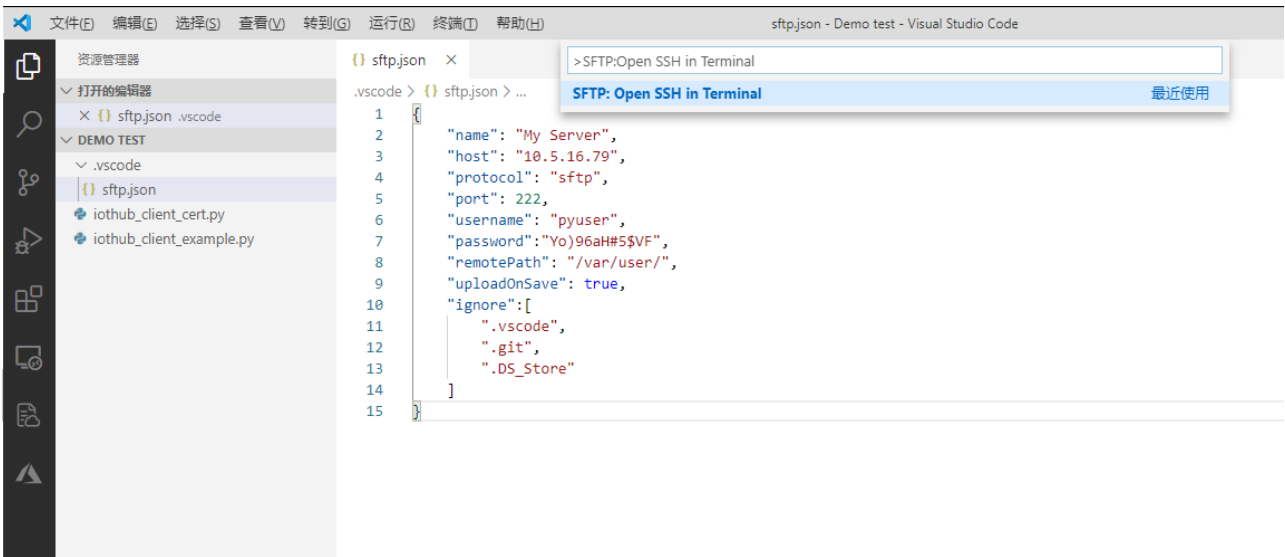
- 安装Azure IoT SDK
使用VS Code打开项目文件夹，在“命令面板”中输入>SFTP:Config 命令打开sftp.json文件。



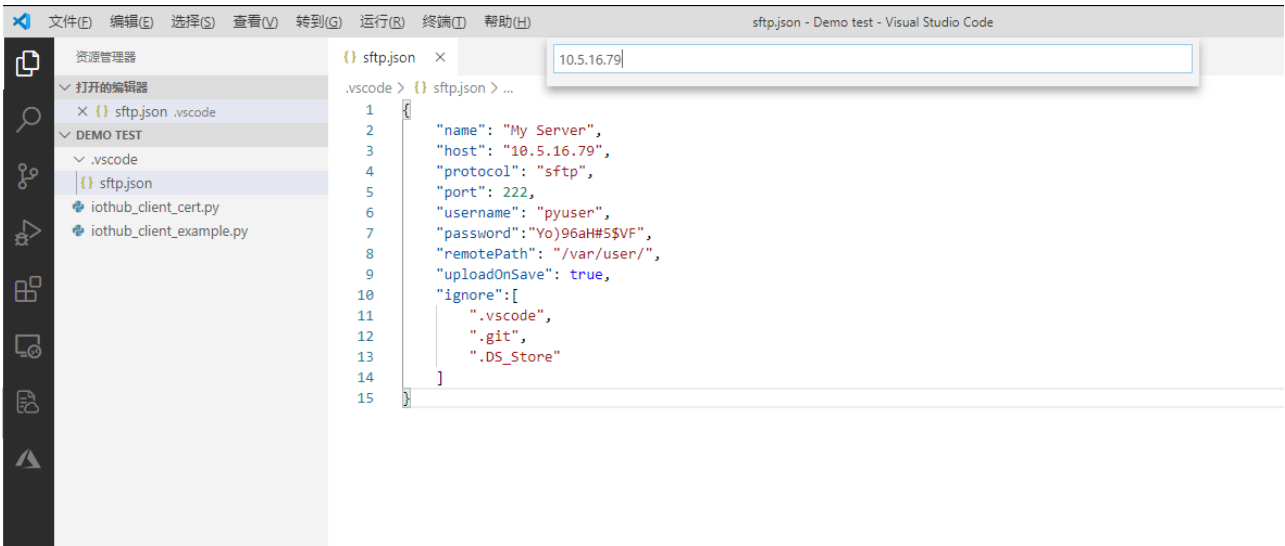
配置sftp.json文件，配置方法见[建立SFTP连接](#)。



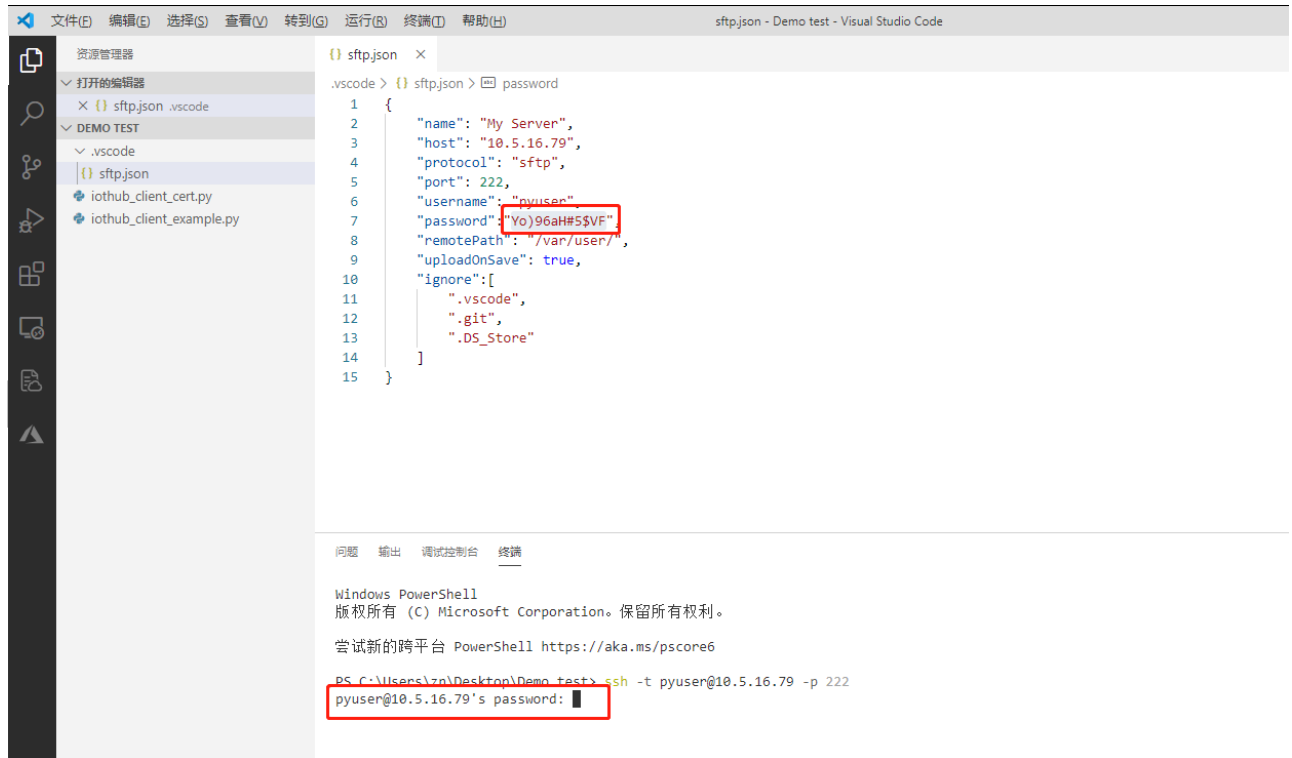
配置完成并保存后在“命令面板”中输入>SFTP:Open SSH in Terminal以连接InGateway。



输入后命令面板会提示你需要输入SFTP服务器的IP地址（即“host”项内容）。



“终端”窗口会提示你需要输入密码，你只需要将sftp.json文件中“password”项复制粘贴到此处即可。



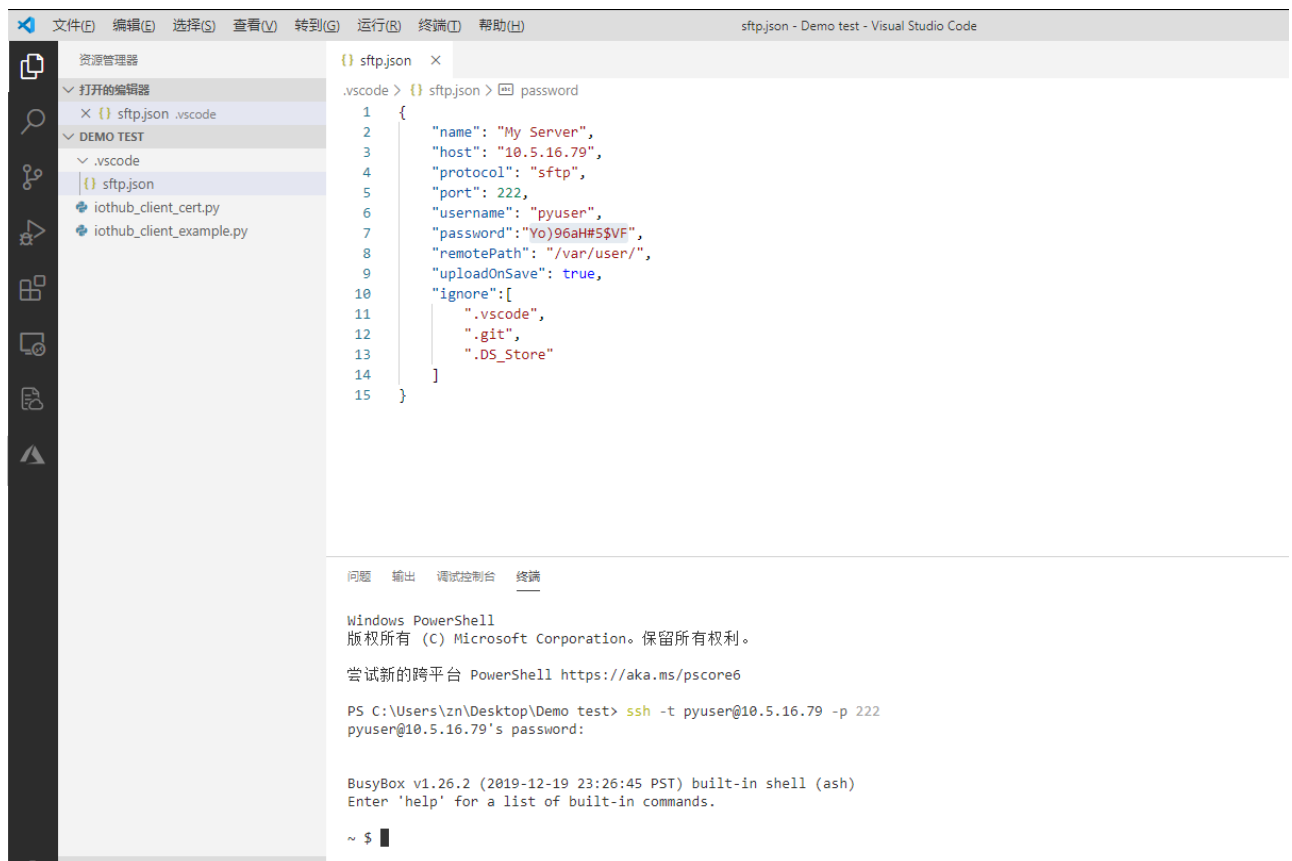
```
.vscode > {} sftp.json > password
1 {
2   "name": "My Server",
3   "host": "10.5.16.79",
4   "protocol": "sftp",
5   "port": 222,
6   "username": "pyuser",
7   "password": "Yo)96aH#5$VF",
8   "remotePath": "/var/user/",
9   "uploadOnSave": true,
10  "ignore": [
11    ".vscode",
12    ".git",
13    ".DS_Store"
14  ]
15 }
```

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\zn\Desktop\Demo test> ssh -t pyuser@10.5.16.79 -p 222
pyuser@10.5.16.79's password: 
```

成功与InGateway建立SFTP连接后如下图所示：



```
.vscode > {} sftp.json > password
1 {
2   "name": "My Server",
3   "host": "10.5.16.79",
4   "protocol": "sftp",
5   "port": 222,
6   "username": "pyuser",
7   "password": "Yo)96aH#5$VF",
8   "remotePath": "/var/user/",
9   "uploadOnSave": true,
10  "ignore": [
11    ".vscode",
12    ".git",
13    ".DS_Store"
14  ]
15 }
```

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

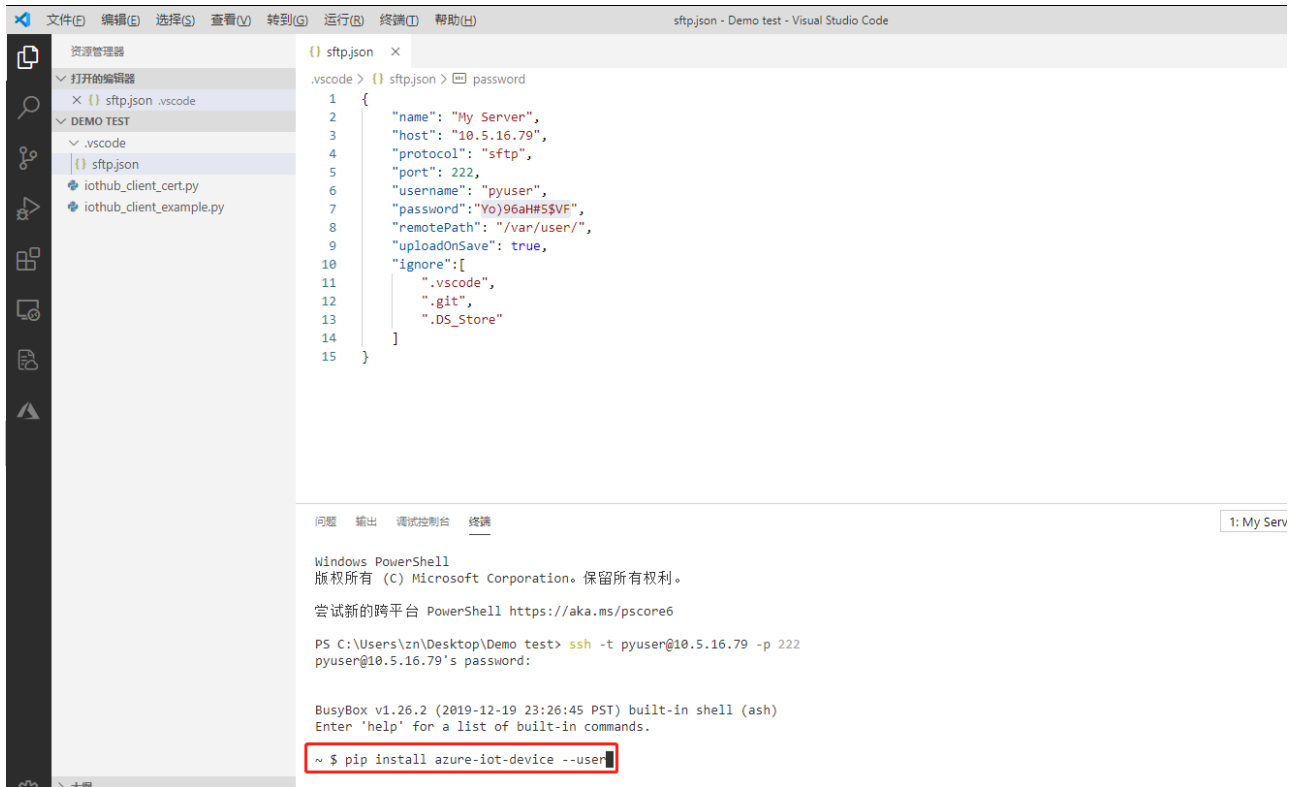
尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\zn\Desktop\Demo test> ssh -t pyuser@10.5.16.79 -p 222
pyuser@10.5.16.79's password:

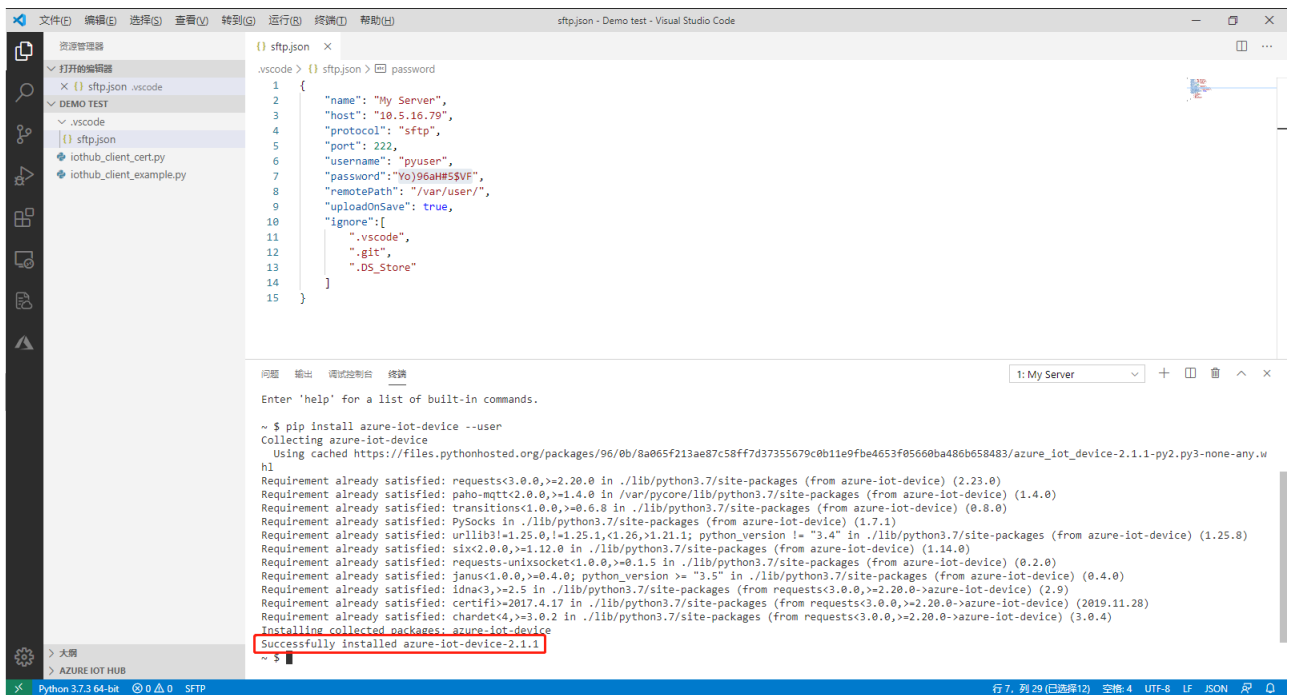
BusyBox v1.26.2 (2019-12-19 23:26:45 PST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ $ 
```

在“终端”中输入`pip install azure-iot-device --user`命令以安装Azure IoT SDK。(安装前请确认InGateway已经联网成功)



安装完成后如下图所示：

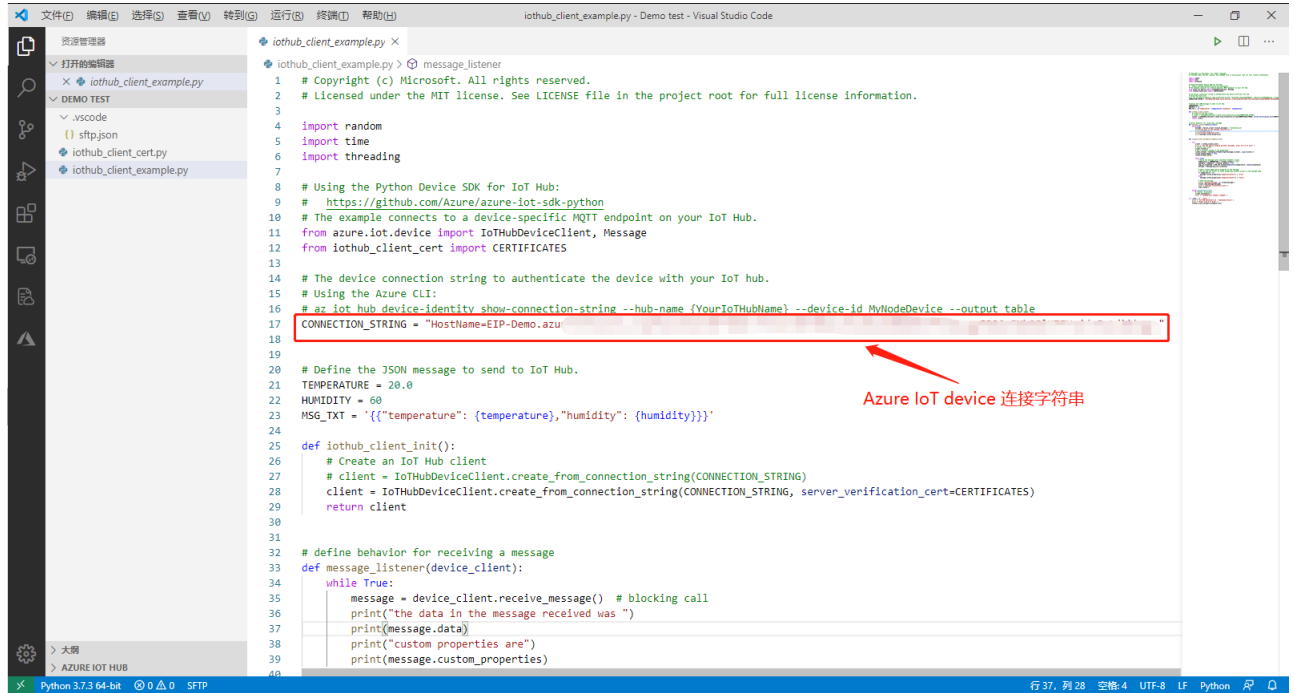


开始测试

- 配置`iothub_client_example.py`
- 使用Azure IoT Tools查看上报数据
- 使用Azure IoT Tools下发数据

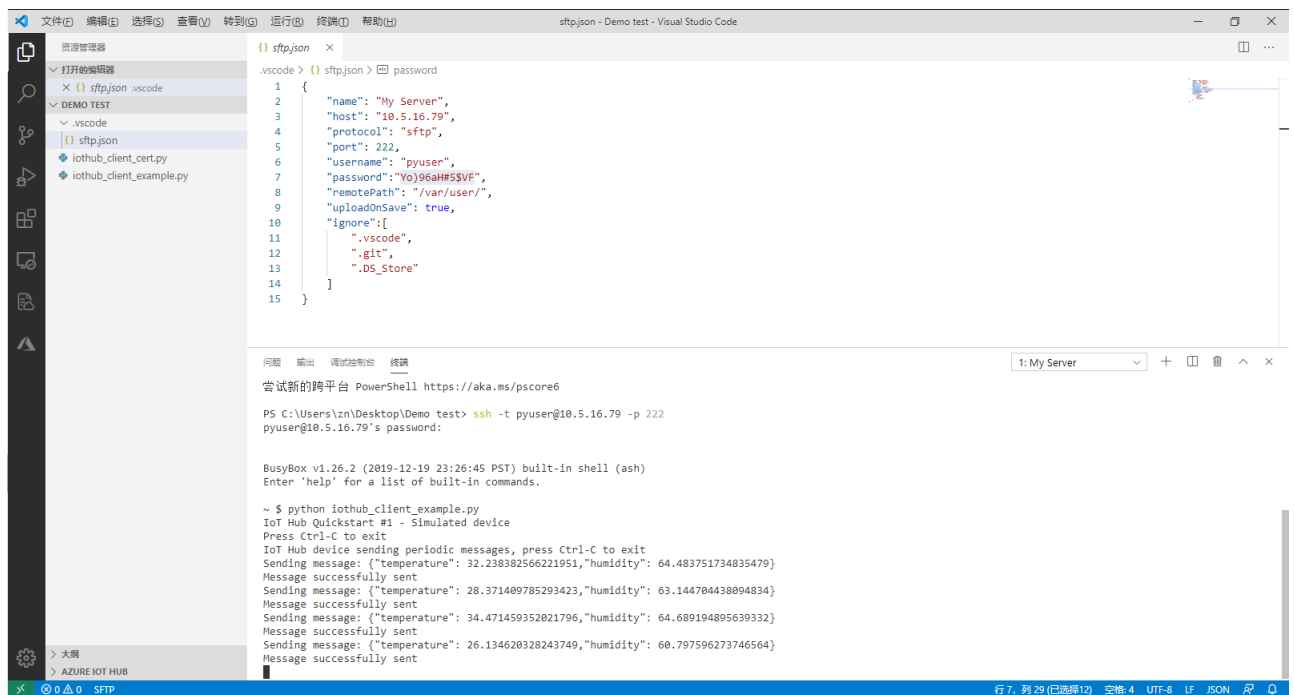
- 步骤1：配置`iothub_client_example.py`

在VS Code中打开项目文件夹并选中`iothub_client_example.py`，根据你的实际情况修改脚本中的`CONNECTION_STRING`参数。



```
1 # Copyright (c) Microsoft. All rights reserved.
2 # Licensed under the MIT license. See LICENSE file in the project root for full license information.
3
4 import random
5 import time
6 import threading
7
8 # Using the Python Device SDK for IoT Hub:
9 # https://github.com/Azure/azure-iot-sdk-python
10 # The example connects to a device-specific MQTT endpoint on your IoT Hub.
11 from azure.iot.device import IoTHubDeviceClient, Message
12 from iothub_client_cert import CERTIFICATES
13
14 # The device connection string to authenticate the device with your IoT hub.
15 # Using the Azure CLI:
16 # az iot-hub device-identity show-connection-string --hub-name {YourIoTHubName} --device-id MyNodeDevice --output table
17 CONNECTION_STRING = 'HostName=EIP-Demo.azure'
18
19
20 # Define the JSON message to send to IoT Hub.
21 TEMPERATURE = 20.0
22 HUMIDITY = 60
23 MSG_TXT = '{"temperature": {temperature}, "humidity": {humidity}}'
24
25 def iothub_client_init():
26     # Create an IoT Hub client
27     # client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)
28     client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING, server_verification_cert=CERTIFICATES)
29     return client
30
31 # define behavior for receiving a message
32 def message_listener(device_client):
33     while True:
34         message = device_client.receive_message() # blocking call
35         print("the data in the message received was ")
36         print(message.data)
37         print("custom properties are")
38         print(message.custom_properties)
```

- 步骤2：使用Azure IoT Tools查看上报数据
建立与InGateway的SFTP连接并运行iothub_client_example.py



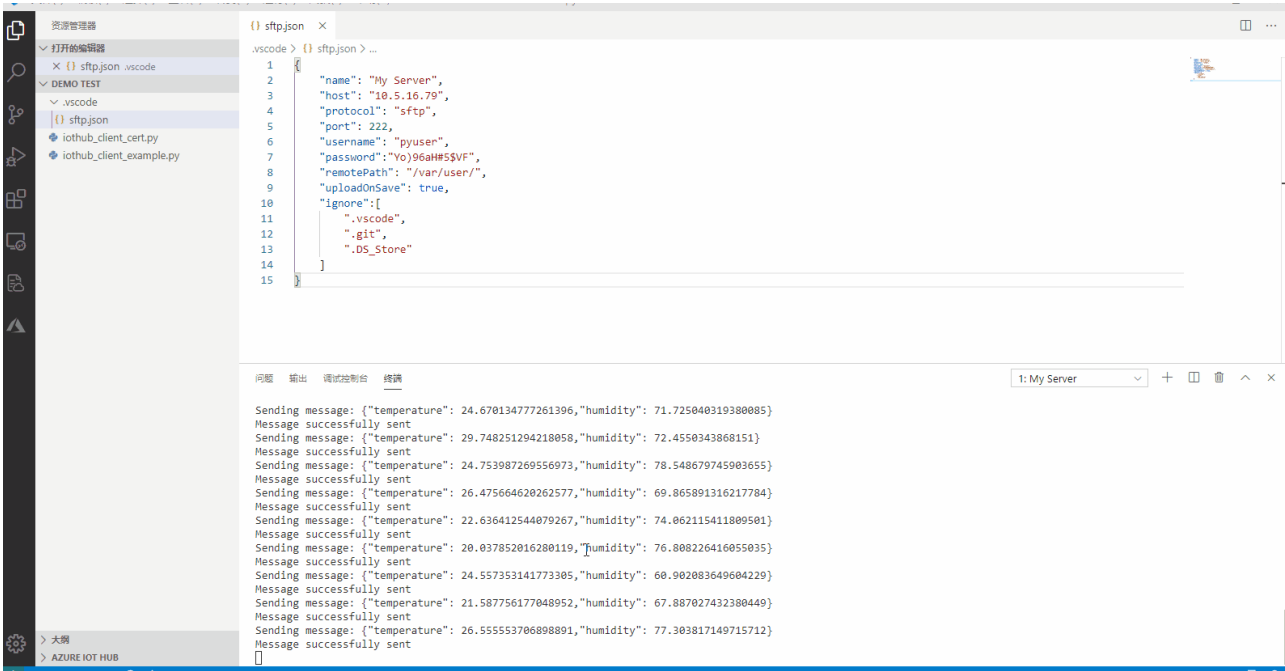
```
1 {
2   "name": "My Server",
3   "host": "10.5.16.79",
4   "protocol": "sftp",
5   "port": 222,
6   "username": "pyuser",
7   "password": "Yo96aH55VF",
8   "remotePath": "/var/user/",
9   "uploadOnSave": true,
10  "ignore": [
11    ".vscode",
12    ".git",
13    ".DS_Store"
14  ]
15 }
```

```
PS C:\Users\zn\Desktop\Demo test> ssh -t pyuser@10.5.16.79 -p 222
pyuser@10.5.16.79's password:

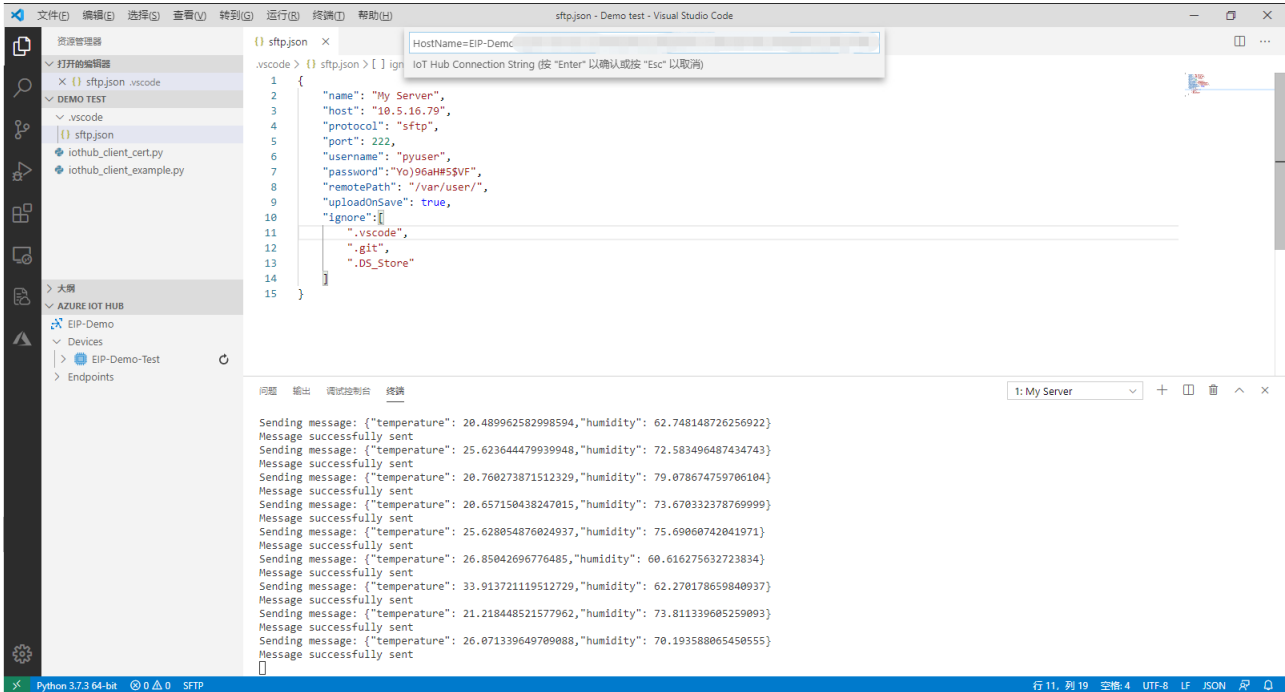
BusyBox v1.26.2 (2019-12-19 23:26:45 PST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ $ python iothub_client_example.py
IoT Hub Quickstart #1 - Simulated device
Press Ctrl-C to exit
IoT Hub device sending periodic messages, press Ctrl-C to exit
Sending message: {"temperature": 32.238382566221951, "humidity": 64.483751734835479}
Message successfully sent
Sending message: {"temperature": 28.371409785293423, "humidity": 63.144704438094834}
Message successfully sent
Sending message: {"temperature": 34.471459352021796, "humidity": 64.689194895639332}
Message successfully sent
Sending message: {"temperature": 26.134620328243749, "humidity": 60.797596273746564}
Message successfully sent
```

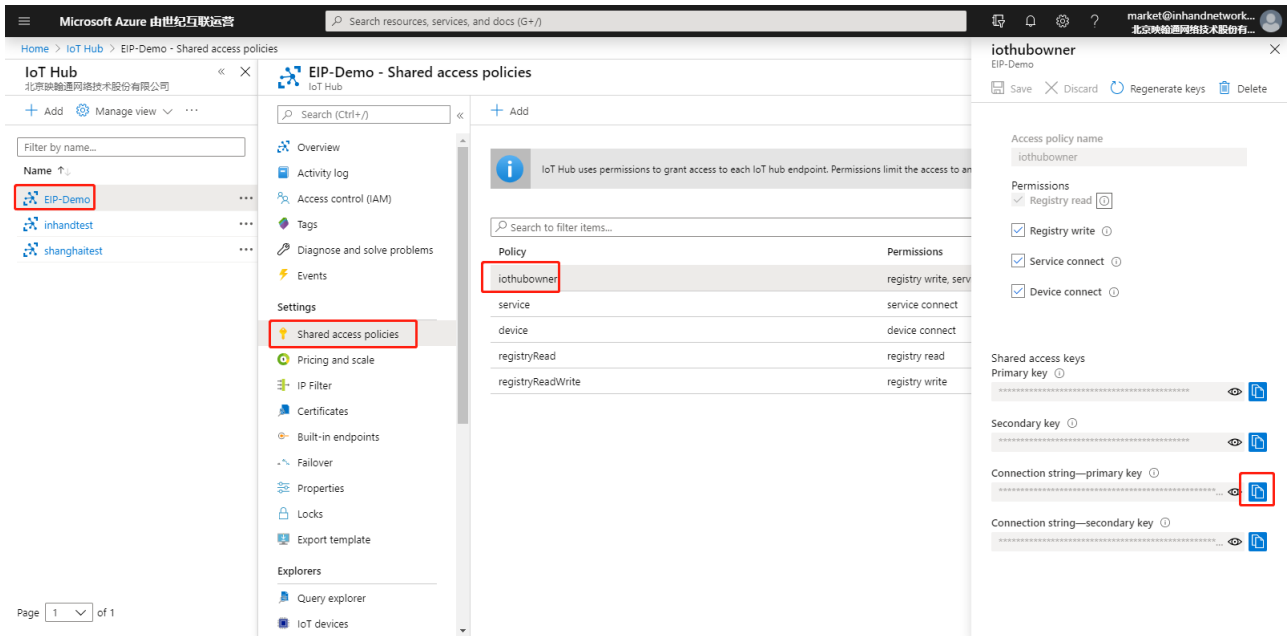
在“AZURE IOT HUB”模块中设置IoT Hub的连接字符串以建立与IoT Hub的连接。



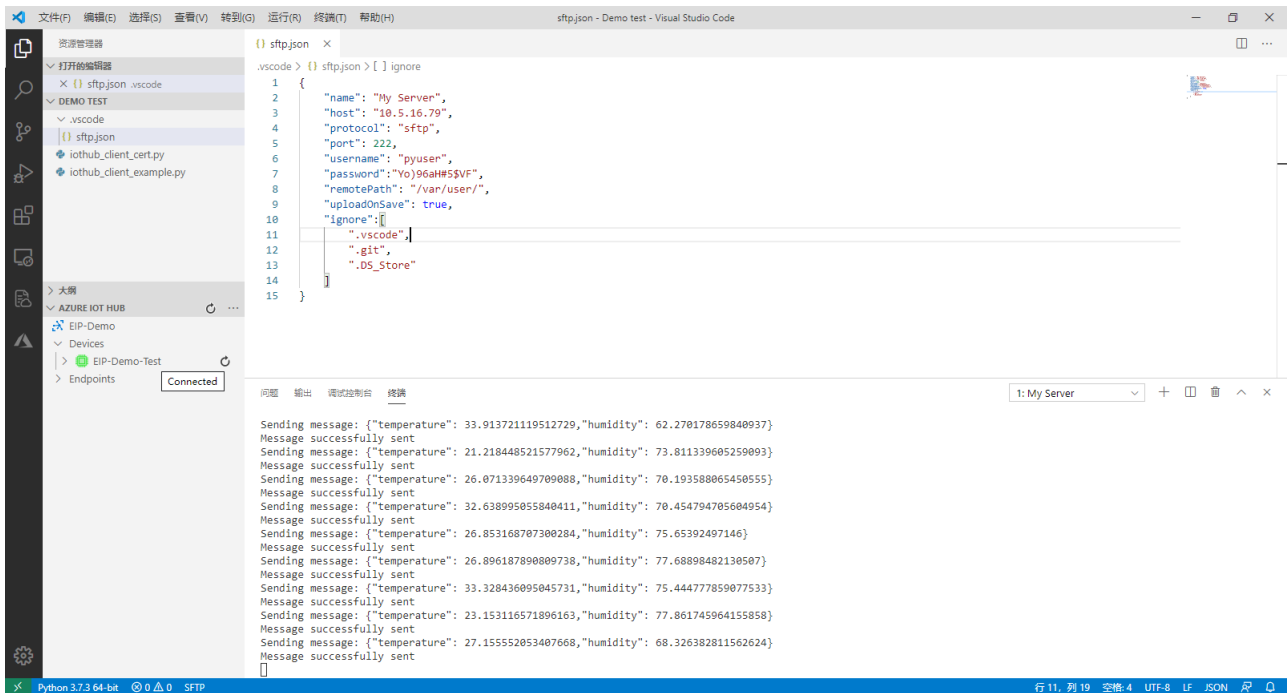
随后会提示你输入IoT Hub Connexion String（IoT Hub连接字符串）。



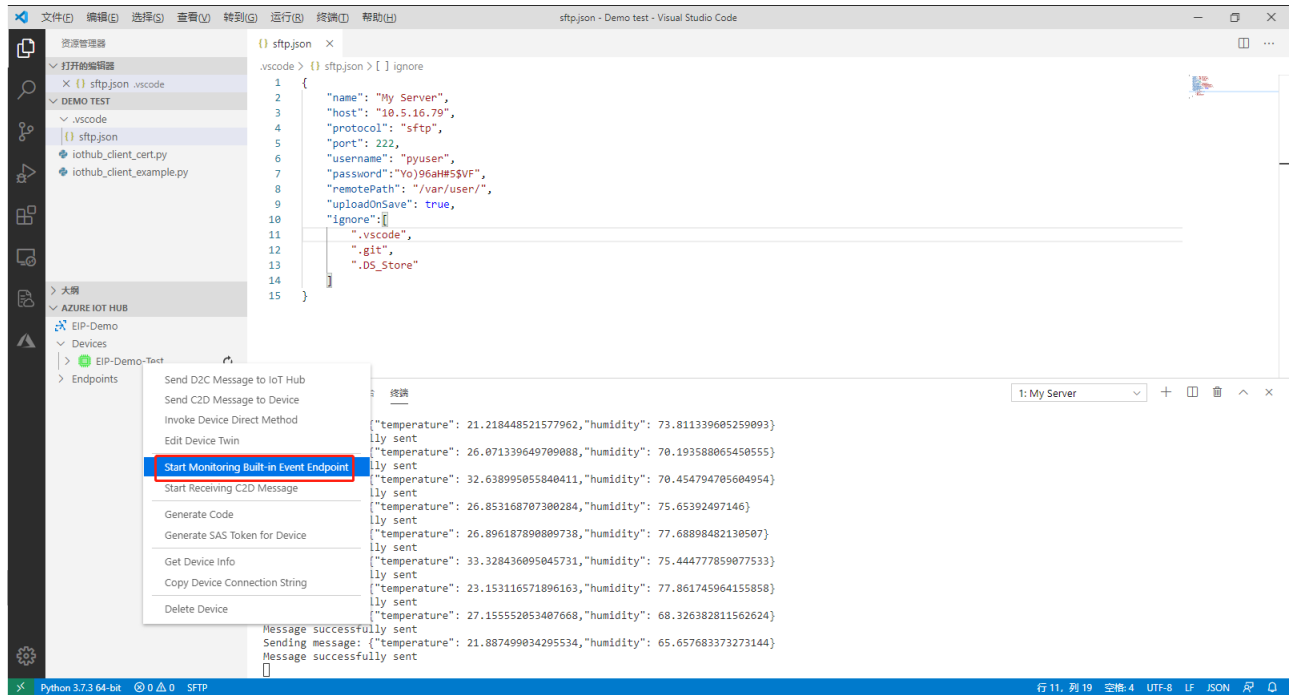
IoT Hub连接字符串可从“Azure IoT Hub”页面复制。



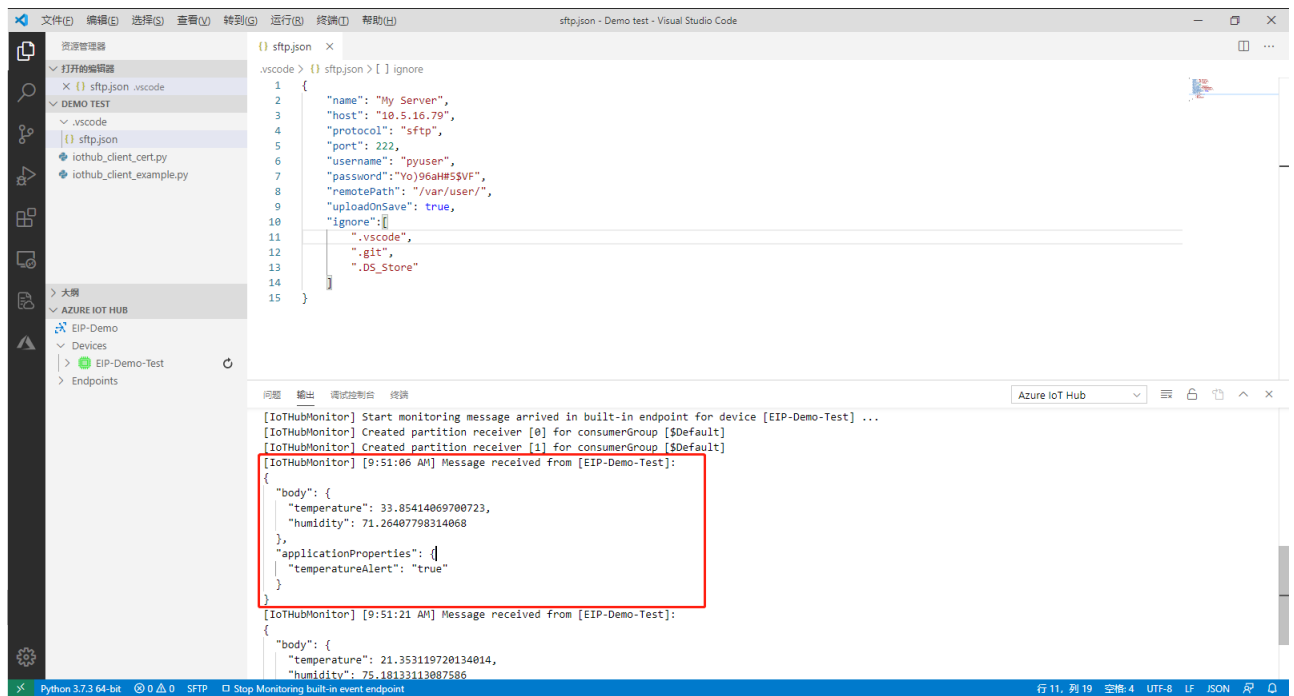
输入IoT Hub Connection String后可以看到该IoT Hub下的“IoT Device”且状态为Connected。



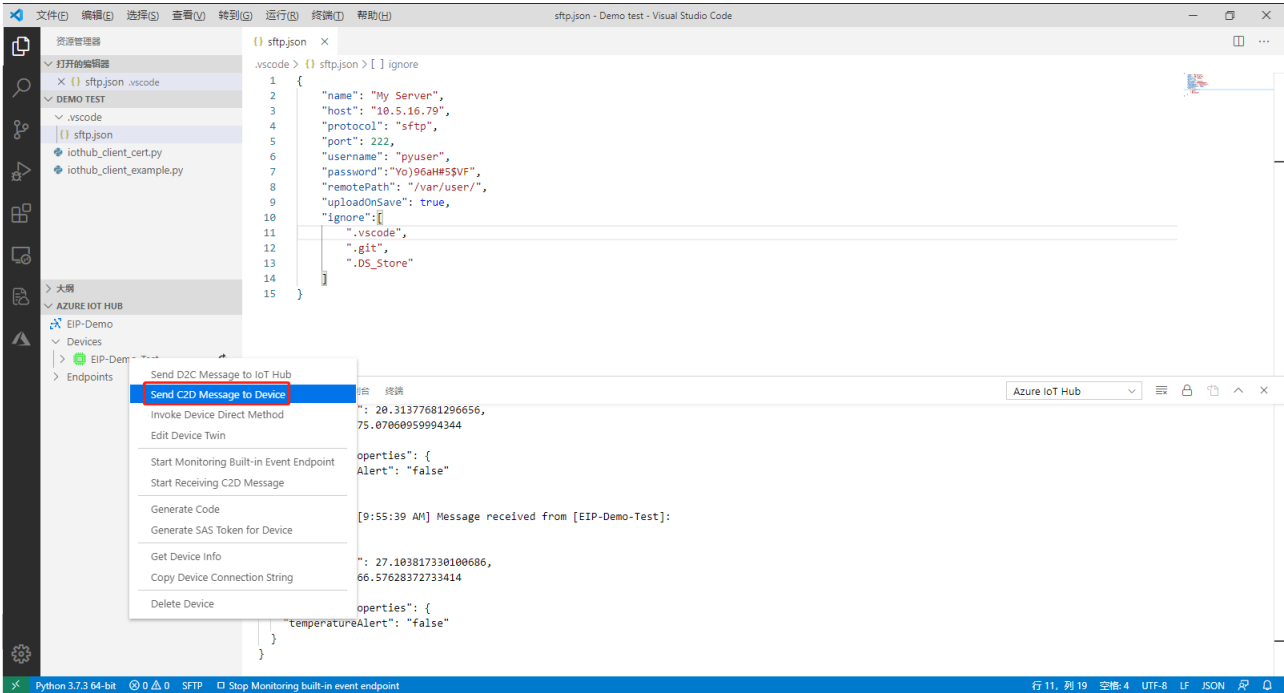
右击“IoT Device”并在菜单中选择Start Monitoring Built-in Event Endpoint以查看InGateway推送到IoT Hub的数据。



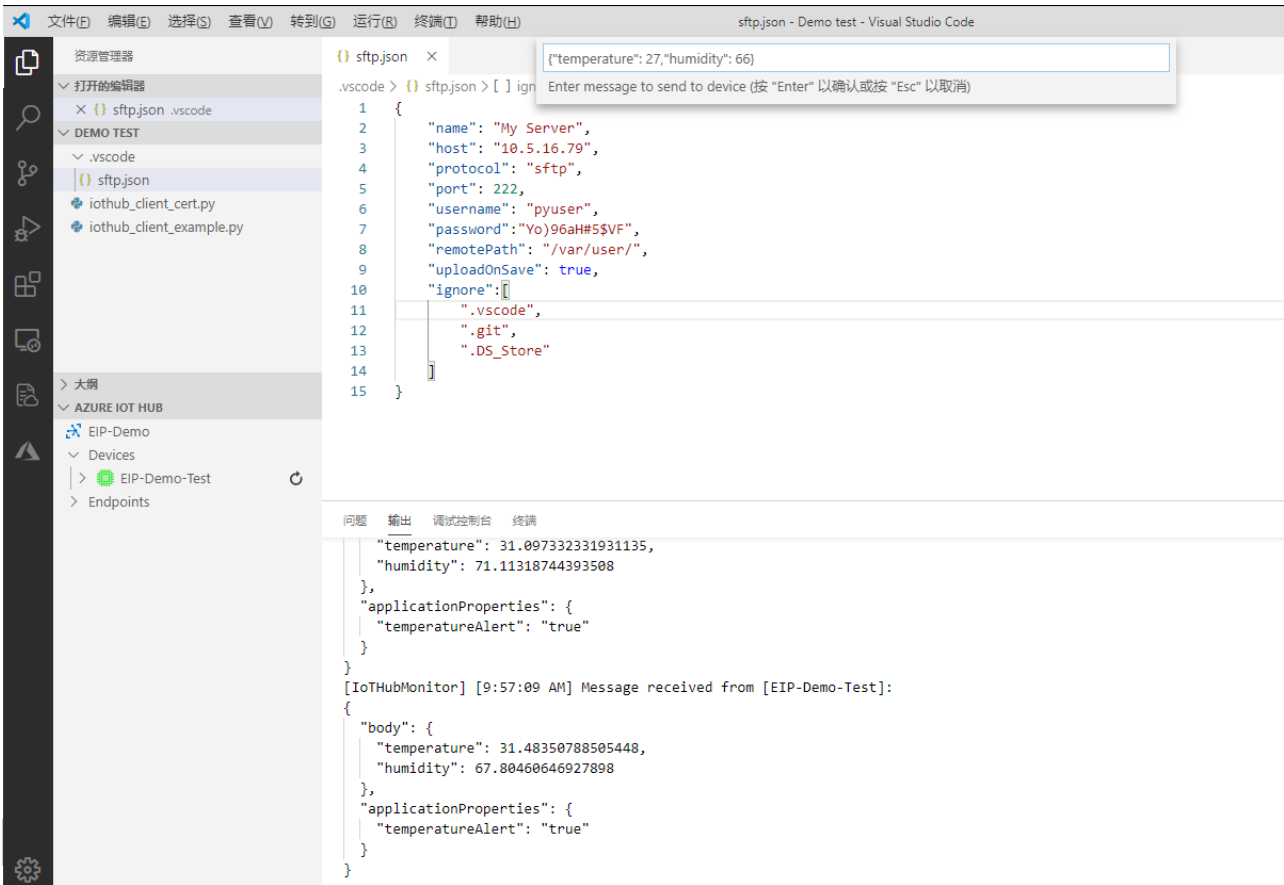
随后在“输出”窗口可以查看IoT Hub接收到的数据。



- 步骤3：使用Azure IoT Tools下发数据
右击“IoT Device”并在菜单中选择 **Send C2D Message to Device** 以下发数据至InGateway。



在下发框中输入需要下发的数据，如{"temperature": 27,"humidity": 66}



在“输出”窗口出现下图所示日志说明数据下发成功：

```
.vscode > {} sftp.json > [ ] ignore
1
2 {
3   "name": "My Server",
4   "host": "10.5.16.79",
5   "protocol": "sftp",
6   "port": 222,
7   "username": "pyuser",
8   "password": "Yo96aH5$VF",
9   "remotePath": "/var/user/",
10  "uploadOnSave": true,
11  "ignore": [
12    ".vscode",
13    ".git",
14    ".DS_Store"
15  ]
16 }
```

```
{
  "applicationProperties": {
    "temperatureAlert": "false"
  }
}
[IoT Hub Monitor] [9:57:55 AM] Message received from [EIP-Demo-Test]:
{
  "body": {
    "temperature": 22.6687936211124,
    "humidity": 78.54943490839443
  },
  "applicationProperties": {
    "temperatureAlert": "false"
  }
}
[C2DMessage] Sending message to [EIP-Demo-Test] ...
[C2DMessage] [Success] Message sent to [EIP-Demo-Test]
```

随后可在“终端”中查看InGateway接收到的下发数据。

```
1: My Server
+
-
x
Sending message: {"temperature": 31.097332331931135,"humidity": 71.113187443935075}
Message successfully sent
Sending message: {"temperature": 31.483507885054479,"humidity": 67.80460646927898}
Message successfully sent
Sending message: {"temperature": 24.521691252272692,"humidity": 66.508254383551310}
Message successfully sent
Sending message: {"temperature": 28.19627961355544,"humidity": 64.525097649039296}
Message successfully sent
Sending message: {"temperature": 22.6687936211124,"humidity": 78.549434908394431}
Message successfully sent
the data in the message received was
b'{"temperature": 27,"humidity": 66}'
custom properties are
{}
Sending message: {"temperature": 21.049248444905904,"humidity": 60.957959694600355}
Message successfully sent
Sending message: {"temperature": 20.255741900928143,"humidity": 72.44941757816153}
Message successfully sent
```

至此，完成了上传数据至Azure IoT Hub并接收Azure IoT Hub下发的数据。

FAQ

Q1：脚本运行一段时间后，数据无法上传至Azure IoT Hub了。

A1：请检查与Azure IoT Hub的连接是否正常，当上报消息超过8000条时可能会导致无法正常上报数据。

