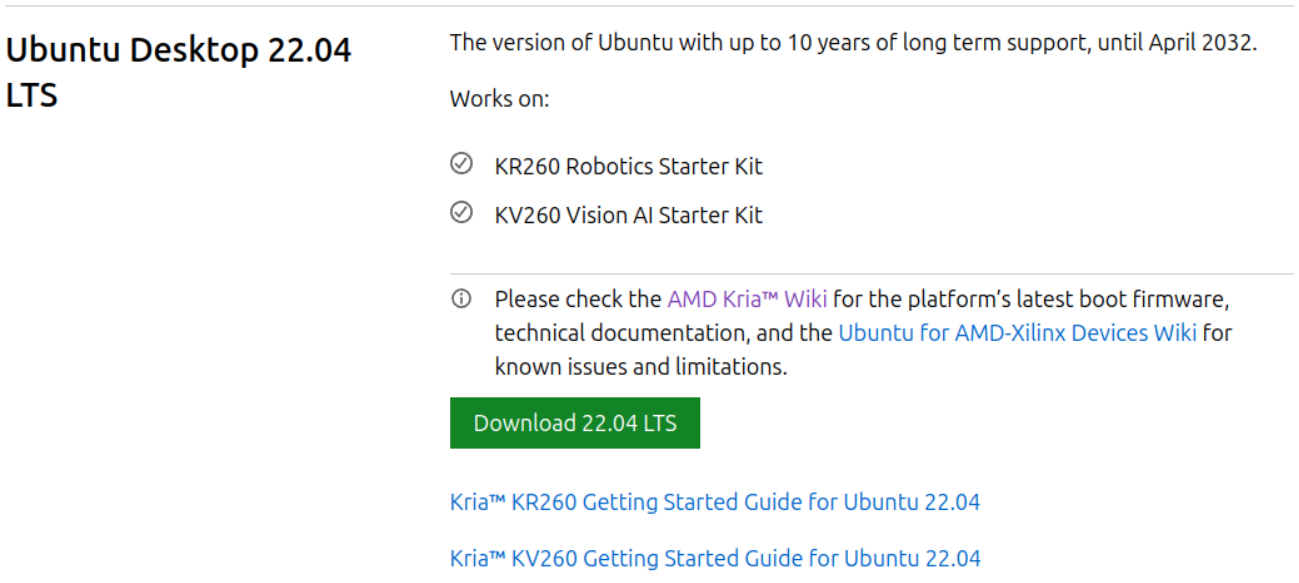
2023.12.28/v0.1 Sanam Shakya

Preparing Ubuntu 22.04 OS for KRIA KV260 board

Download the Ubuntu 22.04 image from the download link



Ubuntu Desktop 22.04 LTS

The version of Ubuntu with up to 10 years of long term support, until April 2032.

Works on:

- KR260 Robotics Starter Kit
- KV260 Vision AI Starter Kit

Please check the AMD Kria™ Wiki for the platform's latest boot firmware, technical documentation, and the Ubuntu for AMD-Xilinx Devices Wiki for known issues and limitations.

Kria™ KR260 Getting Started Guide for Ubuntu 22.04

Kria™ KV260 Getting Started Guide for Ubuntu 22.04

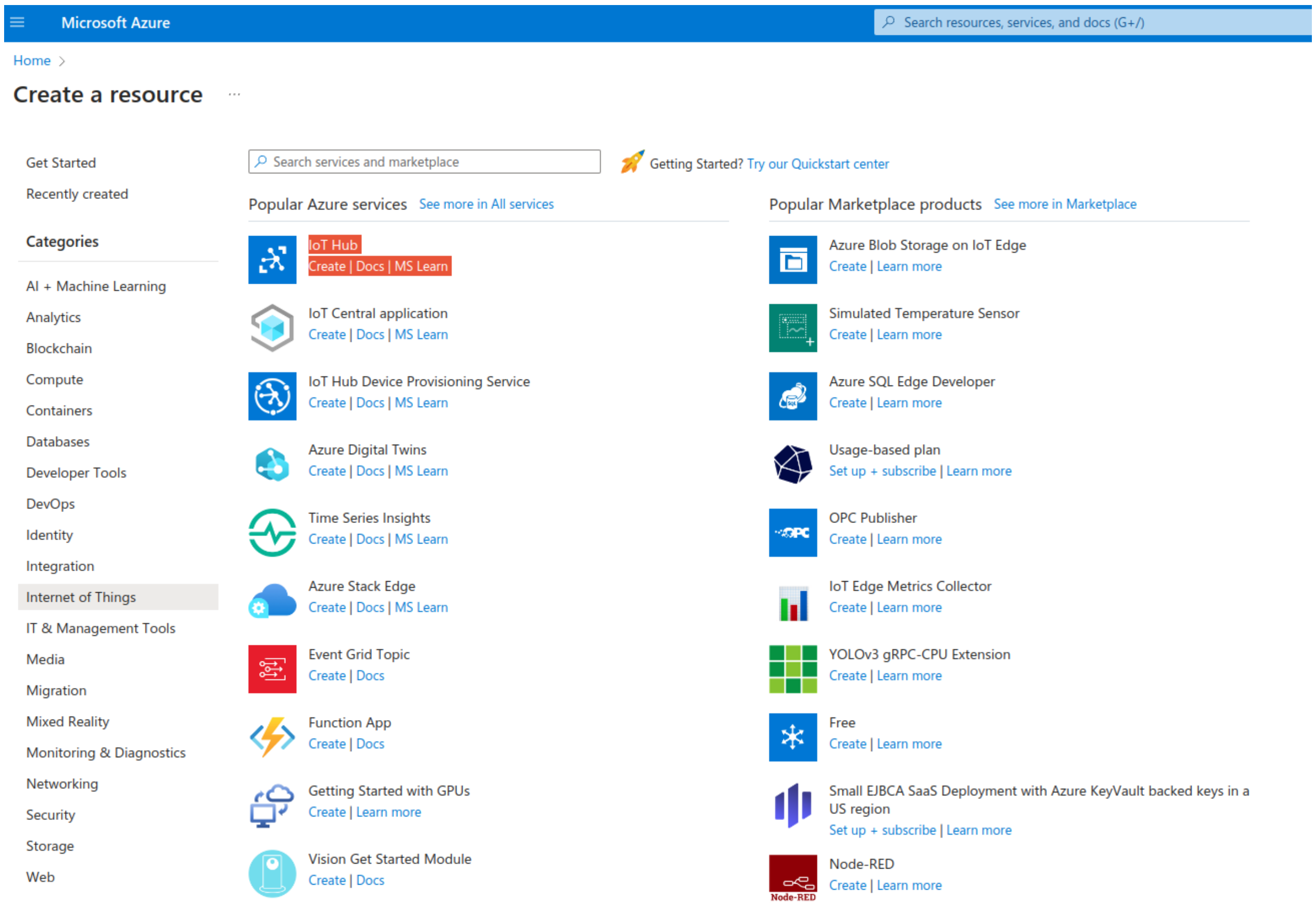
Next, prepare the SD card with the above downloaded Ubuntu image using burning tools like Balena Etcher.

Now boot the KV260 with the SD card with Ethernet and USB to Serial cable connected to board. We will be using Serial console for initial access and debugging and Ethernet network for accessing through SSH and KV260 connected to the internet.

For initial login here are the Login Details:
Username : ubuntu
Password: ubuntu
This will ask to change the password. So update the password and login the system.
After successful login, one can access the KV260 device console.

Create IoT Hub in Azure Portal:

- Go to Azure portal " https://portal.azure.com ".
- Create a resource >> IoT Hub.



Microsoft Azure
Search resources, services, and docs (G+/)
Home >
Search services and marketplace
Getting Started? Try our Quickstart center

Create a resource

Get Started
Recently created

Categories

AI + Machine Learning
Analytics
Blockchain
Compute
Containers
Databases
Developer Tools
DevOps
Identity
Integration
Internet of Things
IT & Management Tools
Media
Migration
Mixed Reality
Monitoring & Diagnostics
Networking
Security
Storage
Web

Popular Azure services — See more in All services

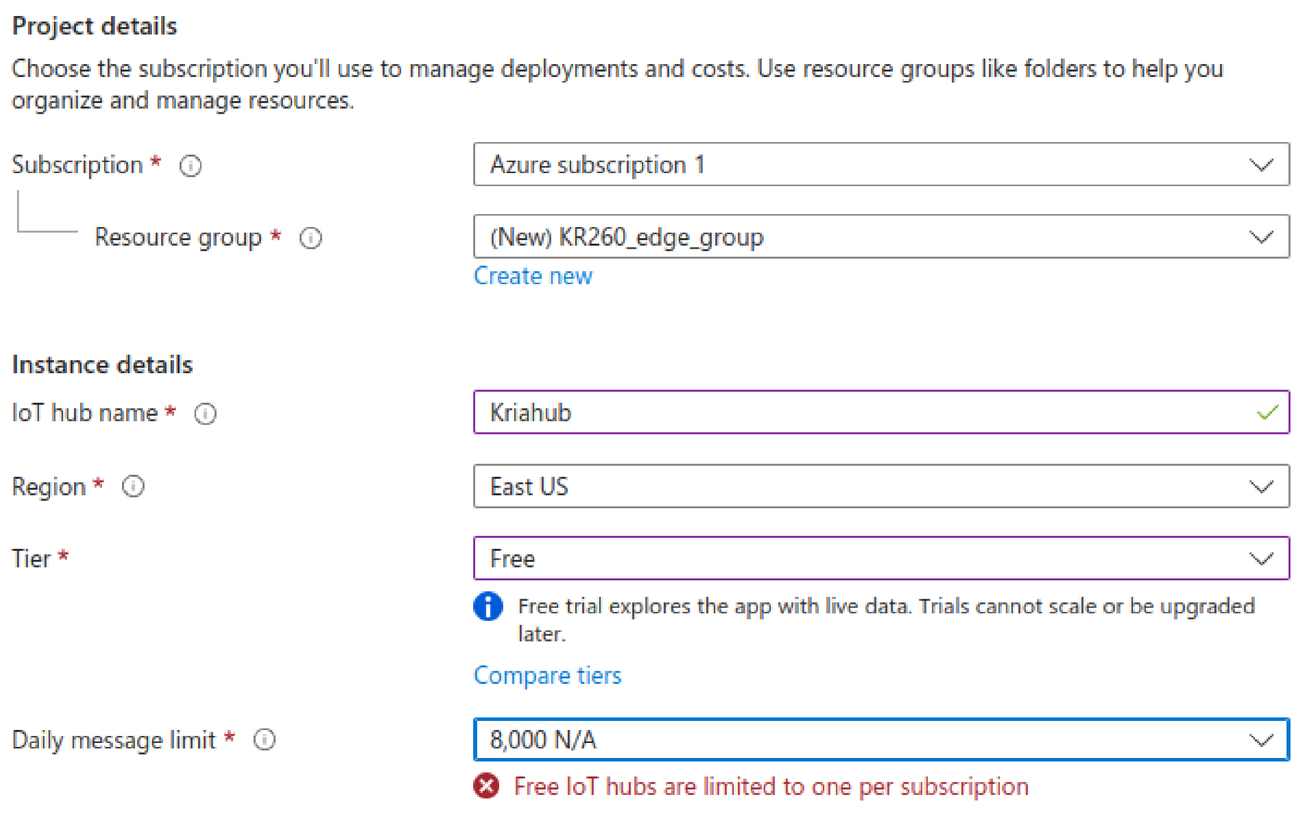
- IoT Hub — Create | Docs | MS Learn
- IoT Central application — Create | Docs | MS Learn
- IoT Hub Device Provisioning Service — Create | Docs | MS Learn
- Azure Digital Twins — Create | Docs | MS Learn
- Time Series Insights — Create | Docs | MS Learn
- Azure Stack Edge — Create | Docs | MS Learn
- Event Grid Topic — Create | Docs
- Function App — Create | Docs
- Getting Started with GPUs — Create | Learn more
- Vision Get Started Module — Create | Docs

Popular Marketplace products — See more in Marketplace

- Azure Blob Storage on IoT Edge — Create | Learn more
- Simulated Temperature Sensor — Create | Learn more
- Azure SQL Edge Developer — Create | Learn more
- Usage-based plan — Set up + subscribe | Learn more
- OPC Publisher — Create | Learn more
- IoT Edge Metrics Collector — Create | Learn more
- YOLOv3 gRPC-CPU Extension — Create | Learn more
- Free — Create | Learn more
- Small EJBCA SaaS Deployment with Azure KeyVault backed keys in a US region — Set up + subscribe | Learn more
- Node-RED — Create | Learn more

Next, create one IoT Hub Service and fill in the necessary details

Next, create one IoT Hub Service and fill in the necessary details



Project details

Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription *	Azure subscription 1
Resource group *	(New) KR260_edge_group

Instance details

IoT hub name *	Kriahub
Region *	East US
Tier *	Free
Daily message limit *	8,000 N/A

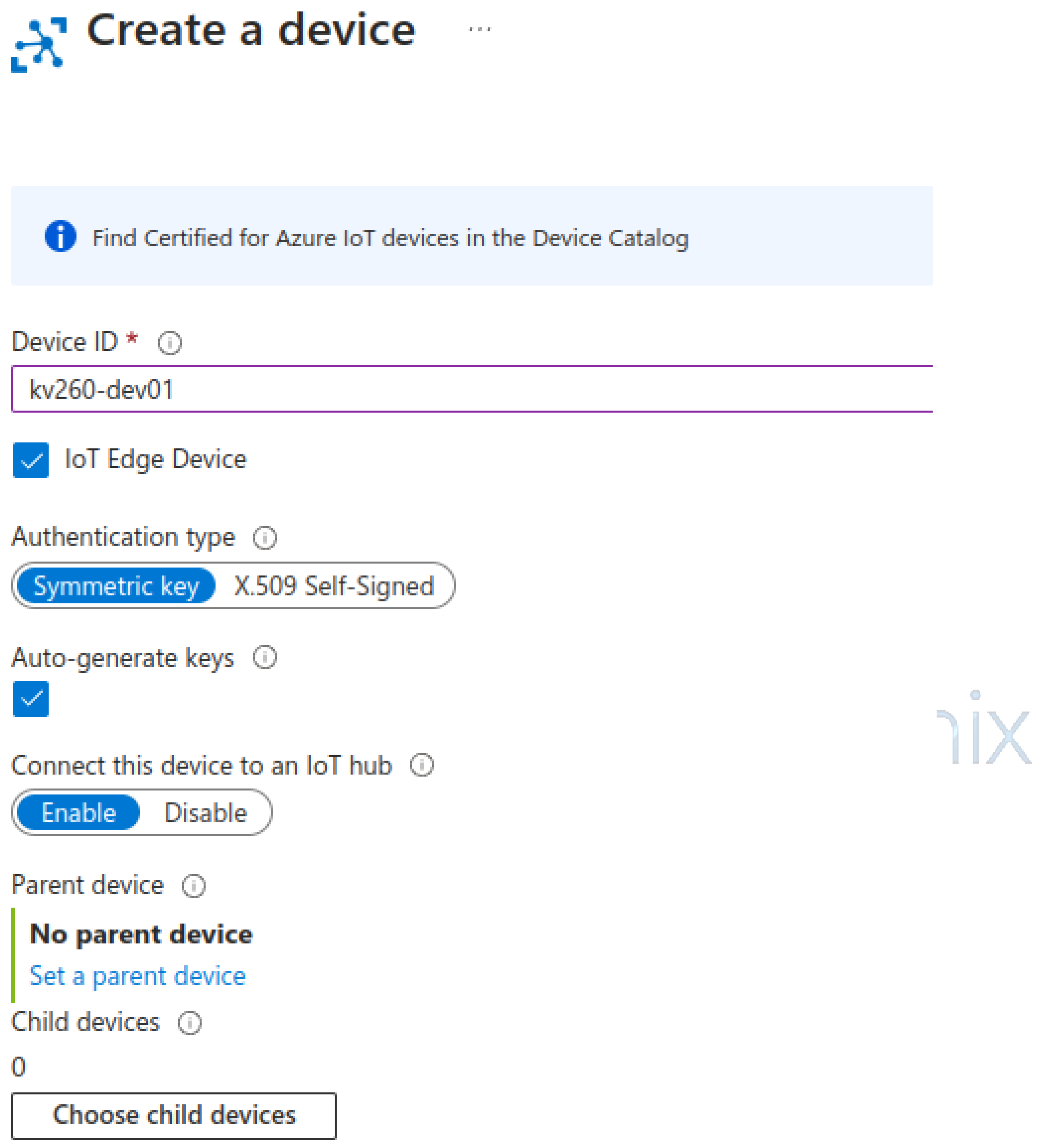
Free trial explores the app with live data. Trials cannot scale or be upgraded later.

Free IoT hubs are limited to one per subscription

- Click on Review+ Create button to create the Azure IoT Hub.
- Next, create a device where you can actually receive some data from the hardware.

Create an IoT Device

Go to the IoT Device and click on new, and give the device ID


Create a device

Find Certified for Azure IoT devices in the Device Catalog

Device ID *: kv260-dev01

[✓] IoT Edge Device

Authentication type: [✓ Symmetric key] X.509 Self-Signed

Auto-generate keys: [✓]

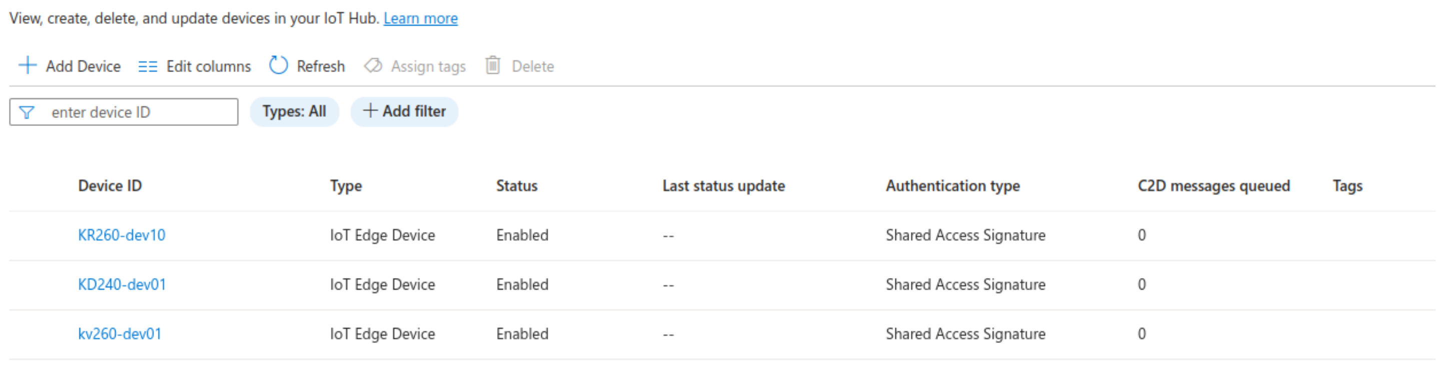
Connect this device to an IoT hub: [✓ Enable] Disable

Parent device: **No parent device**

Set a parent device

Child devices: 0

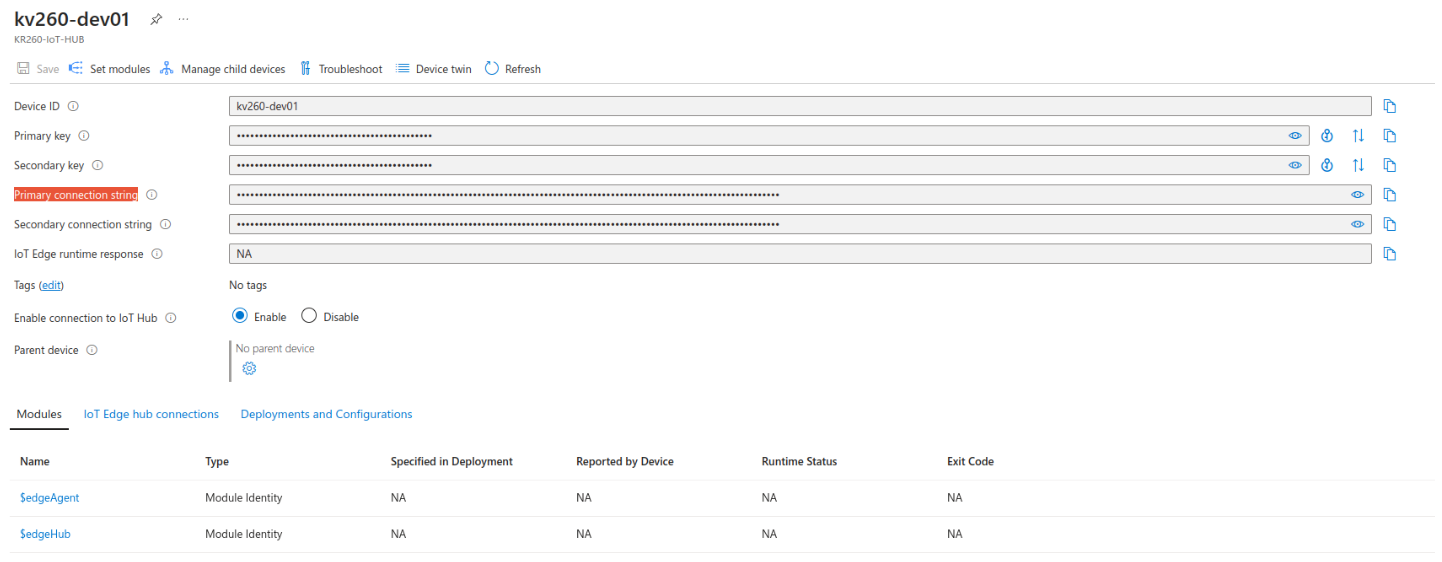
After this device will be available in the IoT hub Device list.



View, create, delete, and update devices in your IoT Hub. Learn more

Device ID	Type	Status	Last status update	Authentication type	C2D messages queued	Tags
KR260-dev10	IoT Edge Device	Enabled	--	Shared Access Signature	0	
KD240-dev01	IoT Edge Device	Enabled	--	Shared Access Signature	0	
kv260-dev01	IoT Edge Device	Enabled	--	Shared Access Signature	0	

Next, look into device information for getting the keys and connection string.



kv260-dev01
KR260-IoT-HUB

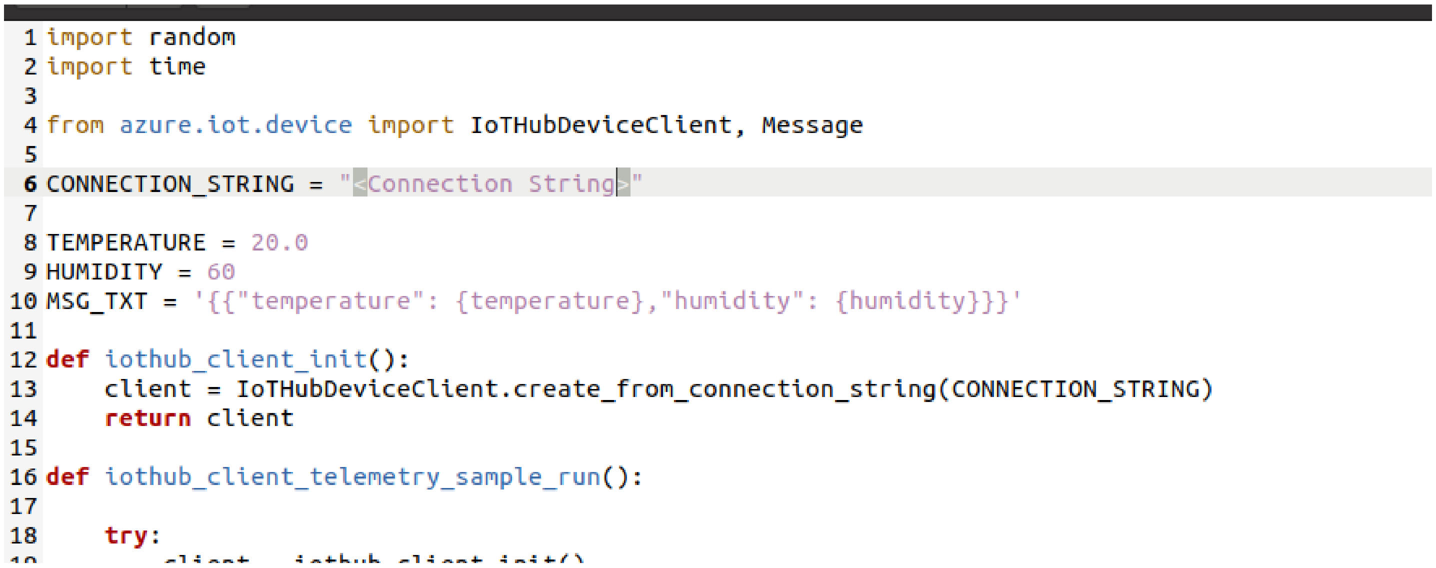
Device ID	kv260-dev01
Primary key	••
Secondary key	••
Primary connection string	•••
Secondary connection string	•••
IoT Edge runtime response	NA
Tags (edit)	No tags
Enable connection to IoT Hub	[✓ Enable] Disable
Parent device	No parent device

Name	Type	Specified in Deployment	Reported by Device	Runtime Status	Exit Code
$edgeAgent	Module Identity	NA	NA	NA	NA
$edgeHub	Module Identity	NA	NA	NA	NA

Copy the “Primary Connection String” which will be used in the python application for sending the sensor data to IoT hub.

Adding python application in KRIA

Copy the simulated_sensor.py example code in KV260 board.
Next update the “CONNECTION STRING” with the above Primary Connection string.

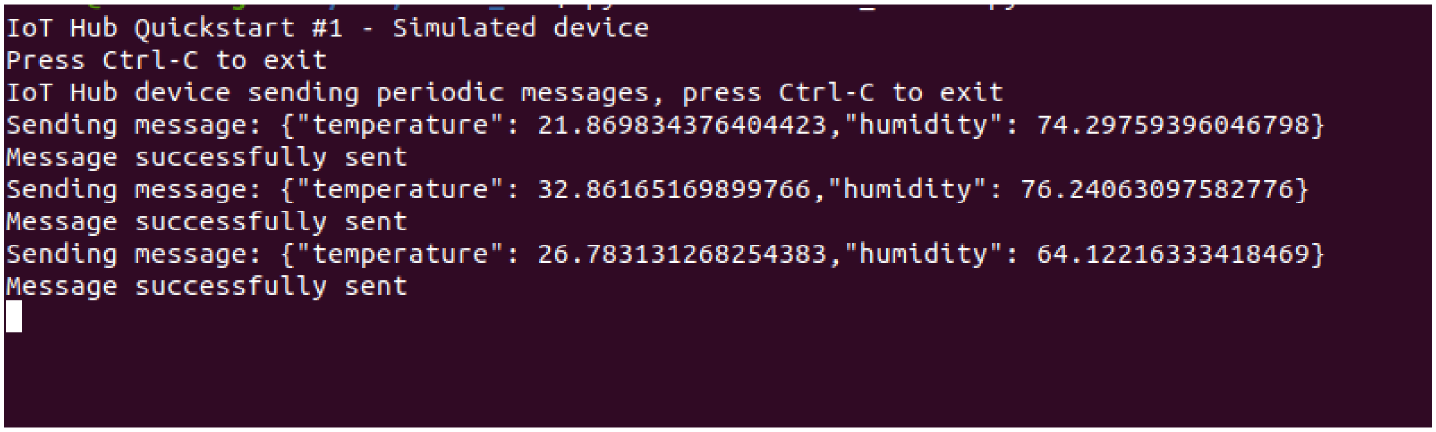


```python
import random
import time

from azure.iot.device import IoTHubDeviceClient, Message

CONNECTION_STRING = "<Connection String>"

TEMPERATURE = 20.0
HUMIDITY = 60
MSG_TXT = '{{"temperature": {temperature},"humidity": {humidity}}}'

def iothub_client_init():
    client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)
    return client

def iothub_client_telemetry_sample_run():

    try:
        client = iothub_client_init()
```

Then run the simulated application in console:

```
python3 simulated_sensor.py
```

Here is the console log after successful message send to Azure IoT hub.

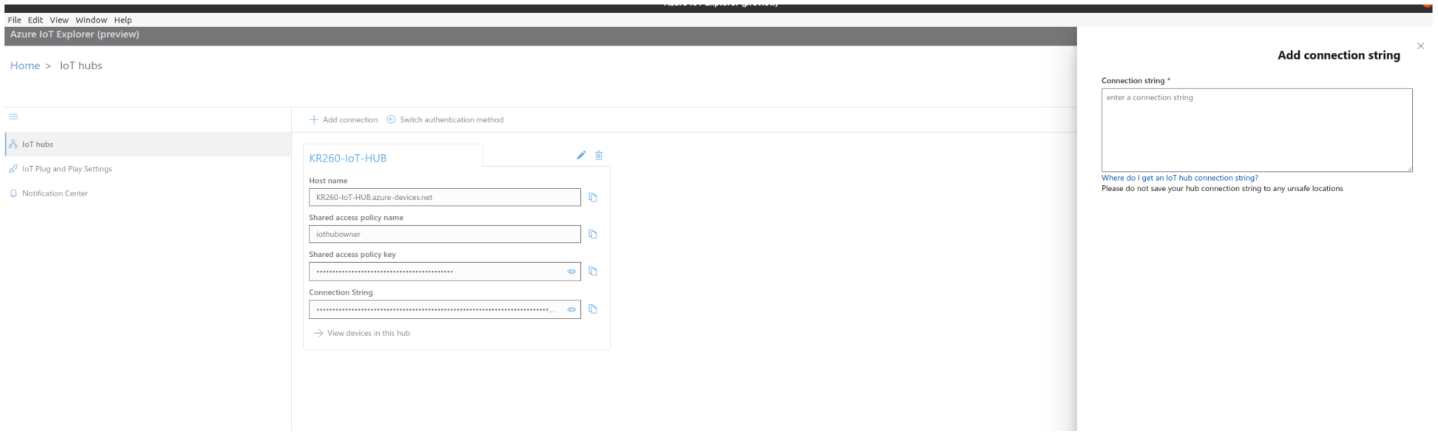


```
IoT Hub Quickstart #1 - Simulated device
Press Ctrl-C to exit
IoT Hub device sending periodic messages, press Ctrl-C to exit
Sending message: {"temperature": 21.869834376404423,"humidity": 74.29759396046798}
Message successfully sent
Sending message: {"temperature": 32.86165169899766,"humidity": 76.24063097582776}
Message successfully sent
Sending message: {"temperature": 26.783131268254383,"humidity": 64.12216333418469}
Message successfully sent
```

Viewing message in Host Machine

For viewing the message published by Azure IoT Device in KV260, one can use Azure IoT explorer available in following link:
https://github.com/Azure/azure-iot-explorer/releases

In IoT HUbs page of the application, in +Add connection copy the connection string for the IoT hub and save the configs:


File Edit View Window Help
Azure IoT Explorer (preview)
Home > IoT hubs
IoT hubs
IoT Plug and Play Settings
Notification Center
Add connection
Switch authentication method
View devices in this hub

KR260-IoT-HUB

Host name: KR260-IoT-HUB.azure-devices.net

Shared access policy name: iothubowner

Shared access policy key: ••

Connection String: ••...

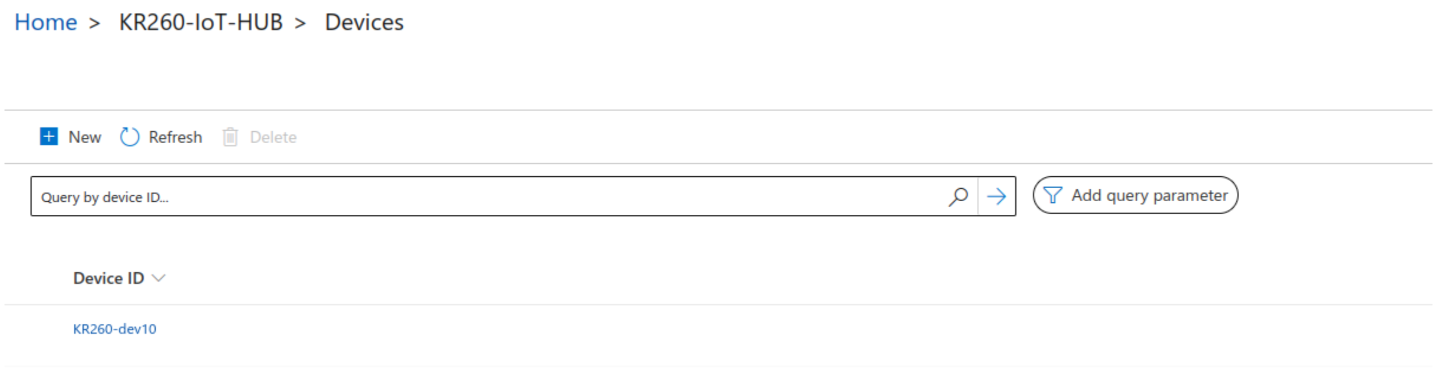
Add connection string

Connection string *: enter a connection string

Where do I get an IoT hub connection string?

Please do not save your hub connection string to any unsafe locations

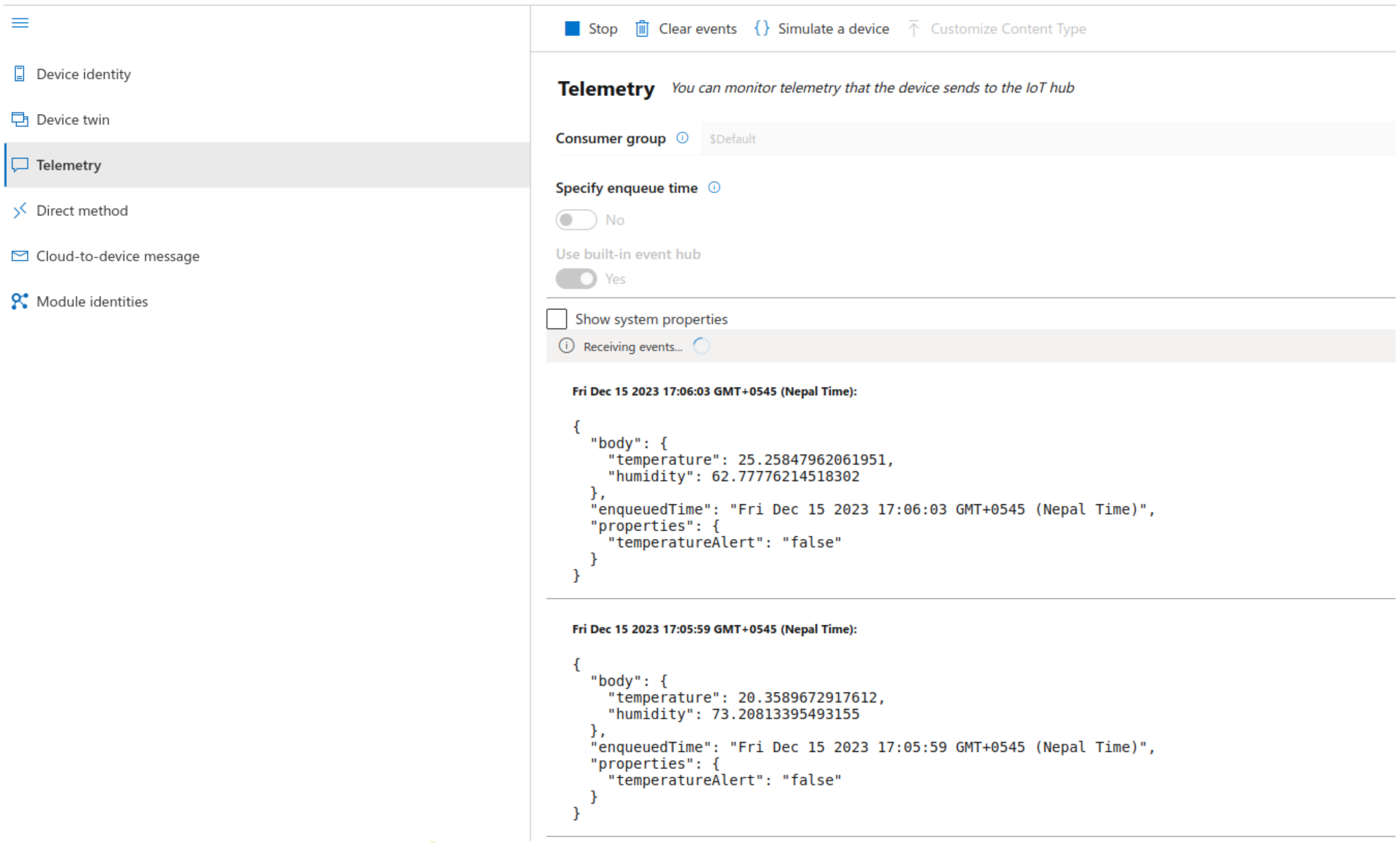
One can find the corresponding device list in the IoT HuB page of Azure IoT explorer application.



Device ID
KR260-dev10

Just click onto the device to view the device information and also the message send by python application running in the KV260 board.

For viewing the message send to device, go to Telemetry and click the >Start button.
After this one can view the message send to the device.



Telemetry *You can monitor telemetry that the device sends to the IoT hub*

Consumer group: $Default

Specify enqueue time: No

Use built-in event hub: Yes

Show system properties

Receiving events...

Fri Dec 15 2023 17:06:03 GMT+0545 (Nepal Time):

```
{
  "body": {
    "temperature": 25.25847962061951,
    "humidity": 62.77776214518302
  },
  "enqueuedTime": "Fri Dec 15 2023 17:06:03 GMT+0545 (Nepal Time)",
  "properties": {
    "temperatureAlert": "false"
  }
}
```

Fri Dec 15 2023 17:05:59 GMT+0545 (Nepal Time):

```
{
  "body": {
    "temperature": 20.3589672917612,
    "humidity": 73.20813395493155
  },
  "enqueuedTime": "Fri Dec 15 2023 17:05:59 GMT+0545 (Nepal Time)",
  "properties": {
    "temperatureAlert": "false"
  }
}
```