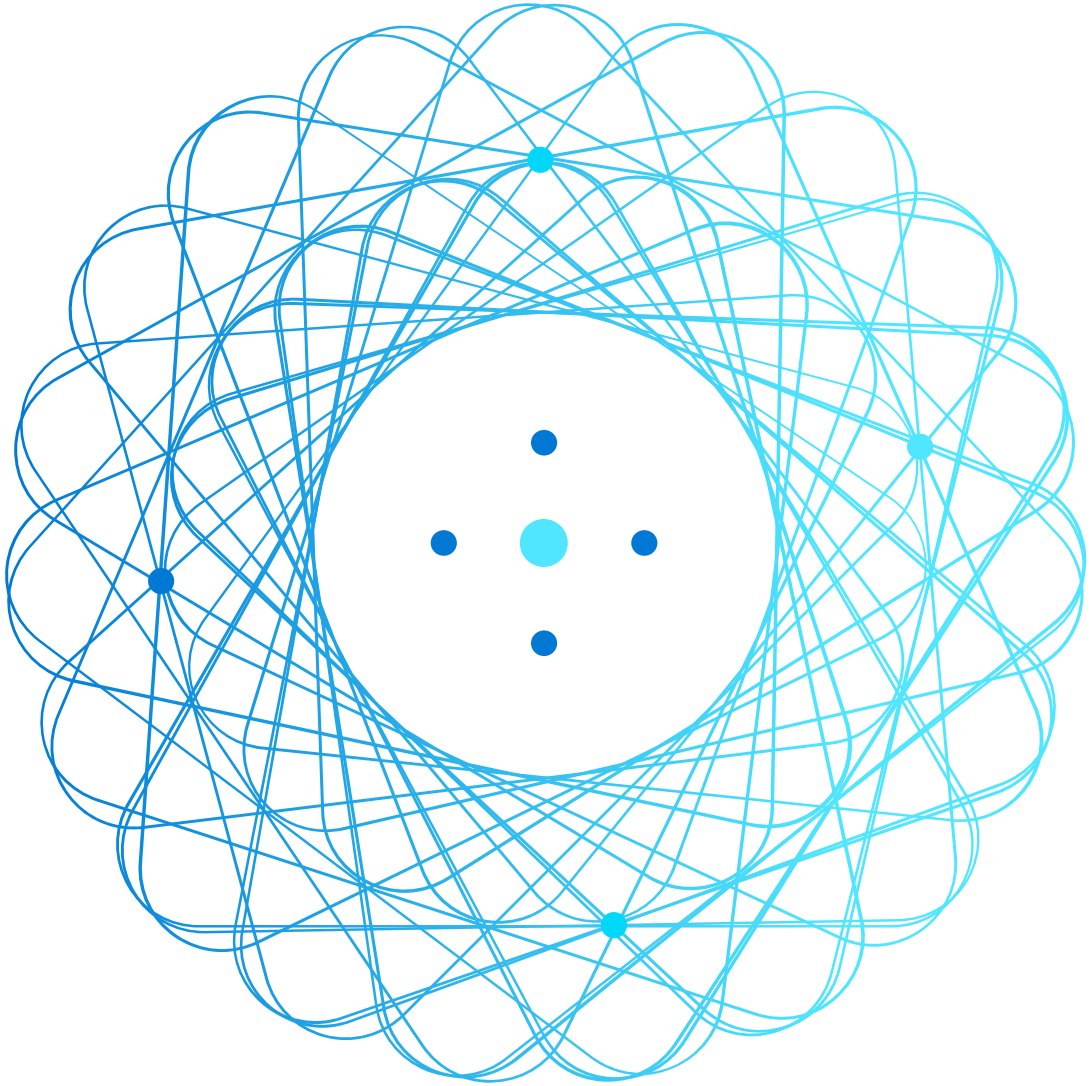
Microsoft Azure

AZ-220T01
Module 2:
Devices and device communication

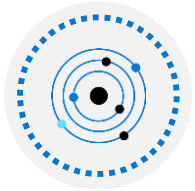


© Copyright Microsoft Corporation. All rights reserved.

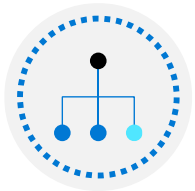
Lesson 1: Learning objectives

Module 2 – Learning objectives

Explain the core features of the IoT Hub service



Describe the lifecycle of an Azure IoT device

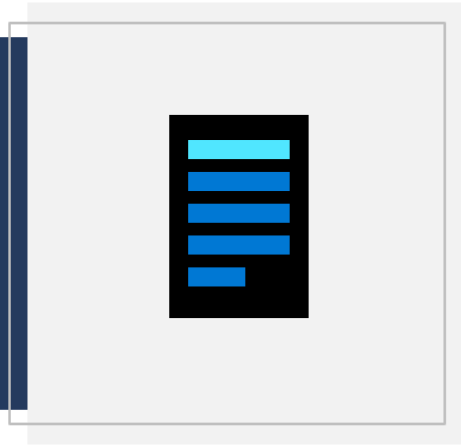


Describe how IoT Hub manages device identities and implements other security features

Register devices with the IoT Hub using the Azure portal, Azure CLI, and Visual Studio Code

Implement the IoT Hub Device and Service SDKs

Lesson 2: IoT Hub concepts



IoT Hub capabilities

IoT Hub capabilities are differentiated by *Tiers*

You can move from Basic to Standard on demand

Capability	Basic tier	Free/Standard tier
Device-to-cloud telemetry	Yes	Yes
Per-device identity	Yes	Yes
Message routing, message enrichments, and Event Grid integration	Yes	Yes
HTTP, AMQP, and MQTT protocols	Yes	Yes
Device Provisioning Service	Yes	Yes
Monitoring and diagnostics	Yes	Yes
Cloud-to-device messaging		Yes
Device twins, Module twins, and Device management		Yes
Device streams (preview)		Yes
Azure IoT Edge		Yes
IoT Plug and Play Preview		Yes

IoT Hub throughput

IoT Hub throughput is differentiated by *Editions*

You can upgrade or downgrade editions at any time

Within an edition, you pay for a certain number of *Units*

Autoscaling possible through an Azure Functions implementation

Tier edition	Sustained throughput	Sustained send rate
B1, S1	Up to 1111 KB/ minute per unit (1.5 GB/day/unit)	Average of 278 messages/minute per unit (400,000 messages/day per unit)
B2, S2	Up to 16 MB/ minute per unit (22.8 GB/day/unit)	Average of 4,167 messages/minute per unit (6 million messages/day per unit)
B3, S3	Up to 814 MB/ minute per unit (1144.4 GB/day/unit)	Average of 208,333 messages/minute per unit (300 million messages/day per unit)

IoT Hub endpoints

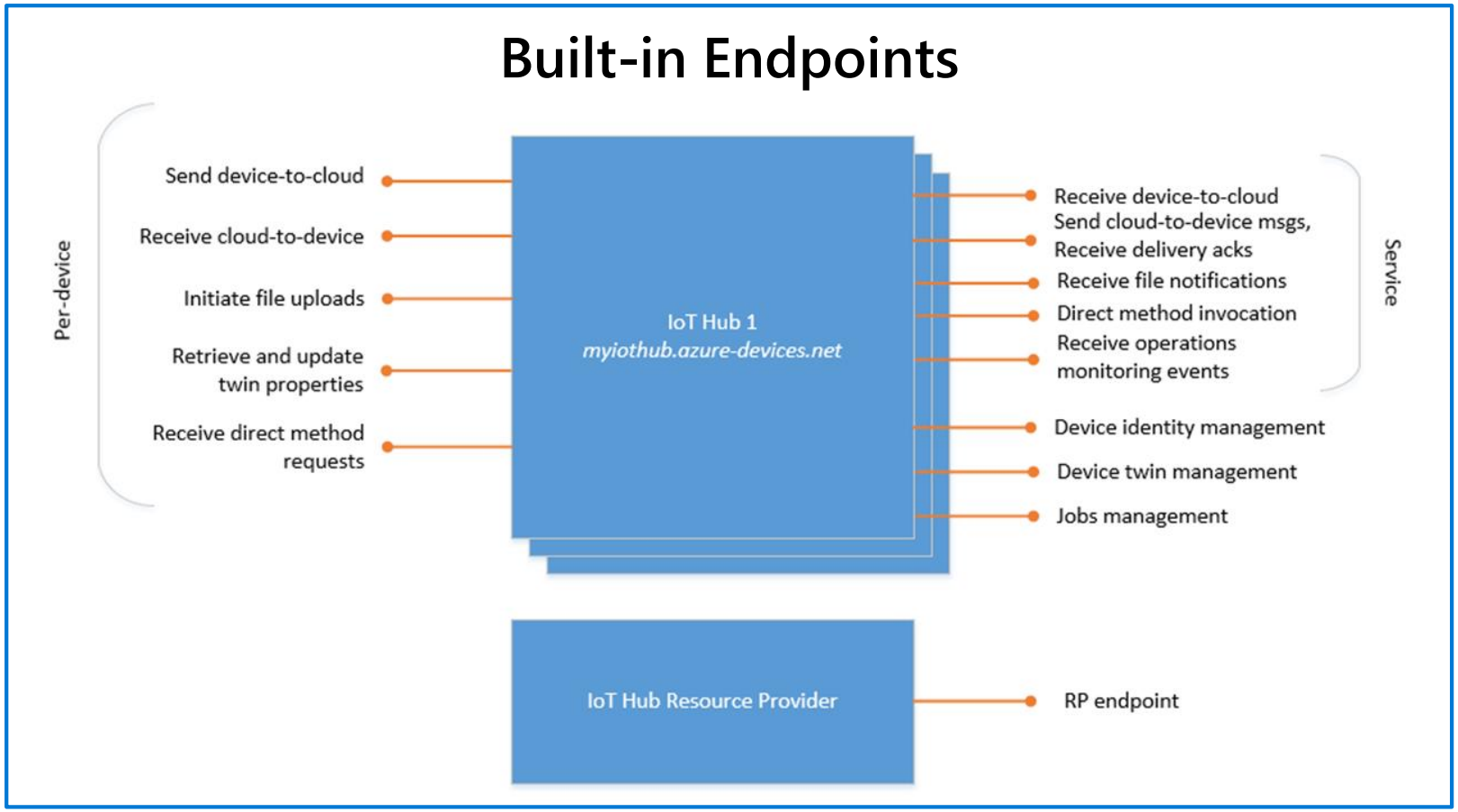
Custom Endpoints:

Azure Storage containers

Event Hubs

Service Bus Queues

Service Bus Topics

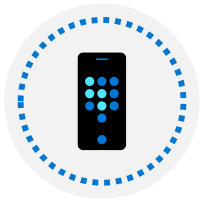


Introduction to IoT Hub security features

Access control and permissions:

Hub-level shared-access policies (default and custom)

Device-level permissions

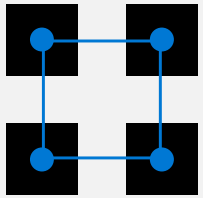


Authentication and security credentials:

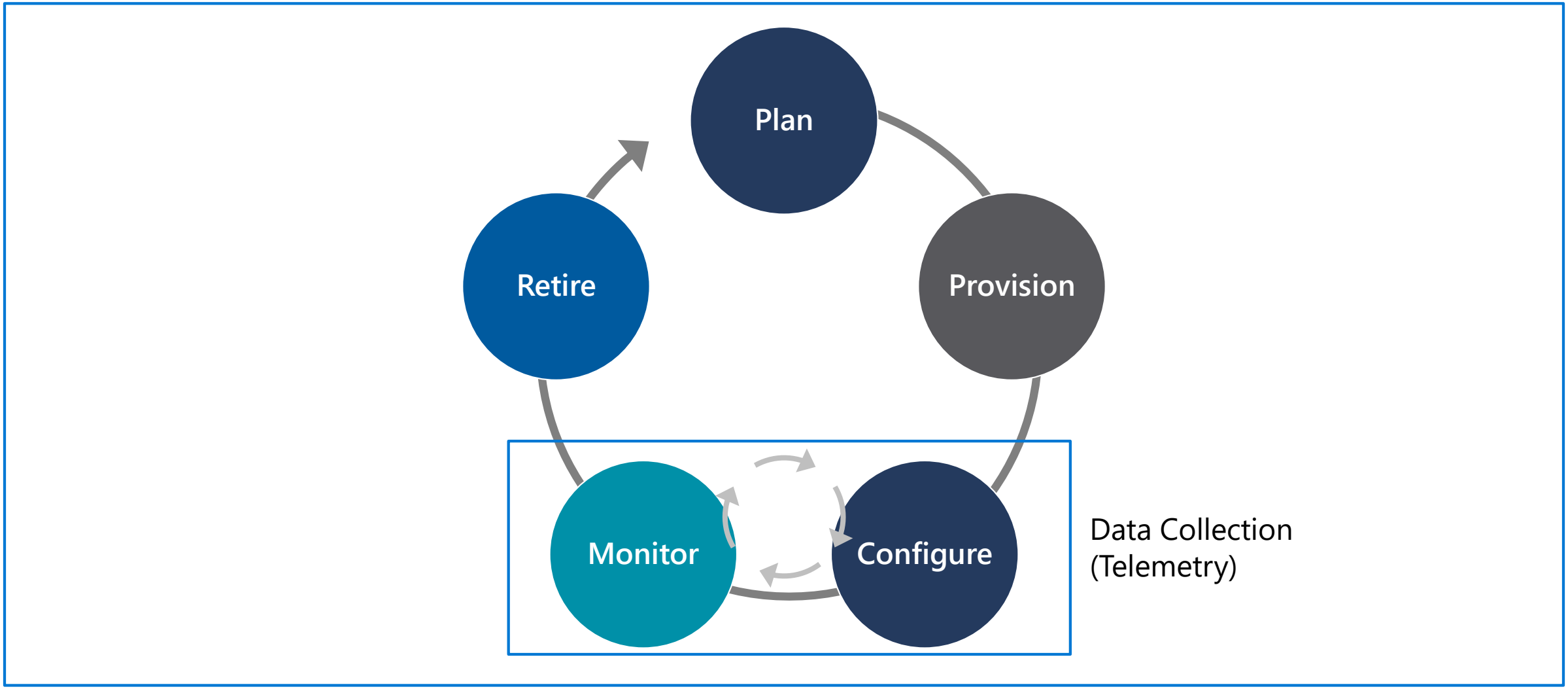
Security tokens (devices and services)

X.509 certificates (devices)

Lesson 3: IoT Device Lifecycle concepts



Device lifecycle terms and concepts



Azure IoT device types

Plan

IoT Devices: Typically a small-scale, standalone computing device that may collect data or control other devices

IoT Edge Devices: IoT Edge devices have the IoT Edge runtime installed; can be used as a field gateway device

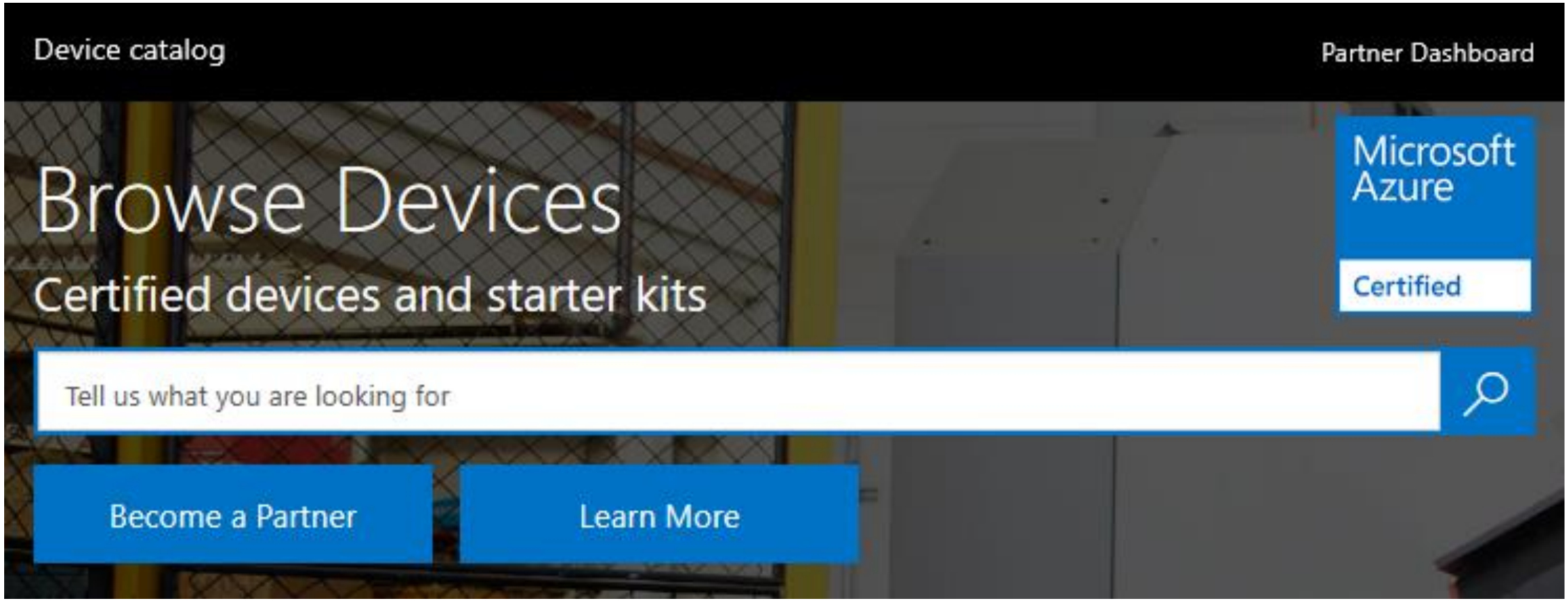
Note that while we often think of “IoT Devices” as hardware, developers can also use “simulated devices,” a software representation of physical devices that run on your local machine or in the cloud

Simulated devices can be used in various stages during the rollout of an IoT solution to represent individual device behaviors or to generate a telemetry workload

Microsoft Certified for IoT Device catalog

Plan

Microsoft maintains an online device catalog that provides a list of hardware devices certified to work with IoT Hub at https://devicecatalog.azure.com/



Azure IoT device registration

Provision

Creating a *device registration* in the IoT Hub *identity registry* requires, at a minimum:

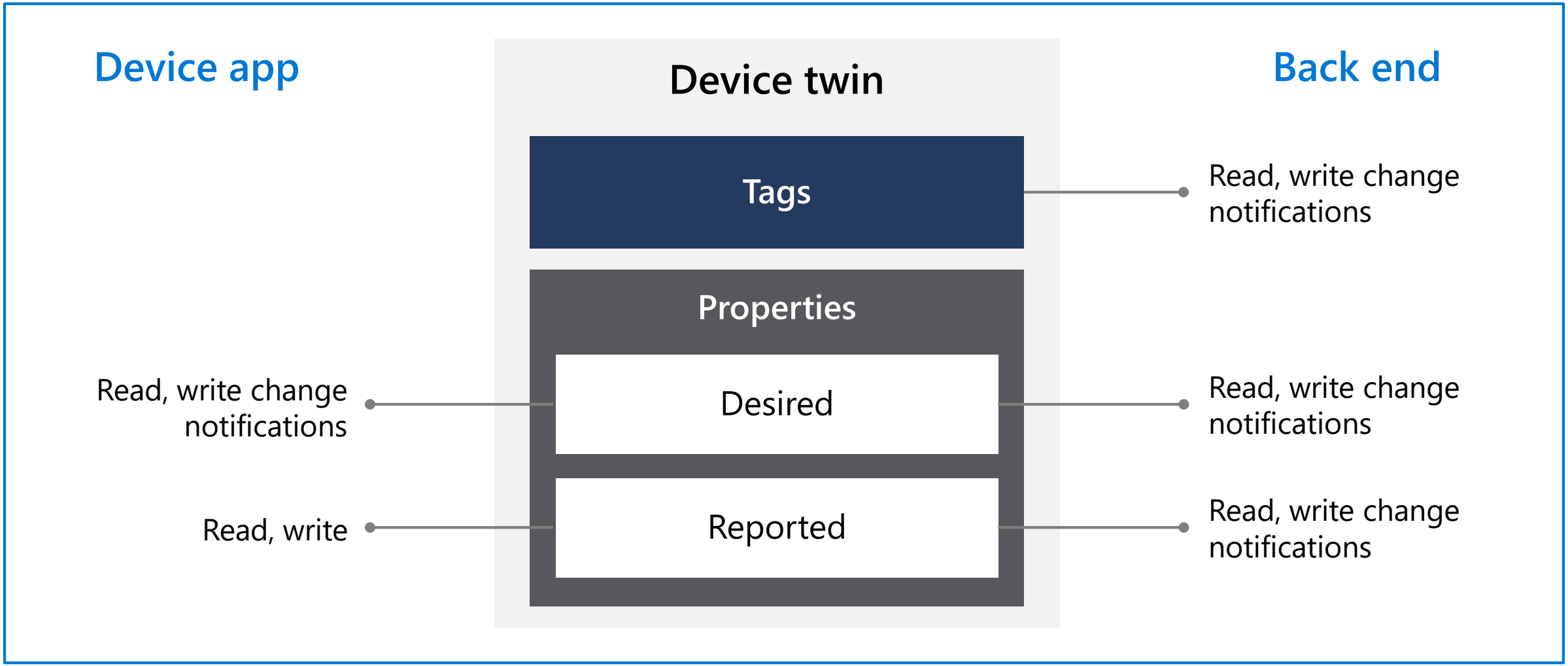
A *device id* – the unique identifier you assign to a device

The *authentication type* – symmetric key or X.509 certificate

Both are used to authenticate the device to the IoT Hub, and both of these will be discussed in more detail later in the course

Introduction to Device Twins

Configure

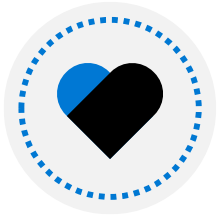


Device monitoring

Monitor

While the device is live, you need to *monitor* it…

Status – what's happening with ongoing operations?

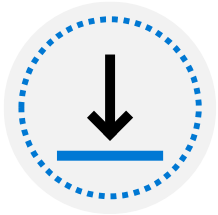


Health – is it working right?

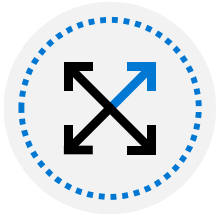
Device retirement

Retire

At the end of a device's lifecycle, it should be *retired…*

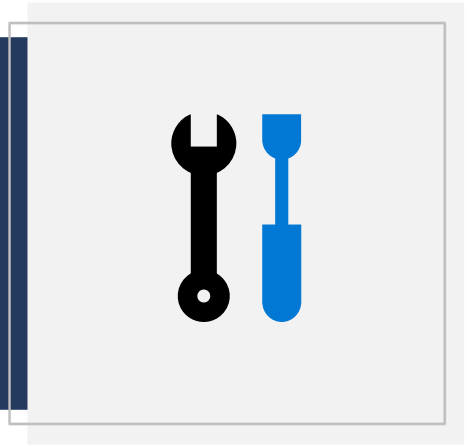


Disable – temporarily removes the ability of the device to communicate with the IoT Hub

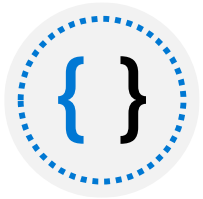


Delete – removes the device registration completely

Lesson 4: IoT Hub Developer tools

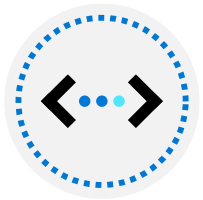


IoT Developer tools overview

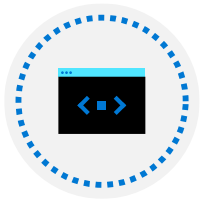


Software Development Kits (SDKs)

Visual Studio

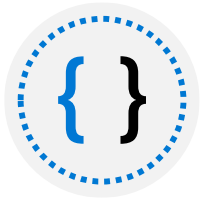


Visual Studio Code



Command-Line Interfaces (CLIs)

Azure IoT Hub SDKs



Benefits of using the SDKs:

Develop a “future-proof” solution with minimal code

Leverage features designed for a complete software solution and focus on your specific need

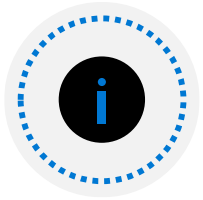
Develop with your preferred language for different platforms

Benefit from the flexibility of open source with support from Microsoft and community

Included SDKs:

IoT Hub Device SDKs

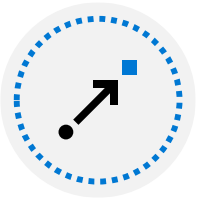
IoT Hub Service SDKs



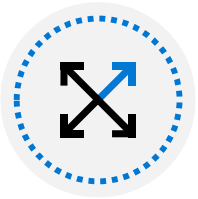
Platform support:

C, .NET (C#), Node.js, Java, and Python

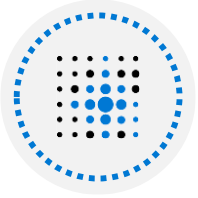
Azure IoT Hub Device SDKs: Languages



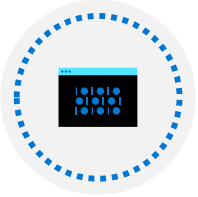
C (easily ported!)



C#



Java

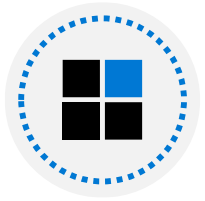


Node.js

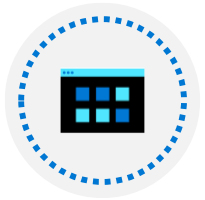
Python

Azure IoT Hub Device SDKs: Example platforms

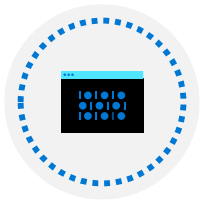
Linux (Ubuntu, Debian, Raspbian)



Windows



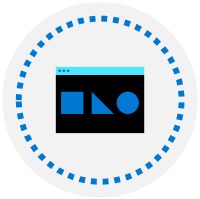
MBED



Arduino:

Huzzah, ThingDev, FeatherM0

FreeRTOS (ESP32, ESP8266)

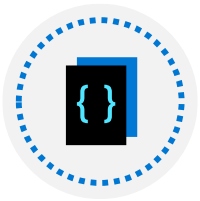


.NET Variations:

NET Framework 4.5

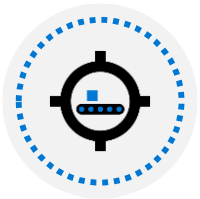
PCL (Profile 7 UWP, Xamarin.iOS, Xamarin.Android)

.NET Standard 1.3



Azure Sphere

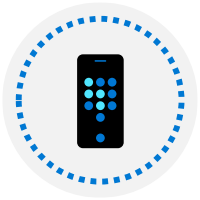
Azure IoT Hub Device SDKs: Protocols



MQTT

MQTT over WebSockets

AMQP



AMQP over WebSockets

HTTPS

Azure IoT Hub Service SDKs

Coding language support:

C

C#

Java

Node.js

Python

Backend Scenarios:

Identity registry

Cloud-to-device messaging

Direct method operation

Querying

Jobs

File uploads

Visual Studio code extensions

Azure IoT Tools collection:

Azure IoT Hub Toolkit

Azure IoT Edge

Azure IoT Device Workbench

Azure CLI tools

Added by Azure CLI extensions for IoT

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

Hub Commands

```
create, delete, show-connection-string, etc.
```

Subgroup commands

```
device-identity, device-twin, etc.
```

Running CLI commands

Example:

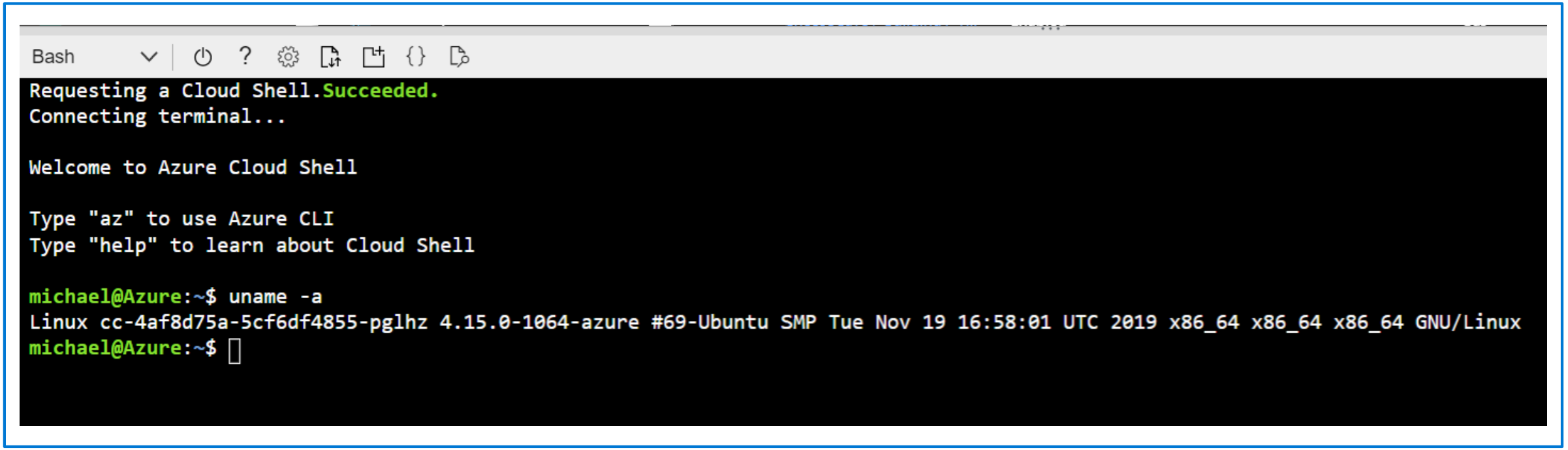
```
az iot hub create --resource-group MyResourceGroup --name MyIotHub
```

Getting help:

```
az iot hub <command name> --help
az iot hub create –help
```

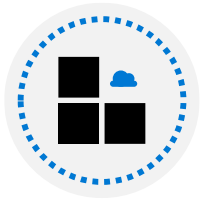
Azure Cloud Shell

A browser-based shell experience running Bash or PowerShell, with many common Microsoft and third-party tools installed

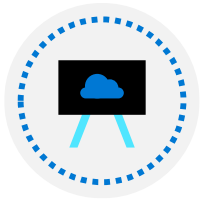


Lesson 5: Device configuration and communication

Device communication



Device-to-Cloud



Cloud-to-Device

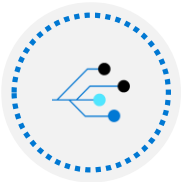
Communication protocols: Protocol comparison

Protocol	Port	When you should use this protocol
MQTT	8883	Use on all devices that do not require to connect multiple devices (each with its own per-device credentials) over the same TLS connection
MQTT over WebSockets	443	
AMQP	5671	Use on field and cloud gateways to take advantage of connection multiplexing across devices
AMQP over WebSockets	443	
HTTPS	443	Use for devices that cannot support other protocols

Communications protocols: Considerations

Cloud-to-device pattern – HTTPS 1.0/1.1 require server polling rather than server push, so cloud-to-device messages are not efficient

Field gateways – MQTT and HTTPS do not support sharing a single TLS connection



Low resource devices – MQTT and HTTPS are lighter-weight libraries compared to the AMQP libraries

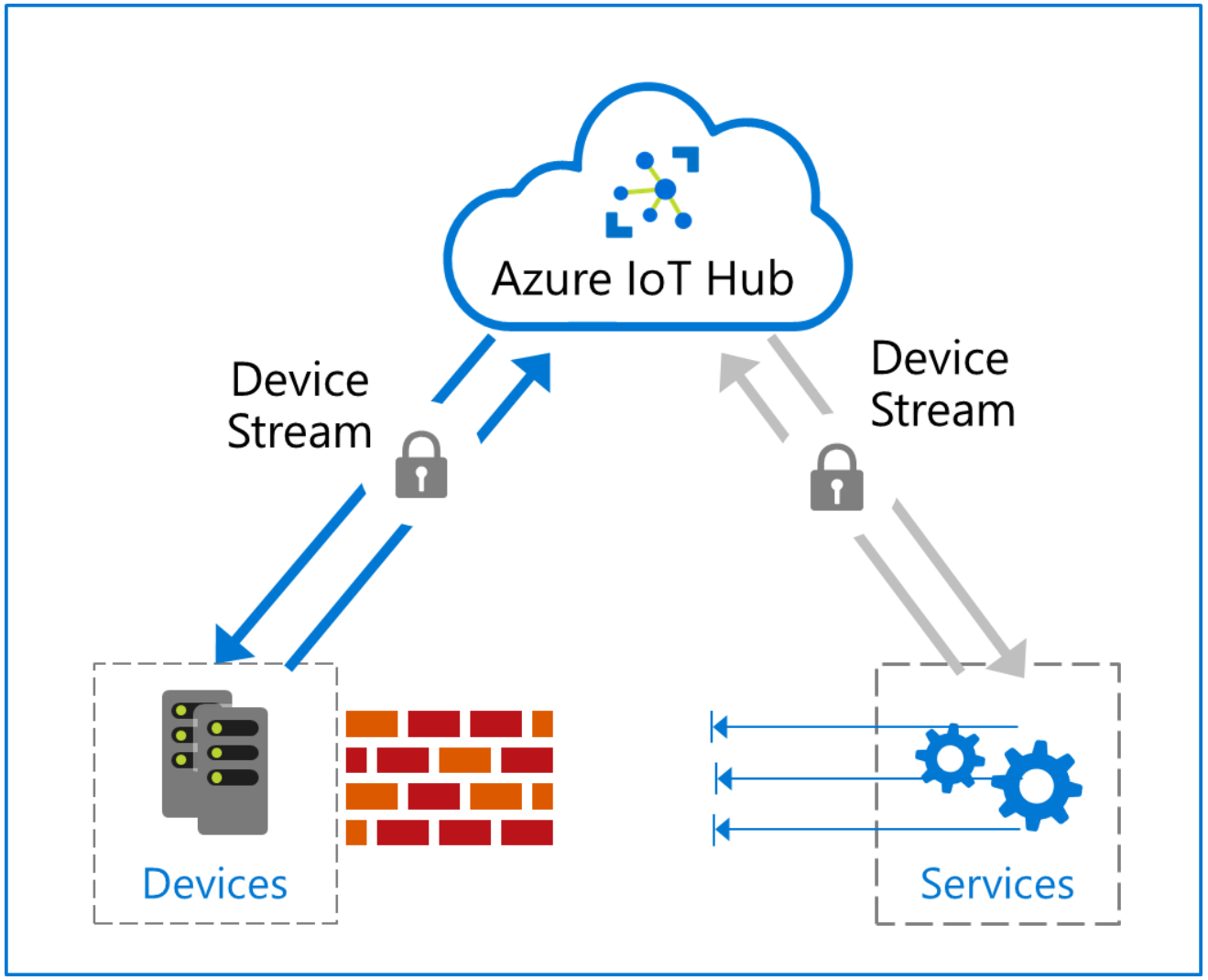
Network traversal – as previously shown, there are specific ports that need to be open that might be an issue

Payload size – MQTT and AMQP are fundamentally binary protocols and thus the payloads are more compact than HTTPS payloads

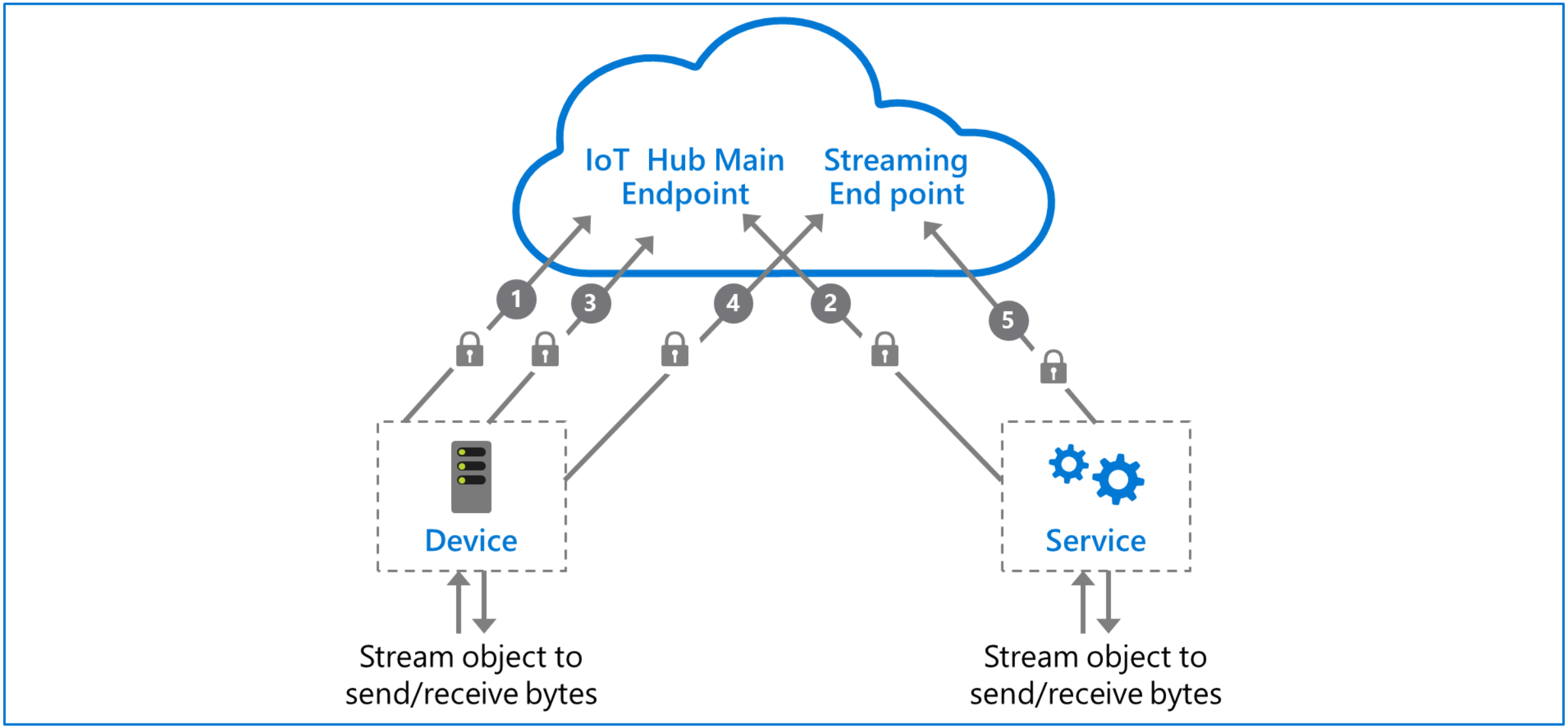
IoT Hub Device Streams (preview) overview

Benefits

Requirements



IoT Hub Device Streams (preview) workflows



Device-side code implementation: Sample projects

Microsoft sample code projects:

Device Streaming Sample

File Upload Sample

Import Export Devices Sample

Keys Rollover Sample

Message Sample

Method Sample

Twin Sample

Xamarin Sample

Message Sample C#:

Message Sample.cs

MessageSample.csproj

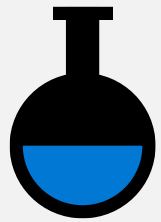
Program .cs

```
private async Task SendEvent()
{
    string dataBuffer;

    Console.WriteLine("Device sending {0} messages to IoTHub...\n", MessageCount);

    for (int count = 0; count < MessageCount; count++)
    {
        _temperature = s_randomGenerator.Next(20, 35);
        _humidity = s_randomGenerator.Next(60, 80);
        dataBuffer = $"{{\"messageId\":{count},\"temperature\":{_temperature},\"humidity\":{_humidity}}}";
        Message eventMessage = new Message(Encoding.UTF8.GetBytes(dataBuffer));
        eventMessage.Properties.Add("temperatureAlert", (_temperature > TemperatureThreshold) ? "true" : "false");
        Console.WriteLine("\t{0}> Sending message: {1}, Data: [{2}]", DateTime.Now.ToLocalTime(), count, dataBuffer);

        await _deviceClient.SendEventAsync(eventMessage).ConfigureAwait(false);
    }
}
```

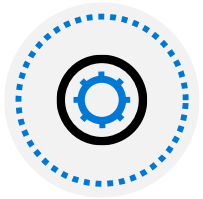
Device-side code implementation: Code example

```
private async Task SendEvent()
{
    string dataBuffer;

    Console.WriteLine("Device sending {0} messages to IoTHub...\n", MessageCount);

    for (int count = 0; count < MessageCount; count++)
    {
        _temperature = s_randomGenerator.Next(20, 35);
        _humidity = s_randomGenerator.Next(60, 80);
        dataBuffer = $"{{\"messageId\":{count},\"temperature\":{_temperature},\"humidity\":{_humidity}}}";
        Message eventMessage = new Message(Encoding.UTF8.GetBytes(dataBuffer));
        eventMessage.Properties.Add("temperatureAlert", (_temperature > TemperatureThreshold) ? "true" : "false");
        Console.WriteLine("\t{0}> Sending message: {1}, Data: [{2}]", DateTime.Now.ToLocalTime(), count, dataBuffer);

        await _deviceClient.SendEventAsync(eventMessage).ConfigureAwait(false);
    }
}
```

Lesson 6: Module 2 labs



Module 2 labs



Lab 3: Setup the development environment:

You will install the VS Code extensions for IoT

You will Install the Azure CLI extensions for IoT

You will verify that the dev environment is working properly

Lab 4: Connect IoT Device to Azure:

You will create a new Device ID within Azure IoT Hub using the Azure CLI

You will configure a simulated device to connect to Azure IoT Hub

You will verify that device telemetry is being received by Azure IoT Hub

Lesson 7: Module 2 review questions

Module review: Question 2.1

Which of the following lists the custom endpoints that IoT Hub supports?

Answer A:
Azure Storage containers, Event Hubs, Service Bus Queues, Service Bus Topics

Answer B:
Azure Blob storage, Azure File storage, Azure Table storage, Azure Cosmos BD

Answer C:
Azure Storage containers, Event Hubs, Azure SQL Database, Azure Data Lake Gen 2

Module review: Question 2.2

Which of the following correctly describes differences between the Basic and Standard tiers of IoT Hub?

Answer A:

The Basic and Standard tiers support the same capabilities, but Standard provides higher throughput at an increased cost.

Answer B:

The Basic tier supports the full list of IoT Hub capabilities, while the Standard tier supports a subset. Equivalent levels of message throughput are available.

Answer C:

The Basic tier does not support IoT Edge devices or cloud-to-device messaging, while the Standard tier supports the full list of IoT Hub capabilities. Both tiers offer the same levels of message throughput.

Module review: Question 2.3

What are the sections of a device twin file?

Answer A:
A device twin includes sections for Tags, Desired properties, Reported properties, and Device identity properties.

Answer B:
A device twin includes sections for Condition, Configuration, Identity, and State.

Answer C:
A device twin includes sections for IoT Hub properties, Device properties, Connection properties, and State properties.

Module review: Question 2.4

What is the purpose of the IoT Hub identity registry?

Answer A:
The IoT Hub identity registry stores information about cloud apps and services that can connect to IoT Hub.

Answer B:
The IoT Hub identity registry stores the device ID and authentication data for devices that can connect to IoT Hub.

Answer C:
The IoT Hub identity registry stores device twin properties for devices that can connect to IoT Hub.

Module review: Question 2.5

A person opens the Azure Cloud Shell, ensures that Bash is selected as the environment, and then runs the following two commands:

1. "az group create --name MyAZ220RG --location westus"
2. "az iot hub create --resource-group MyAZ220RG --name MyIotHub".

Which of the following statements is correct?

Answer A:

The first command will fail to create a resource group because no subscription is provided.

Answer B:

The first command will create a resource group named MyAZ220RG in the westus region.

Answer C:

The second command will fail because no pricing tier is specified.

Module review: Question 2.6

Which of the following lists the categories of developer tools that are commonly used to create IoT solution resources?

Answer A:
Azure IoT Hub, Azure Time Series Insights, Azure Stream Analytics, Azure Data Lake Gen 2.

Answer B:
SDKs, Visual Studio, Visual Studio Code, CLIs.

Answer C:
macOS, Linux, Windows.

Module review: Question 2.7

Which of the following statements about Device Twins is correct?

Answer A:
Each device that is registered with IoT Hub has a Device Twin.

Answer B:
Device twins are XAML documents maintained by IoT Hub.

Answer C:
When you disable a device, the device twin is deleted.

Module review: Question 2.8

A company is investigating how to implement 2-way communication between devices and IoT Hub.

Which of the following answers is a common example of sending information from devices to the cloud?

Answer A:
Device-to-cloud messages for scheduling firmware updates.

Answer B:
Device twin's reported properties for reporting device state information.

Answer C:
Device twin's proposed properties for requesting device state information.

Module review: Question 2.9

A company will be implementing an IoT solution that uses both IoT devices and IoT Edge devices, and they expect to process a little over 5 million messages per day. The company wants to minimize their cost.

What IoT Hub tier and edition should they select and how many units will be required?

Answer A:
B1 with 8 units.

Answer B:
B2 with 1 unit.

Answer C:
S2 with 1 unit.