**MICROSOFT AZURE SERVERLESS TECHNOLOGY**

Hello, and welcome to Azure Serverless Technology and IoT.

1. In the first lesson you learned about two serverless computing solutions in Azure.
   1. Azure functions, and
   2. Azure logic apps.

You'll examine what they are, how they differ, and when you should choose one over the other. You'll be introduced to these tools and services and would be asked to help choose the best one for a given business scenario.

1. In the second lesson, you'll help Tailwind Traders select the right Azure IoT service offering for its business scenarios. By evaluating the services in relation to a set of decision criteria, you learn about what the various services do. How they're different or complimentary, and when to use one or the other.

To get the most out of this module, **you should have an understanding of the concept of :**

1. **orchestration and workflows**
2. **Understanding of the concept of APIs**
3. **A high level of familiarity with the relevant Microsoft products such as Dynamics 365 and Office 365**.

In this module, you will help Tailwind Traders to choose which Azure serverless technology and Azure IoT solutions are right for different business scenarios.

Tailwind Traders, a traditional brick and mortar retailer, has found success selling online. The company sees several opportunities to improve its e-commerce website. For example, it wants to provide more accurate real-time inventory information online to customers who want to visit their local store to purchase an item. The company also wants to respond more proactively to customers who've had a negative experience by providing a new customer retention program. Tailwind Traders suspects that serverless computing can help it provide these services, but it needs help to understand which Azure solutions are right for its business scenarios.

**Serverless computing is a term used to describe an execution environment that's set up and managed for you.** You merely specify what you want to happen by writing code or connecting and configuring components in a visual editor. Then specify the actions that trigger your functionality, such as a timer or an HTTP request. Best of all, you never have to worry about an outage. Your code can scale instantly to meet demand, and you pay based only on the actual usage of your code. By the end of this lesson, you'll be able to choose the serverless computing technology that best addresses Tailwind Traders’ business scenarios.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Identify the Product Options**

There are two serverless computing solutions on Azure :

1. Azure Functions : Accelerate and simply application development with serverless compute
2. Azure Logic Apps : Quickly build powerful integration solutions, was designed with orchestration in mind.

Before we start, let's take another look at the key principles of serverless computing. **Serverless computing is a cloud-hosted execution environment that runs your code but abstracts the underlying hosting environment**. The term serverless computing is a misnomer (istilah yang tidak cocok). *After all, there is a server or a group of servers that executes your code or desired functionality.*The key idea is that you're not responsible for setting up or maintaining the server. You don't have to worry about scaling it when there's increased demand and you don't have to worry about outages. The cloud vendor takes care of all maintenance and scaling concerns for you. You create an instance of the servers, and you add your code. No infrastructure configuration or maintenance is required or even allowed. You configure your serverless apps to respond to events. An event could be a rest endpoint, a periodic timer, or even a message received from another Azure service.

The serverless app runs only when it's triggered by an event. Scaling and performance are handled automatically, and you will build only for the resources you use. You don't even need to reserve resources. Serverless computing is ordinarily used to handle backend scenarios. In other words, serverless computing is responsible for sending messages from one system to another or processing messages that were sent from other systems. It's not used for user facing systems, but rather, it works in the background.

**The core Azure services for serverless are Azure Functions and Azure Logic Apps.** Both solutions help developers to build robust cloud apps with minimal code.

Let's take a closer look at how each of these serverless automation platforms work within Azure.

1. With the **Azure Function** service, you can host a single method or function by using a popular programming language in the cloud that runs in response to an event.

An example of an event might be an HTTP request, a new message in a queue, or a message on a timer. Because of its atomic nature, Azure Functions can serve many purposes in an applications design, functions can be written in many common programming languages, such as C#, Python, JavaScript, TypeScript, Java, and PowerShell. Azure Functions scales automatically and changes accrue (increase,grow) only when a function is triggered. These qualities make **Azure Functions a solid choice when demand is variable.** For example, you might be receiving messages from an IoT solution that monitors a fleet of delivery vehicles. You will likely have more data arriving during business hours.

* Azure Functions can scale out to accommodate these busier times.
* An **Azure Function is a stateless environment**. A function behaves as if it's restarted every time it responds to an event. This feature is ideal for processing incoming data. If a state is required, the function can be connected to an Azure storage account.
* Azure Functions can perform orchestration tasks by using an *extension called durable functions*, which allows developers to chain functions together while maintaining state. The Azure Function solution is ideal when you are concerned only with the code that's running your service and not the underlying platform or infrastructure.
* You use functions most commonly when you need to perform work in response to an event. You do this often via a REST request, timer, or message from another Azure service and when that work can be completed quickly within seconds or less.

Azure Functions works well for this scenario. Let's say you need to execute custom algorithms or perform specialized data parsing and data lookups.

Now let's explore the key components of Logic Apps.

1. Azure Logic Apps

Logic Apps is a low code-no code development platform hosted as a cloud service. This service helps you automate and orchestrate tasks, business processes, and workflows when you need to integrate apps, data, systems, and services across enterprises or organizations.

Logic Apps simplify how you design and build scalable solutions, whether in the cloud, on-premises, or both. This solution covers :

* app integration,
* data integration,
* system integration,
* enterprise application integration, and
* business-to-business (B2B) integration.

Azure Logic Apps is designed in a web-based designer and can execute a logic that's triggered by Azure services without your having to write any code. You build an app by linking triggers to actions with connectors. A trigger is an event such as a timer that causes an app to execute a new message to be sent to a queue or an HTTP request.

A blue globe with black text

Description automatically generated

And an action is a task or step that can be execute. There are logical actions, such as those you would find in most programming languages. Examples of actions include working with variables, decision statements, and loops, and tasks that parse and modify data.

To build enterprise integration solutions with Azure logic apps, you can choose from a growing gallery of over 200 connectors. The gallery includes services such as Salesforce, SAP, Oracle DB, and file shares. If you can't find the action or connector you need, you can build your own by using custom code.

Both Azure Functions and Azure Logic Apps help enable business-logic that automates your Azure workflow. The primary difference **between the two services is their intent**. **Azure Functions is a serverless computer service and Azure Logic Apps is intended to be a serverless orchestration service.**

Although you can use Azure Functions to orchestrate a long-running business process that involves various connections, this was not its primary use case when it was designed. Additionally, the two services are priced differently. ***Azure Functions pricing is based on the number of executions and the running time of each execution. Logic Apps pricing is based on the number of executions and the type of connectors that it utilizes.***

Azure Functions is a serverless compute service that lets you run event-triggered code without having to explicitly provision or manage infrastructure. Azure Functions scales automatically, and charges accrue only when a function is triggered. These qualities make Azure Functions a solid choice when demand is variable. Azure Functions can scale out to accommodate these busier times.

----------------------------------------------------------------------------------------------------------------------------------------------------------  
**Analyze the decision Criteria**

With two viable serverless options, it can be difficult to know which is the best one for the job. In this session, you'll analyze the criteria that experts employ when they're choosing a serverless service to use for a given business need. Understanding the criteria can also help you better understand the nuanced differences between the products.

1. When choosing the most appropriate serverless solution, you should ask yourself if you need to perform an orchestration across well-known APIs. As we noted previously, Azure Logic Apps was designed with orchestration in mind, from the web-based visual configurator to the pricing model. Logic Apps excels at connecting a large array of disparate services via their APIs to pass and process data through many steps in a workflow.

It's possible to create the same workflow by using Azure Functions, but it might take a considerable amount of time to research which APIs to call and how to call them. Azure Logic Apps has already componentized these API calls so that you supply only a few details, and the details of calling the necessary APIs is abstracted away.

1. Although Azure Logic Apps can perform logic, loops, decisions, and so on, if you have a logic intensive orchestration that requires a complex algorithm, implementing that algorithm might be more long-winded and visually overwhelming. With Azure Functions, you can use the full expressiveness of a programming language in a compact form. This led to concisely build complex algorithms or data lookup and parsing operations. You would be responsible for maintaining the code, handling exceptions resiliently, and so on.
2. Next, you need to think about your existing code. If you already have your orchestration or business-logic expressed in C#, Java, Python, or another popular programming language, it might be easier to port your code into the body of an Azure Functions function app.than to recreate it by using Azure Logic Apps. Ultimately, your choice comes down to whether you prefer to work in a declarative environment or an imperative environment. Developers who have expertise in an imperative programming language might prefer to think about automation and orchestration from an imperative mindset. IT professionals and business analysts might prefer to work in a more visual, low code, no code, declarative environment.
3. Azure Logic Apps is best suited for users who are more comfortable in a visual environment that allows them to automate their business processes. Azure Functions is best suited for software developers and those with expertise in an imperative programming language.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Use AZURE FUNCTIONS**

Data about each product that's sold at Tailwind Traders is packaged as a JSON message and sent to an event hub. The event hub distributes the JSON message to subscribers, which allows various systems to be notified. Tailwind Traders wants to upgrade its e-commerce site to include real-time inventory tracking. Currently, the website updates product availability nightly, at 2 AM.

A Windows service that's written in C# contains all of the necessary logic to :

1. retrieve the messages,
2. Parse the JSON,
3. Perform a look-up across multiple databases to find additional product information, and
4. Potentially send notifications to the purchasing department so that they can reorder quantities that fall below certain levels.

The Windows service runs in a virtual machine that's hosted on Azure. Most of the time, this system works fine, however, some products are in high demand and some products are kept in low quantities at each store. Several times a day, customers drive to a store to pick up an item only to find that it's no longer in stock. Instead of running the algorithm nightly, the company wants to run the inventory updater each time a product is purchased.

Let's apply the decision criteria you learned about earlier to find the right option for Tailwind Traders. Because the Tailwind Traders developers’ team has already written the logic in C#, it would make sense to copy the relevant C# code from the Windows service and port it to an Azure function. The developers would bind the function to trigger each time a new message appears on a specific queue. It's possible to implement the same logic in Azure Logic Apps. However, ***because the team has already invested time in building the service in C#, it can use the same code in an Azure function.***

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Use AZURE LOGIC Apps**

Tailwind Traders sends its customers an invitation to participate in a customer satisfaction survey randomly after a purchase. Currently, the customer satisfaction results are aggregated, averaged and charted. However, it's customer service department sees an opportunity to reach out proactively to customers who provide low scores and leave comments with a negative sentiment.

Ideally, negative customer satisfaction scores **would trigger a customer retention workflow**. First, a sentiment analysis would be generated based on the free form comments. An email would be sent to the customer with an apology and a coupon code. And the message would be routed to the Dynamics 365 customer service team, so that it could schedule a follow up email.

Unfortunately, no Tailwind Traders developer resources are available to take on this project. But the customer service team works with several cloud and IT professionals who might be able to construct a solution. Once again, applied the decision criteria you learned about earlier to find the right option for Tailwind Traders.

This is an ideal scenario for Azure logic apps. Connectors already exist for each of the steps outlined in the workflow, because **Azure logic apps is a no code service**, **no developers are needed.** A cloud or IT professional should be able to build and support this workflow. A cloud or IT professional could use existing connectors to perform a sentiment analysis, by *using the Azure Cognitive Services Connector*, Send an email by using the Office 365 Outlook Connector and create a new record and follow up email by using the Dynamics 365 customer service connector.

Using Azure functions as an alternative would take quite a bit of research development and testing for a developer to build a solution that utilizes all these disparate software systems.

----------------------------------------------------------------------------------------------------------------------------------------------------------

1. There is a server (or a group of servers) that execute your code or desired functionality. The key idea is that you are not responsible for setting up or maintaining that server. You don't have to worry about scaling the server or dealing with outages. The cloud vendor takes care of all the maintenance and scaling concerns for you**.**
2. Serverless computing is used to handle backend scenarios. It is responsible for sending messages from one system to another, or processing messages that were sent from other systems. It's not used for end-user facing systems.
3. Azure Functions allows you to host a single method or function using a popular programming language in the cloud that runs in response to an event such as an HTTP request, a new message on a queue, or on a timer.
4. Logic Apps is low-code / no-code development platform hosted as a cloud service that helps you automate and orchestrate tasks, business processes, and workflows. An Azure Logic Apps is designed in a web-based designer and can execute logic triggered by Azure services without writing any code. Azure Logic Apps is a development platform hosted as a cloud service that helps you automate and orchestrate tasks, business processes, and workflows when you need to integrate apps, data, systems, and services across enterprises or organizations.

A screenshot of a computer error

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

----------------------------------------------------------------------------------------------------------------------------------------------------------

**MICROSOFT AZURE SERVERLESS TECHNOLOGY**

Welcome back. In this lesson, you'll help Tailwind Traders select the right Azure IoT service offering for its business scenarios. By evaluating the services in relation to a set of decision criteria, you learn what the various services do, how they're different or complimentary, and when to use one or the other.

Tailwind Traders sees many opportunities to use Azure IoT services across many different facets of their operations, from new product development to logistics and point of sale. IoT bridges the physical and digital worlds by enabling devices with sensors and an internet connection to communicate with cloud-based systems via the Internet. By the end of this lesson, you'll be able to choose the Azure IoT service that best addresses Tailwind Traders business scenarios.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Identify the product options**

IoT enables devices to gather and then relay information for data analysis. Smart devices are equipped with sensors that collect data.

A few common sensors that measure attributes of the physical world include :

* Environmental sensors that capture temperature and humidity levels,
* Barcode QR or Optical Character Recognition scanners (OCR Scanners),
* Geo-location and proximity sensors,
* Light, color, and infrared sensors,
* Sound and ultrasonic sensors,
* Motion and touch sensors,
* Accelerometer and tilt sensors,
* Smoke, gas and alcohol sensors,
* Error sensors to detect when there's a problem with the device,
* Mechanical sensors that detect anomalies or deformations, and
* Flow level and pressure sensors for measuring gases and liquids.

By using Azure IoT services, devices that are equipped with these kinds of sensors and that can connect to the Internet, could send their sensor readings to a specific endpoint in Azure ***via a message***. The messages' data is then collected and aggregated and it can be converted into reports and alerts. Alternately, all devices could be updated with new firmware to fix issues or add new functionality by sending software updates from Azure IoT services to each device.

Let suppose your company manufactures and operates smart refrigerated vending machines. What kinds of information would you want to monitor?

* You might want to ensure that each machine is operating without any errors.
* The machines haven't been compromised.
* The machines' refrigeration systems keep their contents within a certain temperature range and
* that you're notified when products reach a certain inventory level so you can restock the machines.

If the hardware of your vending machines can collect and synthesize information in a standard message. The messages each machine sends can be received, stored, organized, and displayed by using Azure IoT services. The data that's collected from these devices could be combined with Azure AI services to help you predict when machines need proactive maintenance, when inventories will need to be replenished, and new products ordered from vendors. Many services can assist in drive end-to-end solutions for IoT on Azure.

In this lesson, you'll explore Azure IoT Hub, Azure IoT Central, and Azure Sphere.

1. Azure IoT Hub : Build your IoT applications with two way communication

An IoT hub communicates to IoT devices by sending and receiving messages. Azure IoT Hub is a managed service that's hosted in the cloud and that acts as a central message hub for bi-directional communication between your IoT application and the devices it manages.

* is a managed service(MS) that's hosted in the cloud and that acts as a central message hub for bidirectional communication between your IoT application and the devices it manages.
* You can use Azure IoT Hub to build IoT solutions with reliable and secure communications between millions of IoT devices and a cloud-hosted solution back-end.
* You can connect virtually any device to your IoT Hub.

The IoT Hub service supports communications both from the device to the cloud and from the cloud to the device. It also supports multiple messaging patterns, such as device-to-cloud telemetry, file upload from devices, and request to reply to methods to control your devices from the cloud. After an IoT Hub receives messages from a device, it can route that message to other Azure services.

From a cloud-to-device perspective, IoT Hub allows for :

* **Command and Control**. That is, you can have either manual or automated remote control of connected devices so you can instruct the device to open valves, set target temperatures, restart stuck devices, and so on.
* **Tracking,** IoT Hub monitoring helps you maintain the health of your solution by tracking events such as device creation, device failures, and device connections.

1. Azure IoT Central : Connect Manage and monitor your IoT devices

* **Azure IoT Central** builds on top of IoT Hub by adding a dashboard that allows you to connect, monitor, and manage your IoT devices.
* The visual user interface makes it easy to quickly connect new devices and watch as they begin sending telemetry or messages.
* You can watch the overall performance across all devices in aggregate.
* You can set up alerts that send notifications when a specific device needs maintenance.
* Finally, you can push hardware updates to the device.

To help you get up and running quickly, IoT Central provides starter templates for common scenarios across various industries, such as retail, energy, healthcare, and government. You then customize the design starter templates directly in the UI by choosing from existing themes or creating your own custom theme, setting the logo, and so on. With IoT Central, you can tailor the starter templates for the specific data that's sent from your devices, the reports you want to see, and the alerts you want to send.

You can use the UI to control your devices remotely. This feature allows you to push a software update or modify a property of the device. You can adjust the desired temperature for one or all of your refrigerated vending machines from directly inside of IoT Central.

A key part of IoT Central is the use of device templates. By using a device template, you can connect a device without any service site coding. IoT central uses the templates to construct the dashboards, alerts, and so on. Device developers still need to create code to run on the devices, and that code must match the device template specification.

1. Azure Sphere : Actively protect your devices, your business and your customers

* Azure Sphere creates an end-to-end, highly secure IoT solution for customers that encompasses everything from the hardware and operating system on the device to the secure method of sending messages from the device to the message hub.
* Azure Sphere has built-in communication and security features for internet-connected devices.

Azure Sphere comes in three parts.

* 1. The first part is the Azure Sphere Micro-controller Units, or MCU, which is responsible for processing the operating system and signals from attached sensors. This image displays the Seed Azure Sphere MT3620 Development Kit MCU, one of several different starter kits that are available for prototyping and developing Azure Sphere applications.
  2. The second part is a customized Linux Operating System that handles communication with the security service and can run the vendor's software.
  3. The third part is Azure Sphere Security Service, also known as AS3. Its job is to make sure that the device has not been maliciously compromised. AS3 is Microsoft's cloud-based service that communicates with Azure Sphere chips to enable maintenance, update, and control. When the device attempts to connect to Azure, it first must authenticate itself per device, *which it does by using certificate-based authentication*. If it authenticates successfully, AS3 checks to ensure that the device hasn't been tampered with. After it has established a secure channel of communication, AS3 pushes any OS or approved customer-developed software updates to the device.

After the Azure Sphere system has validated the authenticity of the device and authenticated it, the device can interact with other Azure IoT services by sending telemetry and other information. Now that you have looked at how Azure IoT Hub, Azure IoT Central, and Azure Sphere can assist in drive end-to-end solutions for IoT on Azure, next, you will choose the appropriate service to use in different business scenarios.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Analyze the decision criteria**

In this video, you'll analyze the criteria that experts employees when they decide which IoT service to use for a given business need. Understanding the criteria can also help you better understand the nuanced differences between each product. Let's start off by looking at security. Whether you are a manufacturer or customer, you don't want your devices to be maliciously compromised and used for criminal purposes. It's obviously more critical to ensure the integrity of an ATM than, say, washing machine. So, when choosing the correct option, you always look at what is most critical for your needs before making a decision. **When security is a critical consideration in your products design, the best product option is azure sphere,** which provides a comprehensive into in solution for IoT devices.

As we mentioned previously, **Azure Sphere ensures a secure channel of communication between the device and azure by controlling everything from the hardware to the operating system and the authentication process**. This ensures that the integrity of the devices uncompromised. After a secure channel is established, messages can be received from the device securely and messages air software updates can be sent to the device remotely.

Your next decision will be the level of services you require from your IoT solution. If you merely want to connect to your remote devices to receive telemetry and occasionally push updates and you don't need any reporting capabilities, you might prefer to implement Azure IoT hub by itself. Your programmers can still create a customized set of management tools and reports by using the IoT Hub Restful API.

IoT Central integrates with many different azure products, including IoT Hub to create a dashboard with reports and management features. The dashboard is based on starter templates for common industry and usage scenarios. You can use the dashboard that's generated by the starter templates, as is or customize it to suit your needs. You can have multiple dashboards and target them at a variety of users.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Use the IoT HUB**

The Tailwind Traders senior leadership team has decided to partner with a leading appliance manufacturer to create an exclusive high end brand that promises a preemptive maintenance service agreement. This unique feature would differentiate Tailwind Traders' appliances in a crowded competitive market. The feature also makes the brand lucrative, because a yearly subscription would be required. To build a strong brand reputation, the appliances will send telemetry information to a centralized location, where it can be analyzed, and maintenance can be scheduled. The devices will not require remote control. They will merely be sending their telemetry data for analysis, and proactive maintenance. Because Tailwind Traders already has software in place from managing appliance maintenance requests, the company wants to integrate all functionality into this existing system.

Let's apply the decision criteria you learned about earlier to find the right option for Tailwind Traders.

1. Do you think it is critical to ensure that the device are, in this case, each appliance isn't compromised? No, it's preferable, but not critical that the devices aren't compromised. The worst that could happen is a hacker reads the current temperature of the customer's refrigerator, or the number of loads of laundry the washing machine has completed. Even if the customer calls and reports strange behavior with their appliance, a technician could reset or replace the microcontroller. It might not warrant the extra expense, or engineering resources that would be required to employ Azure Sphere.  But why not? Azure Sphere provides a complete solution for scenarios where security is critical. In this scenario, *security is preferred, but not critical*. The appliances can't be updated with new software remotely. The sensors merely report to usage data. As a result, Azure Sphere isn't necessary.
2. In this case, do you think you need a dashboard for reporting and management? No, Tailwind Traders wants to integrate the telemetry data, and all other functionality into an existing maintenance request system. Given the responses to the decision criteria, Azure IoT Hub is the best choice in this scenario. Azure IoT Central provides a dashboard that allows companies to manage IoT devices individually, and in aggregate, view reports, and set up error notifications via a GUI. But, in this scenario, Tailwind Traders wants to integrate the telemetry it collects, and other analysis functionality into an existing software application. Furthermore, the company's appliances will be collecting data via sensors only, and don't need the ability to update settings, or software remotely. Therefore, the company doesn't need Azure IoT Central. Azure Sphere provides a complete solution for scenarios where security is critical. In this scenario, security is preferred, but not critical. The appliances can't be updated with new software remotely. The sensors merely report usage data. As a result, Azure Sphere isn't necessary.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Use the IoT Central**

Tailwind traders own a fleet of delivery vehicles that transport products from warehouses to distribution centers, and from distribution centers to stores and homes. The company is looking for a complete logistic solution that takes data sent from an onboard vehicle computer and turns it into actionable information. Furthermore, shipments can be outfitted with sensors from a third-party vendor to collect and monitor ambient conditions. These sensors can collect information such as temperature, humidity, tilt, shock light, and the location of a shipment.

The goals for this logistic system, our shipment monitoring with real time tracing and tracking, shipment integrity with real time ambient condition monitoring. Security from theft last or damage of shipments, geofencing, route optimization, fleet management, and vehicle analytics, and forecasting for predictable departure and arrival of shipments.

The company would prefer a pre-built solution to collect the sensor and vehicle computer data, and provide a graphical user interface that displays reports about shipments and vehicles.

Let's find the right solution in this case :

1. is it critical to ensure that the device isn't compromised? Ideally each sensor and vehicle computer would be impervious to interference. However, security was not mentioned as a critical concern at this point. The vehicle computers and sensors are built by a third party vendor and, unless Tailwind traders wants to manufacture its own devices, which they don't, the company will be forced to use hardware that's already available.
2. Does Tailwind traders need a dashboard for reporting and management? Yes, a reporting and management dashboard is a requirement. If Tailwind traders use IoT central, the company would actually be using an IoT hub that's pre-configured for its specific needs by the connected logistics starter template. Otherwise, the company would need to do a lot of custom development to build its own cloud-based dashboards and management systems on top of Azure IoT hub. On the other hand, Azure Sphere provides a complete solution for scenarios where security is critical. In this scenario, security is ideal but not a critical priority. Although Azure Sphere provides an end-to-end solution that includes hardware, Tailwind Traders will use hardware from a third-party vendor. So, in this scenario, Azure Sphere is not necessary. In this scenario, you can see that IoT Central is the best choice.

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Use the Azure Sphere**

Tailwind Traders wants to implement a touchless point-of-sale solution for self-checkout. The self-checkout terminals should be above all else, secure. Each terminal must be impervious to malicious code that could create fraudulent transactions, force the company to take systems offline during a heavy shopping period, or send transactional data to a spying organization. The terminals should also report back vital information on the company's health and allow secure updates to its software remotely. After reviewing many possible solutions during a request for proposal process, Tailwind Traders decides that it needs features that vendors have yet to implement. Instead of using an existing solution, the company has decided to work with a leading engineering firm that specializes in IoT solutions. This approach allows the company to build a uniquely secure terminal that gives it a retail platform to build on going forward. Although most of the company's focus is on the terminal itself, Tailwind Traders realizes that it wants a solution that can help it make sense of all the data that will be generated by these terminals across all of its retail stores. And It wants an easy way to push software updates to its terminals.

Let's see whether you can decide on the perfect solution.

1. Do you think it is critical to ensure that the device, or in this case, each point of sale terminal is not compromised? Absolutely. Device security is the primary requirement.
2. Does Tailwind Traders need a dashboard for reporting and management? Yes. The company requires a reporting and management dashboard. Given the responses to the decision criteria, the IoT engineering firm will build a platform on top of both Azure IoT Central and Azure Sphere. Even though no specific starter template is available in Azure IoT central for this scenario, one can easily be adapted to accommodate the kinds of reports the company wants to see and the management operations it wants to perform**. By using IoT Central, Tailwind Traders would actually be using IoT Hub behind the scenes as well.**

----------------------------------------------------------------------------------------------------------------------------------------------------------

1. Azure IoT Hub is a managed service hosted in the cloud that acts as a central message hub for bi-directional communication between your IoT application and the devices it manages. You can use Azure IoT Hub to build IoT solutions with reliable and secure communications between millions of IoT devices and a cloud-hosted solution backend. You can connect virtually any device to your IoT Hub.
2. Azure Sphere creates an end-to-end highly secure IoT solution for customers that encompasses everything from the hardware and operating system on the device to the secure method of sending messages from the device to the message hub**.**
3. Azure IoT Central builds on top of IoT Hub by adding a dashboard that allows you to connect, monitor, and manage your IoT devices. The visual user interface (UI) makes it easy to quickly connect new devices and watch as they begin sending telemetry or error messages.
4. Azure Sphere creates an end-to-end highly secure IoT solution for customers that encompasses everything from the hardware and operating system on the device, to the secure method of sending messages from the device to the message hub.
5. Although Azure IoT Central provides a dashboard that allows companies to manage IoT devices individually and in aggregate via a GUI, in this scenario, the company will integrate the telemetry they collect into an existing software application so there is no need for IoT Central to be deployed. Azure IoT Hub is the best choice in this scenario. It is a managed service hosted in the cloud that acts as a central message hub for bi-directional communication between your IoT application and the devices it manages. You can connect virtually any device to your IoT Hub.
6. If the company uses IoT Central they would also be using IoT Hub which can be preconfigured for their needs by using the Connected Logistics starter template. However, the Azure IoT Hub by itself will not provide a solution. Azure IoT Central is the best choice in this scenario. IoT Central has access to prebuilt templates such as the Connected Logistics starter template which provides an out-of-the-box dashboard that will accomplish many of these requirements. Certain features might need to be reconfigured once the template has been deployed but the truck gateway functionality would be ideal for this scenario.

A screenshot of a computer

Description automatically generated A white text on a white background

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------------------------------------------------------------------------------

1. Logic Apps is a development platform that is hosted as a cloud service. It helps you automate and orchestrate tasks, business processes, and workflows when you need to integrate apps, data, systems, and services across enterprises or organizations.
2. It would make sense to copy existing logic written in C#, from the Windows Service and port it to an Azure Function. It's possible that the same logic could be implemented in Azure Logic Apps. However, since the service already exists in in C#, it can be leveraged for use in an Azure Function.
3. Azure Logic Apps is a development platform hosted as a cloud service that helps you automate and orchestrate tasks, business processes, and workflows when you need to integrate apps, data, systems, and services across enterprises or organizations.
4. You can use Azure IoT Hub to build IoT solutions with reliable and secure communications between millions of IoT devices and a cloud-hosted solution backend.
5. Azure Sphere creates an end-to-end highly secure IoT solution for customers that encompasses everything from the hardware and operating system on the device to the secure method of sending messages from the device to the message hub.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

A screenshot of a cloud service

Description automatically generated A white and blue text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------------------------------------------------------------------------------

Congratulations you have completed azure serverless technology and IoT. You covered two lessons in this module. Let's look back on your journey. You helped Tailwind traders choose the right serverless computing technology for its business scenarios,

* when the company needed to build a solution that pulls code logic from an existing C# window service you chose Azure functions
* when the company needed to orchestrate a workflow to improve customer retention after a negative shopping experience, you chose Azure logic apps.

In both cases, you noted how choosing the other serverless computing service would be possible. However, you try to help the company consider the decision criteria we outlined and choose the right service for the scenario. Without serverless computing tail, when traders would be forced to set up and manage its own computing infrastructure for these business scenarios, the team would have needed to closely monitor the services to determine whether it needed to scale the service, and it likely would have wasted money in the process with either too many or too few computing resources dedicated to the solution. Additionally, it might have had to design right test and maintain custom code to get similar results by helping tail when traders select the right serverless computing solutions, you are able to deploy new functionality to help the company improve customer satisfaction with its E-commerce platform.

Our goal in this lesson was to help tail when traders explore various IoT services from Azure and choose the best service for the company's business scenarios.

* Tailwind traders was able to capture telemetry data from appliances, combine it with some machine learning to predict future maintenance and create a significant value added service for customers by using azure IoT hub.
* The company was able to implement a complete real time logistics system to track deliveries and vehicles by using Azure IoT Central and the Connected Logistics Starter template.
* And finally, it was able to design and build a secure modern point of sale South Checco terminal by using azure sphere

Without azure IoT services. Receiving messages from devices might still be possible, but it would likely be less secure and require custom development to implement a dashboard for reporting and management. It would also be more difficult to push software or firmware updates to each device. IoT is an exciting evolution in computing that bridges the physical and digital worlds. Azure IoT services provide a significant amount of functionality for organizations that want to build device driven and censor driven solutions.

----------------------------------------------------------------------------------------------------------------------------------------------------------