Choose the best Azure IoT service for your application

Introduction

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.

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.

In this module, 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'll learn about what the various services do, how they're different or complementary, and when to use one or the other.

Learning objectives

After you've completed this module, you'll be able to:

- Choose the Azure IoT service that best addresses your business scenario.

Prerequisites

- Familiarity with basic computing concepts and terminology
- Familiarity with cloud computing is helpful but not necessary

A1: 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 code, or optical character recognition (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.
- Flow, level, and pressure sensors for measuring gasses 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 message's 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's 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 are keeping their contents within a certain temperature range.
- 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 send this 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 product ordered from vendors.

Many services can assist and drive end-to-end solutions for IoT on Azure.

Azure IoT Hub

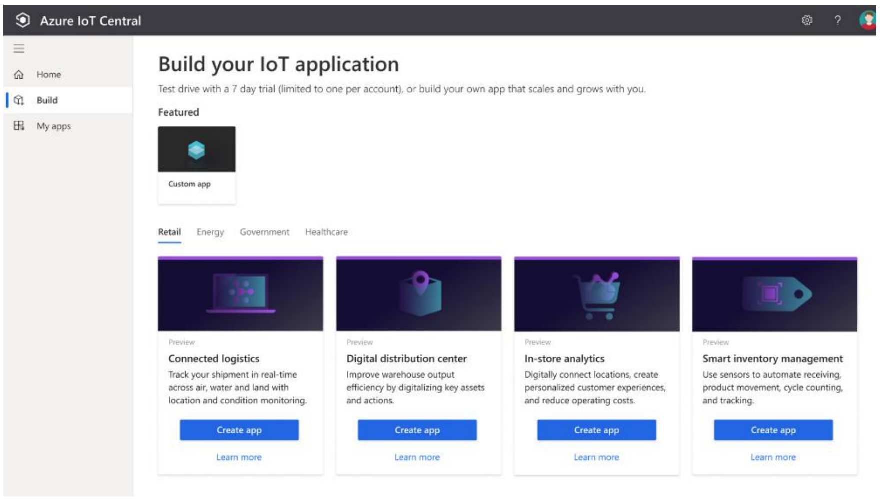
Azure IoT Hub is a managed service that's hosted in the cloud and that acts as a central message hub for bi-directional communication between your IoT application and the devices it manages. You can use Azure IoT Hub to build IoT solutions with reliable and secure communications between millions of IoT 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-reply 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.

IoT Hub monitoring helps you maintain the health of your solution by tracking events such as device creation, device failures, and device connections.

Azure IoT Central



Build your IoT application

Test drive with a 7 day trial (limited to one per account), or build your own app that scales and grows with you.

Connected logistics
Track your shipment in real-time across air, water and land with location and condition monitoring.

Digital distribution center
Improve warehouse output efficiency by digitalizing key assets and actions.

In-store analytics
Digitally connect locations, create personalized customer experiences, and reduce operating costs.

Smart inventory management
Use sensors to automate receiving, product movement, cycle counting, and tracking.

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. You can watch the overall performance across all devices in aggregate, and you can set up alerts that send notifications when a specific device needs maintenance. Finally, you can push firmware 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-side 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.

Azure Sphere

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:

- The first part is the Azure Sphere micro-controller unit (MCU), which is responsible for processing the operating system and signals from attached sensors. The following image displays the Seeed Azure Sphere MT3620 Development Kit MCU, one of several different starter kits that are available for prototyping and developing Azure Sphere applications.
- The second part is a customized Linux operating system (OS) that handles communication with the security service and can run the vendor's software.

- 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. 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 error information.

A2: Analyze the decision criteria

In this unit, we'll analyze the criteria that experts employ 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.

Is it critical to ensure that the device is not compromised?

Not in every case. Manufacturers and customers would rather not have their devices to be maliciously compromised and used for nefarious purposes, however in some cases it's more critical to ensure the integrity than others. An example would be that of an ATM in comparison to a washing machine. When security is a critical consideration in your product's design, the best product option is Azure Sphere, which provides a comprehensive end-to-end solution for IoT devices.

As we mentioned in the previous unit, 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 device is uncompromised. After a secure channel is established, messages can be received from the device securely, and messages or software updates can be sent to the device remotely.

Do I need a dashboard for reporting and management?

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.

However, if you want a pre-built customizable user interface with which you can view and control your devices remotely, you might prefer to start with IoT Central. With this solution, you can control a single device or all devices at once, and you can set up alerts for certain conditions, such as a device failure.

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 template as is or customize it to suit your needs. You can have multiple dashboards and target them at a variety of users.

A3: Use 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 the data 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 pro-active maintenance.

Because Tailwind Traders already has software in place for managing appliance maintenance requests, the company wants to integrate all functionality into this existing system.

Which service should you choose?

Let's apply the decision criteria from the previous unit.

First, is it critical to ensure that the device or, in this case, each appliance, isn't compromised? It's preferable, but not critical, that the devices aren't compromised. The worst that could happen is that 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.

Second decision criterion: do I need a dashboard for reporting and management? In this case, no. Tailwind Traders wants to integrate the telemetry data and all other functionality into an existing maintenance request system. In this case, Azure IoT Central is not required.

So, given the responses to the decision criteria, Azure IoT Hub is the best choice in this scenario.

Why not use Azure IoT Central?

Azure IoT Central provides a dashboard that allows companies to manage IoT devices individually and an 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.

Why not use Azure Sphere?

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.

A4: Use IoT Central

Tailwind Traders owns 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 logistics 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 the temperature, humidity, tilt, shock, light, and the location of a shipment.

A few goals of this logistics system include:

- Shipment monitoring with real-time tracing and tracking.
- Shipment integrity with real-time ambient condition monitoring.
- Security from theft, loss, or damage of shipments.
- Geo-fencing, route optimization, fleet management, and vehicle analytics.
- 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.

Which service should you choose?

Here again, apply the decision criteria that you learned about earlier.

First, is it critical to ensure that the device or, in this case, each appliance, 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.

Second, does Tailwind Traders need a dashboard for reporting and management? Yes, a reporting and management dashboard is a requirement.

Based on these responses to the decision criteria, Azure IoT Central is the best choice in this scenario. The Connected Logistics starter template provides an out-of-box dashboard that will satisfy many of these requirements. This dashboard is preconfigured to showcase the critical logistics device operations activity. Admittedly, the dashboard might need to be reconfigured to remove sea vessel gateways, but the truck gateway functionality would be almost exactly what Tailwind Traders needs.

Why not use IoT Hub?

If Tailwind Traders uses IoT Central, the company would actually be using an IoT hub that's preconfigured 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.

Why not use Azure Sphere?

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.

A5: Use 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 the 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 decides 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.

Which service should you choose?

Again, apply the decision criteria as you've been doing.

First, is it 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.

Next, does Tailwind Traders need a dashboard for reporting and management? Yes, the company requires a reporting and management dashboard.

So, 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.

Why not choose IoT Hub?

By using IoT Central, Tailwind Traders would actually be using Azure IoT Hub behind the scenes as well.

Summary

Our goal in this module was to help Tailwind 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 self-checkout terminal by using Azure Sphere.

Without Azure IoT services, receiving messages from devices might still be possible, but it would likely be much 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 sensor-driven solutions.

Further reading

Azure Sphere development kits provide everything you need to start prototyping and developing Azure Sphere applications. Order a kit and start taking advantage of the rich development experience in Visual Studio. Get started with Azure Sphere.

Choose the best AI service for your needs

Introduction

Artificial Intelligence (AI) is a category of computing that adapts and improves its decision-making ability over time based on its successes and failures. Microsoft Azure provides several AI solutions to choose from, each one depending on the problem you're trying to solve.

Tailwind Traders, a traditional brick-and-mortar retailer that has experienced explosive online sales growth, faces exciting challenges as it seeks to improve its e-commerce and service operations. Microsoft's AI services might be a good fit for one of the company's new initiatives, but Tailwind Traders needs help to better understand which product option is best for each scenario.

In this module, you'll learn about the various Microsoft AI services, and you'll analyze the decision criteria that experts use to select the right service for a specified scenario.

Learning objectives

After completing this module, you'll be able to:

- Choose the Azure AI services that best address your company's business challenges.

Prerequisites

- Familiarity with the concept of *application programming interfaces*, or APIs. Programmers use APIs to interact with the functionality that's contained in code libraries.
- Familiarity with the following additional concepts:
    - *Web API*: An API that's accessible from servers that accept requests via HTTP.
    - *Web API endpoint*: The location of the code library.
    - *REST API*: The design of the URL style that's used to expose the API's functionality.

B1: Identify the product options

AI is a broad classification of computing that allows a software system to perceive its environment and take action that maximizes its chance of successfully achieving its goals. A goal of AI is to create a software system that's able to adapt, or learn something on its own without being explicitly programmed to do it.

There are two basic approaches to AI. The first is to employ a *deep learning* system that's modeled on the neural network of the human mind, enabling it to discover, learn, and grow through experience.

The second approach is *machine learning*, a data science technique that uses existing data to train a model, test it, and then apply the model to new data to forecast future behaviors, outcomes, and trends.

Forecasts or predictions from machine learning can make apps and devices smarter. For example, when you shop online, machine learning powers product recommendation systems that offer additional products based on what you've bought and what other shoppers have bought who have purchased similar items in the past.

Machine learning is also used to detect credit card fraud by analyzing each new transaction and using what it has learned from analyzing millions of fraudulent transactions.

Virtually every device or software system that collects textual, visual, and audio data could feed a machine learning model that makes that device or software system smarter about how it functions in the future.

Azure product options

At a high level, there are three primary product offerings from Microsoft, each of which is designed for a specific audience and use case. Each option provides a diverse set of tools, services, and programmatic APIs. In this module, we'll merely scratch the surface of the options' capabilities.

Azure Machine Learning

Azure Machine Learning is a platform for making predictions. It consists of tools and services that allow you to connect to data to train and test models to find one that will most accurately predict a future result. After you've run experiments to test the model, you can deploy and use it in real time via a web API endpoint.

With Azure Machine Learning, you can:

- Create a process that defines how to obtain data, how to handle missing or bad data, how to split the data into either a training set or test set, and deliver the data to the training process.
- Train and evaluate predictive models by using tools and programming languages familiar to data scientists.
- Create pipelines that define where and when to run the compute-intensive experiments that are required to score the algorithms based on the training and test data.
- Deploy the best-performing algorithm as an API to an endpoint so it can be consumed in real time by other applications.

Choose Azure Machine Learning when your data scientists need complete control over the design and training of an algorithm using your own data. The following video discusses the basic steps required to set up a machine learning system.

Azure Cognitive Services

Azure Cognitive Services provides prebuilt machine learning models that enable applications to see, hear, speak, understand, and even begin to reason. Use Azure Cognitive Services to solve general problems, such as analyzing text for emotional sentiment or analyzing images to recognize objects or faces. You don't need special machine learning or data science knowledge to use these services. Developers access Azure Cognitive Services via APIs and can easily include these features in just a few lines of code.

While Azure Machine Learning requires you to bring your own data and train models over that data, Azure Cognitive Services, for the most part, provides pretrained models so that you can bring in your live data to get predictions on.

Azure Cognitive Services can be divided into the following categories:

- **Language** services: Allow your apps to process natural language with prebuilt scripts, evaluate sentiment, and learn how to recognize what users want.
- **Speech** services: Convert speech into text and text into natural-sounding speech. Translate from one language to another and enable speaker verification and recognition.
- **Vision** services: Add recognition and identification capabilities when you're analyzing pictures, videos, and other visual content.
- **Decision** services: Add personalized recommendations for each user that automatically improve each time they're used, moderate content to monitor and remove offensive or risky content, and detect abnormalities in your time series data.

Azure Bot Service

Azure Bot Service and Bot Framework are platforms for creating virtual agents that understand and reply to questions just like a human. Azure Bot Service is a bit different from Azure Machine Learning and Azure Cognitive Services in that it has a specific use case. Namely, it creates a virtual agent that can intelligently communicate with humans. Behind the scenes, the bot you build uses other Azure services, such as Azure Cognitive Services, to understand what their human counterparts are asking for.

Bots can be used to shift simple, repetitive tasks, such as taking a dinner reservation or gathering profile information, on to automated systems that might no longer require direct human intervention. Users converse with a bot by using text, interactive cards, and speech. A bot interaction can be a quick question and answer, or it can be a sophisticated conversation that intelligently provides access to services.

B2: Analyze the decision criteria

In this unit, you'll analyze the criteria that experts employ when they choose an AI service for a specific business need. Understanding the criteria can also help you better understand the nuanced differences among the products.

Are you building a virtual agent that interfaces with humans via natural language?

Use Azure Bot Service when you need to create a virtual agent to interact with humans by using natural language. Bot Service integrates knowledge sources, natural language processing, and form factors to allow interaction across different channels.

Bot Service solutions usually rely on other AI services for such things as natural language understanding or even translation for localizing replies into a customer's preferred language.

Before you jump in to build a custom chat experience by using Bot Service, it might make sense to search for prebuilt, no-code solutions that cover common scenarios. For example, you can use QnA Maker, which is available from Azure Marketplace, to build, train, and publish a sophisticated bot that uses FAQ pages, support websites, product manuals, SharePoint documents, or editorial content through an easy-to-use UI or via REST APIs.

Likewise, Power Virtual Agents integrates with Microsoft Power Platform so that you can use hundreds of prebuilt connectors for data input. You can extend Power Virtual Agents by building custom workflows with Power Automate, and if you feel that the out-of-the-box experience is too limiting, you can still build more complex interactions with Microsoft Bot Framework.

Do you need a service that can understand the content and meaning of images, video, or audio, or that can translate text into a different language?

Use Azure Cognitive Services when it comes to general purpose tasks, such as performing speech to text, integrating with search, or identifying the objects in an image. Azure Cognitive Services is *general purpose*, meaning that many different kinds of customers can benefit from the work that Microsoft has already done to train and test these models and offer them inexpensively at scale.

Do you need to predict user behavior or provide users with personalized recommendations in your app?

The Azure Cognitive Services Personalizer service watches your users' actions within an application. You can use Personalizer to predict their behavior and provide relevant experiences as it identifies usage patterns. Here again, you could capture and store user behavior and create your own custom Azure Machine Learning solution to do these things, but this approach would require much effort and expense.

Will your app predict future outcomes based on private historical data?

Choose Azure Machine Learning when you need to analyze data to predict future outcomes. For example, suppose you need to analyze years' worth of financial transactions to discover new patterns that could help you create new products and services for your company's clients and then offer those new services during routine customer service calls. When you're working with proprietary data, you'll likely need to build a more custom-tailored machine learning model.

Do you need to build a model by using your own data or perform a different task than those listed above?

Use Azure Machine Learning for maximum flexibility. Data scientists and AI engineers can use the tools they're familiar with and the data you provide to develop deep learning and machine learning models that are tuned for your particular requirements.

B3: Use Machine Learning for decision support systems

The Tailwind Traders e-commerce website allows its customers to browse and purchase items that can be delivered or picked up from a retail store nearest to their location.

The Marketing team is convinced that it can increase sales dramatically by suggesting add-on products that complement the items in a shopper's cart at the point of checkout. The team could hard-code these suggestions, but it feels that a more organic approach would be to use its years' worth of sales data as well as new shopping trends to decide what products to display to the shopper. Additionally, the suggestions could be influenced by product availability, product profitability, and other factors.

The Marketing team's existing data science experts have already done some initial analysis of the problem domain, and have determined that its plan might take months to prototype, and possibly a year to roll out.

Which service should you choose?

Let's apply the decision criteria you learned about in the preceding unit to find the right option.

First, is Tailwind Traders building a virtual agent that interfaces with humans via natural language? No, it is not, so Azure Bot Service is not a good candidate for this scenario.

Second, does Tailwind Traders need a service that can understand the content and meaning of images, video, audio, or translate text into a different language? No, it doesn't, so the relevant Cognitive Services will not help the company.

Third, does Tailwind Traders need to predict user behavior or provide users with personalized recommendations? Yes, it does. However, creating recommendations based on user behavior is only part of the requirement. Tailwind Traders needs to create a complex model that incorporates historical sales data, trending sales data, inventory, and more. It's possible that the Azure Cognitive Services Personalizer service could play a role, but it couldn't handle the entire breadth of the project alone.

Fourth, will the Tailwind Traders app predict future outcomes based on private historical data? Yes, and that is why in this scenario, Azure Machine Learning is likely the best choice.

The success of this effort would depend primarily on the ability of the model to select precisely the right up-sale products to suggest to the shopper. Because the model would need to be tweaked and tuned over time, an off-the-shelf model would likely not suffice.

Finally, it sounds like the Marketing team already employs some data science experts, and the team is willing to make at least a year-long commitment to building, testing, and tweaking the models to be used.

B4: Use Cognitive Services for data analysis

The first generation of the Tailwind Traders e-commerce website was available exclusively in English. However, when the Marketing team sponsored a demographics study for the company's brick-and-mortar locations, it found that, on average, only 80 percent of potential customers speak English. In some neighborhoods, that number falls to 50 percent. The team sees the addition of multiple languages as a wonderful opportunity to serve non-English speakers with the same online e-commerce experience as English speakers.

Which service should you choose?

As in the preceding unit, apply the decision criteria you learned about earlier to find the right option.

First, is Tailwind Traders building a virtual agent that interfaces with humans via natural language? No, it is not, so Azure Bot Service is not a good candidate for this scenario. However, should Tailwind Traders ever implement a customer service agent, it might want to consider using the Translator API to provide real-time translation to help customers who are not English speakers.

Second, does Tailwind Traders need a service that can understand the content and meaning of images, video, audio, or translate text into a different language? Yes, it does. Translating textual content from one language into another is a general purpose task that you can simplify by using the Azure Cognitive Services Translator service. The service is easy to integrate into your applications, websites, tools, and solutions. It allows you to add multilanguage user experiences in more than 60 languages, and you can use it on any hardware platform with any operating system for text-to-text language translation.

Azure Cognitive Services is likely the best option for this scenario, but let's continue applying the decision criteria to make sure.

Third, does Tailwind Traders need to predict user behavior or provide users with personalized recommendations? No, it doesn't, so the Azure Cognitive Services Personalizer is not a good candidate for this scenario.

Finally, will the Tailwind Traders app need to predict future outcomes based on private historical data? No. Although it's possible to create a Machine Learning model for multilanguage translation, it would be expensive and time consuming for Tailwind Traders to attempt to build translation models themselves. The team has neither the deep learning competency nor the linguistic data that's required to train the models.

Now that you've examined all the expert criteria, you can confidently select Cognitive Services as the best product option for this scenario.

B5: Use Bot Service for interactive chat experiences

The Customer Service team has long asked for a virtual agent to handle the vast majority of questions it gets asked. No matter how prominent it makes the answers to the most frequently asked questions on the website, shoppers are impatient and perceive contact in a chat window as saving them time.

The team wants shoppers to feel as though they're interacting with a real human. When it becomes clear that the virtual agent can't provide an answer, the chat session should be transferred to a human.

Providing a virtual agent would decrease the amount of time it takes for all shoppers to receive answers. The virtual agent could answer most questions, which would free up human customer service agents to provide support for more difficult questions or thorny account-related issues.

Which service should you choose?

Once again, apply the decision criteria you're now familiar with to find the right product.

First, is Tailwind Traders building a virtual agent that interfaces with humans via natural language? Yes, it is. Azure Bot Service should be used in this scenario to implement a virtual agent chat experience. Bot Service could benefit from the information on the website's Frequently Asked Questions page, along with thousands of chat sessions that have been stored between shoppers and customer service representatives. Customer Service supervisors can test and tweak the answers to continue to refine the chat experience.

Even though you've likely found the best option for this scenario, keep applying the decision criteria to see whether any additional options might work.

Second, does Tailwind Traders need a service that can understand the content and meaning of images, video, audio, or translate text into a different language? Possibly, yes. In this scenario, Azure Cognitive Services could be used along with Bot Service to build the solution. To expedite implementation, the developers could explore using prebuilt solutions, such as QnA Maker (part of Cognitive Services) or Power Virtual Agents. Also, any Azure Bot solution would likely implement several Azure Cognitive Services, such as Language Understanding (LUIS) and possibly Translator, to translate from the shopper's language to English and back again.

Third, does Tailwind Traders need to predict user behavior or provide users with personalized recommendations? No, it doesn't. Azure Cognitive Services Personalizer is not a good candidate for this scenario.

Finally, will the Tailwind Traders app need to predict future outcomes based on private historical data? No. Although Tailwind Traders *does* have historical data to feed into a model, which would make it possible to use Azure Machine Learning to create a chat solution, another option is already tailored for the chat bot experience.

Summary

Our goal in this module was to help Tailwind Traders explore several AI service offerings from Azure that it can apply to various business opportunities.

You identified a few product options and their capabilities, including Azure Bot Service, Azure Cognitive Services, and Azure Machine Learning. You analyzed certain decision criteria to help yourself choose one option over another depending on the scenario. Then you applied those decision criteria to three Tailwind Traders initiatives, helping the company find the best service option for each scenario.

Without AI services, Tailwind Traders would spend more time and effort on manual tasks, respond to customers less quickly, offer weak product recommendations, and be unable to fully support customers who speak languages other than English.

AI is one focus that could transform every area of a business. Such transformation is limited only by the creativity and imagination of the organization.

Learn more

This module discussed several products and services that you can learn more about:

- For an exhaustive list of services available in Azure Cognitive Services, see What are Azure Cognitive Services?.
- The Cognitive Services Personalizer service was mentioned as a possible solution for one of the scenarios. For more information, see Cognitive Services Personalizer.
- Azure Language Understanding (LUIS) was mentioned as a way to interact with the Bot Service by using natural language. For more information, see Azure Language Understanding.
- QnA Maker was mentioned as a pre-packaged virtual assistant solution available from Azure Marketplace. For more information, see QnA Maker.

Choose the best Azure serverless technology for your business scenario

Introduction

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, and 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.

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.

In this module, you'll learn about two serverless computing solutions on Azure: Azure Functions and Azure Logic Apps. You'll learn what they are, how they differ, and when you should choose one over the other.

Learning objectives

After completing this module, you'll be able to:

- Choose the serverless computing technology that best addresses your business scenario.

Prerequisites

- An understanding of the concept of orchestration and workflows
- An understanding of the concept of application programming interface (API)
- High-level familiarity with relevant Microsoft products such as Dynamics 365 and Office 365

C1: Identify the product options

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. 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 service, and you then 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're billed only for the resources you use. You don't even need to reserve resources.

Serverless computing is ordinarily used to handle *back-end* 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.

In this module, we'll cover two Azure serverless computing services: Azure Functions and Azure Logic Apps.

Azure Functions

With the Azure Functions 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 on a queue, or a message on a timer.

Because of its atomic nature, Azure Functions can serve many purposes in an application's design. Functions can be written in many common programming languages, such as C#, Python, JavaScript, Typescript, Java, and PowerShell.

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. For example, you might be receiving messages from an IoT solution that monitors a fleet of delivery vehicles. You'll 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. And if 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 allow developers to chain functions together while maintaining state.

The Azure Functions solution is ideal when you're concerned only with the code that's running your service and not the underlying platform or infrastructure. You use Azure 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 Logic Apps

Logic Apps is a low-code/no-code development platform hosted as a cloud service. The 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 simplifies 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 (EAI), and business-to-business (B2B) integration.

Azure Logic Apps is designed in a web-based designer and can execute logic that's triggered by Azure services without writing 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, then a new message to be sent to a queue, or an HTTP request. An action is a task or step that can execute. There are logic 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.

What are the differences between these services?

You can call Azure Functions from Azure Logic Apps, and vice versa. The primary difference between the two services is their intent. Azure Functions is a serverless compute 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.

C2: 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 unit, we'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.

Do 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.

Do you need to execute custom algorithms or perform specialized data parsing and data lookups?

With Azure Functions, you can use the full expressiveness of a programming language in a compact form. This lets you concisely build complex algorithms, or data lookup and parsing operations. You would be responsible for maintaining the code, handling exceptions resiliently, and so on.

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 verbose and visually overwhelming.

Do you have existing automated tasks written in an imperative programming language?

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 re-create it by using Azure Logic Apps.

Do you prefer a visual (declarative) workflow or writing (imperative) code?

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.

C3: 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 2:00 AM. A Windows service that's written in C# contains all of the necessary logic to:

- Retrieve the messages.
- Parse the JSON.
- Perform a lookup across multiple databases to find additional product information.
- 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.

Which service should you choose?

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.

Why not choose Azure Logic Apps?

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.

C4: 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, its 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.

Which service should you choose?

In this scenario, Azure Logic Apps is likely the best solution. 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.

Because Azure Logic Apps is a low-code/no-code service, no developers are needed. A cloud or IT professional should be able to build and support this workflow.

Why not choose Azure Functions?

Although it's possible to build the entire solution by using Azure Functions, this approach might be a challenge if no software developers can be committed to this project.

This is an ideal scenario for Azure Logic Apps. Connectors already exist for each of the steps outlined in the workflow. It would take quite a bit of research, development, and testing for a developer to build a solution that utilizes all these disparate software systems.

Summary

In this module, we wanted to help 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# Windows service, we helped it choose Azure Functions.

When the company needed to orchestrate a workflow to improve customer retention after a negative shopping experience, we helped it choose Azure Logic Apps.

In both cases, we noted how choosing the other serverless computing service would be possible. However, we tried to help the company consider the decision criteria we outlined and choose the right service for the scenario.

Without serverless computing, Tailwind 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, write, test, and maintain custom code to get similar results.

By helping Tailwind Traders select the right serverless computing solutions, we were able to deploy new functionality to help the company improve customer satisfaction with its e-commerce platform.

Choose the best tools to help organizations build better solutions

Introduction

Modern software development practices are supported by tools that encompass virtually every aspect of the software development life cycle. Microsoft has created a comprehensive set of tools that help organizations implement DevOps practices, develop solutions, and save money while doing so. In this module, you'll learn how to choose the right tools to support those practices.

Tailwind Traders has experimented with various software development processes and tools. Until now, however, there has been no organizational commitment to shift to a DevOps mindset. Likewise, there has been no planned, coordinated effort to standardize on a set of core tools and processes. Several new initiatives at the company accentuate the need for agile, repeatable, dependable management and deployment of software systems. Tailwind Traders believes that the adoption of DevOps tooling and practices is critical to the company's future success.

In this module, you'll learn about the various software development process tools that Microsoft offers. You'll explore the criteria that experts use to make their choices.

By the end of this module, you'll be able to choose the software development process tools and services that best align with your organization's goals and practices.

Learning objectives

After completing this module, you'll be able to:

- Choose the software development process tools and services that best address specific business scenarios.

Prerequisites

- A development or operations background is valuable but not required.
- Some familiarity with the concept of DevOps and its larger purpose in the organization, goals, outcomes, and so on.
- To help with understanding the value of the tools covered in this module, familiarity with such concepts as:
    - Software development lifecycle
    - Source-code management and version control
    - The various forms of testing
    - Continuous integration and continuous delivery, or CI/CD

    - Continuous deployment
    - Infrastructure as code

D1: Understand your product options

Software developers and operations professionals strive to create working software systems that satisfy the needs of the organization. However, sometimes their short-term objectives are at cross-purposes, which can result in technical issues, delays, and downtime.

DevOps is a concept that combines philosophies and practices to facilitate technical teams as they work toward common goals. To accomplish this alignment, organizations employ practices and processes that automate the ongoing development, maintenance, and deployment of software systems. The aim is to expedite the release of software changes, ensure the ongoing deployability of the system, and ensure that all changes meet a high quality bar.

When done correctly, DevOps practices and processes touch nearly every aspect of a company and the software development lifecycle, including planning, project management, and the collaboration of software developers and operations and quality assurance teams. Tooling automates and enforces most of the practices and processes, making it both difficult and unnecessary to work around.

DevOps requires a fundamental mindset change from the top down. Organizations can't merely install software tools or adopt services and hope to get all of the benefits promised by DevOps.

In this module, we'll focus on the Microsoft tools that can help accomplish some of the DevOps objectives. Organizations that aren't ready to fully embrace the power of DevOps can support technical teams in their cloud development activities. If you're interested in learning more about DevOps in general, Microsoft Learn has several learning paths and modules that can help you.

Microsoft tools enable source-code management, continuous integration and continuous delivery (CI/CD), and automate the creation of testing environments. It seems as though these tools overlap in functionality, so in this module you'll learn about product options, and when to choose one product over another.

Product options

At a high level, there are three primary offerings, each of which is aimed at a specific audience and use case, that provide a diverse set of tools, services, programmatic APIs, and more.

Azure DevOps Services

Azure DevOps Services is a suite of services that address every stage of the software development lifecycle.

- **Azure Repos** is a centralized source-code repository where software development, DevOps engineering, and documentation professionals can publish their code for review and collaboration.

- **Azure Boards** is an agile project management suite that includes Kanban boards, reporting, and tracking ideas and work from high-level epics to work items and issues.
- **Azure Pipelines** is a CI/CD pipeline automation tool.
- **Azure Artifacts** is a repository for hosting artifacts, such as compiled source code, which can be fed into testing or deployment pipeline steps.
- **Azure Test Plans** is an automated test tool that can be used in a CI/CD pipeline to ensure quality before a software release.

Azure DevOps is a mature tool with a large feature set that began as on-premises server software and evolved into a software as a service (SaaS) offering from Microsoft.

GitHub and GitHub Actions

GitHub is arguably the world's most popular code repository for open-source software. Git is a decentralized source-code management tool, and GitHub is a hosted version of Git that serves as the primary remote. GitHub builds on top of Git to provide related services for coordinating work, reporting and discussing issues, providing documentation, and more. It offers the following functionality:

- It's a shared source-code repository, including tools that enable developers to perform code reviews by adding comments and questions in a web view of the source code before it can be merged into the main code base.
- It facilitates project management, including Kanban boards.
- It supports issue reporting, discussion, and tracking.
- It features CI/CD pipeline automation tooling.
- It includes a wiki for collaborative documentation.
- It can be run from the cloud or on-premises

Most relevant for this module, GitHub Actions enables workflow automation with triggers for many lifecycle events. One such example would be automating a CI/CD *toolchain*.

A toolchain is a combination of software tools that aid in the delivery, development, and management of software applications throughout a system's development lifecycle. The output of one tool in the toolchain is the input of the next tool in the toolchain. Typical tool functions range from performing automated dependency updates to building and configuring the software, delivering the build artifacts to various locations, testing, and so on.

With such similarity between many GitHub and Azure DevOps features, you might wonder which product to choose for your organization. Unfortunately, the answer might not be straightforward.

Although both Azure DevOps and GitHub allow public and private code repositories, GitHub has a long history with public repositories and is trusted by tens of thousands of open-source project owners. GitHub is a lighter-weight tool than Azure DevOps, with a focus on individual developers contributing to the open-source code. Azure DevOps, on the other hand, is more focused on enterprise development, with heavier project-management and planning tools, and finer-grained access control.

Note

Your choices are not limited to Azure DevOps Services or GitHub and GitHub Actions. In practice, you can mix and match these services as needed. For example, you can use GitHub repos with Azure Boards for work item tracking.

Azure DevTest Labs

Azure DevTest Labs provides an automated means of managing the process of building, setting up, and tearing down virtual machines (VMs) that contain builds of your software projects. This way, developers and testers can perform tests across a variety of environments and builds. And this capability isn't limited to VMs. Anything you can deploy in Azure via an ARM template can be provisioned through DevTest Labs. Provisioning pre-created lab environments with their required configurations and tools already installed is a huge time saver for quality assurance professionals and developers.

Suppose you need to test a new feature on an old version of an operating system. Azure DevTest Labs can set up everything automatically upon request. After the testing is complete, DevTest Labs can shut down and deprovision the VM, which saves money when it's not in use. To control costs, the management team can restrict how many labs can be created, how long they run, and so on.

D2: Analyze the decision criteria

In this unit, you'll analyze the criteria that experts employ when they choose DevOps tools or services to address specific business needs. Understanding the criteria can also help you better understand the nuanced differences between each product.Do you need to automate and manage test-lab creation?

If your aim is to automate the creation and management of a test lab environment, consider choosing Azure DevTest Labs. Among the three tools and services we've described, it's the only one that offers this functionality.

However, you can automate the provisioning of new labs as part of a toolchain by using Azure Pipelines or GitHub Actions.

Are you building open-source software?

Although Azure DevOps can publish public code repositories, GitHub has long been the preferred host for open-source software. If you're building open-source software, you would likely choose GitHub if for no other reasons than its visibility and general acceptance by the open-source development community.

The remaining decision criteria are specific to choosing between either Azure DevOps or GitHub.

Note

Your choices aren't limited to Azure DevOps Services or GitHub and GitHub Actions. In practice, you can mix and match these services as needed. For example, you can use GitHub repos with Azure Boards for work-item tracking.

Regarding source-code management and DevOps tools, what level of granularity do you need for permissions?

GitHub works on a simple model of read/write permissions to every feature. Meanwhile, Azure DevOps has a much more granular set of permissions that allow organizations to refine who is able to perform most operations across the entire toolset.

Regarding source-code management and DevOps tools, how sophisticated does your project management and reporting need to be?

Although GitHub has work items, issues, and a Kanban board, project management and reporting is the area where Azure DevOps excels. Azure DevOps is highly customizable, which allows an administrator to add custom fields to capture metadata and other information alongside each work item. By contrast, the GitHub Issues feature uses tags as its primary means of helping a team categorize issues.

Regarding source-code management and DevOps tools, how tightly do you need to integrate with third-party tools?

Although we make no specific recommendations about third-party tools, it's important for you to understand your organization's existing investments in tools and services and to evaluate how these dependencies might affect your choice. It's likely that most vendors that create DevOps tools create hooks or APIs that can be used by both Azure Pipelines and GitHub Actions. Even so, it's probably worth the effort to validate that assumption.

D3: Use Azure DevOps to manage the application development lifecycle

The software development team at Tailwind Traders works on many different projects, both for internal and external usage. The team needs to give project sponsors and managers executive level reporting, including burndown charts, track progress against epics, and track custom information that's specific to Tailwind Traders in each work item and bug report.

As Tailwind Traders grows and hires contractors and outside vendors for short-term work, the upper management team wants to ensure that these individuals have access only to the information they need to do their work.

Which services should we choose?

Apply the decision criteria you learned about in the preceding unit to find the right option.

First, does Tailwind Traders need to automate and manage test lab creation? No. So, in this scenario, Azure DevTest Labs is not a candidate, because it isn't intended for this specific use case.

Second, is Tailwind Traders building open-source software? Though it's not stated specifically, Tailwind Traders is building internal and external systems, such as their e-commerce system, which isn't open source. So that isn't a consideration in this scenario.

Third, what level of granularity does Tailwind Traders need for permissions? Earlier, we stated that Tailwind Traders will hire temporary employees and vendors for short-term work, which makes a granular permissions requirement an important consideration for upper management. Based on our description in the preceding unit, this feature would make Azure DevOps a leading candidate. By using Azure DevOps, Tailwind Traders administrators would also have a more robust set of options for controlling permissions across the entire portfolio of work.

Fourth, does Tailwind Traders require a sophisticated project management and reporting solution? Yes, robust project management and reporting features are one of the primary considerations. Here again, because of the amount of work-item customization and reporting the management team wants, Azure DevOps would likely be a good choice.

Fifth, does Tailwind Traders require tight integration with any third-party DevOps tools? Tool integration was not listed as a primary consideration for this scenario. As you learned in the preceding unit, most third-party DevOps tools integrate with both Azure DevOps and GitHub, which makes it likely that the team will find the tools it needs.

D4: Use GitHub to contribute to open-source software

Tailwind Traders hopes to publish an API that would allow third parties to integrate their own inventories of new and used items. This approach would allow Tailwind Traders to offer a wider variety of products directly from their e-commerce site.

Although the internal implementation of the API is closed source, Tailwind Traders wants to create a set of examples that call the API to perform various actions. The team needs a platform to share example code, collect feedback on the API, allow contributors to report issues, and build a community around feature requests.

Which service should you choose?

Apply the decision criteria you learned about earlier to find the right option.

First, does Tailwind Traders need to automate and manage test lab creation? No. In this scenario, Azure DevTest Labs is not a candidate because it isn't designed for this use case.

Second, is Tailwind Traders building open-source software? Yes. As we noted in a previous unit, developers are used to seeing this kind of content available on GitHub. With GitHub, Tailwind Traders developers can publish their code, accept community contributions to improve the code examples, accept feedback and bug reports, and more. Because this scenario involves open-source code, GitHub is a leading candidate.

Third, what level of granularity does the Tailwind Traders team need for assigning permissions? Though it's not stated explicitly, the fact that Tailwind Traders will be accepting community contributions, issuing reports, and generally attempting to build a community of developers around their API examples, the company's permission needs are basic: users can either *view only* or *view and write*. This is another reason why GitHub would be a good candidate for this scenario.

Fourth, does Tailwind Traders require a sophisticated project management and reporting solution? Again, because of the nature of this project, the team doesn't require a sophisticated project management and reporting solution. In this scenario, the strength of Azure DevOps Services isn't required.

Fifth, does Tailwind Traders require tight integration with any third-party DevOps tools? Tool integration wasn't listed as a primary consideration for this scenario and doesn't qualify or disqualify either tool.

GitHub is the best choice for this scenario. Although you could use Azure DevOps to make the repository public, some of the other features that involve the development community, such as feedback or bug reports, would be less accessible.

D5: Use Azure DevTest Labs to manage testing environments

Tailwind Traders wants to be more methodical and careful when it pushes new versions of its e-commerce website to production. The company will expand its quality assurance (QA) team, and it will use the cloud to create and host virtual machines (VMs). Through this approach, it will create testing environments that match the production environment.

The management team has concerns around the costs of a more automated test environment. For instance, it wants to make sure that the QA professionals are not wasting time configuring the testing environment to match the production environment. The team wants to ensure that the VMs are destroyed when they're no longer in use. It wants to limit the number of VMs that each QA professional is allowed to spin up. Also, the team wants to ensure that each environment is configured correctly and consistent with the production environment.

Which service should you choose?

Once again, start by applying the decision criteria you learned about previously to find the right product.

First, does Tailwind Traders need to automate and manage test lab creation? Yes. This looks like a job for Azure DevTest Labs, because it can do everything that the team needs to accomplish in this scenario.

We could continue evaluating the decision criteria, but neither Azure DevOps nor GitHub is needed for this scenario. Remember that either Azure DevOps or GitHub could be used to create product releases that can automatically be included in any VMs that you create for testing purposes.

Summary

The goal in this module was to help Tailwind Traders choose the best DevOps solution for a set of requirements across various software development and testing needs.

We identified various product options and capabilities, including Azure DevOps Services, GitHub (including GitHub Actions), and Azure DevTest Labs. We analyzed the criteria for choosing one option over another for each scenario. Then we applied those criteria to three separate challenges at Tailwind Traders, helping the team determine the best service option for the scenarios.

Without software development services and tools from Microsoft, the Tailwind Traders team might have difficulty in realizing the benefits of such DevOps practices as continuous integration and continuous delivery (CI/CD), source-code management, and work-item management.

DevOps practices and processes have changed the software development landscape, helping to accelerate software development and improve the deployability and quality of software systems. Microsoft offers a wealth of tools that can help organizations implement DevOps practices, experience better collaboration among technical teams, and achieve more consistent results from those teams.

Choose the best tools for managing and configuring your Azure environment

Introduction

By using Azure management tools, administrators, developers, and managers can interact with the cloud environment to perform such tasks as:

- Deploying dozens or hundreds of resources at a time.
- Configuring individual services programmatically.
- Viewing rich reports across usage, health, costs, and more.

Microsoft Azure provides a collection of management tooling options to choose from, depending on the situation.

Tailwind Traders, a traditional brick-and-mortar retailer, is now experiencing explosive growth by selling products online. The company owes much of its success to an ability to quickly and efficiently manage its cloud environment. As it began its cloud journey, Tailwind Traders initially had to choose the right management tools for its business needs.

In this module, you'll explore the array of Azure management tools and the decision criteria that experts use to select the right ones for their specific scenarios.

Learning objectives

After completing this module, you'll be able to:

- Choose the Azure management tools that best address your organization's technical needs and challenges.

Prerequisites

- Familiarity with basic computing concepts and terminology
- Familiarity with cloud computing is helpful but not necessary

E1: Identify the product options

At a high level, there are two broad categories of management tools: visual tools and code-based tools.

Visual tools provide full, visually friendly access to all the functionality of Azure. However, visual tools might be less useful when you're trying to set up a large deployment of resources with interdependencies and configuration options.

When you're attempting to quickly set up and configure Azure resources, a code-based tool is usually the better choice. Although it might take time to understand the right commands and parameters at first, after they've been entered, they can be saved into files and used repeatedly as needed. Also, the code that performs setup and configuration can be stored, versioned, and maintained along with application source code in a source code-management tool such as Git. This approach to managing hardware and cloud resources, which developers use when they write application code, is referred to as *infrastructure as code*.

There are two approaches to infrastructure as code: *imperative* code and *declarative* code. Imperative code details each individual step that should be performed to achieve a desired outcome. By contrast, declarative code details only a desired outcome, and it allows an interpreter to decide how to best achieve that outcome. This distinction is important because tools that are based on declarative code can provide a more robust approach to deploying dozens or hundreds of resources simultaneously and reliably.

Your product options

Microsoft offers a variety of tools and services to manage your cloud environment, each aimed at different scenarios and users. The following video describes some of these options.

The Azure portal

By using the Azure portal, a web-based user interface, you can access virtually every feature of Azure. The Azure portal provides a friendly, graphical UI to view all the services you're using, create new services, configure your services, and view reports. The Azure portal is how most users first experience Azure. But, as your Azure usage grows, you'll likely choose a more repeatable code-centric approach to managing your Azure resources.

The Azure mobile app

The Azure mobile app provides iOS and Android access to your Azure resources when you're away from your computer. With it, you can:

- Monitor the health and status of your Azure resources.
- Check for alerts, quickly diagnose and fix issues, and restart a web app or virtual machine (VM).
- Run the Azure CLI or Azure PowerShell commands to manage your Azure resources.

Azure PowerShell

Azure PowerShell is a shell with which developers and DevOps and IT professionals can execute commands called cmdlets (pronounced *command-lets*). These commands call the Azure Rest API to perform every possible management task in Azure. Cmdlets can be executed independently or combined into a script file and executed together to orchestrate:

- The routine setup, teardown, and maintenance of a single resource or multiple connected resources.
- The deployment of an entire infrastructure, which might contain dozens or hundreds of resources, from imperative code.

Capturing the commands in a script makes the process repeatable and automatable.

Azure PowerShell is available for Windows, Linux, and Mac, and you can access it in a web browser via Azure Cloud Shell.

Windows PowerShell has helped Windows-centric IT organizations automate many of their on-premises operations for years, and these organizations have built up a large catalog of custom scripts and cmdlets, as well as expertise.

The Azure CLI

The Azure CLI command-line interface is an executable program with which a developer, DevOps professional, or IT professional can execute commands in Bash. The commands call the Azure Rest API to perform every possible management task in Azure. You can run the commands independently or combined into a script and executed together for the routine setup, teardown, and maintenance of a single resource or an entire environment.

In many respects, the Azure CLI is almost identical to Azure PowerShell in what you can do with it. Both run on Windows, Linux, and Mac, and can be accessed in a web browser via Cloud Shell. The primary difference is the syntax you use. If you're already proficient in PowerShell or Bash, you can use the tool you prefer.

ARM templates

Although it's possible to write imperative code in Azure PowerShell or the Azure CLI to set up and tear down one Azure resource or orchestrate an infrastructure comprising hundreds of resources, there's a better way to implement this functionality.

By using Azure Resource Manager templates (ARM templates), you can describe the resources you want to use in a declarative JSON format. The benefit is that the entire ARM template is verified before any code is executed to ensure that the resources will be created and connected correctly. The template then orchestrates the creation of those resources in parallel. That is, if you need 50 instances of the same resource, all 50 instances are created at the same time.

Ultimately, the developer, DevOps professional, or IT professional needs only to define the desired state and configuration of each resource in the ARM template, and the template does the rest. Templates can even execute PowerShell and Bash scripts before or after the resource has been set up.

E2: Analyze the decision criteria

In this unit, you'll analyze the criteria that experts employ to help them decide which Azure management tools to use to address their business needs. Understanding the criteria can help you to better understand the nuanced differences among the products.

Do you need to perform one-off management, administrative, or reporting actions?

Azure PowerShell and the Azure CLI are Azure management tools that allow you to quickly obtain the IP address of a virtual machine (VM) you've deployed, reboot a VM, or scale an app. You might want to keep custom scripts for both tools handy on your local hard drive for certain operations that you need to perform multiple times.

By contrast to the Azure CLI and PowerShell, Azure Resource Manager templates (ARM templates) define the infrastructure requirements in your application for repeatable deployments. Although ARM templates aren't intended for one-off scenarios, it's possible to use them for this purpose. However, for one-off scenarios, you may prefer more agile tools like PowerShell, Azure CLI scripts, or the Azure portal.

Keep in mind that ARM templates can include both PowerShell and/or Azure CLI scripts, which will give you the ability to utilize scripts for tasks that may not be possible with the ARM template itself. The ability to combine Azure management tools gives flexibility in choosing the right tool(s) for your particular need.

The Azure portal can perform most, if not all, management and administrative actions. If you're just learning Azure and/or need to set up and manage resources infrequently (or prefer a visual interface for viewing reports), it makes sense to take advantage of the visual presentation that the Azure portal offers.

However, if you're in a cloud management or administrative role, it's less efficient to rely solely on visual scanning and clicking. To quickly find the settings and information you want to work with, the Azure CLI or PowerShell will give you the most flexibility for repeatable tasks.

The last management tool to discuss is the Azure mobile app, which you can access via an iOS or Android phone or tablet. Because it's full featured, it's likely the best choice when a laptop isn't readily available and you need to view and triage issues immediately.

Do you need a way to repeatedly set up one or more resources and ensure that all the dependencies are created in the proper order?

ARM templates define your application's infrastructure requirements for a repeatable deployment that is done in a consistent manner. A validation step ensures that all resources can be created in the proper order based on dependencies, in parallel, and idempotent.

By contrast, it's entirely possible to use either PowerShell or the Azure CLI to set up all the resources for a deployment. However, there's no validation step in these tools. If a script encounters an error, the dependency resources can't be rolled back easily, deployments happen serially, and only some operations are idempotent.

When you're scripting, do you come from a Windows administration or Linux administration background?

If you or your cloud administrators come from a Windows administration background, it's likely you'll prefer PowerShell. If you or your cloud administrators come from a Linux administration background, it's likely you'll prefer the Azure CLI. In practice, either tool can be used to perform most one-off administration tasks.

E3: Use the Azure portal to visually understand and manage your cloud environment

Tailwind Traders uses Azure extensively throughout its entire organization. To make sure that both the technical and executive teams are aware of the company's cloud spend, the director of cloud operations will begin to meet weekly with the chief financial officer (CFO) to talk about their cloud spend.

Conversations might begin at a high level, but the two officers might want to dive deep during the meeting to gain more insight into how Azure resources are being used. Ideally, they would be able to see the data displayed visually, but also be able to run custom reports in real time. Which tool can they use during their meeting?

Which service should you choose?

Apply the decision criteria you learned about in the preceding unit to find the right option.

First, in this scenario, does Tailwind Traders need to perform one-off management, administrative, or reporting actions? Yes, and given the requirement to view data visually and create custom reports during the meeting, the Azure portal is the best choice. The meeting attendees can quickly find answers to their questions by using a wealth of reporting options.

The next two decision criteria don't apply to this scenario, because the director of cloud operations and the CFO won't be deploying or configuring any resources.

The Azure portal is the correct product option for this scenario.

E4: Use Azure PowerShell for one-off administrative tasks

Tailwind Traders employs technologists with many different skills. A team of developers and administrators builds and maintains a collection of intranet applications that are vital to the business. The team members have strong backgrounds in Windows development and network administration.

The team moved its applications to the cloud, and it now needs a way to perform one-off testing, management, and administrative tasks in its intranet environment. The team quickly realized that managing Azure from the portal takes too much time and is not repeatable. Which tool should the company use for one-off tasks?

Which service should you choose?

As you did in the preceding unit, apply the decision criteria you learned about earlier to find the right option.

First, in this scenario, does the Tailwind Traders team need to perform one-off management, administrative, or reporting tasks? Yes. However, the team already knows that it doesn't want to rely on the Azure portal for these one-off actions. Therefore, both Azure PowerShell and the Azure CLI are good options. We'll hone in on which tool the team should use in a moment.

Second, in this scenario, does Tailwind Traders need a repeatable and reliable means of deploying its entire infrastructure? No, not in this scenario. Therefore, Azure Resource Manager templates (ARM templates) are not the right choice.

When the Tailwind Traders team is doing scripting, does it come from a Windows administration or Linux administration background? This team has a Windows administration background. It would likely be most comfortable using Azure PowerShell, because this tool allows it to use the syntax it's most comfortable with to perform one-off administration tasks.

Azure PowerShell is the best choice for this scenario.

E5: Use the Azure CLI for one-off administrative tasks

As we noted in the preceding unit, Tailwind Traders employs technologists with many different skills. The DevOps team is primarily concerned with keeping external systems, such as the company's e-commerce site, up and running. This team has a Linux administration background. It frequently needs to perform administrative tasks related to the health of the cloud environment. The team quickly realized that managing Azure from the portal takes too much time and isn't repeatable. Which tool should it use for one-off tasks?

Which service should you choose?

Once again, apply the decision criteria you learned about earlier to find the right option.

Because this scenario is almost identical to the one in the preceding unit, you can skip over the first two criteria. In other words, you can quickly eliminate Azure Resource Manager templates (ARM templates) and the Azure portal as viable options for this scenario. So, let's go to the third decision criterion.

Choosing the right option in this scenario should be determined by the team's background. Because this team has a Linux administration background, it would likely be most comfortable using the Azure CLI. The Azure CLI allows the team to use the Bash shell and its syntax to perform one-off administration tasks.

The Azure CLI is the best choice for this scenario.

E6: Use the Azure mobile app to manage Azure on the go

Tailwind Traders experiences surges in e-commerce traffic that coincide with national holidays and weekends. In the company's first few years, managers of critical systems had to convene at the office of the director of cloud operations during these important periods. However, now that Tailwind Traders has successfully operationalized most critical systems, the director wants to relax this requirement and allow employees to spend these dates with their families. Is there a product that can help support this scenario?

Which service should you choose?

Let's run through our decision criteria again.

First, does Tailwind Traders need to perform one-off management, administrative, reporting actions? Yes. The real question is, how? A phone or tablet solution could help key employees keep an eye on the health of the cloud environment when they're out of the office. The Azure mobile app is likely a good compromise, because it lets employees be away from work and still perform essential, one-off management and administrative tasks.

We can skip the rest of the decision criteria in this unique scenario. The Azure mobile app is the right choice.

E7: Use ARM templates to deploy an entire cloud infrastructure

Tailwind Traders wants to operationalize their cloud deployments. The company needs a repeatable, reliable way to scale its operations during peak sales periods. Because you'll be choosing a process for scaling your production environment, you need to ensure that your chosen service:

- Is efficient and can potentially create many resources in parallel.
- Creates all dependencies in the correct order.
- Can be used without worrying that it failed in the middle of provisioning the necessary infrastructure.

Which service should you choose?

Let's run through the decision criteria one more time.

First, in this scenario, does Tailwind Traders need to perform one-off management, administrative, or reporting actions? This time, we're not looking to support one-time or one-off management or administration tasks. We're looking for a technology to automate the deployment of an entire infrastructure, as needed.

Second, does Tailwind Traders need a repeatable and reliable way to deploy its entire infrastructure? Yes, this is exactly what the company needs. Our decision criteria lead us to choose Azure Resource Manager templates (ARM templates) for this scenario.

You could use Azure PowerShell or the Azure CLI, but these scripting technologies have significant limitations when it comes to deploying infrastructure. ARM templates can help overcome these limitations.

The third decision criterion assumes that you need to write a script by using imperative code. However, when you use ARM templates, you define your infrastructure declaratively by using JSON code. In some instances, you still might need imperative code for configuration or clean-up tasks. In these cases, you can trigger the execution of scripts by using either Azure PowerShell or the Azure CLI to perform these tasks.

In this scenario, ARM templates are the correct choice.

Summary

Our goal in this module was to help Tailwind Traders choose the right cloud management tools from Microsoft for its various technical needs.

We identified a variety of product options and their capabilities, including the Azure portal, the Azure mobile app, Azure PowerShell, the Azure CLI, and Azure Resource Manager templates (ARM templates).

We analyzed decision criteria for choosing one option over another in specific scenarios.

We then applied those decision criteria to three different Tailwind Traders initiatives, helping the company find the best service option for each scenario.

Without a full suite of management tools, the company would be severely limited in how it interacts with Azure. Fortunately, Azure provides a powerful mix of visual management tools, imperative scripting tools, and declarative infrastructure-as-code tools.

Choose the best monitoring service for visibility, insight, and outage mitigation

Introduction

Modern software systems running in the cloud are complex, and gaining visibility into the health and performance of your application-hosting environment across all of its layers of services is challenging. Fortunately, there are several solutions from Microsoft that can help you react quickly to outages, research intermittent issues, optimize your usage, and be proactive in handling future planned downtime.

Tailwind Traders, a traditional brick-and-mortar retailer, is now experiencing explosive growth by selling products online. The company is seeking to tighten and operationalize control of its cloud environment. It faces several challenges, from needing to optimize its cloud spend and security posture to tracking intermittent issues and planning ahead for upcoming outages. However, the company needs help with choosing the right product option for each of these scenarios.

In this module, you'll learn about the several Microsoft monitoring solutions, and you'll analyze decision criteria that experts use to select the right service for a specific scenario.

Learning objectives

After completing this module, you'll be able to:

- Choose the cloud monitoring service that best addresses your company's business challenges.

Prerequisites

- Familiarity with basic computing concepts and terminology
- Familiarity with cloud computing is helpful but not necessary

F1: Identify your product options

Several basic questions or concerns face all companies that use the cloud.

- Are we using the cloud correctly? Can we squeeze more performance out of our cloud spend?
- Are we spending more than we need to?
- Do we have our systems properly secured?
- How resilient are our resources? If we experience a regional outage, could we fail over to another region?
- How can we diagnose and fix issues that occur intermittently?
- How can we quickly determine the cause of an outage?
- How can we learn about planned downtime?

Fortunately, by using a combination of monitoring solutions on Azure, you can:

- Gain answers, insights, and alerts to help ensure that you've optimized your cloud usage.
- Ascertain the root cause of unplanned issues.
- Prepare ahead of time for planned outages.

The product options

At a high level, there are three primary Azure monitoring offerings, each of which is aimed at a specific audience and use case and provides a diverse set of tools, services, programmatic APIs, and more.

Azure Advisor

Azure Advisor evaluates your Azure resources and makes recommendations to help improve reliability, security, and performance, achieve operational excellence, and reduce costs. Advisor is designed to help you save time on cloud optimization. The recommendation service includes suggested actions you can take right away, postpone, or dismiss.

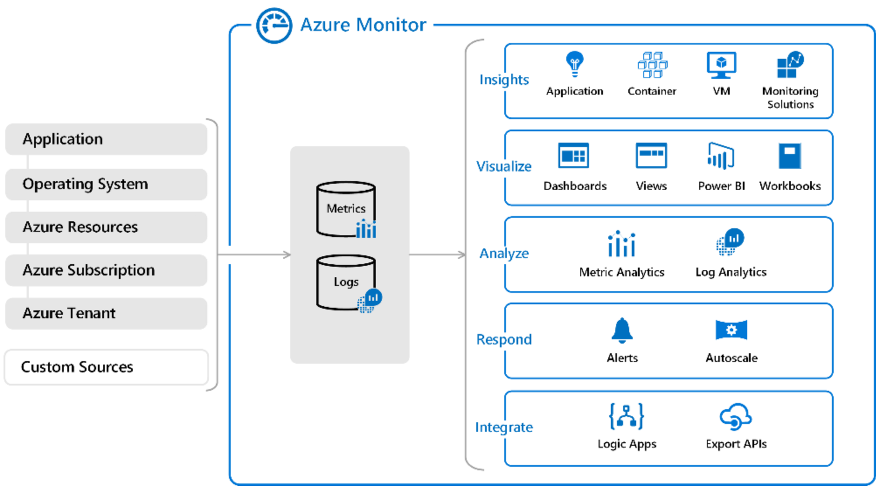
The recommendations are available via the Azure portal and the API, and you can set up notifications to alert you to new recommendations.

When you're in the Azure portal, the Advisor dashboard displays personalized recommendations for all your subscriptions, and you can use filters to select recommendations for specific subscriptions, resource groups, or services. The recommendations are divided into five categories:

- **Reliability**: Used to ensure and improve the continuity of your business-critical applications.
- **Security**: Used to detect threats and vulnerabilities that might lead to security breaches.
- **Performance**: Used to improve the speed of your applications.
- **Cost**: Used to optimize and reduce your overall Azure spending.
- **Operational Excellence**: Used to help you achieve process and workflow efficiency, resource manageability, and deployment best practices.

Azure Monitor

Azure Monitor is a platform for collecting, analyzing, visualizing, and potentially taking action based on the metric and logging data from your entire Azure and on-premises environment.



The following diagram illustrates just how comprehensive Azure Monitor is.

- On the left is a list of the sources of logging and metric data that can be collected at every layer in your application architecture, from application to operating system and network.

- In the center, you can see how the logging and metric data is stored in central repositories.
- On the right, the data is used in a number of ways. You can view real-time and historical performance across each layer of your architecture, or aggregated and detailed information. The data is displayed at different levels for different audiences. You can view high-level reports on the Azure Monitor Dashboard or create custom views by using Power BI and Kusto queries.

Additionally, you can use the data to help you react to critical events in real time, through alerts delivered to teams via SMS, email, and so on. Or you can use thresholds to trigger autoscaling functionality to scale up or down to meet the demand.

Some popular products such as Azure Application Insights, a service for sending telemetry information from application source code to Azure, uses Azure Monitor under the hood. With Application Insights, your application developers can take advantage of the powerful data-analysis platform in Azure Monitor to gain deep insights into an application's operations and diagnose errors without having to wait for users to report them.

Azure Service Health

Azure Service Health provides a personalized view of the health of the Azure services, regions, and resources you rely on. The status.azure.com website, which displays only major issues that broadly affect Azure customers, doesn't provide the full picture. But Azure Service Health displays both major and smaller, localized issues that affect you. Service issues are rare, but it's important to be prepared for the unexpected. You can set up alerts that help you triage outages and planned maintenance. After an outage, Service Health provides official incident reports, called root cause analyses (RCAs), which you can share with stakeholders.

Service Health helps you keep an eye on several event types:

- **Service issues** are problems in Azure, such as outages, that affect you right now. You can drill down to the affected services, regions, updates from your engineering teams, and find ways to share and track the latest information.
- **Planned maintenance** events can affect your availability. You can drill down to the affected services, regions, and details to show how an event will affect you and what you need to do. Most of these events occur without any impact to you and aren't shown here. In the rare case that a reboot is required, Service Health allows you to choose when to perform the maintenance to minimize the downtime.
- **Health advisories** are issues that require you to act to avoid service interruption, including service retirements and breaking changes. Health advisories are announced far in advance to allow you to plan.

F2: Analyze the decision criteria

In this unit, you'll analyze the criteria that experts employ when they choose an Azure monitoring service for a specified business need. By understanding the criteria, you can better assess the nuanced differences among the products.

Do you need to analyze how you're using Azure to reduce costs, improve resilience, or harden your security?

Choose Azure Advisor when you're looking for an analysis of your deployed resources. Azure Advisor analyzes the configuration and usage of your resources and provides suggestions on how to optimize for reliability, security, performance, costs, and operations based on experts' best practices.

Do you want to monitor Azure services or your usage of Azure?

If you want to keep tabs on Azure itself, especially the services and regions you depend on, you want to choose Azure Service Health. You can view the current status of the Azure services you rely on, upcoming planned outages, and services that will be sunset. You can set up alerts that help you stay on top of incidents and upcoming downtime without having to visit the dashboard regularly.

However, if you want to keep track of the performance or issues related to your specific VM or container instances, databases, your applications, and so on, you want to visit Azure Monitor and create reports and notifications to help you understand how your services are performing or diagnose issues related to your Azure usage.

Do you want to measure custom events alongside other usage metrics?

Choose Azure Monitor when you want to measure custom events alongside other collected telemetry data. Custom events, such as those added in the source code of your software applications, could help identify and diagnose why your application is behaving a certain way.

Do you need to set up alerts for outages or when autoscaling is about to deploy new instances?

Here again, you would use Azure Monitor to set up alerts for key events that are related to your specific resources.

F3: Use Azure Advisor

Tailwind Traders wants to optimize its cloud spend. Also, the organization is concerned about security breaches, because it stores customer data and historical purchase data in cloud-based databases. As the organization ramps up its cloud expertise, it wants to better understand its use of the cloud, better understand best practices, and pinpoint "easy wins" where it can tighten up its cloud spend and security practices.

Which service should you choose?

Apply the decision criteria you learned about in the preceding unit to find the right option.

First, in this scenario, does Tailwind Traders need to analyze its Azure usage for the sake of optimization? Yes. Tailwind Traders understands that it might be spending too much, is concerned about its security practices, and wants to have its cloud usage analyzed against industry best practices. Therefore, Azure Advisor is the perfect option for this scenario.

Although you might have found the right product option, let's continue evaluating the decision criteria for this scenario.

Second, in this scenario, does Tailwind Traders want to monitor the health of Azure services that affect all customers or the resources that are deployed on Azure? This scenario isn't concerned with operations. However, Azure Advisor does analyze and provide recommendations for achieving operational excellence.

Third, in this scenario, does Tailwind Traders want to measure custom events alongside other usage metrics? No, measuring custom events isn't mentioned as a requirement and isn't a consideration in this scenario.

Fourth, in this scenario, does Tailwind Traders want to set up alerts for outages or when autoscaling is about to deploy new instances? Again, this scenario isn't concerned with operations. However, Azure Advisor does analyze and provide recommendations for achieving operational excellence.

Azure Advisor is the right product option to help Tailwind Traders better understand and optimize both its cloud spend and its cloud security posture. This product might help the organization with other areas of its cloud usage as well.

F4: Use Azure Monitor

The Tailwind Traders e-commerce website is experiencing intermittent errors, and the team is unsure of the cause. Because of the nature of the errors, the team suspects that it's either a database or caching issue. What are the circumstances surrounding the errors? Does it happen only during peak usage times? What is the state of the team's Azure SQL instance? What is the state of its Redis caching server? How can it trace the issues to a root cause?

Which service should you choose?

As in the preceding unit, apply the decision criteria that you learned about earlier to find the right option.

First, in this scenario, does Tailwind Traders need an analysis of its Azure usage for the sake of optimization? No, optimization isn't the team's objective in this scenario, so Azure Advisor isn't a candidate.

Second, in this scenario, does Tailwind Traders want to monitor the health of Azure services that affect all customers or the resources deployed on Azure? Because this issue happens intermittently, it's unlikely to affect an entire Azure region or service. It's more likely that a logic issue exists somewhere in their e-commerce website code, or another issue is causing database failures or caching locks. In this scenario, the team could use Azure Monitor to pinpoint a specific user session and look at the performance of each service that's involved in the issue.

Third, in this scenario, does Tailwind Traders want to measure custom events alongside other usage metrics? Yes. Software developers can send additional information about the state of the web application via Application Insights to help locate the root cause of the issue. Application Insights relies on the Azure Monitor platform to store custom event information.

Fourth, in this scenario, does Tailwind Traders want to set up alerts for outages or for when autoscaling is about to deploy new instances? No, alerting isn't their objective in this scenario.

Azure Monitor is the best option for helping Tailwind Traders track this intermittent issue. The team can use a wealth of tools to help it gain insight into the application's performance at a high level and dive deep into specific issues.

F5: Use Azure Service Health

Tailwind Traders wants to operationalize its cloud environment. Specifically, its cloud operations team wants to let stakeholders know about upcoming planned downtime in advance. The team also wants its solution architects to be forewarned about any Microsoft plans to sunset services so it can rearchitect its software products accordingly.

When outages do happen, the team wants to quickly ascertain whether the issue is specific to their services or a service interruption that affects many Azure customers. The team also wants to provide key stakeholders with reports that explain how and why the incident occurred, and so on.

Which service should you choose?

Again, apply the decision criteria you learned about earlier to find the right product.

First, in this scenario, does Tailwind Traders need to analyze its Azure usage for the sake of optimization? No, so Azure Advisor isn't a candidate for this scenario.

Second, does Tailwind Traders want to monitor the health of Azure services that affect all customers or the resources deployed on Azure? In this scenario, the requirement is to stay abreast of upcoming planned downtime. Additionally, the team wants to capture official incident reports. For this reason, Azure Service Health is the strongest candidate to choose for this scenario.

Although it's likely that you would choose Azure Service Health, let's continue evaluating the remaining decision criteria.

Third, in this scenario, does Tailwind Traders want to measure custom events alongside other usage metrics? No, measuring custom events isn't mentioned as a requirement and isn't a consideration in this scenario.

Fourth, in this scenario, does Tailwind Traders want to set up alerts for outages or when autoscaling is about to deploy new instances? Setting up alerts for outages is a requirement in this scenario, but creating alerts for other events such as autoscaling are not in scope. Use Azure Service Health to set up alerts that are specific to Azure outages that affect all Azure customers.

Use Azure Monitor to set up alerts for outages and other events that affect only your specific resources.

In this scenario, Azure Service Health is the correct option to choose.

Summary

Our goal in this module was to help Tailwind Traders explore several monitoring service offerings from Azure to apply to a variety of business scenarios.

We identified three product options and their capabilities: Azure Advisor, Azure Monitor, and Azure Service Health. We analyzed decision criteria for choosing one option over another for certain scenarios. Then we applied those decision criteria to three different challenges faced by Tailwind Traders, helping them find the best service option for the scenario.

Without monitoring services, Tailwind Traders would spend more money on its cloud environment, be unsure about its cloud security posture, have difficulty pinpointing issues in its application logic, and be unable to plan ahead for outages or supply formal outage reports to stakeholders.

Azure monitoring services provide a comprehensive array of features to help improve your cloud operations.