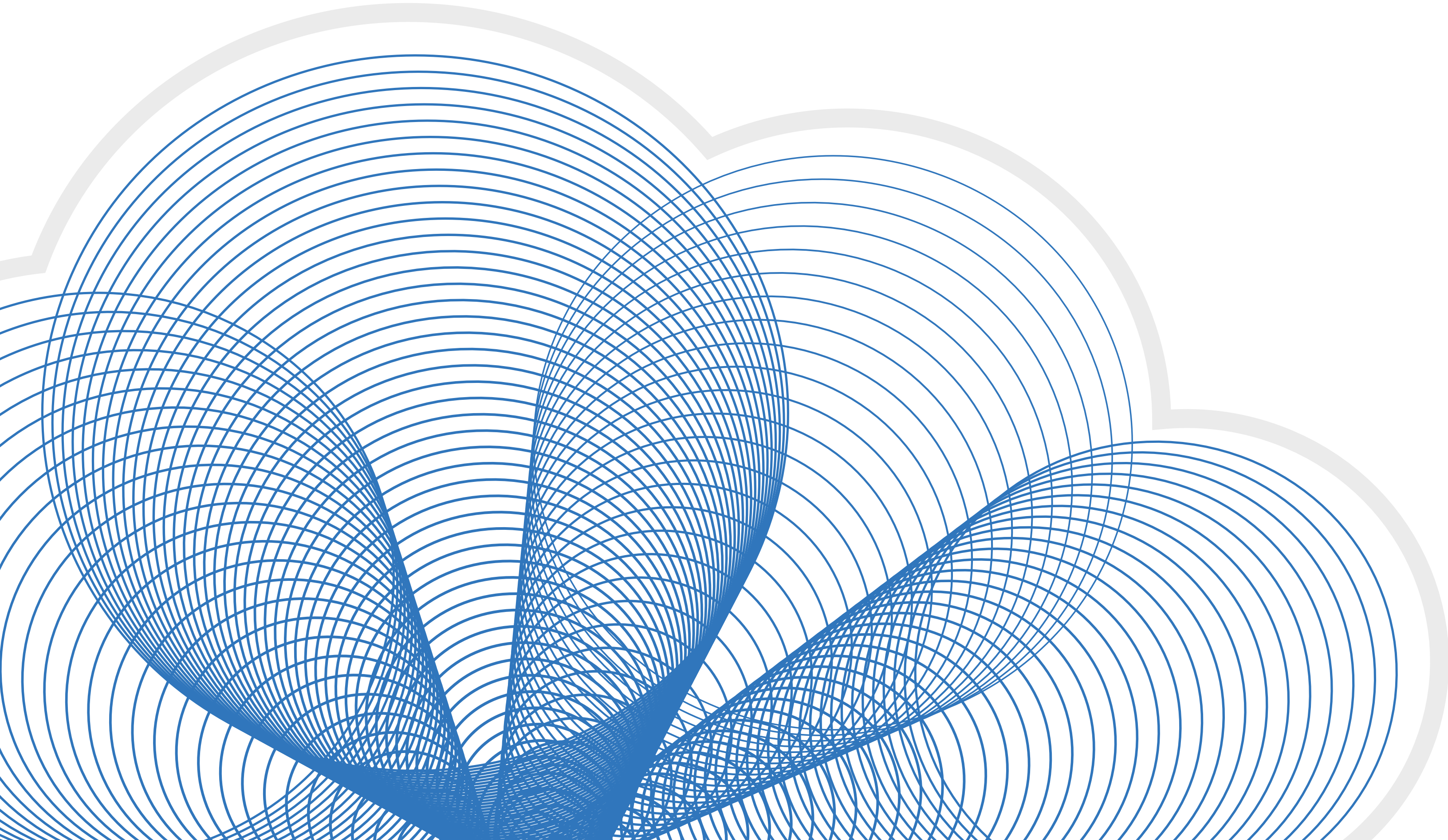
SECOND EDITION

E-Book series

Microsoft Azure

The Developer's Guide to Azure

New content on:
Cloud-native apps, dev tools,
DevOps, data & AI, and security



The Developer's Guide to Azure

01 /

Getting started with the Azure application platform

02 /

Developer tools and developer cloud

03 /

Cloud-native applications

04 /

Connecting your application with data

05 /

Adding intelligence to your application

06 /

Securing your application

07 /

Deploying your services and optimizing costs

08 /

Microsoft Azure in action

09 /

Summary and resources

The Developer's Guide to Azure

The Developer's Guide to Azure is designed for developers and architects who are embarking on their journey into Microsoft Azure. In this guide, you will learn how to get started and choose services that are appropriate for your scenarios.

From creating websites, databases, and desktop and mobile applications, to integrating the latest technologies into your application, Azure does the heavy lifting for you. Azure services are designed to work together so you can build complete solutions that last for the lifetime of your application.

Whether you are just starting out, writing code for fun, or a professional developer, developing with Azure puts the latest cloud technology and best-in-class developer tools at your fingertips. You can easily build for the cloud in your favorite language.

Azure provides a broad range of services that enable you to build rich applications and solutions so you can focus on apps, not infrastructure.

We're here to help

As you begin your cloud journey, you might run into some roadblocks. Fortunately, finding help is easy due to the popularity of Azure. We have compiled the following comprehensive list of helpful resources:

With the Azure support plans, you will get access to Azure technical support teams, guidance for cloud design, and assistance with migration planning. You can even acquire a support plan that guarantees a response from the technical support teams within 15 minutes.

The official Azure documentation and guides give you an overview of everything in Azure and provide deep insights through the documentation of each feature.

Service Level Agreements (SLAs) inform you about the uptime guarantees and downtime credit policies for Azure.

@Azure on Twitter is the account to follow for news and updates from the Azure team and community.

@AzureSupport on Twitter is operated by skilled Azure engineers who respond quickly to issues that you tweet to them.

Azure Community Support provides a place for discussions with the Azure community and contains answers to community questions.

Azure Advisor automatically makes personalized recommendations for your Azure resources, including what you need to do to be more secure, have higher availability, increase performance, and reduce costs.

Azure Service Health gives you a personalized view of the health of your Azure services.

Stack Overflow provides answers to Azure questions and includes many active posts by members of the Azure engineering teams.

Stay up to date with the latest releases and product announcements on Azure at Azure updates.

What can Azure do for you?

With Azure, you can get your work done faster, take your skills to the next level, and imagine and build tomorrow's applications today.

Azure provides a broad range of services that enable you to build rich applications and solutions for the cloud in your preferred language. Across our services we have 1000+ new capabilities, from AI to Kubernetes, to containers, databases, and more, to ensure we’re keeping you ahead of the curve. Azure provides end-to-end developer experience that helps you create reliable, global, and secure applications faster. You can build your applications using your favorite programming languages, open-source frameworks, and tools, and host them on Azure. There is a vast collection of sample applications available to help you get started and to inspire you with ideas for your projects.

The Developer's *Guide to Azure* will give you guidance and explain the benefits of hosting your application on Azure.

By the time you finish reading this guide, you will be able to:

- Automate your development process and be more productive.
- Spend less time doing repetitive tasks and more time creating reliable and secure app experiences your users will love, using best-in-class development tools and integrated DevOps, including Visual Studio, Visual Studio Code, GitHub, and Azure DevOps.
- Grow your expertise and skills using the resources that are presented in this guide.
- Build on the code, languages, tools, platforms, and frameworks you already know and use.
- Add new skills at your own pace and connect with a global community of your developer peers to advance your knowledge and career.
- Use different hosting services that Azure offers: Azure Virtual Machines, Azure App Service, and containers and featured services.
- Build modern applications with a cloud-native architecture.
- Connect your application to data and include AI capabilities.
- Turn your ideas into reality!

Let's begin our journey by learning how to get started with the Azure application platform.

01 /

Getting started with the Azure application platform

You have made the decision to build applications on the cloud, and now you can't wait to begin!

Getting started on Azure is incredibly easy. All you need to do is to sign up for an Azure free account.

With your Azure free account, you'll receive the following, and you won't be billed unless you choose to upgrade:

- 12 months of popular free services
- USD 200 credit to explore any Azure service for 30 days
- 25+ services, always free

Simply choose which programming languages, tools, platforms, and frameworks you want to use, and then start running your applications on Azure. In this section, we will cover the following topics:

- Where to host your application
- What to use, and when
- How to make your application more performant

Let's start by talking about where you can host your application on Azure.

Where to host your application

Azure offers services designed to provide what you need to deliver and scale every application. When you use Azure services to run your application, you get scalability, high availability, a fully managed platform, and database services. Azure also offers the following options for running your application.

PaaS

Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

Azure App Service

Azure App Service allows you to host your applications in a fully managed application platform loved by all developers. Azure App Service provides you with a collection of hosting and orchestrating services that share features and capabilities. For example, all services in App Service have the capability to secure an application using Azure Active Directory and can use custom domains.

As one of the most widely used Azure services, Web Apps can host your web applications or APIs. A web application is basically an abstraction of a web server, such as Internet Information Services (IIS) or Tomcat, that is used to host HTTP-driven applications. Web Apps can host applications written in .NET, Node.js, Python, Java, or Go, and you can use the available extensions to run even more languages.

If you have an existing application that you wish to migrate to Azure, there is a vast number of tooling options you can use, including Azure Migrate, Azure App Service migration assistant, PowerShell scripts for assessing and migrating .NET sites, and ASP.NET app containerization and migration to Azure App Service.

LEARN MORE

Official Azure developer documentation

QUICK START

Create an ASP.NET Core web app in Azure

Azure Spring Cloud

Azure Spring Cloud makes it easy to deploy Spring Boot microservice applications to Azure without any code changes. It is a fully managed service that lets you focus on building and running applications without the need to manage infrastructure. You can deploy your JARs or code, and Azure Spring Cloud will automatically wire your apps with the Spring service runtime. Azure Spring Cloud is jointly built, operated, and supported by Microsoft and VMware while still plugging into platform services to allow observability during operation.

You can develop and deliver Java apps using fully managed Spring Cloud components, including service discovery, configuration management, and distributed tracing. Azure Monitor provides deep insights into application dependencies and operational telemetry, providing aggregate metrics for a holistic view of how different services interact. Powerful visualization tools built into the Azure portal enable you to monitor average performance and error rates, along with rich detail into platform events that may be relevant to performance decreases or errors. This allows you to detect issues before they impact users and continuously improve your application performance.

A large portfolio of Spring starters provides native integration with Azure services such as Azure Cosmos DB, Azure Active Directory, and Azure Key Vault. With Spring starters, you can make your application more secure and connect it to various data sources, all out of the box.

Azure Spring Boot is a fully managed service with scalable global infrastructure. This allows you to focus on code with no need to manage infrastructure, and reduce downtime and deployment risk with turnkey support for blue-green deployments.

QUICK START

Deploy your first Azure Spring Cloud application

Deploy Spring microservices to Azure

Containers

While much more lightweight, containers are similar to virtual machines (VMs), and you can start and stop them in a few seconds. Containers also offer tremendous portability, which makes them ideal for developing an application locally on your machine and then hosting it in the cloud, in test, and later in production.

You can even run containers on-premises or in other clouds—the environment that you use on your development machine travels with your container, so your application always runs in the same ecosystem.

Scale and orchestrate containers with Azure Kubernetes Service

Azure Kubernetes Service (AKS) makes it simple to create, configure, and manage a cluster of VMs that are preconfigured to run containers, with support for both Linux and Windows containers.

This means that you can use your existing skills to manage and deploy applications that run in containers on Azure.

AKS reduces the complexity and operational overhead of managing a Kubernetes cluster by offloading much of that responsibility to Azure. As a hosted Kubernetes service, Azure handles critical tasks such as health monitoring and maintenance.

In addition, you pay only for the agent nodes within your clusters, not for the masters. As a managed Kubernetes service, AKS provides automated Kubernetes version upgrades and patching, easy cluster scaling, a self-healing hosted control plane (masters), and cost savings, since you only pay for running agent pool nodes.

With Azure handling the management of the nodes in your AKS cluster, there are many tasks that you don't have to perform manually, such as cluster upgrades. Because Azure handles these critical maintenance tasks for you, AKS does not provide direct access (such as with SSH) to the cluster.

QUICK START

Get started with Azure Kubernetes Service

Host containers in Azure App Service Web App for Containers

Web App for Containers helps you to easily deploy and run containerized web applications at scale.

Just pull container images from Docker Hub or a private Azure Container Registry, and Web App for Containers will deploy the containerized application with your preferred dependencies to production in seconds. The platform automatically takes care of operating system (OS) patching, capacity provisioning, and load balancing. You can run Docker containers on Linux and Windows using Web App for Containers.

With Web App for Containers, your applications are hosted using a predefined application stack based on a Docker container. The Docker containers, both Windows and Linux, can be deployed from any Docker registry, such as Docker Hub, Azure Container Registry, and GitHub.

Azure Container Registry

Once you have created a container image to run your application in, you can store that container in Azure Container Registry, which is a highly available and secure storage service, specifically built to store container images.

Azure Container Registry is great for storing your private Docker images.

You can also use Container Registry for your existing container development and deployment pipelines. Use the acr build command to build container images in Azure. You can either build on demand or fully automate builds with source code commit and base image update build triggers.

Virtual machines

Hosting your application in a VM in Azure Virtual Machines provides you with a lot of control over how you host your application. However, you're responsible for maintaining the environment, including patching the OS and keeping antivirus programs up to date.

You can use a VM to test the latest preview version of Visual Studio without getting your development machine "dirty."

Azure DevTest Labs and Azure Lab Services provide the ability to set up lab environments in Azure. These services enable developer teams to more easily manage developer VM resources and costs in the cloud.

Azure DevTest Labs allows you to set up an environment for your team. Users connect to VMs in the lab and use them for their day-to-day work and short-term projects. This enables the lab admin to analyze costs and usage, as well as setting policies to optimize the team's costs.

Azure Lab Services lets you create managed lab types. The service handles all the infrastructure management for the lab, from spinning up VMs to handling errors and scaling the infrastructure.

Azure Batch

If you need to run large-scale batch or high-performance computing (HPC) applications on VMs, you can use Azure Batch.

Batch creates and manages a collection of thousands of VMs, installs the applications you want to run, and schedules jobs on the VMs. You don't need to deploy and manage individual VMs or server clusters; Batch schedules, manages, and autoscales your jobs so you only use the VMs you need.

Batch is a free service, so you only pay for the underlying resources consumed, such as VMs, storage, and networking.

Batch is well suited to running parallel workloads at scale, such as financial risk models, media transcoding, VFX, 3D image rendering, engineering simulations, and many other compute-intensive applications. Use Batch to scale out an application or script that you already run on workstations or an on-premises cluster, or develop software as a service (SaaS) solutions that use Batch as a compute platform.

QUICK START

Get started on Azure Batch with these step-by-step tutorials

Azure Arc: Hybrid and multi-cloud environments

Azure offers world-class tools and cloud services that empower developers to build the applications of the future. However, your company probably has a hybrid environment and wants the same best-in-class development and management experiences for your applications in Azure and in your on-premises environments, edge locations, and even on other clouds.

Azure Arc offers simplified management, faster app development, and consistent Azure services anywhere. As a developer, you can architect, design, and deploy applications anywhere without sacrificing central visibility, security, and control. You can get Azure innovation and cloud benefits by deploying consistent Azure data, application, and machine learning services on any infrastructure.

Some of the key features of Azure Arc include:

- Work faster with Azure turnkey applications services such as App Service, Web Apps, Logic Apps, API Management, and Event Grid across clouds, datacenters, and at the edge.
- For your databases, deploy Azure Arc-enabled Azure SQL and PostgreSQL Hyperscale on any Kubernetes distribution and on any cloud.
- Use your favorite tools and existing DevOps practices anywhere and build iteratively.
- Reduce errors with consistent policy-driven application deployment and cluster operations at scale from source control and templates.
- Take advantage of elastic scale, consistent management, and cloud-style billing models anywhere.

QUICK START

Azure Arc Jumpstart

Azure services and products for app development

As shown in *Table 1.1*, Azure services and products for app development are designed to work together and are highly optimized for developer productivity:

Azure services and products for app development	Feature
Azure App Service	Makes building and managing web applications and APIs easy with a fully managed platform and features like autoscaling, deployment slots, and integrated web authentication
Azure Spring Cloud	Makes it easier to develop and deploy Spring Boot applications, with dynamic scaling, security patching, and out-of-the-box monitoring
Azure Functions	Makes event-driven programming simple, with state-of-the-art autoscaling, and triggers and bindings to integrate with other Azure services
Azure Logic Apps	Produces automated workflows for integrating apps, data, services, and backend systems with a library of more than 400 connectors
Azure Event Grid	Simplifies event-based applications with a single service for managing the routing of events from any source to any destination
Azure API Management	Provides a unified management experience and full observability across all internal and external APIs

Table 1.1: Azure application services along with features

Let's dive further into Azure App Service features in the next section.

Azure App Service features

Azure App Service is one of the key services in Azure that you can use to host your applications created with popular frameworks (.NET, .NET Core, Node.js, Java, PHP, Ruby, or Python) in containers, or running on any OS. Azure App Service also adds the power of Azure to your applications, including security, load balancing, autoscaling, and automated management.

Each of these services brings unique capabilities to the table, but they all share some common features.

Scaling

Azure App Service runs on **App Service plans**, which are abstractions from VMs. One or more VMs run your Azure App Service, but since Azure takes care of them, it is not necessary for you to know which ones. You can, however, scale the resources that run your Azure App Service.

You can either choose a higher pricing tier (ranging from free to premium) or increase the number of application instances that are running. It's even possible to have Azure App Service automatically scale the number of instances for you, based on a schedule or metric, such as CPU, memory, or HTTP queue length.

Deployment slots

After deploying a new version of your application to a deployment slot, you can test whether it works as expected and then move it into your production slot.

By setting up staging environments in Azure App Service, you can route a percentage of traffic from your production application to a **deployment slot**. For example, if you move 10 percent of your users to the new version of your application in the deployment slot, you can see whether the new features are functioning as expected and whether users are using them.

When you are satisfied with how the new version of your application is performing in the deployment slot, you can carry out a "swap," which exchanges the application in the deployment slot with that in your production slot. You can also swap from the development slot to a staging slot, and then to the production slot. Before doing this, the swap operation verifies that the new version of your website is warmed up and ready to go. When this has been confirmed, the swap operation switches the slots, and your users now see the new version of the application—with no downtime. You can also swap back and revert the deployment of the new version.

You use deployment slots within environments such as development, test, or production. You don't use deployment slots as environments because they all reside in the same App Service plan.

Those deployment slots should be separated for security, scaling, billing, and performance. You can swap deployment slots manually through the Azure CLI and Azure API Management. This allows DevOps tools to perform swap operations during a release.

Continuous deployment

To publish your application to App Service, you can use services such as Jenkins and Octopus Deploy. You can also use the continuous deployment (CD) feature in App Service.

The process does the following:

1. Retrieves the latest source code from the repository that you indicate
2. Builds the code according to a template that you pick (ASP.NET, Node.js, Java, and so on)
3. Deploys the application in a staging environment and load-tests it
4. Deploys the application to production after approval (you can indicate whether you want to use a deployment slot)

This makes it possible for you to create a build-test-release pipeline in App Service.

Connect to on-premises resources

You can connect external resources such as data stores to your app services. Depending on your requirements, you can connect to services on-premises through many mechanisms, such as:

- Azure Hybrid Connections
- Azure Virtual Networks
- Azure ExpressRoute

These resources do not need to be located in Azure—they can be anywhere, such as on-premises or in your own datacenter.

Custom domains and App Service certificates

When you spin up an application in App Service, it exposes a URL—for example, `https: //<your_custom_name>.azurewebsites. net`. Most likely, you will want to use your own custom domain, which you can do by mapping that domain name to App Service.

TUTORIAL

Map an existing custom DNS name to Azure App Service

Additionally, you can ensure that your application is served over HTTPS by using an SSL/TLS certificate. Just bring your own certificate, buy one directly from the Azure portal, or create a free App Service managed certificate. When you buy an SSL certificate from the Azure portal, you will receive an App Service certificate. You can configure this to be used by your custom domain bindings.

App Service managed certificates are free, are issued by DigiCert, and offer the option to secure your web applications hosted using a custom subdomain. They are also managed by App Service and are renewed automatically.

QUICK START

Purchase and configure an SSL certificate in this walk-through

App Service Environment

In a multi-tier web application, you often have a database or services used by your application in Web Apps. Ideally, you want these services to be exposed only to the application and not to the internet. Given that it provides the entry point for your users, however, the application itself is often internet-facing.

To isolate these support services from the internet, you can use Azure Virtual Network. This service wraps your support services and connects them to your application in Web Apps so that the support services are exposed only to the application, and not to the internet. This documentation describes the Azure App Service VNet integration feature and how to set it up with apps in App Service.

Sometimes, you want even more control. Maybe you want your application to be wrapped in a virtual network in order to control access to it. Perhaps you want it to be called by another application in Web Apps and be part of your back end. For this scenario, you can use an Azure App Service Environment. This affords you a very high scale and gives you control over isolation and network access.

Snapshot Debugger for .NET

Debugging applications can be difficult, especially if the application is running in production. With the Application Insights Snapshot Debugger feature of Azure Monitor, you can take a snapshot of your in-production applications when code that you are interested in executes.

The Snapshot Debugger lets you see exactly what went wrong without impacting the traffic of your production application. The Snapshot Debugger can help you to dramatically reduce the time it takes to resolve issues that occur in production environments. Additionally, you can use Visual Studio to set snap points to debug step by step. You can view the results in the Azure portal or within Visual Studio.

Automatic OS and .NET Framework patching

Because you are using a fully managed platform, you don't manage your own infrastructure at all, but you do benefit from automatic OS and framework patching.

Azure Functions

With Azure Functions, you can write the code you need for a solution without worrying about building a full application or the infrastructure to run it. A function is a unit of code logic that's triggered by an HTTP request or an event in another Azure service, or is based on a schedule.

Some of the key features of Azure Functions include:

Feature	Description
Automated and flexible scaling	Keep the focus on adding value instead of managing infrastructure.
Choice of programming language	Write functions using your choice of C#, Java, JavaScript, Python, or PowerShell.
End-to-end development experience	Choose from building and debugging to deploying and monitoring with integrated tools and built-in DevOps capabilities.
Simplified integration	Easily integrate with Azure services and SaaS offerings.
Pay-per-use pricing	Pay only for the time spent running your code, with a Consumption hosting plan.

Table 1.2: Capabilities of Azure Functions

Input and output bindings connect your function code to other services, including Azure Storage, Azure Cosmos DB, Azure Service Bus, and even third-party services such as Twilio and SendGrid. Using Azure Functions, you can build small pieces of functionality quickly and host them in an elastic environment that automatically manages scaling.

With Azure Functions, it is possible to pay only for functions that run, rather than having to keep compute instances running all month. This is also called serverless because it only requires you to create your application—you don't have to deal with any servers or even the scaling of servers. You can write Azure Functions in .NET, JavaScript, Java, and a growing list of languages.

An application that uses Azure Functions activates a function every time a new image file is uploaded to Azure Blob storage. The function then resizes the image and writes it to another Blob storage account. Data from the blob that triggered the function is passed into the function as the myBlob parameter, which includes the blob URL. Use the outputBlob output binding parameter to specify which blob to write the result to. There's no need to write the plumbing for connecting to Blob storage—you just configure it.

QUICK START

Create your first Azure function using the Azure portal

Cold start/warm start

Cold start is a term used to describe the behavior of an application and its tendency to take longer to start up after it has been inactive for a period of time. For functions running in Consumption and App Service plan pricing models, when a function app has been inactive for a period of time, it will automatically scale to zero instances. When new events come in, a new instance needs to be specialized with your application running on it. Specializing a new instance may take some time (latency) before the first event can be handled.

To eliminate cold start latency, you can use the Azure Functions Premium plan and configure the number of pre-warmed instances. The Azure Functions application will maintain the specified number of pre-warmed instances to more readily scale up to handle new events.

Azure Logic Apps

You can orchestrate business logic with Logic Apps by automating a business process or integrating with SaaS applications.

Just like in Azure Functions, Logic Apps can be activated by an outside source, for instance, a new message. Weaving together API calls to connectors, you can create a (possibly complex) workflow that can involve resources both in the cloud and on-premises.

Logic Apps has many connectors to APIs that can connect to Azure SQL Database, Salesforce, SAP, and so on.

You can also expose your own APIs or functions as connectors to use in a logic app, making it possible for you to easily perform actions against external systems in your workflow or have your logic app activated by one of them.

The following is an example of a workflow in Logic Apps:

1. The logic app is activated when an email containing a shipping order arrives in Microsoft 365.
2. Using the data in the email, the logic app checks on the availability of the ordered item in SQL Server.
3. Using Twilio, the logic app sends a text message to the customer's phone indicating that the order was received and that the item has been shipped.

QUICK START

Get started with Azure Logic Apps

Just like Azure Functions, Logic Apps is serverless and scaled automatically, and you pay for it only when it is running.

Power Apps

Power Apps is a suite of apps, services, and connectors, as well as a data platform, that provides a rapid development environment to build custom apps for your business needs. Using Power Apps, you can quickly build custom business apps that connect to your data stored either in the underlying data platform (Microsoft Dataverse) or in various online and on-premises data sources (such as SharePoint, Microsoft 365, Dynamics 365, and SQL Server).

Apps built using Power Apps provide rich business logic and workflow capabilities to transform your manual business operations into digital, automated processes. Furthermore, apps built using Power Apps have a responsive design and can run seamlessly in a browser and on mobile devices (phone or tablet).

Power Apps makes the business app-building experience accessible to everyone by empowering users to create feature-rich, custom business apps without writing code. For professional developers, Power Apps also provides an extensible platform that allows developers to programmatically interact with data and metadata, apply business logic, create custom connectors, and integrate with external data.

QUICK START

Official Power Apps documentation

Power Automate

Microsoft Power Automate is a SaaS offering for automating workflows across the growing number of applications and SaaS services that business users rely on. While Logic Apps is aimed more toward a developer audience, Microsoft Power Automate is targeted toward business users, administrators, and office workers.

Microsoft Power Automate offers an easier path toward simple integration workflows. Logic Apps offers the ability to extend Power Automate with more advanced workflow capabilities. An example of additional capabilities that Logic Apps offers is the ability to run inline code within the workflow.

QUICK START

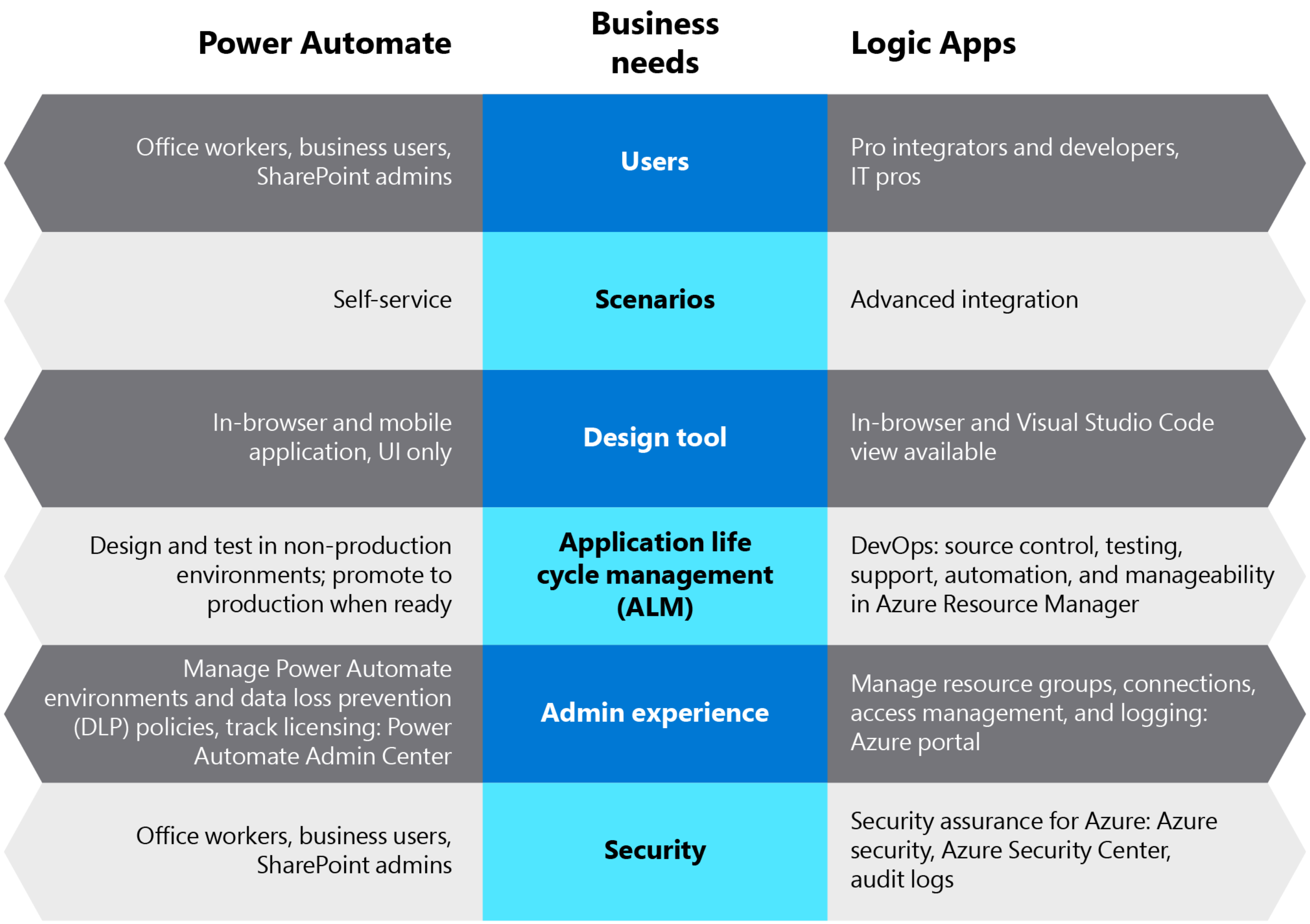
Learn more about Power Automate

Logic Apps versus Power Automate

Both Microsoft Power Automate and Logic Apps offer designer-first integration services that can create workflows.

Here's a side-by-side comparison to help determine when to use Power Automate or Logic Apps for a particular integration scenario:

Both services can integrate with various SaaS and enterprise applications; however, they are each targeted at different users.



Power Automate	Business needs	Logic Apps
Office workers, business users, SharePoint admins	**Users**	Pro integrators and developers, IT pros
Self-service	**Scenarios**	Advanced integration
In-browser and mobile application, UI only	**Design tool**	In-browser and Visual Studio Code view available
Design and test in non-production environments; promote to production when ready	**Application life cycle management (ALM)**	DevOps: source control, testing, support, automation, and manageability in Azure Resource Manager
Manage Power Automate environments and data loss prevention (DLP) policies, track licensing: Power Automate Admin Center	**Admin experience**	Manage resource groups, connections, access management, and logging: Azure portal
Office workers, business users, SharePoint admins	**Security**	Security assurance for Azure: Azure security, Azure Security Center, audit logs

Figure 1.1: How to choose between Power Automate and Logic Apps

API Management

API Management allows you to create API gateways for existing back-end services in a consistent manner.

With API Management, you can publish APIs to external, partner, and internal developers to unlock the potential of their data and services. Essentially, you can use Azure API Management to take any back end and launch a full-fledged API program based on it.

Some of the common uses of API Management include:

- **Securing mobile infrastructure** with API access keys, preventing DOS attacks by using throttling, or using advanced security policies like JWT token validation.
- **Enabling ISV partner ecosystems** by offering fast partner onboarding through the developer portal and building an API facade to decouple from internal implementations that are not ripe for partner consumption.
- **Running an internal API program** by offering a centralized location for the organization to communicate about the availability and latest changes to APIs, gating access based on organizational accounts—all based on a secured channel between the API gateway and the back end.

FURTHER LEARNING

Azure APIs + Microservices Guide

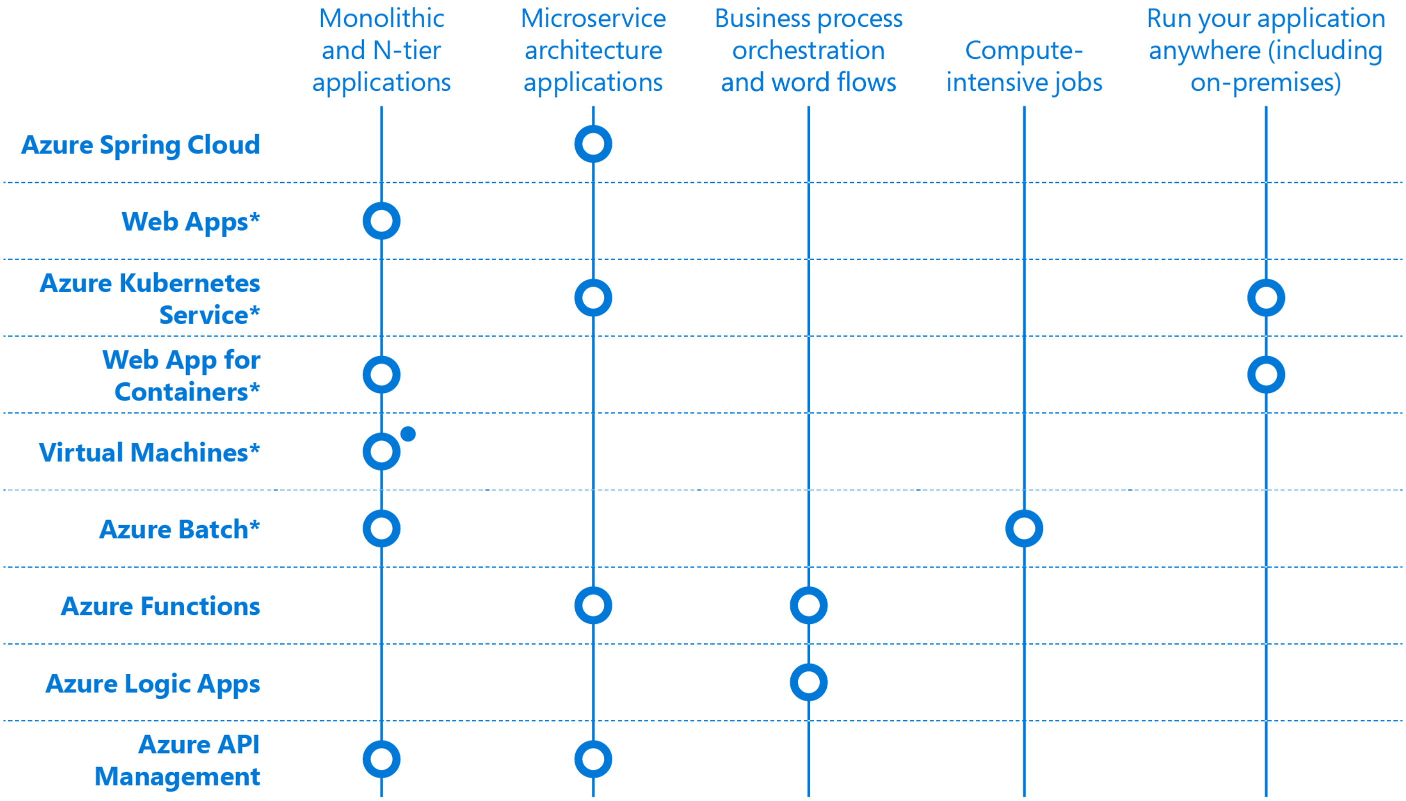
Azure API Design E-Book

API Management in a Hybrid and Multi-Cloud World

What to use, and when

Some of the services that run your application in Azure can work together in a solution, while others are more suited to different purposes.

While this can make it difficult to pick the right services, *Figure 1.2* will help you to identify which services in Azure are right for your situation:



	Monolithic and N-tier applications	Microservice architecture applications	Business process orchestration and word flows	Compute-intensive jobs	Run your application anywhere (including on-premises)
Azure Spring Cloud		○			
Web Apps*	○				
Azure Kubernetes Service*		○			○
Web App for Containers*	○				○
Virtual Machines*	○●				
Azure Batch*	○			○	
Azure Functions		○	○		
Azure Logic Apps			○		
Azure API Management	○	○			

* Services with an asterisk have a free tier that you can use to get started at no cost.

● For lifting and shifting existing applications to Azure.

Figure 1.2: A quick summary on choosing an Azure service in various scenarios

Using events and messages in your application

Modern, globally distributed applications often have to deal with large amounts of messages coming in, so they need to be designed with decoupling and scaling in mind. Azure provides several services to help with event ingestion and analysis as well as messaging patterns. These services are also vital for creating intelligent applications that make use of AI.

Service Bus

The core of messaging in Azure is [Service Bus](). Service Bus encompasses a collection of services that you use for messaging patterns. The most important services are Azure Service Bus queues and topics.

Service Bus queues

Service Bus queues decouple systems from one another. For example, a web application receives orders from customers and needs to invoke a web service to process the orders. The web service will take too long to process the orders, perhaps up to five minutes.

One way to solve this problem is to use a queue to decouple the web application from the web service. The web application receives the order and writes it in a message on a Service Bus queue. Then, the web application informs the user that the order is being processed. The web service takes messages from the queue, one by one, and processes them. When the web service has processed an order, it sends an email notification to the customer that the item has been ordered.

By decoupling the systems, the web application can work at a different speed from the web service, and both can be scaled individually to the application's needs.

DOC

`Get started with Service Bus queues`

Service Bus topics

Just like Service Bus queues, Service Bus topics are a form of application decoupling.

Here are the differences between them:

- With a queue, multiple applications write messages to the queue, but only one application at a time can process a message.
- With a topic, multiple applications write messages to the topic, and multiple applications can process a message at the same time.

Applications can create a subscription on the topic that indicates what type of messages they're interested in. Just like queues, topics have features, including duplicate detection and a dead letter subqueue, to which messages are moved when they fail to be processed correctly.

DOC

`Get started with Service Bus topics`

Event Hubs

Event Hubs can help enterprises capture massive amounts of data to analyze or transform and move for later use.

Event Hubs is designed for massive data ingestion. It effortlessly handles millions of messages per second. It retains messages for up to 7 days or indefinitely by writing messages to a data store using the Event Hubs Capture feature.

You can use Event Hubs to filter data with queries as it comes in and output it to a data store such as Azure Cosmos DB. You can even replay messages.

QUICK START

Send events to and receive events from Azure Event Hubs

Event Grid

Event Grid offers a different type of messaging—a fully managed publish and subscribe service that hooks into almost every service in Azure as well as into custom publishers and subscribers.

This is different from working with Service Bus queues and topics, for which you'd need to poll the queue or topic for new messages. Event Grid automatically pushes messages to subscribers, making it a real-time, reactive event service.

Services inside and outside of Azure publish events when a new blob is added, for example, or when a new user is added to an Azure subscription. Event Grid detects these events and makes them available to event handlers and services that subscribe to the events, as shown in *Figure 1.3*:

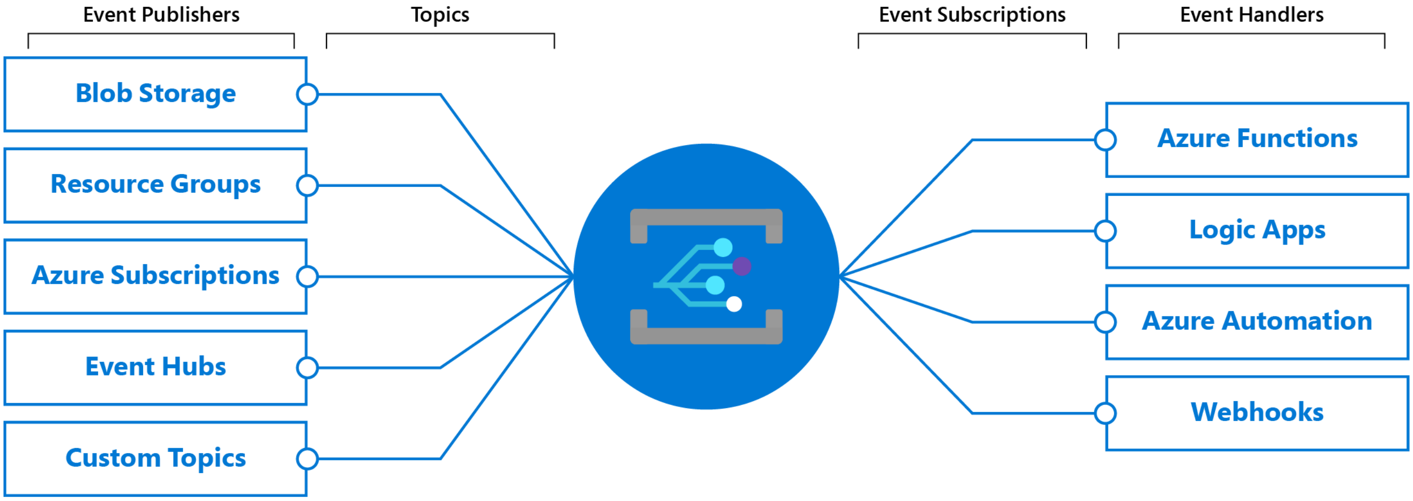


Figure 1.3: Flow of events from event publishers to event handlers

Event handlers could be Functions or Logic Apps, which can then act on the data in the event.

Another important aspect of Event Grid is that it is serverless. This means that, like Logic Apps and Functions, Event Grid scales automatically and does not need an instance of it to be deployed. You just configure and use it, and pay only when it's used.

You can use Event Grid if you want an email notification every time someone is added to, or removed from, your mailing list in Mailchimp. Event Grid is used to activate an application in Logic Apps and configured to listen to changes to the Mailchimp mailing list. Event Grid then signals to Logic Apps to send an email containing the name of the person who has been added or deleted and the action that was performed.

TUTORIAL

Monitor virtual machine changes with Event Grid and Logic Apps

Azure SignalR Service

You can use Azure SignalR Service to simplify the process of adding real-time web functionality to applications over HTTP that enables services to push content updates to connected clients. The service is based on ASP.NET Core SignalR and is offered as a standalone, fully managed service in Azure.

SignalR can update connected applications in real time over HTTP without the need for the applications to poll for updates or submit new HTTP requests. This enables you to create seamless web experiences that update information on the fly. For example, an auction application might use SignalR to refresh the latest bid as soon as it happens, without completely refreshing the page or constantly polling for information.

Hosting a SignalR server yourself is not a simple task, and it can be difficult to scale and secure properly. When you use the fully managed Azure SignalR Service, setup is easy, and security, availability, performance, and scalability are all managed for you.

QUICK START

Create a chat room with SignalR

Azure messaging services

Azure provides a myriad of options to perform messaging and to decouple applications. Which one should you use, and when? *Figure 1.4* summarizes the differences to help you choose.

	Event ingestion	Device management	Messaging	Multiple consumers	Multiple senders	Use for decoupling	Use for publish/ subscribe	Max message size
Service Bus queues*			○		○	○		1 MB
Service Bus topics*			○	○	○	○		1 MB
Event Hubs*	○		○	○	○	○		256 KB
Event Grid*	○		○	○	○		○	64 KB
SignalR Service*			○	○	○			64 KB

* Services with an asterisk have a free tier that you can use to get started at no cost.

Figure 1.4: A summary of Azure services for events and messages: what to use and when

Azure Communication Services

Azure Communication Services is a platform with rich communication APIs for deploying voice, video, chat, or SMS capabilities within your applications across any device, on any platform, using the same reliable and secure infrastructure that powers Microsoft Teams. You can add communication features to your applications without being an expert in communication technologies such as media encoding and real-time networking. Azure Communication Services supports various communication formats:

- Voice and video calling
- Rich text chat
- SMS

Some of the features of Azure Communication Services include:

- Deliver video, voice, chat, SMS, and telephony experiences anywhere your customers are (across your applications, websites, and mobile platforms).
- Use a reliable global platform trusted by millions of people daily.
- Reach more customers without compromising security using a secure and compliant cloud.
- Connect people across web and mobile apps. Add communication workflows to apps with flexible SDKs and APIs for common platforms and programming languages, including iOS, Android, Web, .NET, and JavaScript.

QUICK START

Add chat to your app

Add 1:1 video calling to your app

Send an SMS message

Making your application more performant

After your application is up and running in Azure, you want it to be as performant as possible. Azure provides a range of services that can help you with that.

Azure Traffic Manager

Many modern applications have users all over the world. Providing a performant experience for everyone is challenging, to say the least. The most obvious problem you need to deal with is latency, the time it takes for a signal or a request to travel to a user. The farther away users are from your application, the more latency they experience.

Azure Traffic Manager scales across regions, helping to reduce latency and provide users with a performant experience, regardless of where they are.

Traffic Manager is an intelligent routing mechanism that you put in front of your Web Apps applications. Web Apps acts as an endpoint, which Traffic Manager monitors for health and performance.

When users access your application, Traffic Manager routes them to the Web Apps application in their proximity that is most performant.

Including Traffic Manager in your architecture is a great way to improve the performance of your application.

Azure Front Door

Your users may be spread across the globe and, at times, they may be traveling. This can make it difficult to ensure that they have a performant experience and that your application is available and secure, regardless of location.

Azure Front Door can help.

This service can route traffic from users to the most performant application endpoint for them to improve performance. Front Door can route to available endpoints, while avoiding endpoints that are down.

Traffic Manager does this as well, but in a different manner to Front Door. Front Door works at OSI layer 7 or the HTTP/HTTPS layer, while Traffic Manager works with DNS. In other words, Front Door works on the application level, and Traffic Manager works on the network level. This is a fundamental difference that determines the capabilities of the services.

Because of this difference, Front Door does a lot more than route users to available and performant endpoints.

Front Door allows you to author custom web application firewall (WAF) rules for access control to protect your HTTP/HTTPS workload from exploitation based on client IP addresses, country codes, and HTTP parameters.

If you need help choosing between Front Door and Traffic Manager, consider the following guidance:

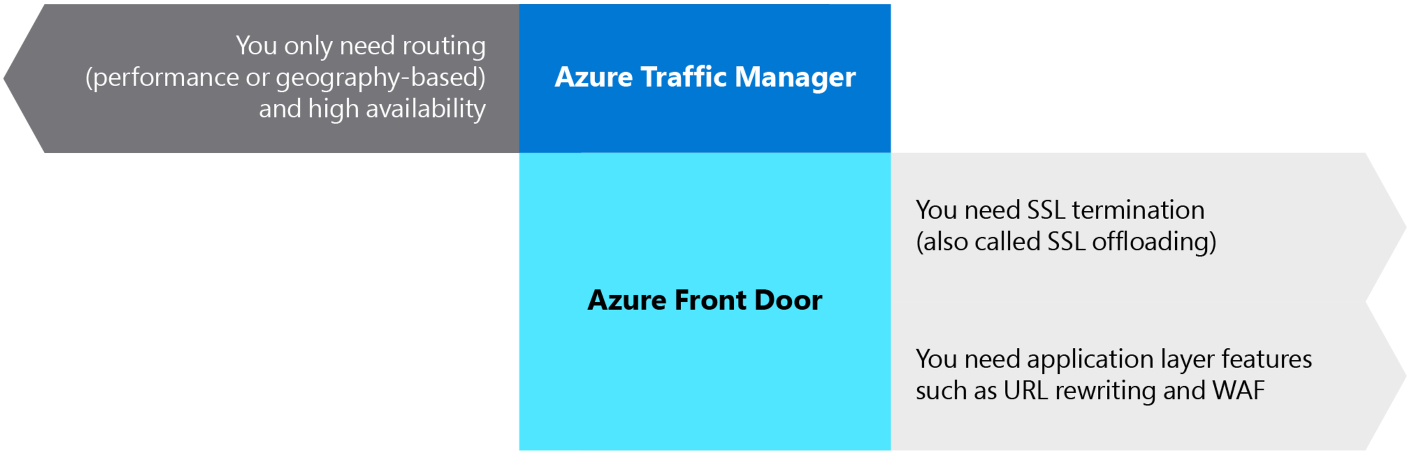

You only need routing (performance or geography-based) and high availability
Azure Traffic Manager
Azure Front Door
You need SSL termination (also called SSL offloading)
You need application layer features such as URL rewriting and WAF

Figure 1.5: Choosing between Front Door and Traffic Manager

Other capabilities of Front Door include:

- **URL-based routing**

  This allows you to route requests for different URLs to different back-end pools (applications that receive traffic, such as Web Apps). For instance, `http: //www.contoso. com/users/*` goes to one pool, and `http: //www.contoso. com/ products/*` goes to another.

- **URL rewrite**

  This enables you to customize the URL that you pass on to the back-end pool.

- **SSL termination**

  With this, you can secure your traffic from end to end, from the browser to the application in the back-end pool.

- **Session affinity**

  When you want users to be sent to the same endpoint every time, session affinity is useful. This is important in cases where the session state is saved locally on the back end for a user session.

Additionally, Front Door enables you to create rate-limiting rules to battle malicious bot traffic. These are just some of the unique capabilities of Front Door.

Azure Content Delivery Network

One of the Azure services that can help you to make your application faster is Azure Content Delivery Network.

You upload your static files—videos, images, JavaScript, CSS, and even static HTML files—to a data store, such as Azure Blob storage, and then couple Content Delivery Network to that.

Content Delivery Network will then take those static files and replicate them to hundreds of points of presence (PoP) all over the world. All you need to do in your application is change the reference to the static files to a different URL.

For example, the previous reference might have `been ~/images/image.png`, and it would now be `https: //example. azureedge.com/image.png`.

Not only is this easy to do, but it also improves the performance of your application in the following ways:

- It offloads serving content from your application. Since it is now served by Content Delivery Network, it frees up processing cycles for your application.
- It brings static content physically closer to your users by distributing it to PoPs all over the world.

You can benefit from Content Delivery Network in web applications as well as in mobile and desktop applications. One way to use Content Delivery Network is to serve videos for a mobile application. Since videos can be large, you don't want to store them on the mobile device—and neither do your users. Using Content Delivery Network, the videos are served from the PoP. Since it is close to the user, this also improves performance.

QUICK START

`Get started with Azure Content Delivery Network`

In the next chapter, you will look at the Microsoft developer ecosystem, including the Visual Studio family of IDEs, GitHub, and Azure DevOps.

02 /

Developer tools and developer cloud

The most comprehensive developer tools and cloud

The Microsoft developer ecosystem, including the Visual Studio family of IDEs, coupled with the power of DevOps platforms—GitHub, Azure DevOps, and cloud services in Microsoft Azure—provides the most comprehensive end-to-end developer experience.

> *"Microsoft has the world's most beloved developer tools with Visual Studio, and with GitHub hosts the developer community where the world comes together to build software. Developers can use their favorite languages, open-source frameworks, and tools to code and deploy code to the cloud from anywhere, collaborating in a secure way, and integrating different components in no time."*
>
> **– Scott Guthrie**

Let's take a look at the key tooling and platforms that comprise Microsoft's developer ecosystem:

- **Visual Studio and Visual Studio Code**: World-class IDEs built for everyone and runnable anywhere.
- **GitHub and Azure DevOps**: Community-based, open-source, and enterprise-grade work item tracking, CI/CD pipelines, artifact storage, and more.
- **Microsoft Azure**: Azure is an excellent cloud provider, offering the ability to host .NET, Java, JavaScript/Node.js, Python, and more.

Now it's time to explore the Visual Studio family of tools, GitHub and Azure DevOps, and platform integration services in more detail.

Visual Studio and Visual Studio Code

With Visual Studio and Visual Studio Code, you can develop your application wherever and however you like. From game development to web applications and even Linux apps written in C++ or .NET, Visual Studio continues to be the IDE of choice for developers working on Windows. Visual Studio Code is one of the most popular editors for developers working on any operating system and building apps with any programming language or framework.

Visual Studio

Visual Studio is an integrated, full-featured development environment providing teams and individuals with the end-to-end toolset for development, testing, debugging, and deployment. Visual Studio offers a set of innovative and intelligent features that allows individual developers and entire teams alike to be more productive. IntelliSense and IntelliCode bring smarter code completion into the IDE and simplify repetitive tasks like refactoring. Diagnostics and debugging features, such as Snapshot Debugging and live integration with Azure Application Insights, offer full visibility into your app's execution history and debug state wherever it runs. Built-in Git and GitHub integration allows for seamless collaboration: developers can create and clone repos, manage branches, and resolve merge conflicts right inside the IDE. Visual Studio Live Share lets developers collaborate in their IDE in real time by setting up a shared network session, allowing participants to edit and work together as if they were sitting side by side.

Whether developers want to deploy their app locally, to their own servers, or to Azure, Visual Studio makes the process consistent and easy to set up. Deployment to Azure can be configured from within the IDE, regardless of whether it's published directly, via FTP, through a CI/CD pipeline, to Azure PaaS services, or through a Docker/Kubernetes container configuration.

Visual Studio for Mac

Visual Studio for Mac is a full-featured IDE for developers on macOS who build applications, games, and services for iOS, Android, macOS, the cloud, and the web. Modern technologies and frameworks like .NET, Unity, C#, and F# enable you to rapidly innovate with a world-class IDE. Visual Studio for Mac is designed natively for the Mac, while being powered by the same code editor, compiler, IntelliSense code completion, and refactoring experience that you know and love from Visual Studio on Windows.

Visual Studio 2022

Visual Studio's upcoming release, Visual Studio 2022 coming in late 2021, will be full of performance enhancements and a wide range of features to boost personal and team productivity. With 64-bit Visual Studio 2022, developers can scale to large and complex solutions without running out of memory. Innovative features such as Hot Reload for .NET and C++ apps, Live

Preview for XAML apps, Web Live Preview for ASP.NET apps, and IntelliCode whole-line completion will enable developers to be productive in their app development lifecycle. Improved Git and GitHub tooling and Live Share with integrated chat will enable developers to collaborate seamlessly.

Visual Studio 2022 will also come with the latest innovative tools for modern app development. Visual Studio 2022 will have full support for .NET 6 and its unified framework for web, client, and mobile apps for both Windows and Mac developers. This includes the .NET Multi-platform App UI (.NET MAUI) for cross-platform client apps on Windows, Android, macOS, and iOS. Developers will be able to use ASP.NET Blazor web technologies to write desktop apps via .NET MAUI. For C++ developers, Visual Studio 2022 will include robust support for C++ workloads with new productivity features, C++20 tooling, and IntelliSense. We're also integrating support for CMake, Linux, and Windows Subsystem for Linux (WSL) to make it easier for you to create, edit, build, and debug cross-platform apps.

DOCS

Visual Studio

Visual Studio 2022 Roadmap

Visual Studio 2019 for Mac

Visual Studio Code

Visual Studio Code is a cross-platform code editor with binaries for Windows, macOS, and Linux. Many of the features that make Visual Studio great can also be found in Visual Studio Code, from classic IntelliSense to newer features like IntelliCode and Live Share.

Leveraging an ecosystem of 30,000 (and growing) first-party and third-party extensions and themes, Visual Studio Code can be customized to the needs of each developer and supports working with virtually every programming language and framework, and tools like package managers. Extensions for Azure allow building, deploying, and managing applications using a variety of Azure services with a few clicks, while collaboration is made simpler with extensions for GitHub pull requests and issues. You can even develop your own custom extensions that satisfy any unique needs you or your team may have. Visual Studio Code also allows developers to work with remote hosts and machines, via extensions like Remote-SSH and Remote-Containers (for Docker containers), as well as GitHub Codespaces.

Visual Studio Code is completely free to use on any platform and is based on an open-source codebase.

GETTING STARTED

Visual Studio Code

Download Visual Studio Code

Build, release, and deploy with GitHub + Azure DevOps

The Azure DevOps product management and engineering teams at Microsoft joined GitHub under a single leadership team to deliver on coordinated roadmaps. This renewed focus on both offerings ensures that GitHub will continue to rise as the platform of choice for code management and CI/CD mechanics, while ensuring Azure DevOps continues to deliver the mature software development lifecycle features to its users.

The GitHub and Azure DevOps team recognizes that one solution doesn't always fit all, which is why Microsoft allows customers to adopt hybrid GitHub and Azure DevOps environments. Two of the most well-known hybrid solutions are:

- Azure Boards-GitHub integration
- Azure Pipelines-GitHub integration

DOCS

Azure DevOps Public Roadmap

GitHub Public Roadmap

Planning and tracking

Both GitHub and Azure offer management of the product roadmap and backlogs as a part of agile methodologies.

GitHub offers issues to track ideas, enhancements, tasks, or bugs. GitHub also offers project management with tagging, milestones, and Kanban boards to drive a project forward.

If a more structured process is desired, Azure Boards can be integrated into GitHub or used with other Azure DevOps services. Azure Boards supports agile methodologies, including Agile, Scrum, and Kanban. Azure Boards enables you to track work with Kanban boards, backlogs, team dashboards, and custom reporting. For roadmap planning, Delivery Plans can be added to Azure DevOps from the Visual Studio Marketplace, providing everything a team needs to track a feature from ideation to production.

GitHub Boards

Project boards on GitHub can help you organize and prioritize your work using a Kanban approach to work management. These boards are flexible and can be used to track specific feature work, software roadmaps, and even release checklists.

The primary components of project boards include issues, pull requests, and notes. These components are all visualized on the board as cards within one or more columns. Cards can contain relevant metadata for issues and pull requests, such as status, assignees, and who opened them. Notes can be used to create task reminders, reference specific issues or pull requests, or any other information that may be relevant to the board.

Project boards come in three different configurations:

- User-owned project boards, related to personal repos
- Organization-wide project boards, which can contain issues and pull requests on up to 25 repositories within an organization
- Repository project boards, which are scoped to issues and pull requests within a single repository

Project boards can also be automated, allowing for cards to move from one status to another. Trigger-based workflows enable cards to assume specific statuses such as To do, In progress, or Done. Typically the triggers consist of simple events, such as issues being created, new pull requests being opened, issues being closed, or pull requests being merged.

Azure Boards

Planning your work and tracking your progress are important tasks—and Azure Boards can help you complete them.

In Azure Boards, you can create a complete backlog of work items (such as user stories) and plan them in sprints so your team can work iteratively to finish the tasks.

The whole planning system is optimized for working in an agile way. It even includes Kanban boards for managing your work.

Everything can be customized to work in the best possible way for your teams, whether using Scrum, another agile method, or the Capability Maturity Model Integration (CMMI) process. You can create and manage tasks, features, user stories, bugs, requirements, issues, change requests, and more.

Repository and dev workflows

GitHub enables developers to share code and packages through its core repository capabilities, GitHub Packages, and npm. Azure Repos provides both Git and Team Foundation Version Control (centralized) repositories and Azure Artifacts for packages. Both GitHub and Azure DevOps integrate with Azure Container Registry, which provides a fully managed and optionally geo-replicated instance for Docker images and Helm Charts.

GitHub Repos

Repos are the heart of GitHub. Using standard Git format, you can manage your project's files, as well as discussing and managing your project's work. You can restrict who is allowed to view or contribute to a repository by changing its visibility level. You may select public (default) or private, which keeps access restricted to users you wish to have access.

If you are using the free version of GitHub, you are able to use unlimited public repositories with access to a full feature set, or unlimited private repositories with a limited feature set. The limitations are in the form of scoped access to GitHub Community support, Dependabot® alerts, storage limits, and other constraints. Premium tiers offer the ability to increase minutes/month on GitHub Actions and storage on GitHub Packages storage along with advanced features around pull request management, protected branches, and repository insight graphs. Please see GitHub's products listing for more information.

Azure Repos

Azure Repos uses standard Git. This means that you can use it with any Git tool and IDE, including Visual Studio and Visual Studio Code, as well as Git for Windows, macOS, Linux, Eclipse, and IntelliJ. When you follow the Git workflow, you usually begin by creating your own branch of the code to, for instance, add a feature. Once you finish this, you commit your code to create a pull request for that branch and submit it to the server. Users can see, review, test, and discuss this pull request. Once it's good enough to be pulled into the main branch, the request is accepted, and your development branch can be deleted.

With Azure Repos, you have a rich toolset to support the Git workflow. You can link work items, such as user stories or bugs, to pull requests so you know what each change is about. You can have discussions about committed code and even comment on changes within code. Azure Repos also enables voting on changes in code, so a change only gets accepted once everyone on the team agrees to it.

Azure Repos offers unlimited private Git repositories.

GitHub Packages

GitHub Packages is a software package hosting service that allows you to host your own packages privately or publicly. These packages can then be used in your projects or made available to other users.

Packages supports many commonly used package managers, such as npm, RubyGems, Apache Maven, Gradle, Docker, and NuGet. In addition, GitHub offers container registry support for hosting Docker or OCI images. Access tokens are required to publish, install, or delete packages, keeping the lifecycle management of your packages secure.

GitHub Packages also offers automation support. You can integrate Packages with GitHub Actions, GitHub APIs, and webhooks to create DevOps workflows including your code, continuous integration (CI), and deployment all in one interface.

Azure Artifacts

You can host all sorts of packages on Azure Artifacts, including NuGet, npm, Maven, Python, and universal packages. You can even use the Azure Artifacts feed to store packages from public sources, such as nuget.org and npmjs.com. When you store packages from public sources on your feed, you'll be able to keep using them even if they're no longer available on the public feed. You'll also be able to leverage Azure Artifacts to review and validate each package for security purposes within your feed. This is especially useful for mission-critical packages.

Follow these simple steps to use Azure Artifacts:

1. Create an Azure Artifacts feed.
2. Publish your package to the feed.
3. Consume the feed in your favorite IDE, such as Visual Studio.

CI/CD

Both GitHub Actions and Azure Pipelines provide fully automated CI and continuous deployment (CD) capabilities. Users can define multiple environments, each with their own approval rules, secrets, and security permissions. For more complex scenarios, or for developers using repositories outside of GitHub, Azure Pipelines provides access to code from outside of GitHub along with centralized management of workflow templates and other features focused on enabling secure deployments at scale.

GitHub Actions and Azure Pipelines can be used individually or together. Many companies will choose to automate builds using pipelines, and use GitHub Actions to automate non-build workflows. You can also store and serve internal packages and containers with the GitHub Package Registry while keeping compiled binaries and other artifacts in Azure Artifacts. Both products offer deep integration with Azure through an extensive library of tasks and actions in their respective marketplaces.

GitHub Actions

GitHub Actions helps you automate tasks within your project that relate to the overall software development lifecycle of your code. These actions are event-driven, which allows you to configure one or more commands to run after a specific event has occurred. Each event triggers a workflow, which comprises one or more jobs. Each job can have one or multiple actions, allowing for highly configurable workflows to be constructed.

Actions uses YAML syntax to define workflows, events, jobs, and steps within those jobs. Any workflow created for your project will be stored in the repository under a specific workflows folder (.github/workflows). Creating a workflow is as simple as opening a new file and filling in some basic information, such as the workflow name, the event(s) that trigger(s) the workflow, and one or more jobs. An example of a basic workflow can be seen below:

```
name: sample-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v2
      - run: npm install -g bats
      - run: bats -v
```

This will, on pushing new code to the repository, check out the code, set up Node.js, install the bats package in the global npm registry, and validate the installation by requesting the version of the package.

GitHub Actions has a wide variety of built-in actions for your use, along with many more being contributed by the community daily. Over 9,000 actions can be found by visiting the GitHub Marketplace and searching for Actions.

Azure Pipelines

Azure Pipelines provides a lot of value in a small amount of time. It enables CI for compiling and testing code when changes come in, as well as CD for deploying applications after changes are compiled and tested successfully. We encourage every organization to explore CI and CD, as these processes improve code quality and reduce deployment efforts.

Azure Pipelines can help with CI and CD by offering build and deployment pipelines. Each contains steps to compile and test your code and deploy it to one or more environments. The beauty of Azure Pipelines is that it works with any type of code, no matter where you store it—from C# on Azure Repos to Java on Bitbucket, to Node.js on GitHub, or any other language, Git, or SVN repository.

Azure Pipelines works very well with Azure services to deploy your application in an Azure web app, for instance. It also works with any service that runs in any other environment, such as Google Cloud Platform, AWS, or even on-premises in your own datacenter. If you're already using CI tools such as Jenkins or Spinnaker, you can easily bring your existing builds and pipelines to Azure and take advantage of dynamic agent plug-ins to reduce infrastructure requirements and costs.

Make your pipelines as simple or as complex as you want. Ideally, you want to automate as much as you can, from the creation and destruction of your infrastructure to the deployment and testing of your application. Pipeline tasks are available for almost everything, and you can access more tasks as extensions to Azure DevOps in the Visual Studio Marketplace.

Security features

When developing an application, security needs to be integrated into your DevOps process as much as CI, testing, and work item tracking. In many CI/CD platforms, proactive security scanning can be enabled through the use of custom scripts or plug-ins to the platform itself. With GitHub, security is always at the top of your mind. From helping to integrate security into your workflows to proactively scanning your repositories for potential vulnerabilities in your code or libraries, GitHub's powerful platform tools help take the guesswork out of writing and maintaining secure code.

DevSecOps

DevOps best practices of CI and CD rely on heightened collaboration between software engineers and operations teams to accelerate software development. While DevOps delivers on the promise of faster software development, digital leaders face issues with the security and compliance of their code, workflows, and infrastructure. Azure and GitHub provide the tools for any organization to implement DevSecOps, the evolution of DevOps, where developer, security, and operations teams foster a culture of collaboration to achieve continuous security.

Between GitHub and Azure, we have a suite of tools and services to help. Azure and GitHub provide the building blocks to develop and scale DevSecOps practices:

- Shift security left with GitHub Advanced Security, allowing issues to be detected as soon as they are introduced into the codebase.
- Understand the runtime behavior of your apps and infrastructure with Azure Monitor.
- Leverage secret scanning with GitHub.
- Create policy as code with Azure Policy.
- Secure your application using secret management with Azure Key Vault.
- Uncover vulnerabilities and dependencies with GitHub code scanning and Dependabot.
- Integrate production instances with Azure Security Center, your home for security information and status.

With GitHub and Azure, it's never been easier to kickstart and scale your own DevSecOps practices with our unified solution. This complete toolset offers ways for you to remove bottlenecks clogging your delivery pipeline and provide the necessary controls for compliance and security. By uncovering vulnerabilities earlier, your teams save time remediating issues and achieving compliance, while also minimizing any associated costs. So, they can focus on their main goals: propelling innovation with efficient and secure software delivery.

GitHub Advanced Security

GitHub Advanced Security is a suite of advanced application security capabilities that helps developers find and fix issues within their workflow. GitHub Advanced Security consists of code scanning and secret scanning capabilities along with Security Overview.

Code scanning provides an automated security review with every Git push. It provides accurate, actionable security reviews within the developer workflow and offers an opportunity to fix issues before merging code. Code scanning is powered by GitHub's CodeQL engine and integrates with any static application security testing (SAST) engine while providing the same user experience that developers love.

Secret scanning watches your repositories for known secret formats and notifies you as soon as secrets are found. Secret scanning supports 45+ commonly used secret patterns (including Azure, AWS, Google Cloud, npm, Stripe, and Twilio) and custom secret patterns.

Security Overview provides a high-level view of the application security risks to which a GitHub organization is exposed. This view includes code scanning, secret scanning, and Dependabot results, along with associated risks.

GitHub Dependabot

GitHub Dependabot creates automated pull requests to help keep your dependencies secure and up to date. Dependabot monitors security advisories for Ruby, Python, JavaScript, Java, .NET, PHP, Elixir, and Rust. Pull requests are created immediately in response to new advisories. When there's remediation for a new security threat or an updated version of referenced components, Dependabot creates pull requests that include release notes, changelogs, commit links, and vulnerability details. These pull requests are then reviewed and committed by developers or maintainers—keeping dependencies secure and up to date.

If you want to improve the quality of your software and learn more about automating your build and release processes, download and read these free resources:

DOCS

Effective DevSecOps

GitHub Advanced Security

GitHub Dependabot

Better together: Visual Studio + GitHub + Azure

Visual Studio, GitHub, and Azure provide developer teams with the tools, platform, and service to support modern app development. By combining the advanced development, testing, and debugging features of Visual Studio with the open-source knowledge base and automation power of GitHub, Microsoft gives development teams all the tools they need to work efficiently and collaboratively. GitHub Enterprise and Azure DevOps make it easier to manage progress, measure team metrics, and optimize processes. Visual Studio and GitHub work seamlessly with Azure, providing developers with the most advanced toolset for code-to-cloud workflows.

With the combined power of Visual Studio, Azure, and GitHub in Visual Studio subscriptions, you are able to bundle key tools and services to empower your developers to quickly and efficiently deliver modern apps:

- Tools like CI and CD built into GitHub and Azure DevOps allow automating workflows such as running automated tests and deploying to test environments, increasing the agility and productivity of development teams.
- Collaboration platforms provide developers with access to communicate and coordinate with team members and leverage the collective knowledge of the open-source community.
- Advanced cloud services give developers more tools to innovate and experiment with building apps for a variety of platforms.

DOCS

Visual Studio documentation

Visual Studio + GitHub

GitHub Codespaces

Codespaces provides fully configurable cloud development environments, available in your browser, through Visual Studio Code, or any other suitable editors like Emacs or Vim.

Codespaces includes everything developers need to work with a specific repository, including an editor, a terminal, support for common languages, frameworks, and databases. It is completely configurable, allowing you to create a customized development environment for your project, and allowing developers to personalize their experience with extensions and dotfile settings.

Codespaces offers many benefits to teams, including:

- **A standardized environment**: You can create one Codespaces configuration that will allow anyone using your repository to have a consistent experience when working with your code. This experience is consistent for users accessing Codespaces via both Visual Studio Code and the browser-based experience, as seen in *Figure 2.1*:

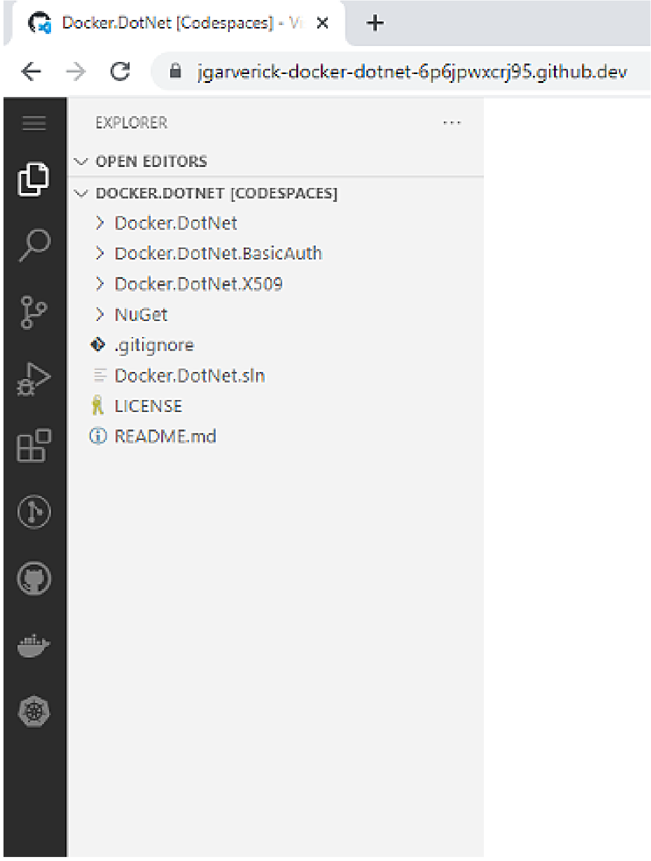


Figure 2.1: Browser experience for Codespaces

- **Fast and personal onboarding**: Once you have your environment [configured](), new developers can use the Code drop-down button on any GitHub repository within an organization and select 'Open with Codespaces' to trigger a new dev container's creation. While initial startup times vary by repository architecture, background processes ensure the new codespace will be created with all the appropriate dependencies and configured.
- **A secure environment**: Developing in the cloud allows you to keep one source of truth for your repository. If all of your developers are using Codespaces, this eliminates the need to clone the repository locally or having to install dependencies locally as root. There is also an option to configure GPG signing of Git commits from within Codespaces as an additional layer of proof, citing which developer authored a change.

DOCS

GitHub Codespaces

Dev Containers in Codespaces

Using GitHub Codespaces in Visual Studio Code

Use your preferred programming language

Developers have their tools, languages, and frameworks of choice, and Azure supports a wide array of developer options. From .NET to Java, JavaScript, Python, and more: you have the ability to develop in your stack of choice and the flexibility to use different languages and frameworks. Azure supports running applications written in these languages seamlessly.

.NET and Azure

If your preferred development stack is .NET, then the Visual Studio family of IDEs and GitHub have you covered. Using anything from .NET Core to legacy .NET Framework versions (4.x), to newer revisions like .NET 5 and 6, all editors offer first-class support for various installations of SDKs. IntelliSense, package management, and integrations with local and cloud-based services all work right out of the box.

For Azure support, the Visual Studio family offers a best-in-class experience when developing for targets such as Azure App Service, Cognitive Services, Blob Storage, Event Hubs and Event Grid, Cosmos DB, and others. This experience is also available within GitHub Codespaces by installing the required extensions for any Azure services you may be using. Leveraging your .NET skills and Azure, you can begin writing code that will host your application on Azure, consume other cloud services from your application, and even take advantage of modern serverless architectures to increase the resiliency and scalability of your application.

DOCS

Key Azure Services for .NET developers

Configure Visual Studio for Azure development with .NET

Configure Visual Studio Code for Azure development with .NET

.NET development on Azure configuration checklist

JavaScript, Python, and Java on Azure

JavaScript and Node.js, Python, and Java developers are also set up for success in Azure. Visual Studio and GitHub offer extensive support for developing, building, and deploying applications written in these languages and frameworks. Simply install the runtime of your choice and any corresponding extensions and you're immediately able to start coding.

From Java runtimes to different versions of Node.js or Python, you can rest assured that your development experience will be consistent using Visual Studio, GitHub, and Azure. Azure App Service (web and function apps) offers support for applications written in any programming language, including Docker containers.

For Java developers, Azure App Service offers Java 11 on both Linux and Windows, allowing developers to run JAR files or even WAR files through Apache Tomcat v8.5 and v9.0 hosted in App Service. Linux services also support the two latest LTS versions of Java, while Windows services support the three latest LTS versions.

Python support is available for 2.7, 3.6, and 3.7 on App Service, giving developers the flexibility to deploy their apps using a targeted runtime. There is also support for running apps using Gunicorn as well as hosting apps written in WSGI frameworks such as Django and Flask.

DOCS

Azure App Service: Node.js, Python, and Java support

SDKs and command-line tools

Azure includes a collection of SDKs for .NET, Node.js, Python, Java, and more. These are built on a common core for easy use of Azure services, with a focus on consistency, familiarity, and language idiomaticity.

Command-line tools are also available for managing Azure services and applications using scripts. The Azure CLI is a set of commands used to create and manage Azure services that is compatible with Windows, macOS, and Linux. Likewise, Azure PowerShell is a set of cmdlets that allows you to manage Azure resources from the PowerShell command line.

Regardless of the development platform or deployment target, Microsoft has a rich offering of development tools and cloud services to meet your needs. From Visual Studio 2022 to Visual Studio Code to GitHub Codespaces, there is an IDE that will rise to the challenge and enable your productivity.

The software development lifecycle is also completely covered with platform offerings from GitHub as well as Azure DevOps. Tracking your work, building and deploying your code, testing, and managing artifacts are easily handled. With GitHub, integrated DevSecOps features such as Dependabot and GitHub Advanced Security allow you to rest easy knowing that vulnerabilities are always being scanned for.

With Microsoft Azure, you have a world of cloud services at your fingertips. Runtime support for languages such as Java, Python, and Node.js in addition to .NET expands the possibilities of what you can build in Azure, as well as where your code is deployed. Whether from the IDE or the command line, interacting with Azure as a part of the development cycle is facilitated for everyone.

In *Chapter 3*, *Cloud-native applications*, we will be taking a look at cloud-native application development, which further expands on the core architectural patterns, components, and methodologies for creating applications that leverage the full potential of the cloud.

03 /

Cloud-native applications

What do we mean by cloud native?

Almost any application can run in the cloud one way or another. Even for legacy applications, you can use virtual machines and have them running in the cloud. But to take full advantage of what the cloud offers, you will need applications to be cloud-native. Designing applications for the cloud allows you to use the best of what the cloud can offer and benefit from the latest innovations in the public cloud.

When transitioning to the cloud, there are many options. Whether the choice is to replatform, refactor, rearchitect, or even rewrite the application to fully leverage the scalability and elasticity of cloud services, Azure can help you. For cloud-native applications to be intelligent, AI can be used to provide advanced insights. Being cloud native also means embracing a potential global footprint and tuning individual microservices or cloud services for optimal performance under load.

The software development lifecycle is consistently shifting to the left, moving feedback loops closer to developers and validating functionality very quickly. As this trend continues, you need to be faster, smarter, and more adaptable. Developers want to go from ideation to code running in the cloud seamlessly. Microsoft combines development tools with Azure to create a highly efficient inner loop for cloud-native development. This combination provides everything you need as a developer, from source code management, editors and IDEs, and infrastructure as code to container registries and continuous integration/continuous deployment (CI/CD) toolchains, all designed to work together. Whether you want to quickly deploy code from Visual Studio or Visual Studio Code or use a CI/CD pipeline from GitHub or Azure DevOps, the process is fast and simple. You can quickly deploy your code and test new functionality and features in Microsoft Azure.

NOTE

Shift to the left is a practice in software development in which teams focus on increasing quality by testing earlier and deploying more frequently.

Speed of development is not the only challenge; you also want your applications to be resilient and scalable. High availability and uptime are very important in the digital-first era. If an application is not available, customers will quickly lose interest and go to a competitor whose application is more reliable.

Over the past several years, you must have seen story after story about data breaches and compromised applications that expose sensitive information. As more businesses embrace digital transformation, the resiliency and security of application infrastructure and data are of the utmost importance. Microsoft Azure offers built-in tools that can help you create highly available geo-distributed applications, along with intelligent threat protection in real time. With scalable services, you can create applications that can withstand high demand and achieve cost savings at the same time.

Using cloud-native design patterns you can achieve the agility, reliability, scalability, and security demanded by the next generation of applications. Developing with managed services in mind takes the guesswork out of the infrastructure layer and allows developers to focus on what matters—solving business problems by producing world-class applications.

QUICK START

Build cloud-native applications in Azure

Cloud-native components

By having a cloud-native approach, app developers can overcome the challenges they face every day around agility, reliability, and security. Not all applications are the same, and in some cases, it is important to prioritize certain characteristics over others. Regardless of the need, however, there are several basic building blocks for cloud-native applications, which include:

- Microservices
- Containers
- Functions
- APIs
- DevOps

When it comes to a cloud-native architecture, it can be implemented in two different ways—optimizing for infrastructure control or optimizing for developer productivity.

Infrastructure control provides control on what you are running, where you are running it, and how it performs. When you have requirements such as the need to use a legacy code base or custom libraries that require runtime access, you want to optimize for infrastructure control and use containers with Kubernetes to orchestrate them.

If you do not have such constraints, you can optimize for developer productivity architecture. Then you can build microservices using event-driven functions without having to worry about servers, allowing you to focus on code, rather than infrastructure.

When applications are built as **microservices**, the release velocity can increase because changes to any component are easier to make. Microservices make applications easier to scale and faster to develop, enabling innovation and accelerating the time to market. These microservices can be delivered as **containers**, which package application code and dependencies together to increase portability, or as **functions**, which offer an event-driven compute-on-demand experience that extends with capabilities to implement code triggered by events. Often, these microservices are surfaced as lightweight **APIs** and are shipped using DevOps processes and tools to automate build, test, and delivery (see *Chapter 1*, *Getting started with Azure Application Platform*, and *Chapter 2*, *Developer tools and developer cloud*).

QUICK START

Azure Architecture Center – Cloud Design Patterns

Kubernetes on Azure

A common pattern of modern applications is to run microservices hosted in containers. Containers are lightweight, standalone, executable packages of software that include everything needed to run an application: code, runtime, system tools, system libraries, and settings.

Microsoft Azure offers many choices on how to run containers. One of them is Kubernetes, also known as K8s, an open-source orchestrator that allows you to automate the deployment, scaling, and management of containerized applications.

When running Kubernetes on Azure, there are several options to choose from, including:

- Azure Kubernetes Service (AKS)
- Azure Red Hat OpenShift
- Arc-enabled Kubernetes
- Application services on top of Kubernetes

Let's look at each of these options and see its benefits.

Azure Kubernetes Service

As a central pillar of cloud-native practices, AKS is an enterprise-grade, managed Kubernetes service. Because it is a fully managed system, Azure handles critical tasks such as health monitoring and maintenance, while developers can focus on code.

AKS has built-in best practices like Azure Advisor notifications to help optimize your Kubernetes deployments with real-time personalized recommendations. It features multi-layer security across operating systems, compute resources, data, networking with consistent configuration, identity, secret integration with Azure Key Vault, and policy management. Also, AKS increases operational efficiency with support for automatic cluster upgrades and the ability to schedule service maintenance operations to off-peak hours.

AKS makes it simple to deploy a managed Kubernetes cluster in Azure as it reduces the complexity and operational overhead of managing Kubernetes by offloading much of that responsibility to Azure. When deploying an AKS cluster, the Kubernetes management plane, control plane nodes, and worker nodes are deployed and configured for you.

QUICK START

Deploy an AKS cluster using the Azure CLI

Kubernetes on Azure

Azure Red Hat OpenShift

Red Hat OpenShift is an enterprise Kubernetes container platform created by Red Hat. OpenShift is all about choosing a platform to power your applications today by automating processes and reducing complexity, allowing you to rapidly deliver without roadblocks. When running Kubernetes in production, you often need to add additional services to get basic functionality such as authentication, logging, and CI/CD. Those are separate components that you must integrate, manage, and keep up to date on your own. Furthermore, when using new features to build modern applications such as Serverless and Service Mesh, this means even more integration and testing.

Azure Red Hat OpenShift allows developers to focus on what matters. You can take advantage of the enhanced user interface for application topology and builds in the web console. It allows you to build, deploy, configure, and visualize containerized applications and cluster resources more easily.

You can bring code from your Git repository or an existing container image and build it using source-to-image (S2I) builds or deploy solutions from the developer catalog, such as OpenShift Service Mesh, OpenShift Serverless, or Knative.

TUTORIAL

Create an Azure Red Hat OpenShift 4 cluster

Deploy an application from source to Azure Red Hat OpenShift

Azure Red Hat OpenShift provides built-in CI/CD. You can create automated builds, tests, and deployments of applications using OpenShift Pipelines, a serverless **CI/CD** system designed to create and scale a pipeline using GitHub Actions, or using existing pipelines.

QUICK START

Azure Red Hat OpenShift

Arc-enabled Kubernetes

With Azure Arc-enabled Kubernetes, you can attach and configure Kubernetes clusters located either inside or outside Azure and bring cloud operations anywhere.

Azure Arc can manage applications running on Kubernetes at scale through advanced DevOps techniques like GitOps. It provides a single pane of glass with central visibility through the Azure portal and governance and compliance of your applications and Kubernetes clusters through Azure Policy.

NOTE

GitOps is a way to operate infrastructure in continuous delivery by using tools developers are already familiar with, like Git and CI/CD tools.

QUICK START

Azure Arc-enabled Kubernetes

Application services on top of Kubernetes

Developers have been taking advantage of the Azure application platform services for the last few years to develop modern apps. These purpose-built services enable developers and operations teams to focus on the business requirements and not have to spend additional resources managing the infrastructure behind these services. To enable developers of on-premises applications to experience that same productivity gain, Microsoft has enabled Kubernetes as a target for running many Azure platform services, managed through Azure Arc. The platform services supported are:

- App Service
- Functions
- Logic Apps
- Event Grid
- API Management

If the project requires you to have more control over the infrastructure (like running on-premises, on the Edge, or in a different cloud), this can be accomplished with Azure Arc. Any Kubernetes cluster connected through Azure Arc is a supported Azure App Service target. Thanks to Azure Arc and portable app services, we can run the same code anywhere and create hybrid and multi-cloud applications.

You can develop and innovate faster with Azure's suite of application services as it comes with features and options that are optimized for developer productivity. Capabilities such as deployment slots, blue-green deployments, web consoles, App Service Editor, and extensive logs, among others, can make your life a little easier.

QUICK START

Set up an Azure Arc-enabled Kubernetes cluster to run App Service, Functions, and Logic Apps

Serverless on Azure

Serverless is a way of hosting your applications in the cloud while completely abstracting the underlying infrastructure. It is about increasing productivity by focusing on the code that powers your application without taking care of infrastructure. Publish the code to Azure, and Azure will take care of scaling, operating, and securing the underlying application code.

Microsoft Azure offers several serverless services, including:

- Azure Functions
- Azure Logic Apps
- Azure Static Web Apps
- Azure Event Grid

Let's look at these options more closely and see what they bring to the table.

Azure Functions

Azure Functions is a serverless solution in Azure that allows you to focus on the code that matters most to you, while Azure Functions handles the rest.

Azure Functions makes event-driven programming easier with state-of-the-art autoscaling, and triggers and bindings to integrate with other Azure services. You can run a piece of code in response to an event and scale without having to worry about infrastructure. By doing so, Azure functions provide increased agility, improved resilience, and scalability.

It comes with support for multiple languages and integration with other services like Azure Key Vault and Azure DevOps and provides flexible hosting options to support critical enterprise workloads. All of this is integrated with development tooling: Visual Studio Code, Visual Studio, and other editors such as IntelliJ or Eclipse.

DOC

Getting started with Azure Functions

Serverless on Azure

Azure Logic Apps

Azure Logic Apps is a cloud-based platform for creating and running automated workflows for integrated apps, data, services, and back-end systems with a library of more than 450 connectors.

You can quickly develop highly scalable integration solutions, allowing you to easily connect to any system or data source.

QUICK START

Create integration workflows with Azure Logic Apps in the Azure portal

Azure Static Web Apps

With Azure Static Web Apps, you have the flexibility to deploy static HTML, JavaScript, and CSS files from a GitHub repository or another source into a managed, Azure-hosted website. A free tier exists for hobby or non-commercial needs, and a standard tier allows for more production-appropriate needs. SSL certificates are included, as are at least two custom domains and at least three staging environments.

QUICK START

Building your first static site with Azure Static Web Apps

Azure Event Grid

Azure Event Grid simplifies event-based applications with a single service for managing the routing of events from any source to any destination.

You can easily build applications using an event-based architecture by subscribing to a source and defining event handlers or Webhook endpoints to which you can send domain events. Event Grid has built-in support for events coming from Azure services, but also supports your own events through custom topics.

QUICK START

Install Event Grid extension on Azure Arc-enabled Kubernetes cluster

Cloud native and open source

It can be said that cloud native and open source have a symbiotic relationship. Cloud-native apps are built on open-source technologies when possible, focusing on architectural modularity and enabling them to be platform-independent. "Cloud native" means flexibility of deployment targets. Microsoft invests in open-source software in several ways in order to help Azure users build the best possible solutions. There are several areas of investment:

- Ensuring open-source technologies perform well on Azure
- Open-source communities (Apache, Linux, the .NET Foundation)
- Tools and integrations to help with Azure deployments
- DAPR – Distributed Application Runtime, a lightweight framework for building modern distributed applications

With added commitments to supporting open-source functionality in Azure as well as the advent of more modern runtimes for service-based applications such as DAPR, open-source technologies are an essential building block for individuals and organizations looking to build solutions that run on Azure.

DOCS

Check out featured open source projects and products

Blazor | Build client web apps with C# | .NET

Open Service Mesh

ANNOUNCEMENT

Gain flexibility to run open-source applications your way with Microsoft Azure

Overview of open source on Azure

Open source has fundamentally changed software development. With the increase in open-source adoption, it must run smoothly on Azure.

More than half of all the cores on Azure are running Linux, and Microsoft has heavily invested in enhancing its performance, reliability, security, and resiliency. Azure provides you with enterprise-grade, managed open-source software databases like MySQL and MariaDB, Linux operating systems, analytics, and machine learning services to bring AI to your application.

Microsoft's contributions span several areas: Linux, the Kubernetes ecosystem with projects like DAPR or Open Service Mesh, programming languages, web frameworks, and technologies such as .NET, Node.js, Python, PHP, and many more.

DOC

Open source on Azure

Open-source cloud projects

Three cloud-oriented, open-source projects that Microsoft participates in are as follows:

- **Kubernetes**: Microsoft works closely with the Kubernetes project, sharing knowledge, contributing, and shaping its future. Microsoft is now the third-leading corporate contributor and works to make Kubernetes more enterprise-friendly and accessible.
- **Java**: Microsoft participates in and contributes back to the Java community. Microsoft's Build of OpenJDK is a long-term support (LTS) distribution of OpenJDK that is open-source and free for anyone to deploy anywhere. This allows you to focus on developing Java applications and delivering value without being concerned with licensing requirements or costs.
- **The .NET Framework**: It is a cross-platform and open-source development platform. You can develop many types of applications with .NET, across all operating systems and any device. You can also choose from object-oriented languages such as Visual Basic and C# or include functional programming with F#. There is also managed support for C++. With .NET, you can build once and run anywhere. The .NET open-source ecosystem is directly supported by the .NET Foundation, a non-profit organization dedicated to cultivating an innovative and commercially friendly community around .NET.

QUICK START

What is Kubernetes?

Microsoft Build of OpenJDK™

.NET Framework

.NET Foundation

Flexibility of choices for tools, languages, and integrations

Azure lets you build and run your apps on your terms, in any environment, with integrated support for open-source tools, languages, and frameworks. You can develop your apps using your favorite tools, languages, and third-party integration. Also, you can deploy Docker-based containers running any app to Azure App Services, PowerShell scripts to Azure Functions, static websites, and many more.

With AKS, you can run virtually any code by installing any library or runtime. Even more managed services like Azure Functions provide support for a wide range of programming languages.

There are also integrations within Visual Studio and Visual Studio Code that allow you to deploy, test, and run applications in the cloud.

Distributed Application Runtime: DAPR

Stateless microservices are a common pattern for architecting modern cloud-native applications, and DAPR can help you accelerate development.

DAPR is an open-source project whose goal is to simplify writing microservices by providing building blocks that abstract away common challenges of distributed applications such as service-to-service invocation, HTTP and gRPC requests, secret and state management, and input and output bindings. It provides support for multiple programming languages, such as .NET, Java, Python, or Go. DAPR applications can be run on-premises, in any cloud, or on an edge device without having to change a line of code from one to another.

DAPR simplifies distributed applications and acts as a glue that binds the application to the infrastructure capabilities.

QUICK START

Get started with Dapr

How to build cloud-native applications on Azure

Depending on the requirements of the software you are building, you may have to take different architectural approaches to comply with business needs. Microsoft Azure offers tools and services that can accommodate any scenario.

Some examples of possible scenarios are:

- Business-critical applications
- API-first applications
- Real-time data processing
- Geo-distributed applications

Let's take a closer look at each scenario and see what it requires.

Business-critical applications

There are several things you need to consider when it comes to building modern business-critical applications:

- Scaling in order to handle increased traffic and bursts
- Low latency so users have the same experience anywhere in the world
- High availability to achieve maximum uptime

A design example for modern business-critical applications is shown in *Figure 3.1*:

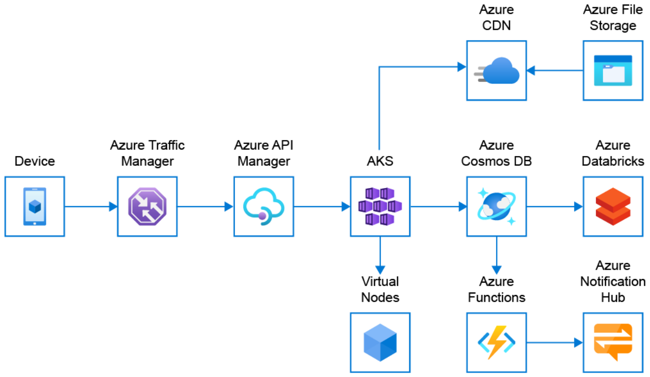


Figure 3.1: Business-critical application design in Azure

You can have instant, elastic scaling that handles traffic and sales bursts without managing infrastructures using a number of services—for example, you can use AKS. Low-latency data access from anywhere in the world for fast, robust user experiences can be achieved with Azure Cosmos DB. Finally, for high availability, you can place services across multiple datacenters and ensure that an application has absolutely no downtime.

API-first applications

With an API-first approach, the biggest challenge is how to secure, govern, and catalog the APIs. You can create an API gateway and developer portal in a matter of minutes and publish APIs easily for internal or external use.

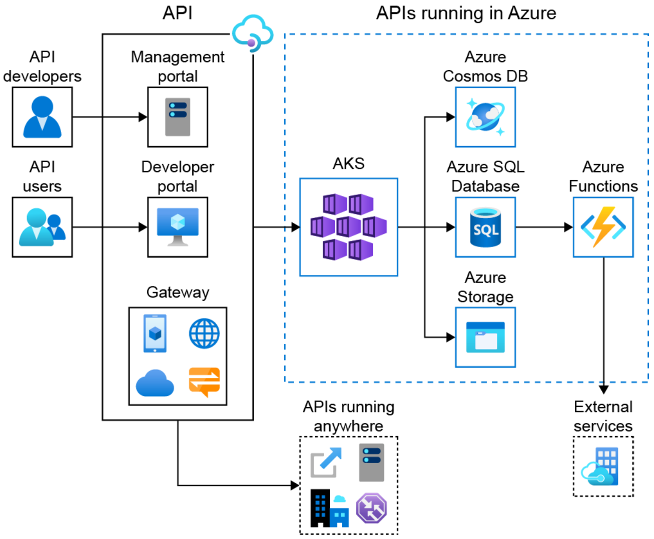


Figure 3.2: API-first design in Azure

This approach allows you to easily handle any data schema change and adapt quickly to rapid changes. You can connect to back-end services running anywhere and manage, secure, and optimize all APIs in one place no matter where they run.

Real-time data processing

Real-time data processing can be a challenge when multiple data sources are in play. Azure offers tools that can help with real-time data ingestion and processing pipelines, capable of detection and notification within seconds.

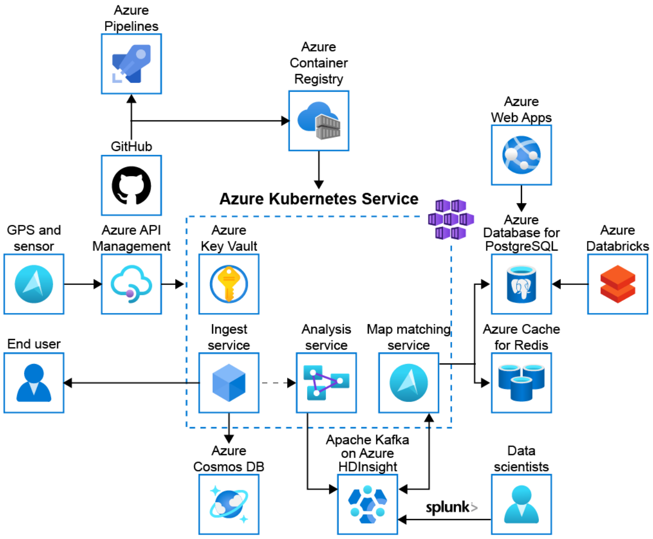


Figure 3.3: Real-time data processing design in Azure

This way, any data change can be processed just a moment after the change occurred. With a secure API gateway, you can connect to back-end services running anywhere. Elastic provisioning of compute capacity, without the need to manage the infrastructure, allows you to focus on data and provide cost savings in the process.

Geo-distributed applications

The geo-distributed approach has two main goals:

- Provide a seamless experience to its users anywhere in the world.
- Provide high availability and maximum uptime.

In *Figure 3.4*, you can see a possible architecture for a geo-distributed application. AKS is placed in several different regions and network traffic is controlled with Traffic Manager to direct the user to the closest available AKS instance to complete the request.

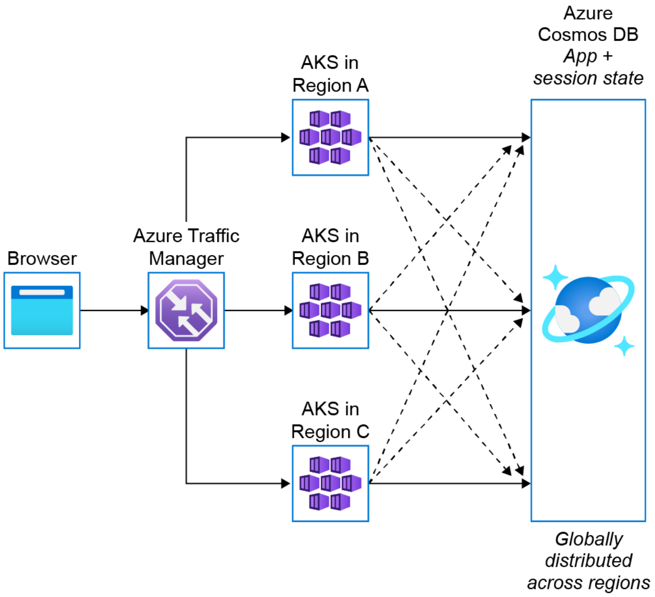


Figure 3.4: Geo-distributed application design in Azure

The Microsoft Azure infrastructure is composed of more than 160 datacenters in more than 60 different regions. With services such as Azure Front Door or Traffic Manager, you can create globally distributed applications. This way, you can ensure that applications are up and running even with issues in one of the datacenters. As the application is hosted in multiple locations, users are routed to the closest instance, providing minimum latency and delay.

Building cloud-native applications is all about leveraging purpose-built cloud services that allow you to develop faster, deploy more frequently, innovate, and be scalable. Microsoft Azure offers dozens of services to choose from when designing and building cloud-native applications. These services come with support for many programming languages, frameworks, and runtimes, and offer excellent integration capabilities. They also come with robust built-in features that can help you monitor, secure, and improve your applications.

Whether in the cloud or on-premises, the management of infrastructure such as AKS is easier than ever using Azure Arc. And with Azure Arc-enabled services like App Services, you can safely deploy applications to local clusters for testing, knowing they will operate the same way in the cloud.

With investments in key areas in the open-source community, Microsoft continues to build on the strength of using open-source solutions, especially with cloud-native applications. From language support for many major programming languages to contributions to the community, like DAPR, the opportunity to embrace open-source offerings in your own application is more prevalent than ever.

Armed with a library of design patterns, business cases, and means to implement them, your journey to design and develop a cloud-native application is greatly improved when using Azure. With design frameworks such as the Cloud Adoption Framework and the Well-Architected Framework, you have solid field-tested guidance on how to efficiently adopt cloud technologies and develop applications with a focus on cloud-native traits.

There are several things that all applications have in common, with the goal of microservices and cloud-native to abstract away the complexities of many of them, such as where and what is running the application. Components such as CPU, memory, and networking are a few examples, with much of this chapter focused on those. However, there is one additional component that has not been addressed. Where will your application store its data?

In the next chapter, we will dive deeper into the various designs and architectures that can be used for data storage and how that plays into the business's data estate. Once application data is stored, it's then the job of data scientists and engineers to utilize machine learning and artificial intelligence to discover and transform the data into meaningful conclusions and decision points. But storing data comes with many caveats, such as compliance, security, and balancing costs. We'll explore all these topics next.

04 /

Connecting your application with data

Azure has your data needs covered

As a developer or architect, at some point in the application design and creation process, a decision must be made as to what type of data should be collected, along with its format and where it will be stored. Additionally, an important key parameters like costs, performance, growth, security, compliance, and the data lifecycle need to be considered in determining the perfect data solution for the system. When progressing through all the options, realize it is highly likely that no single data solution will be able to meet the final needs of the users of the system.

Data is valuable and a strategic asset. Taking a big-picture approach and being open to new ideas around data storage and data processing can lead to incredible opportunities for taking applications to new heights and ensuring a stable and organized data estate.

Whether the solution is a traditional relational database-based application; a stream-based analytics solution; a data mart or data warehouse; or a storage facility for structured and unstructured data, slow- or fast-flowing data, or small data or big data via Lambda or Kappa architectures, Azure has an answer!

Azure makes it simple to mix and match data solutions of varying volumes, varieties, and velocities while providing out-of-the-box world-class performance, security, and governance. By abstracting data estate complexities, developers can focus on solving business problems and delivering value.

What can Azure do for your data?

By choosing Azure data solutions to store or process corporate data, businesses gain access to fully managed Platform as a Service (PaaS) services that free up valuable time and resources, time and resources that can be focused on new ways to delight business users and unlock data insights and business opportunities. By removing the hardware and software management components, architects and developers can focus on designing impressive, data-focused business solutions.

Developers can take advantage of industry-leading innovations, such as built-in security with automatic monitoring and threat detection, automatic tuning for improved performance, and turnkey global distribution and replication. Moreover, any cloud investments are protected by financially backed service-level agreements (SLAs).

Whatever the business needs dictate, Azure data solutions will help get applications to production faster, scale them widely, and manage them easily, all while maintaining security and compliance with laws and regulations. Not only that, but Azure's services, such as machine learning and artificial intelligence, are designed to be integrated together easily with a few clicks of the mouse.

Additionally, Azure data solutions can help solve some of these important questions:

- What data do you have?
- Is it trustworthy?
- Can people access the data needed to make the right decisions?
- How can you enable faster business insights?
- What's the compliance exposure?

Why trust Azure with your data?

No matter what role you may play in the lifecycle of the data estate, it is everyone's responsibility to be cognizant of its security and integrity. When it comes to data in Azure, Microsoft's trusted data principles put the control in your hands:

- You control your data.
- You choose your data location.
- Microsoft secures your data.
- Microsoft defends your data.

For example, Azure allows the data location to be chosen from several best-of-class datacenters around the world to meet any compliance or regulatory requirements that businesses may be required to adhere to.

In terms of security, many Azure services support a Bring Your Own Key (BYOK) feature that allows for the encrypting of data using privately owned keys. Additionally, Microsoft defends the data stored in Azure from known bad actors using Indicators of Compromise (IoCs) that are compiled from a vast set of inputs from the Microsoft Graph Security API.

Finally, Microsoft has been continually recognized as a leader by Gartner for the past 7 years in a row.

Where to store your data?

Today's data storage options are numerous. Picking the right one for the target application can be a daunting task even for developers and architects who have been around since the creation of traditional relational database management systems (RDBMSes).

Today's business problems typically require much more than what RDBMS can provide. Navigating the potential solutions for a new or potential modernization of an application is a vital skill for developers and architects.

Azure provides many types of data stores with the flexibility to support any data storage and processing scenario where it is on-premises, hybrid, in the cloud, or on the Edge. For quick prototyping and proof-of-concept tasks, many of these services have a free tier or 30-day trial period.

As we progress through the various solutions, we will make frequent reference to the Azure Architecture Center, which will provide you a helpful reference architectures to help visualize potential data solutions.

NOTE

```
Almost all storage options
mentioned in this section can be
used as activators and bindings
for Azure Functions.
```

Let's now take a closer look at each storage option.

Azure SQL Database

If the requirement is to utilize traditional relational databases with tables, columns, and rows to store data, Azure SQL Database is a great choice. However, there are many other open-source options you will explore later in this chapter.

Azure SQL Database is a relational database system like on-premises Microsoft SQL Server. Azure SQL Database runs in the cloud, so it's fully managed, performant, scalable, automatically backed up, and includes many other advanced features, such as flexible and responsive serverless compute as well as Hyperscale.

Databases in Azure SQL Database are extremely reliable and robust and offer an SLA that guarantees 99.99 percent uptime. From a cost standpoint, consider the fact that running SQL PaaS and IaaS workloads on Azure can be up to 86% less costly than other cloud platforms, which makes yet another compelling argument for using Azure.

Because Azure SQL Database communicates over the same protocol as SQL Server, the same familiar tools developers used previously, such as SQL Server Management Studio (SSMS), Azure Data Studio, and Visual Studio Code, will continue to work seamlessly.

Some other advanced features include:

- Geo-replication, based on Always On Availability Group (AOAG) technology, which replicates data to other geographical regions in real time.
- Dynamic data masking, which masks sensitive data for certain users at runtime.
- Auditing, which provides a complete audit trail of all the actions that are performed on the data.
- Automatic database tuning, which monitors the performance of the database and tunes it automatically.
- Transparent Data Encryption (TDE), which adds a security layer to protect data at rest from unauthorized or offline access.
- Always Encrypted, which allows developers to store and query encrypted data within the database, and protects sensitive data by encrypting it on the client side, never allowing the data or the cryptographic keys to appear in plaintext inside the database engine.

Get more compatibility with Azure SQL Managed Instance

When migrating databases from a source such as an on-premises hardware-based instance or a virtualized instance of SQL Server, the application may be using features of SQL Server that may not be available in Azure SQL Database. In these cases, it may be possible to migrate to Azure SQL Managed Instance. Because of its design, Azure SQL Managed Instance provides many more features that provide parity with SQL Server and yet provides the benefits of a fully managed service. For example, features such as Linked Server and SQL Agent are supported by Azure SQL Managed Instances, but not by Azure SQL Database.

Azure Database for MySQL, PostgreSQL, and MariaDB

In addition to Azure SQL Database, Azure provides managed data solutions for MySQL, PostgreSQL, and MariaDB. As with all Azure PaaS services, simply create a new instance and focus on building applications, not the underlying infrastructure. Additionally, many of the same cloud-based platform features, such as scalability and security in Azure SQL Database, are also available in these open-source alternatives.

Microsoft is committed to making Azure the best cloud for OSS by bringing together community and Azure innovations. This commitment is most visible with PostgreSQL, where the Azure engineering team has made multiple contributions to the global PostgreSQL open-source project.

All three of these offerings provide the following features:

- Built-in high availability with no additional cost
- Predictable performance
- Scaling as needed within seconds
- Secured protection of sensitive data at rest and in motion
- Automatic backups and point-in-time restore for up to 35 days
- Enterprise-grade security and compliance
- Support for Python, PHP, Node.js, Java, Ruby, .NET, and more

Azure Database for MySQL

Azure Database for MySQL is a relational database service powered by the MySQL Community Edition database engine. It is a fully managed database-as-a-service offering that can handle mission-critical workloads with predictable performance and dynamic scalability.

See **Intelligent apps using Azure Database for MySQL** for a reference architecture using Azure Database for MySQL.

Azure Database for PostgreSQL

Azure Database for PostgreSQL is a fully managed database instance based on the community version of the open-source PostgreSQL database engine. Azure Database for PostgreSQL supports several of the latest major PostgreSQL versions and includes many popular PostgreSQL extensions. As a managed service, enjoy AI-powered performance optimization and enterprise-grade security and compliance, including enhanced security capabilities with Azure Defender. As with other open-source offerings, it is scalable and offers flexibility and high availability with up to 99.99% SLA.

Build apps at any scale with high-performance, horizontal scaling with Azure Database for PostgreSQL Hyperscale (Citus). Hyperscale supports databases with sizes of up to 100 TB with fast backup and restores and rapid scale up and out capabilities. Use Azure Arc-enabled PostgreSQL Hyperscale to run on the infrastructure of choice, including hybrid and multi-cloud options, while benefiting from Azure cloud features.

See **Intelligent apps using Azure Database for PostgreSQL** for a reference architecture using Azure Database for PostgreSQL.

Flexible Server

Azure Databases for MySQL and PostgreSQL support a new deployment model—Azure Database for MySQL Flexible Server and Azure Database for PostgreSQL Flexible Server. Flexible Server provides more options when it comes to configuration and customization, for example, more support for platform features and additional configuration parameters for fine-grained tuning. Compared to the single-server deployment option, Flexible Server provides better cost optimization with start-stop capabilities and burstable instances. Burstable instances are great for scenarios where you don't have steady workloads and/or need more compute power only at peak times.

The Azure Database for PostgreSQL Flexible Server deployment option provides maximum flexibility and control with a choice of a single zone or zone-redundant high availability, with up to 99.99 SLA and the ability to leverage custom maintenance windows for planned database maintenance. Enjoy a simplified end-to-end deployment experience with a low cost of ownership. Reduce the time to market with tight integration with Azure Kubernetes Service, Azure App Service, and more.

Azure Database for MariaDB

Azure Database for MariaDB is a relational database service based on the open-source MariaDB Server engine. It is a fully managed database-as-a-service offering that can handle mission-critical workloads with predictable performance and dynamic scalability.

Azure Arc-enabled data services

Supporting applications with data solutions that reside on-premises or in other cloud providers without continuous or direct Azure connectivity doesn't have to be difficult. Begin the journey to Azure by exploring Azure Arc-enabled services.

Companies have significant, existing investment on-premises, and are looking for a consistent experience as they expand to the cloud and the edge. Many organizations are taking a hybrid approach when it comes to the cloud.

Microsoft is committed to providing a true hybrid experience that's seamless and enables access to the latest innovations, regardless of where the data lives. Azure has been built to enable seamless hybrid capabilities when it comes to development, deployment, and management across on-premises servers, public clouds, and edge devices. Azure Arc now enables Azure data services to run anywhere, on any infrastructure across on-premises, Edge, and third-party clouds.

Azure Arc organizes, governs, and secures Windows, Linux, SQL Server, and Kubernetes clusters across datacenters, the edge, and multi-cloud environments right from Azure, by surfacing them in the Azure portal for single pane of glass management via many of the features afforded by the Azure Resource Manager (ARM).

Azure Arc-enabled data services enable cloud advantages (such as scalability and self-service provisioning, and unified management) in on-premises environments. It also enables them to be always current by receiving frequent updates.

Database services currently enabled by Azure Arc include Azure SQL (Azure SQL Managed Instance) and Azure Database for PostgreSQL Hyperscale.

See Azure Arc hybrid management and deployment for Kubernetes clusters and Manage configurations for Azure Arc-enabled servers for reference architectures using Azure Arc.

Azure Cosmos DB

Azure Cosmos DB is a fast and flexible NoSQL database built for cloud-native applications of any scale. Its key features include:

- Guaranteed availability and speed at any scale, with SLAs for 99.999 percent availability and less than 10ms latency on reads and writes.
- Open APIs for MongoDB, Cassandra, and Graph data; and a Core (SQL) API with SDKs for .NET, Java, Node.js, and Python.
- Serverless and autoscale options automatically match resources to demand without having to plan or manage capacity.
- The free tier enables the development and testing of applications with free database operations and storage for the lifetime of the account.
- Geo-replication, which distributes data to any Azure region globally in real time for higher availability and low latency.
- Automatic indexing of data and flexible schema, which simplifies data ingestion and distribution.
- Out-of-the-box no-ETL analytics with Azure Synapse Link, enabling advanced analytics over real-time operational data stored in Azure Cosmos DB, with no performance impact or data movement.

In addition to all these features, Azure Cosmos DB offers five data consistency levels that allow the fine-tuning of the distributed data system. Choose from models ranging from strong to eventual consistency.

With speed and flexibility, applications of any size or scale will benefit from superior performance, elasticity, and reliability.

See Gaming using Cosmos DB and Globally distributed applications using Cosmos DB for reference architectures using Cosmos DB.

QUICK START

Get started with Azure Cosmos DB

Azure Storage

Azure Storage is one of the most reliable and performant services in Azure. Azure Storage offers five types of storage that all benefit from the following shared features:

- Geo-redundancy, which replicates data to different datacenters, allowing for recovery if a disaster causes an individual datacenter to fail
- Encryption of data at runtime
- Custom domains

The five Azure Storage types are Blob, Table, Queue, Files, and Disk Storage (as shown in *Figure 4.1*):

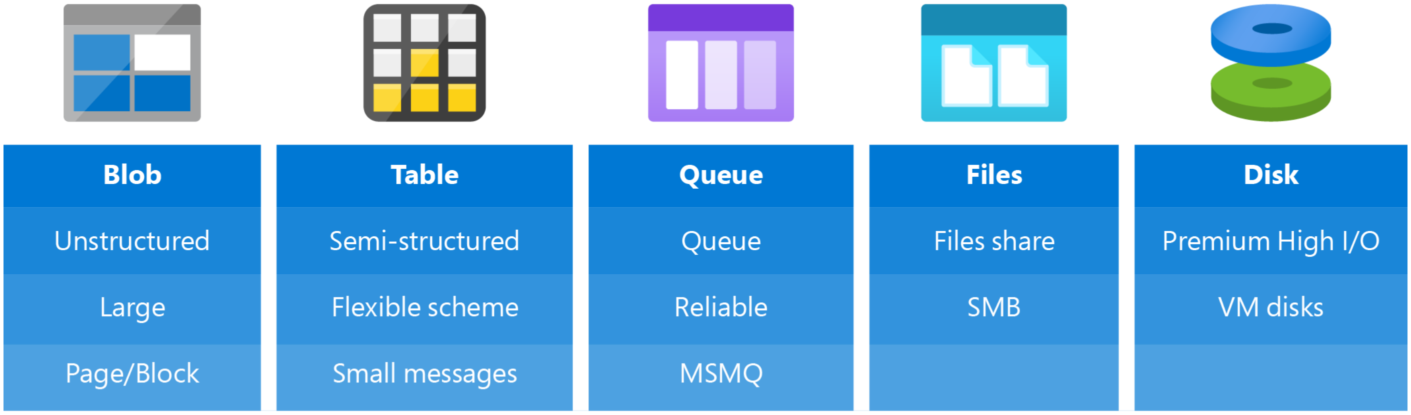
Let's take a closer look at each storage type and what it offers.

Blob Storage

Azure Blob Storage stores large, unstructured data—literally, blobs of data. This data can be video, image, audio, text, or even virtual hard drive (VHD) files for VMs.

There are three types of blobs, page, block, and append blobs:

- **Page blobs** are optimized for random read and write operations and are perfect for storing a VHD.
- **Block blobs** are optimized for the efficient uploading of large amounts of data. These are perfect for storing large video files that don't change frequently.
- **Append blobs** are optimized for append operations, such as for storing operation logs that can't be updated or deleted.



Blob	Table	Queue	Files	Disk
Unstructured	Semi-structured	Queue	Files share	Premium High I/O
Large	Flexible scheme	Reliable	SMB	VM disks
Page/Block	Small messages	MSMQ		

Figure 4.1: Data storage types

QUICK START

Get started with Azure Blob Storage

Table Storage

Azure Table Storage is an inexpensive, extremely fast NoSQL key-value store. Key-value stores are very flexible. For example, one key can contain data describing an order and another key could describe customer information. Table Storage does not have defined data schemas, making it very flexible.

DOC

Get started with Azure Table Storage

Queue Storage

Azure Queue Storage is an unusual type of storage. While it's used to store small messages of data, its main purpose is to serve as a queue. Queues are used by putting messages in them and allowing other processes to pick them up. A Queue-Based Load Leveling pattern decouples the message sender from the message processor, resulting in performance and reliability benefits. Queue Storage is found in previous versions of Windows such as MSMQ.

TUTORIAL

Get started with Azure Queue Storage

Files

You can use Azure Files as a drive from which to share files. It uses the Server Message Block (SMB) protocol, meaning it can be used with Windows and Linux and accessed from either the cloud or on-premises systems when a direct path to Azure is available. By design, Azure Files SMB shares are blocked from the internet. Like the other services in Azure Storage, Azure Files is scalable and inexpensive.

DOC

Get started with Azure Files

Disk Storage

Azure Disk Storage is like File Storage but specifically meant for high I/O performance. It is perfect for use as a drive in a VM that requires high performance to run SQL Server, for instance. Disk Storage is available only in the premium pricing tier of Azure Storage.

Azure Data Lake Storage

The previous data stores were intended for regular application use or for use with VMs. Azure Data Lake Storage, on the other hand, is storage for big data applications. It provides massively scalable, secure data lake functionality built on Azure Blob Storage. Use it to store large amounts of data in its native format—structured, unstructured, or anything in between. The point of Data Lake Storage is to hold your raw data so that it can be analyzed, transformed, and moved when needed.

The following are the main characteristics of Azure Data Lake Storage:

- Unlimited storage capacity. A single file can be larger than one petabyte in size—200 times larger than what other cloud providers offer.
- Scalable performance to accommodate massively parallel analytics.
- Data can be stored in any format, without a schema.

This is a very different approach from the traditional data warehouse, where data schemas are defined upfront.

It is also possible to store the massive amounts of data generated from Internet of Things (IoT) devices (collecting temperature data, for example) in Data Lake Storage. By using Azure Data Lake Storage, it is possible to filter the data and create a view of it with multiple time granularities. Storing the data in Data Lake Storage is inexpensive and allows data to be stored for many years at a very low cost.

See Modern data warehouse for small and medium business for a reference architecture with Azure Data Lake Storage.

QUICK START

Get started with Azure Data Lake Storage using the Azure portal

Host static websites on Azure Storage

Another exciting feature of Azure Storage is static website hosting. This static website feature only uses Blob Storage as its data store and can be used to host a static website on Azure Storage. Simply upload the files of the static website to Blob Storage and indicate which file is the default document (such as index.html) and which one is the error document (such as 404. html). The website will run quickly for very little cost—in fact, only pay for the storage used, since the static website feature doesn't cost anything extra. Additionally, when using geo-redundancy (which is enabled by default), the website will be up and running even if the primary datacenter fails.

Azure data analytics solutions

Just as important as where and how an application stores data is how that data is analyzed to get business insights. Azure provides cloud-scale analytics solutions that cover all major data analytics scenarios and, in most cases, are faster and less costly than other cloud providers.

Whether the solution requires simple data movement and transformations, real-time analytics, or big data analysis driven by AI and machine learning models that requires days, weeks, or months of processing, Azure data analytics solutions enable businesses to get valuable and actionable insights from data to drive business outcomes.

No matter how large, small, or complex the data might be, Azure has a solution that can meet any data analytics requirement.

As shown in *Figure 4.2*, Azure provides solutions for ingestion, storage, operations, preparation, serving, and visualization. Each of these will be covered in more detail from a developer's perspective in the following sections.

For more architecture diagrams for data solutions, reference the [Azure Architecture repo](#).

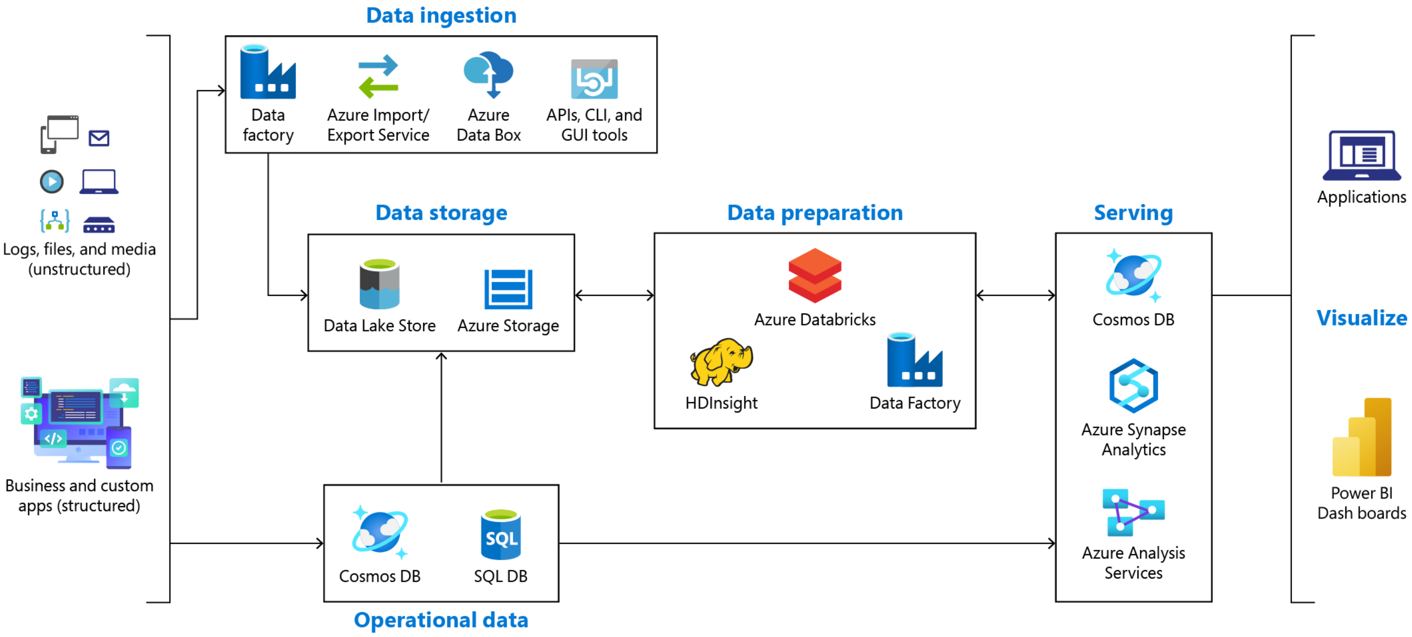


Figure 4.2: Azure Data Analytics solutions diagram

Azure Synapse Analytics

Azure Synapse Analytics (formerly Azure SQL Data Warehouse) brings together limitless enterprise data warehousing and big data analytics, providing the freedom to query data based on business needs, using either serverless or dedicated resources—at scale. Azure Synapse Analytics brings these worlds together with a unified experience to ingest, explore, prepare, manage, and serve data for immediate business intelligence and machine learning needs

Data engineers can use a unified, code-free visual environment for managing data pipelines, and business analysts can securely access datasets and use Power BI to build dashboards in minutes.

With Azure Synapse Analytics, enabling BI and machine learning is a breeze. It is deeply integrated with Power BI and Azure Machine Learning to expand the discovery of insights from data and apply machine learning models.

Azure Synapse Analytics offers a range of benefits. Some of these are:

- The flexibility of choosing to use serverless or dedicated resources.
- Using linked services and the 95+ native connectors, ingest data from data sources no matter where data resides—in Azure, in other clouds, or on-premises.
- Using pipelines, you can handle data transformations, data flows, define schedules, and so on.
- Bring together both relational and non-relational data, such as Cosmos DB and Azure Data Lake Storage.
- Perform interactive, batch, streaming, and predictive analytics with a rich T-SQL experience.
- Perform advanced analyses with Apache Spark using Python, Scala, R, and .NET.
- Apply an Apache Spark pool and Synapse pipelines in Azure Synapse Analytics to access and move data at scale.
- Deep integration of Azure Machine Learning, Azure Cognitive Services, and Power BI.
- Link Power BI workspaces to Azure Synapse Analytics workspaces to enable the ability to query and report on data via Power BI integration.

See Real-Time Analytics on Big Data Architecture for a sample architecture using Azure Synapse Analytics.

TUTORIAL

Creating a Synapse workspace

QUICK START

Create an Azure Synapse Analytics SQL pool

Azure Data Factory

With Azure Data Factory, you can create a comprehensive pipeline that performs a complete extraction, transformation, and loading (ETL) process.

Data Factory can help you move data from on-premises to the cloud, within the cloud, or to on-premises—it doesn't matter where your data resides. Data Factory also provides a variety of built-in connectors that you can use to easily connect to various data sources, including SQL Server, Azure Cosmos DB, Oracle, and many more.

While moving data, you can also filter it, clean it up, or transform it with an activity in the pipeline, such as the Apache Spark activity. In addition, Data Factory allows the scheduling and monitoring of pipelines, as well as a path to lift and shift SQL Server Integration Services (SSIS) packages to the cloud.

See Hybrid ETL with Azure Data Factory for a reference architecture using Azure Data Factory.

QUICK START

Create a data factory using the Azure Data Factory UI

Azure Synapse Link

Azure Synapse Link is Microsoft's preferred solution for analytics on top of Cosmos DB data.

Azure Synapse Link for Azure Cosmos DB is a cloud-native hybrid transactional and analytical processing (HTAP) capability that enables the running of near real-time analytics over operational data in Azure Cosmos DB. Azure Synapse Link creates a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.

Azure Analysis Services

Azure Analysis Services is used to create a semantic model of data, so users can access it directly with visualization tools such as Power BI. This is a managed cloud service—it is scalable, data is stored redundantly, and when not using it, you can pause the service to minimize costs.

With Azure Analysis Services, you can provide modeled data directly to users in a very performant way. Users can query millions of records in seconds because the model lives completely in-memory and is periodically refreshed. You can get data into the semantic model from anywhere; data sources can be in any cloud or on-premises.

See Enterprise business intelligence for a reference architecture using Azure Analysis Services.

QUICK START

Create an Azure Analysis Services server using the Azure portal

Azure Data Lake Analytics

Another Azure service for performing data analytics tasks is Azure Data Lake Analytics. With this service, you can analyze, process, and transform potentially massive amounts of data from Azure Storage and Azure Data Lake Storage.

Azure Data Lake Analytics allows you to create and submit jobs that query data, analyze it, or transform it. You can write these jobs in U-SQL, which is an SQL-like language, and extend U-SQL with Microsoft R and Python.

Only pay for the jobs that are submitted and run, and the service scales automatically depending on the power the jobs need. Azure Data Lake Analytics is typically used for long-running analytics jobs against massive amounts of data.

See Scalable Data Science with Azure Data Lake: An end-to-end Walkthrough for a sample of using Azure Data Lake Analytics.

TUTORIAL

Create your first U-SQL script through the Azure portal

Azure Stream Analytics

The Azure Stream Analytics service is used to analyze, query, and filter real-time streaming data. Stream Analytics can get its data from many services, including Azure Blob Storage, Azure Event Hubs, and Azure IoT Hub. You can analyze the data by using a simple SQL-like language or custom code. After querying and filtering the stream of data, Stream Analytics can natively output the result to many Azure services, including Azure SQL Database, Azure Storage, and Azure Event Hubs.

See Stream processing with Azure Stream Analytics for a reference architecture using Azure Stream Analytics.

QUICK START

Create a Stream Analytics job using the Azure portal

Azure Time Series Insights

You can use Azure Time Series Insights to get quick insights on large amounts of typically IoT-type data. This service gets data from Azure Event Hubs, IoT Hub, and custom reference inputs, and it retains that data for a specified amount of time.

With Azure Time Series Insights, users can query and analyze data through a visualization tool as soon as it comes in. Time Series Insights not only analyzes data but also ingests and holds it for a while. Time Series Insights is optimized for IoT and time-based data, and contains its own data visualization tool.

Check out the Azure IoT reference architecture which includes Azure Time Series Insights.

DEMO

Explore a Time Series Insights demo environment from your browser

Azure Databricks

Azure Databricks provides a unified analytics platform with a host of tools and capabilities. Within Databricks, you can run optimized versions of Apache Spark to do advanced data analytics.

In addition to Spark-based analytics, Databricks provides interactive notebooks and integrated workflows and workspaces that you can use to collaborate with the entire data team, including data scientists, data engineers, and business analysts, all of whom have access to specialized tools for their specific requirements.

Databricks is fully integrated with Azure Active Directory, which gives you the ability to implement granular security. With Databricks, you can perform Spark-based data analytics on data that comes from a variety of places. Additionally, you can plug Databricks into Power BI to create and show powerful dashboards.

See Ingestion, ETL, and stream processing pipelines with Azure Databricks for a sample reference architecture.

EXERCISE

Run a Spark job on Azure Databricks using the Azure portal

HDInsight

HDInsight allows you to run specialized clusters of open-source data analytics tools. The advantage of running these tools in HDInsight is that they're managed, which means you don't have to maintain VMs or patch operating systems. Plus, these tools can scale and easily connect to one another, other Azure services, and on-premises data sources and services.

You can run potentially massive, specialized clusters of different types, such as an Apache Hadoop cluster. It enables you to process and analyze data with Hadoop tools such as Hive, Pig, and Oozie. You can spin up an Apache HBase cluster, which provides a very fast NoSQL database, or you can create an Apache Storm cluster, which is geared toward analyzing data streams, and provides a framework for processing and analyzing massive amounts of data. HDInsight can also run a cluster for Microsoft Machine Learning Server (previously Microsoft R Server). Finally, you can create a cluster that runs Apache Kafka, which is a publish-subscribe messaging system used to build applications with queueing mechanisms.

See Interactive querying with HDInsight for a sample reference architecture using HDInsight.

TUTORIAL

Extract, transform, and load data using Apache Hive on HDInsight

Azure Data Explorer

Use Azure Data Explorer for a fast, fully managed, and highly scalable data analytics service for real-time analysis of large volumes of data streaming from applications, websites, IoT devices, and more.

Azure Data Explorer allows you to natively export Kusto queries that were explored in the web UI to optimized dashboards.

See Azure Data Explorer interactive analytics and Streaming using HDInsight for reference architectures.

Azure Data Studio

Azure Data Studio is a cross-platform database tool for data professionals using on-premises and cloud data platforms on Windows, macOS, and Linux.

Azure Data Studio offers a modern editor experience with IntelliSense, code snippets, source control integration, and an integrated terminal. It's engineered with the data platform user in mind, with the built-in charting of query result sets and customizable dashboards.

The source code for Azure Data Studio and its data providers is available on GitHub under a source code EULA that provides rights to modify and use the software, but not to redistribute it or host it in a cloud service.

Power BI

Power BI is a suite of business analytics tools that deliver insights throughout an organization. Connect to hundreds of data sources, simplify data prep, and drive unplanned analysis. Produce beautiful reports, and then publish them for the organization to consume on the web and across mobile devices.

With Azure services and Power BI, your application can turn data processing efforts into analytics and reports that provide real-time insights. Whether data processing is cloud-based or on-premises, straightforward, or complex, single-sourced, or massively scaled, warehoused, or real-time, Azure and Power BI have the built-in connectivity and integration to bring business intelligence efforts to life.

Power BI has a multitude of Azure connections available, and the business intelligence solutions allow for the creation of data insights unique to the business. Use Power BI to connect as few as one Azure data source, or many, and then shape and refine the data to build customized reports.

See Enterprise business intelligence for a reference architecture using Power BI.

Common Data Model

The Common Data Model simplifies bringing together data from multiple systems and creating a shared data language for business and analytical applications to use. The Common Data Model metadata system makes it possible for data and its meaning to be shared across applications and business processes such as Microsoft PowerApps, Power BI, Dynamics 365, and Azure.

Table 4.1 shows a comparison of the various capabilities offered by the Azure suite of data analytics services:

	Ingest data	Transform data	Query and filter streaming data	Provide an in-memory semantic model for users	Allow users to query data and create dashboards	Analyze data for users
Azure Synapse Analytics	○	○	○	○	○	○
Data Factory*	○	○				
Analysis Services*		○		○		
Data Lake Analytics*		○				○
Stream Analytics*		○	○			
Time Series Insights*		○		○	○	○
Azure Databricks*		○	○			
Azure HDInsight*		○	○	○		○

* Services with an asterisk have a free tier to get started with at no cost.

Table 4.1: Data Analytics in Azure

Azure Purview

The growing amount of data you have today makes it very difficult to discover and catalog. With Azure Purview, you can have a central location for your data catalog where you can register data sources across the enterprise. This enables better data understanding and a central location that provides a seamless data consumption process.

As data becomes fundamental for maximizing business value, managing and governing that data across growing on-premises, multi-cloud, and SaaS environments is essential.

Azure Purview sets the foundation for effective data governance by helping with automated metadata discovery, AI-powered classifiers, and end-to-end lineage. With the help of built-in connectors and Apache Atlas APIs, unify data across on-premises, across clouds, or in external SaaS apps such as Dropbox and Slack.

At the heart of Azure Purview is the data map, which provides automated scanning and classification of metadata at cloud scale.

Data consumers across the business interact with the data discovered in the Data Map using the Purview Data Catalog. The Data Catalog enables effortless discovery for data consumers by offering capabilities such as semantic browse and search, business glossaries, and visual data lineage.

Chief data officers and compliance teams can gain a bird's-eye view of their data, especially sensitive data, with Purview Data Insights, to assess data usage across the organization.

Additionally, Azure Purview enables the following:

- Improved data tracking and understanding through metadata. Data remains at its source location, but a copy of the metadata is added to Azure Purview with reference to the data source location. As metadata is indexed, a data source search is easy and understandable to users.
- Eliminates operational silos by enabling business and technical data analysts, data scientists, and data engineers to find trustworthy, valuable data.
- Tracks data through lineage, thereby allowing a view into data moves from one source to another through ETL processes and pipelines.
- Provides a better understanding of data changes and how data is affected by different processes and applications.
- Using classification and sensitivity label insights, the business can better track sensitive information.

See Data governance with Profisee and Azure Purview for a reference architecture using Azure Purview.

Azure IoT Solutions

The recent explosion of IoT devices, such as sensors, drones, and cameras, is driving significant growth in data at the edge. And the promise of 5G means even more data, from more endpoints, in more places, providing even more of a need for intelligence and analytics to be processed as close to the source as possible.

These applications and features of these IoT devices are virtually unlimited, but they all tend to have one thing in common: they generate a lot of data. In many cases, it is not practical to forward that data to Azure for processing, and it makes more sense to do the processing on the device, or even more likely, a specific edge device.

Azure provides a robust and scalable platform for managing these devices and then getting valuable insights from them.

Refer to the Azure IoT reference architecture to see how all the following solutions work together.

Azure IoT Hub

At the core of Azure IoT is Azure IoT Hub, a flexible cloud Platform as a Service (PaaS) that connects, monitors, and manages IoT devices in a secure and scalable manner.

You can use IoT Hub to ingest massive amounts of messages that typically come from IoT devices, such as messages that contain data from temperature sensors. What's more, IoT Hub establishes two-way communication with devices and allows for the execution of code on devices.

When devices send messages to Azure IoT Hub, it can either store them or route them to another service, such as Azure Event Grid, for analysis or action.

You can also create bidirectional communication tunnels using device streams. Azure IoT Hub device streams facilitate the creation of secure bidirectional TCP tunnels for a variety of cloud-to-device communication scenarios.

Azure IoT Hub Device Provisioning Service

The Azure IoT Hub Device Provisioning Service (DPS) provides Azure IoT Hub with zero-touch, just-in-time provisioning of devices to the appropriate IoT hub without intervention, allowing for the provisioning of devices in a secure and scalable manner.

Azure IoT Central

Azure IoT Central is a Software as a Service (SaaS) offering that enables IoT developers to create rich IoT applications just by navigating through wizards.

There's no need to perform any coding or in-depth configuration—IoT Central does it all by provisioning and configuring everything needed.

Azure IoT Central provides a jump-start in creating and running new IoT solutions via several industry-specific app templates for retail, energy, healthcare, and government. Alternatively, it is also possible to start with a blank, custom app template to build a fully customized solution with Azure IoT Central. Through app templates, it is possible to get up and running very quickly, without the need for years of programming experience. However, if further control and customization are needed, it is possible to tweak the solution according to solution requirements.

Azure IoT Edge

In modern IoT applications, data processing can occur in the cloud or on the device side. Device-side processing is referred to as "edge computing”. Edge computing is very useful for circumstances where it is not possible to rely on a connection to the cloud.

Azure IoT Edge is managed from IoT Hub, enabling the seamless transition of workloads to the edge. This reduces the time spent by devices sending messages to the cloud and allows offline scenarios as well as faster reactions to status changes.

It's possible to run many Azure services at the edge to help with certain scenarios—and the list of available services keeps growing. For example, using Azure IoT Edge, it is possible to push the Azure SQL Edge modules to devices and gain the power to stream, collect, and process that data right on the edge device itself, whether it is in a connected, semi-connected, or a disconnected environment. *Table 4.2* lists just some of these:

Service goal	Use this on Azure IoT Edge
Build and deploy AI models	Machine Learning
Customize computer vision models for use cases	Custom Vision Service
Process real-time streaming data	Stream Analytics
Process events using serverless code	Azure Functions
SQL Edge	Azure SQL Edge
Comply with Industry 4.0 interoperability standards	Azure Industrial IoT
Build custom logic	Custom module

Table 4.2: Services available on Azure IoT Edge

Using Azure IoT Edge, it is possible to run machine learning algorithms locally and provide instant feedback on their findings to local applications.

See IoT Edge data storage and processing for a reference architecture using IoT Edge.

Azure SQL Edge

Microsoft has extended the most secure Microsoft SQL engine and machine learning to the edge with Azure SQL Edge. By using Azure SQL Edge, devices gain the ability to:

- Stream, store, and analyze time series data using time-windowing, aggregation, and filtering capabilities, and achieve deeper insights by combining data types such as time series and graphs.
- Enjoy consistent app development and management, from the cloud to datacenters to the edge—develop once and deploy anywhere.
- Conduct real-time scoring, detect anomalies, and apply business logic at the edge using the built-in machine learning (ML) capabilities.
- Process data at the edge before forwarding it to the datacenter and cloud storage to optimize network bandwidth and cost.
- Deploy and update from Azure or the enterprise portal for consistent security and turnkey management. With SQL Edge, gain high availability and disaster recovery, as well as industry-leading data protection and security tools.
- Expand device architecture coverage to include ARM-based devices on top of x64-based architecture, choose Windows or Linux as the operating system, and run SQL Edge in a connected or disconnected environment.

Azure Digital Twins

The world of IoT tends to lend itself to many IoT devices and sensors that function in connection with environments and business systems. In many cases, extra context must be added when attempting to understand the data IoT devices capture. This can include the environment in which the various devices operate.

For instance, for a temperature sensor, temperature data by itself doesn't disclose much information without some extra context. When basic temperature data is supplemented with external data, such as which room the sensor is in, what data other devices in the room provide, how many people are in the room, or how people are moving in the room, a much better picture of what the temperature data means is created. By utilizing Azure Digital Twins, it is possible to more effectively model the physical environment in which the IoT devices reside.

Leveraging their domain expertise on top of Azure Digital Twins' flexible modeling, enterprises can build contextually aware solutions. With Azure Digital Twins' next-generation IoT solutions, customers can track the past, control the present, and predict the future.

There are many scenarios where Azure Digital Twins can be helpful:

- Predicting maintenance needs for a factory
- Analyzing real-time energy requirements for an electrical grid
- Optimizing the use of available space for an office
- Tracking daily temperatures across several states
- Monitoring busy drone paths
- Identifying autonomous vehicles
- Analyzing occupancy levels for a building
- Finding the busiest cash register in a store

Azure Digital Twins allows for the management of permissions to data and devices in the context of the physical world. By leveraging Azure Active Directory (Azure AD), it is possible to specify that only certain users can access data from a certain physical location.

QUICK START

`Get started by finding available rooms using Azure Digital Twins`

Azure Security Center for IoT

Azure Security Center for IoT provides threat prevention and analysis for every device, for both IoT Edge and IoT Hub, across the entire IoT solution. It provides unified visibility and control, along with adaptive threat prevention with built-in intelligent threat detection and response for your workloads wherever they reside—the edge, on-premises, in Azure, or in other clouds.

Windows for IoT

Windows 10 IoT is a member of the Windows 10 family that brings enterprise-class power, security, and manageability to IoT. It leverages Windows' embedded experience, ecosystem, and cloud connectivity, allowing organizations to create their IoT with secure devices that can be quickly provisioned, easily managed, and seamlessly connected to an overall cloud strategy.

Windows 10 IoT comes in two editions:

- **Windows 10 IoT Core** is the smallest member of the Windows 10 family. While only running a single app, it still has the manageability and security expected from Windows 10.
- **Windows 10 IoT Enterprise** is a full version of Windows 10 with specialized features to create dedicated devices locked down to a specific set of applications and peripherals.

Before attempting to manufacture a device, it's best to first prototype the device with Windows 10 IoT Core to gain an understanding of what features will be needed and what configurations are required when it's time to manufacture.

Windows Server IoT 2019

Windows Server IoT 2019 is a full version of Windows Server 2019 that comes with enterprise manageability and security for your IoT solutions. Windows Server IoT 2019 comes with all benefits of Windows ecosystems, offering a seamless experience with familiar tools for both development and management.

Acquire further knowledge on using Azure IoT solutions in the following free learning path:

RESOURCE HUB

Securely connect IoT devices to the cloud

Whatever your requirements for handling data are, Azure has multiple options to choose from. For storing data, you can choose different data storage options. For transactions, you can choose between services that can handle structured or unstructured data, or both. When it comes to data analytics, you can choose between different services that offer a premium data science experience. In IoT, you can again choose between different services that can accommodate any scenario. The best part is that all these services come with built-in integration with other Azure services and offer a seamless developer experience. Azure data services enable you to enrich your application, focus on what matters, and develop faster and with higher quality.

Learn more about data and data analytics in Azure by downloading and reading the following free e-books:

RESOURCE HUB

Proof of concept playbook

Azure Synapse Analytics Toolkit

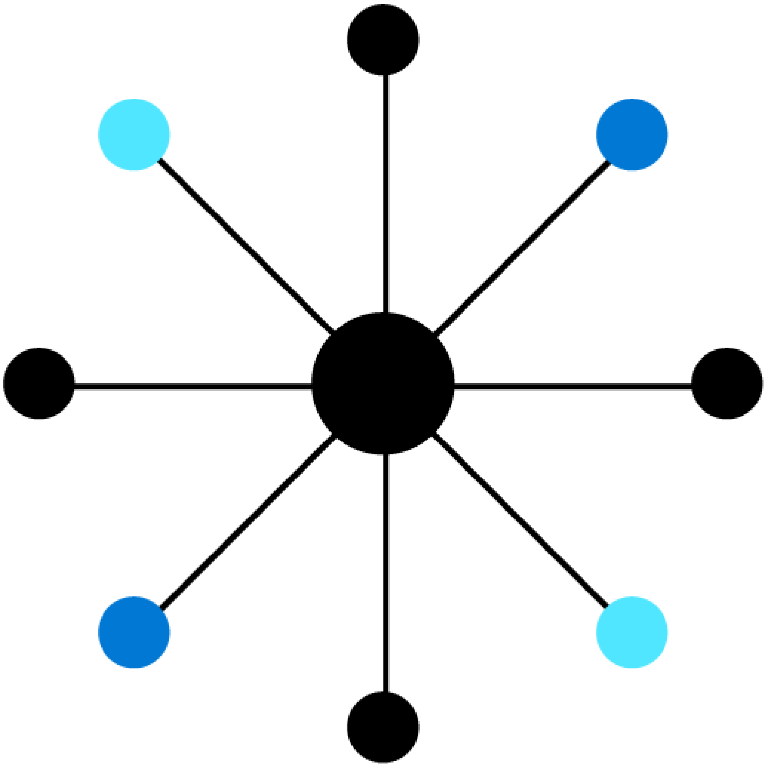
05 /

Adding intelligence to your application

The role of AI in modern application development

Artificial intelligence (AI) brings human-like capabilities into software. The ability to use AI and use it well can distinguish a software product from its competitors, streamline business processes, and reduce manual labor costs.

AI capabilities can be categorized into four pillars: visual perception, natural language processing, speech, and decision-making.



Visual perception

Visual perception allows the application to see. AI can interpret and process images and video streams using Azure services such as Custom Vision, Face, and Computer Vision that can automate image and video analysis for many purposes, including security and regulatory compliance. For example, AI can identify people to ensure that they are in approved work areas, identify whether they are wearing a hard hat in designated areas, and use spatial analysis to ensure a safe distance between a worker and dangerous machinery.

Natural language processing

Natural language processing (NLP) allows an application to understand written and spoken language. One of the most common application of NLP is in digital assistant software. In addition, NLP is also used in online chatbots, language translation, and speech output (synthesis) scenarios. Leveraging Azure services such as Text Analytics, Speech, and LUIS can improve overall customer satisfaction. For example, implementing a chatbot on the company website that can detect and converse with a user in their preferred language. It can also identify the sentiment of an interaction as being positive or negative to recognize potential areas of improvement and automation on the website.

Speech

Speech AI recognizes speech input and can synthesize spoken output. When combined with NLP, it enables a human-computer interaction that is known as conversational AI. As with the previous example of a website chatbot that interprets text, the same technology applies to the spoken word. The Azure Speech service provides speech-to-text, text-to-speech, speech translation, voice assistants, and speaker recognition capabilities.

Decision-making

Decision-making AI can use data from past experiences to apply correlations to current situations and take appropriate actions. Use Azure Machine Learning to develop custom models, or take advantage of the powerful inferencing engine readily available in the Anomaly Detector service. For example, the health and efficiency of factory equipment directly impacts the profitability of a company. Therefore, the company can use sensors on the equipment to gather real-time telemetry to obtain identifiable trends that lead to failure. These trends are encapsulated in a trained model and applied to the factory's incoming telemetry data stream. This practice enables predictive maintenance, warding off unexpected downtime and expense.

Why choose Azure AI?

Azure AI provides curated services built upon decades of research and responsible AI practices. Azure AI provides the tools and technology to responsibly deliver mission-critical AI solutions on your terms.

On your terms

Azure AI focuses on empowering developers of all skill levels and enabling them to use the tools and languages of their choice. Azure bridges the gap by making AI accessible for all skill sets.

Data scientists can use familiar tools to analyze data and train models using Azure Machine Learning. In addition, there are free-to-use standard libraries such as PyTorch, TensorFlow, and scikit-learn as well as notebook experiences that use popular languages like R and Python.

Those entirely new to the AI and machine learning (ML) space can choose from a comprehensive set of domain-specific pre-trained models or use AutoML to determine the best solution for a problem space. In addition, visual tools are available to ease into AI, such as Azure Machine Learning designer, Custom Vision, and Form Recognizer.

In addition to supporting the most popular ML frameworks, Azure AI also provides model portability to various form factors from devices, phones, databases, and cloud services. Furthermore, Azure AI services allow exporting of models in the ONNX format, which is optimal for model distribution.

Mission-critical

Azure AI services enable organizations to deploy mission-critical AI solutions with confidence as they are hardened for the most demanding workloads. It provides features for high-use and in-demand services at Microsoft, such as Office, Xbox, and Microsoft Teams. For example, over the course of a few months, more than 1.8 million hours of meetings were transcribed in real time with Microsoft Teams. Over 1 billion PowerPoint slides are consumed each day, and over 80 million personalized experiences are delivered daily on Xbox. Performing at scale is one of the main focuses of Azure AI.

Mature AI/ML organizations can leverage MLOps (also known as DevOps for machine learning) to ensure development, collaboration, model versioning, validation, deployments, monitoring, and governance through efficient and reproducible pipelines.

Responsible

Lastly, Azure AI has an uncompromising commitment to responsible AI. Azure AI is built upon the most trusted cloud, Azure. As a result, Azure boasts the best privacy controls, responsible AI capabilities, and the most compliance certifications of any cloud in the world.

This chapter provides an overview of Azure AI and mixed reality services, including:

- Azure Applied AI Services
- Azure Cognitive Services
- Azure Machine Learning
- Developer tooling for AI
- Mixed reality

Azure Applied AI Services

Azure AI is a set of services built upon the Azure infrastructure. These services are deeply integrated within Azure's data, app development, and compute services and tools. Azure Applied AI includes services like Azure Bot Service, Azure Form Recognizer, and Azure Cognitive Search. The goal of Azure Applied AI is to provide responsible AI capabilities to modernize existing business processes in an accelerated yet secure fashion. Additionally, you can visit Azure Resources for AI developers for self-paced learning resources.

Azure Bot Service

Azure Bot Service provides the tools and frameworks necessary for building conversational AI solutions, such as a virtual assistant for a website. The complexity of interpreting the context of an interaction with each user and providing rich, sensible, human-like responses is greatly simplified with AI.

The Bot Framework SDK is available for C#, Java, JavaScript, and Python. Developing a bot using the Bot Framework SDK does not require any Azure resources, as local development is possible via the SDK and the Bot Framework Emulator tool. If a more code-free experience is desired, the Bot Framework Composer tool provides a visual canvas to author bots. Azure Bot Service also provides integrations across multiple channels/products and device form factors without affecting the bot code. Using native integration with Azure Cognitive Services gives the bot the ability to speak, listen, and understand users.

Digital virtual assistants can be built using Power Virtual Agents with little or no code. With Power Platform and Azure Bot Service, you can use PaaS or SaaS platforms and start developing in a way that makes sense for you. There is also seamless integration between Power Virtual Agents and Azure Bot Service. This integration enables professionals from multiple disciplines to collaborate on a single solution, democratizing the creation of some parts of the conversational experience and accelerating innovation.

Azure Bot Service makes it easy to create a bot and provides the following support:

- It provides a way for hosting and managing bots built using the Microsoft Bot Framework.
- It integrates natively with Cognitive Services.
- It allows you to connect bots to customers' channels (Facebook, Microsoft Teams, Slack, and more).
- It is a fully managed service in Azure.

EXERCISE

Create a bot with Azure Bot Service

Azure Form Recognizer

Azure Form Recognizer is a data extraction service that applies advanced ML to accurately extract text, key-value pairs, tables, and structures from documents. Custom Azure Form Recognizer models can be trained by manually labeling a few sample documents and then implementing them into a production environment, either on-premises or in the cloud. Azure Form Recognizer also provides prebuilt models for many popular documents, such as receipts, business cards, invoices, and identity documents.

Azure Form Recognizer also offers flexible and secure deployments to ingest data from documents in the cloud or at the edge. Data extracted can then be enhanced by applying search indexes, business automation workflows, and more. As with all Azure services, Azure Form Recognizer takes advantage of Azure's built-in enterprise-grade security to protect data and model assets.

Azure Cognitive Search

Azure Cognitive Search is the only cloud search service with built-in AI capabilities that can enrich all types of information, such as vision, language, speech, or even custom models to identify and explore relevant content at scale. In addition, Azure Cognitive Search uses decades of experience with the Microsoft natural language stack currently integrated into Bing and Office products. As a result, developers can spend more time innovating and less time maintaining a complex cloud search solution.

Azure Cognitive Search provides developers with approachable infrastructure, APIs, and tools to build a rich search experience over private, heterogenous content in web, mobile, and enterprise applications.

There are many options available for working with Azure Cognitive Search and great features to make searching easier for users, including the following:

- Geo-search lets users explore data based on the proximity of a search result to a physical location.
- Language analyzers from Apache Lucene and NLP from Microsoft are available in 56 languages.
- Semantic search capabilities, powered by deep learning models that understand user intent, surface and rank the most relevant search results.
- Monitoring and reporting provide details surrounding search terms and how performing the search was.
- User experience features, such as sorting and paging search results, and intelligent filtering.

QUICK START

Create your first Azure Cognitive Search index in the portal

Azure Metrics Advisor

Azure Metrics Advisor is an AI analytic service that proactively monitors metrics and diagnoses issues in time series data. The service automates the process of applying models to your data. It provides a set of APIs and web-based workspace for data ingestion, anomaly detection, granular analysis, and diagnostics without requiring knowledge of ML concepts.

Azure Metrics Advisor enables building AIOps, predictive maintenance, and business monitoring applications on top of the service. Azure Metrics Advisor can:

- Analyze multidimensional data from multiple data sources
- Identify and correlate anomalies
- Configure and fine-tune the anomaly detection model used on your data
- Diagnose anomalies and help with root cause analysis
- Provide real-time notifications through email, web, Teams, and Azure DevOps hooks

GET STARTED

Azure Metrics Advisor

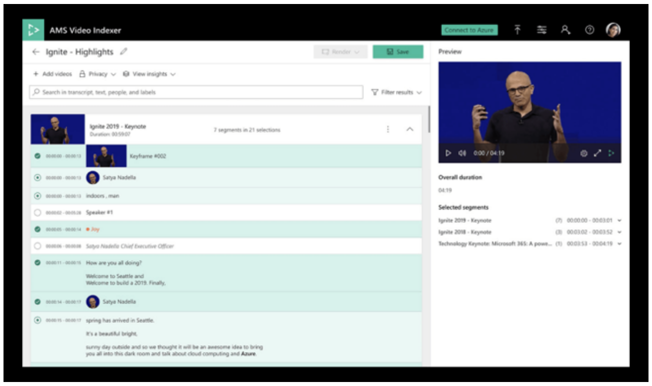
Azure Video Analyzer

Azure Video Analyzer is built upon Azure Media Services and Azure Cognitive Services. The Azure Video Analyzer service can analyze and extract face, language, vision, and speech data from audio and video files using a pre-built model. Media files may be uploaded to the service using the Video Analyzer portal or the API.

In addition to many others, Azure Video Analyzer has the following key capabilities:

- Transcribing of text in a video. The resulting transcript can be refined manually and used to train Azure Video Analyzer to recognize industry terms such as "DevOps."
- Tracking faces and identifying who is in the video and at what timestamp. Azure Video Analyzer has the same capability for audio, for which it recognizes who is speaking and when.
- The service recognizes visual text in a video, such as text on a slide, and makes it part of the transcript.
- Azure Video Analyzer can perform sentiment analysis, identifying when something positive, negative, or neutral is said or displayed.

Figure 5.1 shows the results layout in the Azure Video Analyzer service:



How are you all doing?

Welcome to Seattle and
Welcome to build a 2019. Finally,

spring has arrived in Seattle.

It's a beautiful bright,

sunny day outside and so we thought it will be an awesome idea to bring you all into this dark room and talk about cloud computing and **Azure**.

Figure 5.1: Demonstrating the results in the Video Analyzer portal

Figure 5.1 displays the result of creating a transcript of audio and video from a media file. The transcript is editable and translatable to other languages. Azure Video Analyzer has also recognized text on the slide behind the speakers and marks it as "OCR." Azure Video Analyzer provides this functionality for individual applications by embedding the Cognitive Insights widget.

TRY IT OUT

Upload your first video to Azure Video Analyzer

Azure Immersive Reader

Azure Immersive Reader is an inclusive tool designed for implementing proven techniques to improve reading comprehension for new readers, language learners, and people with learning differences such as dyslexia. Azure Immersive Reader also supports translations to more than 100 languages.

With the Azure Immersive Reader client library, you can leverage the same technology used in Microsoft Word and Microsoft OneNote to improve accessibility in custom applications.

QUICK START

Make your first Azure Immersive Reader request

In this section, we looked at Azure Applied AI services and tools. Learn more at Azure Resources for AI developers to follow a self-paced learning path. In the next section, we will look at Azure Cognitive Services.

Azure Cognitive Services

Azure Cognitive Services is a set of pre-trained, customizable AI models based on Microsoft AI research, enabling access to sophisticated language, vision, decision-making, and speech capabilities through simple API calls. Azure Cognitive Services does not require previous ML experience for integration. In addition, many of these existing models are extensible by training with custom data to fit specific knowledge domains.

Cognitive Services provides a robust set of APIs to incorporate ML and AI into applications.

Table 5.1 shows a series of categories and a list of APIs that are currently available. Note that this list keeps growing, and we will cover a few of these services in the following sections.

Decision	Language	Speech	Vision
Anomaly Detector Content Moderator Personalizer	Language Understanding QnA Maker Text Analytics Translator	Speech-to-Text Text-to-Speech Speech Translation Speaker Recognition	Computer Vision Custom Vision Face API

* All services have a free tier that you can use to get started.

Table 5.1: A quick glance at the Cognitive Services APIs

Custom services, such as Custom Vision and Language Understanding, provide pre-configured ML models and a visual interface to train custom models using domain-specific data and imagery specific to the application under development.

In addition to these services, Cognitive Research Technologies contain innovative APIs and SDKs for researchers and developers looking for emerging cognitive capabilities. One such experimental service is Project Gesture, which enables the AI model to recognize gestures such as waving a hand and use them as actionable feedback in application user experiences.

Let's take a closer look at some Cognitive Services offerings.

RESOURCE HUB

Hands on with AI

Access a 4-week learning path to grow your skills and get certified

Decision

In this category, we look at Cognitive Services offerings that help you make decisions based on trends in data; we will cover the Anomaly Detector and Personalizer services.

Anomaly Detector

Anomaly Detector allows embedding of anomaly detection capabilities into applications. For example, automated alerts, triggering emergency workflows, or providing a visual cue on a monitoring interface for users to watch incoming time series data for inconsistencies.

Using Anomaly Detector doesn't require any prior experience in ML. The RESTful API enables developers to integrate the service into an application and process it quickly.

Personalizer

Personalizer helps applications choose the most relevant content to show a user based on their user behavior, collective trends, and real-time information provided by the current context.

Content can be any unit of information, such as text, images, URLs, or emails. Personalization helps boost application usability and enhances user satisfaction, as the Personalizer service is based on reinforcement learning capabilities.

Language

Azure Cognitive Services also has services to help you with language understanding and comprehension.

Language Understanding (LUIS) service

Use the LUIS service to understand the semantic meaning of what users are saying on social media, in chatbots, or speech-enabled applications. For example, let's look at how a user can book a flight using LUIS.

A list of examples of what a user may say (utterances) is provided to the LUIS service to train a model. These examples could be "Book a flight to Seattle" or "Cancel my flight to Washington D.C." From these utterances, the intent of the user must be determined. An intent represents a task or action that the user wants to accomplish. Thus, the intent is the purpose or the goal of a user's utterance.

After the LUIS service creates an ML model based on the examples provided, it can extract information from the natural language that users put in.

STEP BY STEP

Create a new application in the LUIS portal

Translator

Translator is an AI service for real-time and batch text and document translation. This service provides translations across 90 languages and dialects, powered by the latest innovations in neural machine translation.

Translator supports various use cases, such as translation for call centers, web page localization, and internal enterprise communications.

With Translator, you can create customizable translations by building custom models to handle domain-specific terminology. Translator also offers secure and flexible deployment options, including deployment of Translator as a containerized application.

QUICK START

Get started with Translator

This section looks at services for transcribing speech to text and converting text to speech to enable conversational experiences. This functionality is made available in applications via the Speech SDKs available in multiple programming languages and via the full REST API. As with many Azure AI services, multiple model deployment options are available. By deploying models as a container, data does not need to leave the cloud to be processed, resulting in a more secure solution as the power of Speech services is brought closer to the data.

Speech-to-Text

Speech-to-Text is an AI service that accurately transcribes spoken audio to text. It allows quick and accurate audio transcription in more than 85 languages and variants.

Speech-to-Text enables the creation of customized models to enhance accuracy for domain-specific terminology. It can extract additional value from spoken audio by enabling search or analytics on transcribed text or facilitating an action based on sentiment.

QUICK START

`Get started with Speech-to-Text`

Text-to-Speech

Text-to-Speech is an AI service that converts text to lifelike speech. It enables building apps and services that speak naturally through speech synthesis.

With Text-to-Speech, choose from more than 250 voices and 70 languages and variants. Differentiate applications using a customized voice and use voices with different speaking styles and emotional tones to fit specific use cases. The application of Text-to-Speech is common in text readers and customer support chatbots.

With the level of customization available in Text-to-Speech, create lifelike voices that are unique to an organization.

QUICK START

`Get started with Text-to-Speech`

Vision

In this category of Cognitive Services, we look at APIs that help you extract information and create meaning from images and videos.

Custom Vision

The Custom Vision service can train custom models based on images specific to the application domain. Creating a Custom Vision model is as simple as uploading and labeling a few images to provide the service with training data. While only five images are required to begin the training process, more images can be uploaded and tagged to create a more accurate model.

You can use custom models by making calls to the Custom Vision API and providing it new images—the service will identify objects it has been trained to seek.

The model created with the Custom Vision service can be deployed to the "intelligent edge," meaning the model and API can run somewhere other than the cloud, such as on an on-premises server in a Docker container or on a separate device, such as a phone. This flexible deployment option enables disconnected scenarios as these services run locally and do not require an internet connection. A local deployment also has a relatively small footprint—only the model and the API need to be deployed, not the training data.

In addition to Custom Vision, the Computer Vision API offers built-in intelligence to process images and return information based on visual features. In addition, Computer Vision boasts in-demand AI services such as optical character recognition (OCR), Image analysis, and Spatial analysis.

QUICK START

Create your own Custom Vision project

In summary, Microsoft Azure offers the most comprehensive set of Cognitive Services offerings. These services are pre-trained, customizable AI models that are all based on Microsoft AI research. These services do not require prior machine learning or AI experience and are integrated into applications via SDKs and REST APIs. Trained models are also portable and consumable on-premises and in the cloud.

Azure Machine Learning

Azure Machine Learning empowers developers and data scientists with a wide range of productive experiences for building, training, and deploying machine learning models more quickly. It also helps accelerate the time to market and foster team collaboration with industry-leading MLOps (DevOps for machine learning).

First, let's introduce ML.

What is machine learning?

ML is a field of computer science that gives computers the ability to learn without being explicitly programmed. ML is achieved using one or more algorithm technologies, such as neural networks, deep learning, and Bayesian networks.

What's involved in ML? *Figure 5.2* shows the basic workflow of ML:

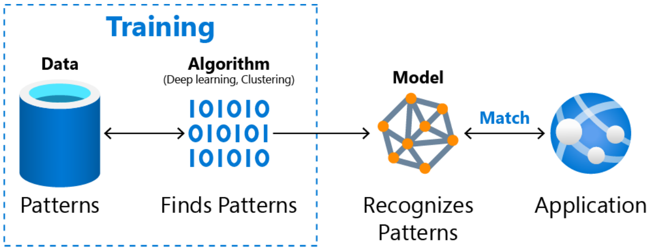


Figure 5.2: Basic workflow of ML

The ML process works as follows:

- Data containing patterns is collected and prepared for the ML algorithm.
- The ML algorithm is used to train a model to identify these patterns.
- The trained model is deployed so that it can be used to recognize patterns in new datasets.
- Applications use services or libraries to use the trained model and take actions based on the results.

The crucial part of this process is that it is iterative. Thus, the ML model can be improved constantly by training it with new data and adjusting the algorithm to distinguish correct results from wrong ones.

RESOURCE HUB

A data scientist's guide to building on top of Azure AI

Azure Machine Learning service

Azure Machine Learning is applied in various scenarios, such as predictive analytics, data recommendations, and data classification. This platform appeals to new and existing data engineers and data scientists as it supports many popular languages, such as Python, R, and the Azure CLI, and open-source technologies such as TensorFlow, PyTorch, and scikit-learn. In addition, AutoML and Azure Machine Learning designer provide a low-code/no-code entry system for those who need little assistance getting started with ML concepts.

Azure Machine Learning is a complete service that offers end-to-end capabilities. Prepare data; train, test, and deploy models, and track their lifecycle through the model registry. For example, a data scientist creates a notebook to train and register a model. This notebook can be run within [Azure Machine Learning workspace](), [Synapse Notebooks](), or [Azure Databricks](). The data scientist can then deploy the model on a Kubernetes container cluster in [Azure Kubernetes Service]().

QUICK START

Get started with Azure Machine Learning by using the Azure portal

Azure Machine Learning studio

[Azure Machine Learning studio]() is the web portal experience for data scientists and developers. Azure Machine Learning studio combines no-code (designer) and code-first (notebook) experiences as an inclusive data science platform. Users can choose their experience based on the type of project and the level of their expertise. Azure Machine Learning studio also offers an automated ML experience where multiple ML experiments are run in parallel to identify the ideal algorithm for a scenario, all via the studio's intuitive user interface.

Azure Machine Learning designer

Azure Machine Learning designer is the code-free approach to prep data and train, test, deploy, manage, and track machine learning models. There is no programming required—each step is visually constructed using drag-and-drop modules.

The Azure Machine Learning designer is a feature of the Azure Machine Learning workspace resource. This workspace acts as a centralized place to work with and store all the artifacts related to ML.

In the designer, a project begins by creating a pipeline from scratch or by starting with one of the many prebuilt samples, including one for predicting flight delays and another for customer relationship management (CRM) prediction:

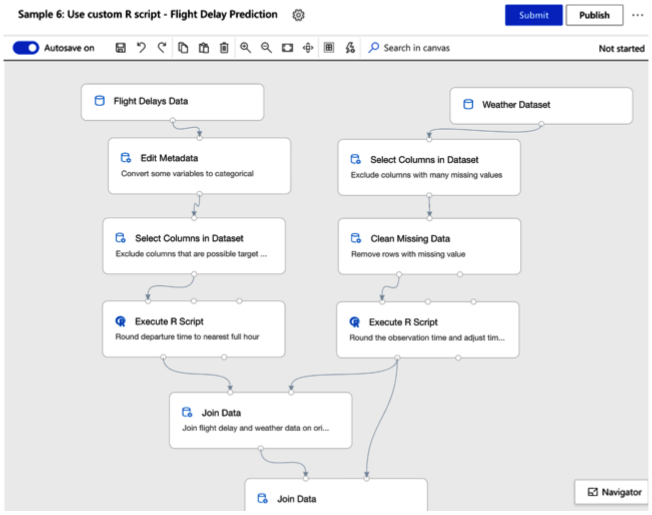

Sample 6: Use custom R script - Flight Delay Prediction
Submit
Publish
Autosave on
Search in canvas
Not started
Flight Delays Data
Weather Dataset
Edit Metadata
Convert some variables to categorical
Select Columns in Dataset
Exclude columns with many missing values
Select Columns in Dataset
Exclude columns that are possible target ...
Clean Missing Data
Remove rows with missing value
Execute R Script
Round departure time to nearest full hour
Execute R Script
Round the observation time and adjust tim...
Join Data
Join flight delay and weather data on ori...
Join Data
Navigator

Figure 5.3: Using a custom R script for flight delay prediction

Figure 5.3 shows a pipeline in Machine Learning designer that uses a customized R script to predict whether a scheduled passenger flight will be delayed by more than 15 minutes. This particular pipeline was created using the "Sample 6: Use Custom R Script—Flight Delay Prediction" sample pipeline.

TUTORIAL

Predict automobile prices with the designer

AutoML

Automated machine learning, also referred to as AutoML, automates the time-consuming, iterative tasks of ML model development. Traditional ML model development is resource-intensive, requiring significant domain knowledge and time to produce and compare dozens of models. This process is automated with AutoML, resulting in obtaining production-ready ML models quickly and efficiently.

During training, Azure Machine Learning spawns several pipelines in parallel to analyze various algorithms and parameters. Each iteration produces a model with a training score. The higher the score, the better the model is considered to "fit" your data. AutoML will stop once it hits the exit criteria defined in the experiment.

In this section, we looked at the tools and capabilities that are part of Azure Machine Learning. Azure Machine Learning can save time, improve model accuracy, and enable reliable deployments when building custom models.

Developer tooling for AI

This section looks at two main frameworks that can help developers infuse AI into their applications; ML.NET and AI Toolkit for Azure IoT Edge. First, let's explore these frameworks.

ML.NET

ML.NET is an open-source and cross-platform ML framework with support for macOS, Windows, and Linux. ML.NET brings ML to .NET developers, allowing them to integrate ML into new or existing web, mobile, desktop, gaming, and Internet of Things (IoT) applications.

ML.NET Model Builder provides an easy-to-understand visual interface to build, train, and deploy custom ML models. Prior ML expertise is not required. Model Builder supports AutoML, which automatically explores different ML algorithms and settings to help you find the one that best suits your scenario.

QUICK START

`Get started developing on Azure AI`

AI Toolkit for Azure IoT Edge

Using ML models locally on devices (the intelligent edge) delivers a powerful advantage: it enables disconnected, local processing on a device without relying on an internet connection or incurring the latency of a web service call to get results.

The AI Toolkit for Azure IoT Edge provides tooling to package machine learning models in Azure IoT Edge-compatible Docker containers and expose those models as REST APIs. The Docker containers are deployed as an IoT edge module on the device and run by to the local IoT Edge runtime infrastructure.

The AI Toolkit for Azure IoT Edge contains examples for getting started, is fully open source, and is available on GitHub.

In summary, using these frameworks allows developers to reuse all their knowledge and skillsets to start building intelligent algorithms and experiences without having to start from scratch. With ML.NET and Azure services such as Azure Machine Learning, developers can be productive quickly and easily.

Mixed reality

Applications are no longer limited to a 2D environment. The world is now the canvas for applications. Various IoT sensors, mixed reality, and computer vision are combined with spatial intelligence to bring data to life in 3D.

Azure Spatial Anchors

In the world of mixed reality, integrate digital information within the context of a physical environment. As an example, render a hologram of a game character on the kitchen counter. With Azure Spatial Anchors, digital content is placed in physical locations and consumed by users using your choice of devices and platforms.

Here are some example use cases enabled by Spatial Anchors:

- **Multi-user experiences** make it easy for people in the same place to participate in shared mixed-reality experiences.
- **Way-finding** is a method of connecting two or more spatial anchors and creating a relationship between them. These connected points of interest make an experience where the user must interact with them to complete a task.
- **Persisting virtual content in the real world** can enable a user to place an object (such as a calendar) on a room wall that people can see using a phone application or a HoloLens device.

Azure Spatial Anchors works with applications built on Unity, ARKit, ARCore, and Universal Windows Platform (UWP) and are consumed with a HoloLens device, iOS-based devices supporting ARKit, and Android-based devices supporting ARCore.

As with all Azure services, Azure Spatial Anchors and spatial data can be made accessible to users through Azure Active Directory.

TUTORIAL

Get started by sharing spatial anchors across sessions and devices

Remote Rendering

When using 3D models in scenarios such as design reviews and medical procedure plans, they need to be as detailed as possible—every detail matters.

With Remote Rendering, 3D models are rendered in the cloud and streamed to devices in real time—with no compromise in visual quality.

Azure Kinect DK

Azure Kinect DK is a developer kit with advanced AI sensors that provides sophisticated computer vision and speech models. Kinect contains depth sensors, a spatial microphone array with a video camera, and an orientation sensor as an all-in-one small device with multiple modes, options, and SDKs.

The Azure Kinect DK development environment consists of multiple SDKs:

- A **Sensor SDK** for low-level sensor and device access
- A **Body Tracking SDK** for tracking bodies in 3D
- A **Speech Cognitive Services SDK** for enabling microphone access and Azure cloud-based speech services

This chapter covered the many Azure services and tools that can help build intelligent apps and services. The choice to consume a prebuilt model or to develop a new custom model is available to all skill levels. Azure AI services allow you to build on your own terms and deploy mission-critical workloads with enterprise-grade security and scalability.

06 /

Securing your application

How can Azure help to secure your application?

Have you ever had a security incident with one of your applications? You might have had one without even knowing it. With Azure, you can protect data, applications, and infrastructure with built-in security services that include security intelligence to help you rapidly identify evolving threats, enabling you to respond promptly.

Azure can also help you implement a layered, in-depth defense strategy across identity, data, hosts, and networks. With services like Azure Security Center, you can get an overview of your security posture, protect against threats, and view recommendations for how to improve security.

Most importantly, you'll be notified as soon as there might be a security incident—so you'll always know whether there is a threat. This way, you can take immediate steps to secure your assets.

In this chapter, we will cover the following topics to help you understand how Azure can help you to secure your application:

- Identity
- Application security
- Posture management
- App access and connectivity
- Logging and monitoring
- Encryption

Identity

An important part of your application's security is authenticating users before they can use it—but authentication is not an easy thing to implement. You need to store user identities and credentials somewhere, implement credentials management, create a secure authentication handshake, and so on. In this section, we will look at some of the services and tools that Azure offers to make it easy for you to authenticate your users and secure your applications.

Microsoft identity platform

Microsoft identity platform (Azure Active Directory) provides all of the things previously listed, and more, out of the box. You store your user identities in Azure AD and have users authenticate against it, redirect them to your application only after they have been authenticated. Azure AD takes care of password management, including resolving common scenarios such as forgotten passwords. Azure AD Conditional Access takes this further to enable organizations to set intelligent policies for granular access control.

Since Azure AD is used by millions of applications every day—including the Azure portal, Outlook.com, and Microsoft 365—it's able to more readily detect and act on malicious behaviors using Azure AD Identity Protection. For instance, if a user were to sign in to an application from a location in Europe and then, one minute later, sign in from Australia, Azure AD would flag this as malicious behavior and ask the user for additional credentials through multi-factor authentication.

The Azure AD enterprise identity service provides single sign-on (SSO) and multi-factor authentication to help protect your users from 99.9 percent of cybersecurity attacks.

Azure AD Application Proxy

Azure AD Application Proxy provides SSO and secure remote access for web applications hosted on-premises. Applications that you would likely want to publish include SharePoint sites, Outlook Web Access, or other line-of-business (LOB) web applications. These on-premises web applications integrate with Azure AD, the same identity and control platform used by Microsoft 365. End users can access your on-premises applications the same way they access Microsoft 365 and other SaaS applications integrated with Azure AD.

Azure AD Identity Protection

Azure AD Identity Protection is a cloud-based tool that helps organizations protect user identities, as well as detecting and investigating identity-based risks. The tool also enables the export of risk detection data to your Security Information and Event Management (SIEM) tool.

Azure AD Identity Protection is powered by intelligent algorithms that analyze 6.5 trillion daily signals from Azure AD, Microsoft accounts, and Xbox accounts. This analysis enables you to take advantage of Azure AD Identity Protection to identify many risk types, such as leaked credentials, unfamiliar sign-in properties, malware-linked IP addresses, atypical travel, and many more.

DOCS

Learn more about Azure AD Identity Protection

Key Vault

As part of your security architecture, you need a secure place to store and manage certificates, keys, and other secrets. Key Vault provides this capability. With Key Vault, you can store the secrets that your applications use in a single secured, central location that leverages the FIPS 140-2 Level 2 validated Hardware Security Module (HSM).

DOCS

What is Key Vault?

One example of using Key Vault with a web application is to use it to securely store a connection string. Your application would get the connection string from Key Vault instead of the configuration system. This way, administrators can control the secrets, and developers never need to worry about them. Key Vault also stores SSL and other certificates used to secure the traffic to and from your applications over HTTPS.

Managed identities for Azure resources

How do you keep credentials out of your code completely? You can start by using Key Vault, but where do you store the credentials to connect to Key Vault? The managed identities for Azure resources feature provides a solution. You can use managed identities for a lot of services in Azure, including Azure App Service. You simply fire up a managed identity with a button to enable your application to acquire Azure AD tokens at runtime and then use those credentials to access other services, including Key Vault, Azure SQL Database, and Azure Storage. Credentials are managed completely by the infrastructure. Your application can simply authenticate with other services without you having to worry about protecting or rotating credentials.

LEARN MORE

How to use managed identities for Azure resources in App Service and Azure Functions

Application security

Application security is about securing your applications, their data, and the interactions between the different components of your apps. In the previous section, we looked at authenticating your users. In this section, we cover the different Azure services that enable you to secure your applications.

Azure Front Door

Azure Front Door is a global, scalable entry point that uses the Microsoft global edge network to create fast, secure, and highly scalable web applications. With Azure Front Door, you can transform your global consumer and enterprise applications into robust, high-performing personalized modern applications with content that reaches a global audience through Azure.

Azure Front Door provides enterprise-grade global load balancing to boost your applications' reliability, performance, and security. With Azure Front Door, you can always keep traffic on the best path to your application, improve your service scale, reduce latency, and increase throughput for your global users with edge load balancing, SSL offload, and application acceleration.

Azure Front Door also offers a modern content delivery network (CDN) with built-in security. It protects you against network- and application-layer attacks at the edge with Web Application Firewall, Bot Protection, and DDoS Protection.

Application Gateway

Application Gateway is a dedicated virtual appliance that provides an application delivery controller (ADC) as a service. It offers various layer 7 load balancing capabilities for your application and allows customers to optimize web farm productivity by offloading CPU-intensive SSL termination to Application Gateway. Application Gateway also provides other layer 7 routing capabilities, including the round-robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single application gateway.

Azure Web Application Firewall

Azure Web Application Firewall (WAF) is a managed, cloud-native service that provides powerful protection for your web applications. Azure WAF helps protect your web applications from malicious attacks and common web vulnerabilities, such as SQL injection and cross-site scripting.

Azure WAF detects malicious attacks, as defined in the OWASP Core Rule Set, and blocks those attacks from reaching your application. It also reports on attempted or ongoing attacks so that you can see active threats to your application, thereby providing an extra layer of security.

LEARN MORE

Read about Azure Web Application Firewall

Azure WAF can also be seen as an overlay service on top of Application Gateway and Azure Front Door. To help you decide which service to use for your scenario, *Figure 6.1* shows a simple flow chart:

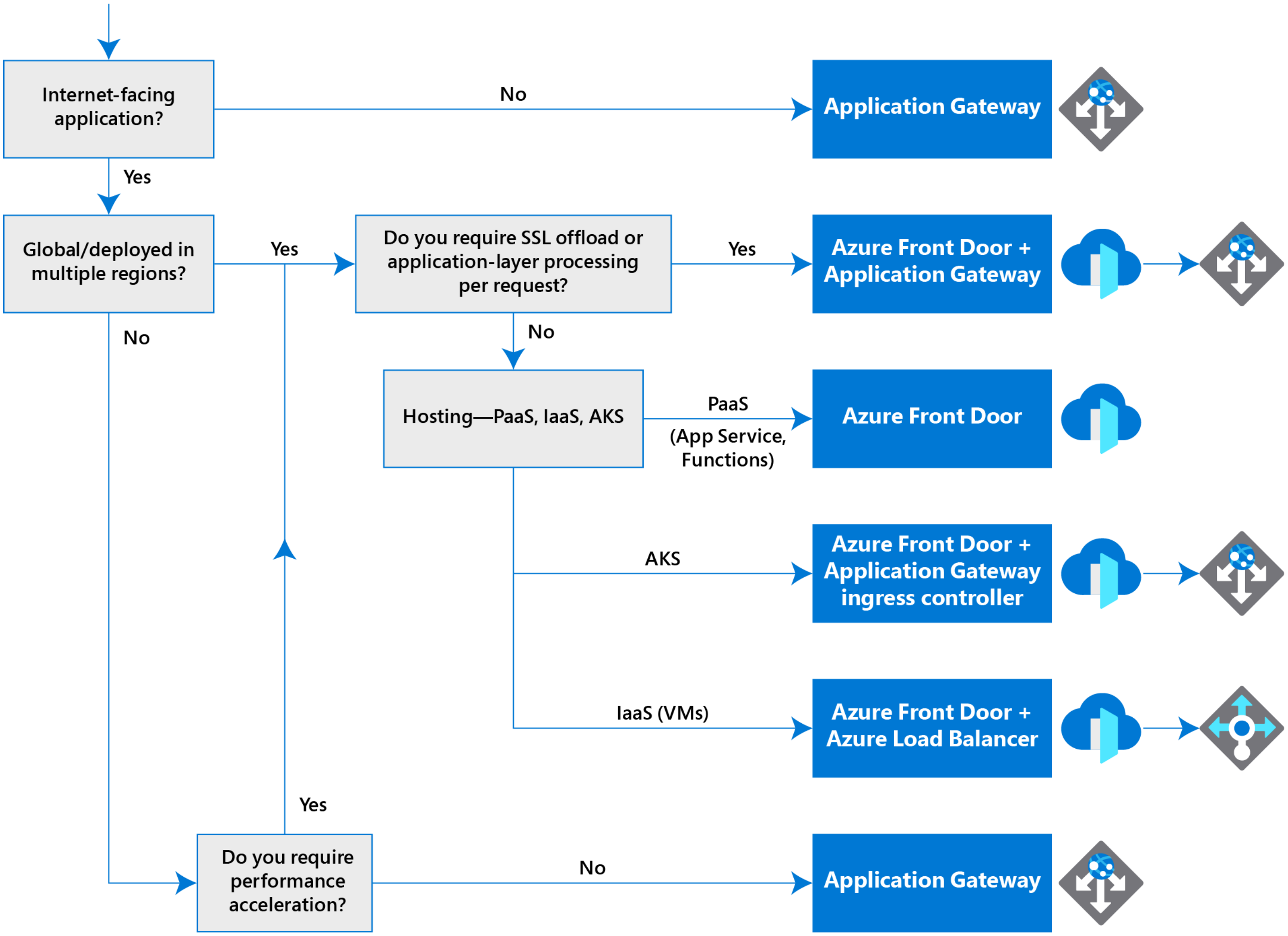

Internet-facing application?
No
Application Gateway
Yes
Global/deployed in multiple regions?
Yes
Do you require SSL offload or application-layer processing per request?
Yes
Azure Front Door + Application Gateway
No
No
Hosting—PaaS, IaaS, AKS
PaaS
(App Service, Functions)
Azure Front Door
AKS
Azure Front Door + Application Gateway ingress controller
IaaS (VMs)
Azure Front Door + Azure Load Balancer
Yes
Do you require performance acceleration?
No
Application Gateway

Figure 6.1: Decision tree for choosing the right Azure service based on your requirements

Azure Firewall

Azure Firewall is a managed, cloud-native network security service that protects your Azure Virtual Network resources. It's a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability.

Azure Firewall enables you to have centralized network- and application-level connectivity policy controls as well as intelligence-based traffic filtering. Azure Firewall has built-in TLS inspection for your selected encrypted applications and offers the ability to detect and block malicious traffic through an advanced IDPS engine.

You can use Azure Firewall to secure your Azure virtual networks in hybrid connectivity scenarios through deployments behind VPN and ExpressRoute gateways.

LEARN MORE

Read about Azure Firewall

Azure DDoS Protection

You've heard about it on the news, and you certainly don't want it to happen to your enterprise: applications are targeted by distributed denial of service (DDoS) attacks all the time. These types of attacks are becoming more common and can overwhelm your application to the point that no one can use it anymore. DDoS Protection offers protection from DDoS attacks through a free tier (Basic) and a paid tier (Standard).

You don't have to do anything to enable the Basic tier—it's automatically enabled for every customer as part of the Azure platform. This service protects your applications against the most common DDoS attacks by performing real-time monitoring and mitigation, and it provides the same defenses as those used by Microsoft Online Services (MOS).

The Standard tier provides additional mitigation capabilities that are tuned specifically to Azure Virtual Network resources. It's simple to enable, and you don't have to change your applications—everything is done at the network level. Plus, with the Standard tier, you can customize the Basic tier protection with your own policies that focus on your specific use cases and applications.

DOCS

Azure DDoS Protection

Posture management

Securing your application is a dynamic challenge that requires you to have the right tools to monitor and investigate threats quickly and efficiently. This is where you will need tools such as Azure Security Center and Azure Defender to have a centralized view for monitoring and policy controls. Depending on your needs and the security requirements of your organization, you might be able to address all your requirements with Azure Security Center, or you might need to look into Azure Defender.

Azure Security Center

[Azure Security Center](#) provides unified security management and advanced threat protection across hybrid cloud workloads. It offers centralized policy controls to limit exposure to threats and rapidly find and fix vulnerabilities.

In addition, Security Center supports integration with third-party solutions and can be customized with automation and programming capabilities.

You can use Security Center to analyze the security state of your compute resources, virtual networks, storage and data services, and applications.

Continuous assessment helps you discover potential security issues, such as systems with missing security updates or exposed network ports. A list of prioritized findings and recommendations can trigger alerts or other guided remediation.

DOCS

`Azure Security Center`

Azure Defender

Azure Defender is a cloud-native tool that provides threat protection for workloads running in Azure, on-premises, and in other clouds. It's natively integrated with Azure Security Center and can integrate with your existing security workflows, such as SIEM solutions and vast Microsoft threat intelligence, to streamline threat mitigation.

Azure Defender protects your hybrid cloud workloads against threats. Azure Defender enables extended detection and response (XDR) capabilities to protect your workloads against threats such as remote desktop protocol (RDP) brute-force attacks and SQL injections.

You can use Azure Defender to ensure the security of your Azure resources. It protects your data in Azure VMs, on-premises, and in other clouds, and it detects unusual attempts to access storage accounts and malware uploads to Azure Storage. Azure Defender can also scan container images in Azure Container Registry for vulnerabilities and protect Azure Kubernetes Service instances.

DOCS

Azure Defender

Protect your web apps and APIs with Azure Defender

BLOG

Azure Defender in the deployment process

App access and connectivity

In this section, we will look at the Azure services and tools that can enable you to secure the connectivity of, and access to, your application. We will review tools and services to help you secure your APIs and connect securely to your virtual machines (VMs), and we'll see how you can securely connect your on-premises network to Azure.

Azure Bastion

Azure Bastion is a fully managed PaaS offering that provides secure and seamless RDP and SSH access to your VMs directly through the Azure portal. Azure Bastion is provisioned directly in your virtual network and supports all the VMs in your virtual network using SSL (Secure Socket Layer) without any exposure through public IP addresses.

With Azure Bastion, you can limit the public exposure of your VMs' IP addresses. Exposing the Bastion host as the primary exposed public access point helps to reduce public internet exposure and limit threats such as port scanning and other types of malware targeting your VMs.

LEARN MORE

Read about Azure Bastion

API Management

APIs should be secure. This is true for APIs you create yourself as well as those from third-party vendors. To assist in making your APIs secure, you can use API Management. This is basically a proxy you put in front of APIs that adds features such as caching, throttling, and authentication or authorization.

With API Management, you secure an API by requiring users to create a subscription to it. This way, applications need to authenticate before they can use your API. You can use various authentication methods, including access tokens, basic authentication, and certificates. Additionally, you can track who's calling your API and block unwanted callers.

API Management supports multiple pricing tiers up to 99.95% SLA guarantee. The Consumption pricing tier offers the ability to have the API Management service automatically scale to handle the load.

While security is critical, API Management offers other capabilities that can help streamline your development and testing workflow, such as test data response mocking, publishing multiple API versions, introducing non-breaking changes safely with revisions, and giving developers access to your API's autogenerated documentation, catalog, and code samples.

TUTORIAL

Get started with API Management

Azure VPN Gateway

One of the many options for connecting Azure to your on-premises network is VPN Gateway. This lets you set up an encrypted site-to-site (S2S) VPN connection between an Azure virtual network and your on-premises network.

Because the traffic is encrypted, it's secure—even when it travels over the public internet. VPN Gateway can send encrypted traffic between Azure virtual networks over the Microsoft network.

You can also create encrypted point-to-site (P2S) connections from your computer to Azure. This way, you have your own private, secure connection to Azure even when you're on the road.

TRY IT OUT

```
Get started by creating an Azure
VPN gateway with PowerShell
```

Azure DNS private zones

The DNS is responsible for translating (or resolving) a service name to its IP address. Azure DNS is a hosting service for DNS domains, providing name resolution using the Azure infrastructure. In addition to internet-facing DNS domains, Azure DNS now supports private DNS domains as a preview feature with Azure DNS private zones. Security benefits arising from private DNS zones include the ability to create a split DNS infrastructure. This enables you to create private and public DNS zones with the same names without exposing internal names. In addition, the use of DNS private zones removes the need to introduce custom DNS solutions that could increase the overall attack surface with independent updating and management requirements.

DOCS

```
Read more about DNS private
zones
```

Cross-premises VPNs

Azure supports two types of cross-premises VPN connections: P2S VPN and S2S VPN. A P2S VPN connection lets you create a secure connection to your virtual network from an individual client computer. This type of connection is established from the client computer, which is useful for telecommuters who want to connect to Azure virtual networks from a remote location. A P2S VPN is also useful when you have only a few clients who need to connect to a virtual network. In contrast, an S2S VPN connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address.

DOCS

```
Read more about P2S VPNs
```

```
Read more about S2S VPNs
```

Azure ExpressRoute

Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a secure private connection facilitated by a connectivity provider without traversing the public internet. With ExpressRoute, you can establish a private connection to Microsoft cloud services, such as Azure, Microsoft 365, and Dynamics 365.

Azure Load Balancer

You can use load balancers to increase the availability of applications. Azure supports both external and internal load balancers, which can be used in a public or internal configuration.

In addition, you can configure load balancers to support high-availability (HA) ports, where an HA ports rule is a variant of a load balancing rule configured on the internal standard load balancer. You can provide a single rule to load balance all TCP and UDP flows arriving on all ports of an internal load balancer.

LEARN MORE

Read about load balancer and HA ports rules

Logging and monitoring

Being able to record and explore logging and monitoring data is a critical part of any security strategy. You need tools and services that enable you to monitor and investigate threats, issues, and risks as they arise. In the following sections, we will review the Azure tools and services that can help you collect and analyze logging and monitoring data.

Azure Sentinel

To get a good overview of the security status of your organization and all of its users, applications, services, and data, you can use a security information and event manager (SIEM) and security orchestration automated response (SOAR) platform. Azure now offers an AI-powered SIEM and SOAR solution in the form of Azure Sentinel.

Use Azure Sentinel to collect data from your organization, including data about users, applications, servers, and infrastructure assets such as firewalls and devices running in the cloud and on-premises. It's easy to collect data from your organization with the built-in connectors. As data is being collected, Azure Sentinel detects security threats and minimizes false positives with its smart machine learning algorithms.

When there is a threat, you'll be alerted and can investigate it with AI, utilizing decades of cybersecurity work at Microsoft. You can respond to incidents with Azure Sentinel's built-in workflow orchestration and task automation.

QUICK START

`Get started with Azure Sentinel`

Azure Monitor

Azure Monitor enables basic monitoring for Azure services by collecting metrics, activity logs, and diagnostic logs. The metrics collected provide performance statistics for different resources, including the operating system associated with a VM.

The activity log will show you when new resources are created or modified. You can view this data with one of the explorers in the Azure portal and send it to Log Analytics for trends and detailed analysis, or you can create alert rules that will proactively notify you of critical issues.

Azure Monitor Logs

Azure Monitor Logs contains different kinds of data organized into records with different sets of properties for each type. Azure Monitor Logs is especially useful for performing complex analysis across data from a variety of sources.

Log Analytics within Azure Monitor allows you to write, execute, and manage Azure Monitor log queries in the Azure portal.

Azure NSG flow logs

A feature of Network Watcher, Azure NSG flow logs allow you to view information about ingress and egress IP traffic through a network security group (NSG). Flow logs can be analyzed to gain information and insights into network traffic and security, as well as performance issues related to traffic.

While flow logs target NSGs, they are not displayed in the same way as other logs and are stored only within a storage account.

Application Insights

Application Insights is an extensible application performance management (APM) service for web developers on multiple platforms. It includes powerful analytics tools to help you diagnose issues and understand what users do with your application. It works for applications on a variety of platforms hosted on-premises or in the cloud, including .NET, Node.js, and J2EE.

Application Insights integrates with your DevOps process and has connection points to a variety of development tools. It can monitor and analyze telemetry from mobile applications by integrating with Visual Studio App Center.

Azure Security and Compliance Blueprint

Azure Security and Compliance Blueprint for HIPAA/HITRUST – Health Data & AI provides tools and guidance to help deploy a platform as a service (PaaS) environment for compliance with the Health Insurance Portability and Accountability Act (HIPAA) and the Health Information Trust Alliance (HITRUST).

This PaaS offering supports ingesting, storing, analyzing, and interacting with personal and non-personal medical records in a secure, multi-tier cloud environment deployed as an end-to-end solution. The blueprint showcases a common reference architecture that could be applied to use cases beyond healthcare and is designed to simplify the adoption of Azure.

DOCS

Read more about the Azure Security & Compliance Blueprint

Azure security technical and architectural documentation

Azure maintains a large library of security technical documentation that supplements security information with individual services. White papers, best practices documents, and checklists are included on the Azure security information page.

Also covered are core public cloud security topics in diverse areas, including network security, storage security, compute security, identity and access management, logging and auditing, cloud workload protection, PaaS security, and more.

Learn more about Azure security by using the following free resources:

RESOURCE HUB

Read more about Azure security

DOCS

Secure your cloud applications in Azure

Azure security baseline for App Service

Encryption

By default, your data is encrypted in Azure when it's stored in Azure SQL Database, Azure Synapse Analytics (formerly Azure SQL Data Warehouse), Azure Database for MySQL, Azure Database for PostgreSQL, Azure Storage, Azure Cosmos DB, or Azure Data Lake Storage. All this encryption works automatically, and you don't need to configure anything when you use it.

To help meet your security and compliance requirements, you can use the following features to encrypt data at rest:

- Azure Disk Encryption encrypts Windows and Linux infrastructure as a service (IaaS) VM boot and data volumes using customer-managed keys.
- Azure Storage encryption automatically encrypts data prior to persisting in Azure Storage, and then automatically decrypts the data when you retrieve it.
- Azure client-side encryption supports encrypting data within client applications before uploading it to Azure Storage or other endpoints and then decrypting data when downloading it to the client.
- Transparent Data Encryption (TDE) encrypts SQL Server, Azure SQL Database, and Azure Synapse Analytics data files. Data and log files are encrypted using industry-standard encryption algorithms. Pages in a database are encrypted before they're written to disk and are decrypted when they're read.
- Always Encrypted encrypts data within client applications prior to storing it in Azure SQL Database. It allows the delegation of on-premises database administration to third parties and maintains separation between those who own and can view the data and those who manage it but should not access it.
- Azure Cosmos DB requires no action from you—user data stored in Azure Cosmos DB in non-volatile storage (solid-state drives) is encrypted by default, and there are no controls to turn it on or off.

You can use the following features to encrypt data in transit:

- **VPN Gateway** can be used to encrypt traffic between your virtual network and your on-premises location across a public connection, or between Azure virtual networks.
- **TLS/SSL encryption** protects data when it's traveling between cloud services and customers. TLS (Transport Layer Security) provides strong authentication, message privacy, and integrity.
- **SMB 3.0 encryption** in VMs running Windows Server 2012 or later can be used to make transfers secure by encrypting data in transit over Azure virtual networks.

Over the last few pages, we looked at the different services and tools that Azure offers to help you secure your application. As we have seen, securing your application in Azure is a shared responsibility between you and Azure. Azure provides lots of good tools to secure your application and data by default, such as out-of-the-box encryption and SSL certificates for your apps. Now, it's your turn to use the services and tools we reviewed in this chapter to secure your application in Azure.

07 /

Deploying your services and optimizing costs

How can Azure help you deploy your services and optimize costs?

Azure has cloud services for every type of organization, including those that need Azure for their datacenters. You can deploy your applications either on the public Azure cloud, on-premises, or even on other cloud platforms. You can also manage your on-premises applications hosted on virtual machines (VMs) or in Kubernetes seamlessly from the Azure portal through the use of Azure Arc. You get to choose how portable your applications should be.

With Azure Arc-enabled app services, you can host Azure app services within a Kubernetes cluster that has been onboarded with Arc. Currently in public preview, this allows you to deploy to targets such as web apps, logic apps, and functions within a cluster and manage them like any app service from the Azure portal. This means you can utilize Kubernetes for your application needs and use the Azure portal for a single-pane-of-glass management experience. It also enables you to deploy and run Azure services in Kubernetes clusters that are on-premises or even on different clouds, expanding your ability to deploy and centralizing management in a convenient one-stop shop.

It's also possible to develop applications in containers and deploy them in containers, on-premises, or in the Azure cloud. Additionally, you can script your complete infrastructure by means of infrastructure as code (IaC) using tools such as Azure Resource Manager templates, Bicep, and Terraform.

Let's explore these options in more detail.

Infrastructure as code

IaC captures environment definitions as declarative code, such as JSON documents, for automated provisioning and configuration. This enables you to use the same version control used for source code with infrastructure deployment templates.

There are many benefits of using IaC:

- It lowers the potential for human error while deploying and managing infrastructure.
- It deploys the same template multiple times to create identical dev, test, and production environments.
- It means the cost of development and test environments can be reduced by creating them on demand.

The more you automate and catalog with version control, the higher confidence levels in the reliability and quality of your IaC definitions will rise. When consistency is reached, the risk threshold is lowered, allowing more frequent deployments. This paves the way for acceleration in other areas, such as canary testing, A/B or blue/green deployments, and more.

Choosing an IaC provider is a decision that should not be made lightly. While providers have similarities between them, each may have nuances that could help (or hurt) your overall automation strategy. Your particular use cases for IaC will ultimately guide you to the appropriate choice for your application.

Azure Resource Manager templates

All Azure services introduced in this guide are based on Azure Resource Manager, which you can use to document your environment using IaC thanks to Azure Resource Manager templates. These templates are JSON files that describe what you want to deploy and what the parameters are.

It's easy to create Azure Resource Manager templates in Visual Studio and Visual Studio Code using Azure resource group project templates. You can also generate Azure Resource Manager templates from the Azure portal by clicking the Automation Script button, which is available on the menu bar of every resource in the Azure portal. This creates the Azure Resource Manager template for the given resource and even generates code for building the resource using the Azure CLI, PowerShell, .NET, and others.

After you have an Azure Resource Manager template, you can deploy it to Azure by using PowerShell, the Azure CLI, Visual Studio, and the Azure portal. Alternatively, you can automate its deployment in a continuous deployment pipeline using Azure DevOps or GitHub Actions.

A great example of deploying resources to the cloud using Azure Resource Manager is the Deploy to Azure button found in many GitHub repositories.

Introducing Bicep

Bicep is a new flavor of IaC, developed and published by Microsoft. It uses the same underlying Azure Resource Manager model to create infrastructure, but uses a YAML-based syntax to provide greater readability.

The domain-specific language behind Bicep declaratively defines resources and allows you to have a first-class authoring experience in Visual Studio and Visual Studio Code.

Some benefits of using Bicep over other IaC methods include:

- Support for all Azure resource types, even those in preview
- A simple syntax that makes Bicep files easier to read and manage in comparison to JSON
- A first-class authoring experience, including rich type-safety and IntelliSense
- A flexible model that allows you to break code up into reusable modules, allowing for sets of resources to be deployed in one module
- Integration with other Azure services, such as Policy, template specs, and Blueprints
- A stateless run model, where no infrastructure state (or state files) are required
- Open-source and free to use

While still in its early stages, Bicep holds promise in terms of readability and composition, especially if you are already used to YAML formatting with other deployment types such as Kubernetes. The stateless nature of Bicep may also be appealing to those who may not like managing state files, especially if there is a need for remote state storage.

DOCS

Set up Bicep development and deployment

Compare Bicep and JSON side by side using Bicep Playground

Terraform with Azure

Hashicorp's Terraform is an open-source tool for provisioning and managing cloud infrastructure. Terraform's template-based configuration files enable you to define, provision, and configure Azure resources in a repeatable and predictable manner.

Terraform is great for deploying infrastructure across multiple cloud providers and on-premises. This enables consistent tooling for managing each infrastructure definition within Azure and across other cloud providers.

Terraform's template-based configuration files enable you to define, provision, and configure Azure resources in a repeatable and predictable manner.

Terraform shares a lot of capabilities with Azure Resource Manager templates. However, it also includes the ability to create reusable modules for deploying and configuring infrastructure. These modules can be shared across multiple Terraform projects or even used multiple times within the same project. This can be leveraged to save a lot of time automating infrastructure deployments.

Additional IaC tools

There are additional IaC tools that can be used. You can bring your existing skills and tools, including Ansible and Chef, to provision and manage Azure infrastructure directly.

Azure Blueprints

It's easy to use Azure Resource Manager templates, resource groups, user identities, and access rights and policies to design and create a complete infrastructure. But how do you keep all of these things together? And how do you keep track of which environments each piece of infrastructure has been deployed to and which version of the artifact is deployed now?

Organize all your infrastructure artifacts with Azure Blueprints. Azure Blueprints provides a mechanism that allows you to create and update artifacts, assign them to environments, and define versions. You can store and manage these artifacts as well as manage their versions and relate them to environments.

This will help you to organize your infrastructure and create a context for Azure Resource Manager templates, user identities, resource groups, and policies.

Azure Blueprints enables you to simplify large-scale Azure deployments by packaging key environment artifacts into a single blueprint definition. Then, you can easily apply the blueprint to new subscriptions and environments, including fine-tune control and management through versioning.

QUICK START

Get started by defining and assigning an Azure blueprint in the Azure portal

Tracking your Azure usage

With Azure products and services, you only pay for what you create and the Azure resources you use. It's important to keep track of what you are using and the costs involved.

Azure Cost Management and Billing

Using Azure Cost Management and Billing, you can monitor and control Azure spending and optimize your Azure resource usage. Azure Cost Management gives you the tools to plan, analyze, and reduce your spending to maximize your cloud investment.

Reports in Azure Cost Management show the usage-based costs of Azure services and third-party Azure Marketplace offerings. Costs are based on negotiated prices and factor in reservations and Azure Hybrid Benefit discounts. Collectively, the reports show your internal and external costs for usage and Azure Marketplace charges.

You can use the Azure portal or various APIs for export automation to integrate cost data with external systems and processes. Automated billing data export and scheduled reports are also available.

TUTORIAL

Optimize costs from recommendations

Azure Advisor

Azure Cost Management also works with Azure Advisor to provide cost optimization recommendations. Azure Advisor helps you optimize and improve efficiency by identifying idle and underutilized resources.

Azure Advisor, for example, monitors your VM usage for seven days and then identifies underutilized VMs. VMs whose CPU utilization is five percent or less, and whose network usage is seven MB or less for four or more days, are considered low-utilization VMs.

Creating a billing alert

Azure Cost Management alerts can be used to monitor your Azure usage and spending. Cost alerts are automatically generated based on when Azure resources are consumed. Alerts show all active cost management and billing alerts together in one place.

When your consumption reaches a given threshold, alerts are generated by Azure Cost Management. There are three types of cost alerts:

- **Budget alerts** notify you when spending, based on usage or cost, reaches or exceeds the amount defined in the alert condition of the budget. Azure Cost Management budgets are created using the Azure portal or the Azure Consumption API.
- **Credit alerts** notify you when your Azure credit monetary commitments are consumed. Monetary commitments are for organizations with enterprise agreements. Credit alerts are generated automatically at 90% and 100% of your Azure credit balance. Whenever an alert is generated, it's reflected in cost alerts and in the email sent to the account owners.
- **Department spending quota alerts** notify you when department spending reaches a fixed threshold of the quota. Spending quotas are configured in the EA portal.

Cost alerts can easily be viewed within the Azure portal. All alerts will show the alert type. A budget alert shows the reason why it was generated and the name of the budget it applies to. Each alert shows the date it was generated, its status, and the scope (subscription or management group) that the alert applies to.

The possible statuses for alerts include "active" and "dismissed." An active status indicates that the alert is still relevant. A dismissed status indicates that someone has marked the alert to set it as no longer relevant.

Select an alert from the list to view its details. Alert details show more information about the alert. If a recommendation is available for a budget alert, then a link to the recommendation is also shown. You can also navigate to **Cost analysis**, where you can explore costs relating to the alert's scope.

How to use Azure Billing APIs

Azure Billing APIs can be used to pull usage and resource data into your preferred data analysis tools. These APIs are implemented as a resource provider and are part of the family of APIs exposed by Azure Resource Manager.

There are three Azure Billing APIs available:

- **Usage API**: Used to get consumption data for an Azure subscription
- **RateCard API**: Used to get meter (AKA resource) metadata information along with prices
- **Invoice API**: Used to download invoices

Azure Billing APIs are exposed as REST APIs that can be integrated into custom application scenarios. They can be used to get better insights into your cloud spend during the month, your estimated consumption, and a few other things. They can be queried and stored in a database for later use, or they can be integrated into reporting solutions such as Power BI to give greater flexibility when accessing and displaying billing data.

As we've seen throughout this chapter, there are many options available for not just cloud services, but also for codifying configurations using IaC via several different providers. Using platform-native tools such as Azure Cost Management and Azure Advisor, you can get a clear picture of your spending as well as areas for right-sizing. Alerts can be created to monitor specific spending trends, and reporting is easy using the built-in Azure Billing APIs.

08 /

Microsoft Azure in action

Navigating the Azure portal

In this section, you will learn how to develop your first web application and database in Azure. For those who are new to Azure, we have provided you with a quick tour of Azure, starting with the Azure portal.

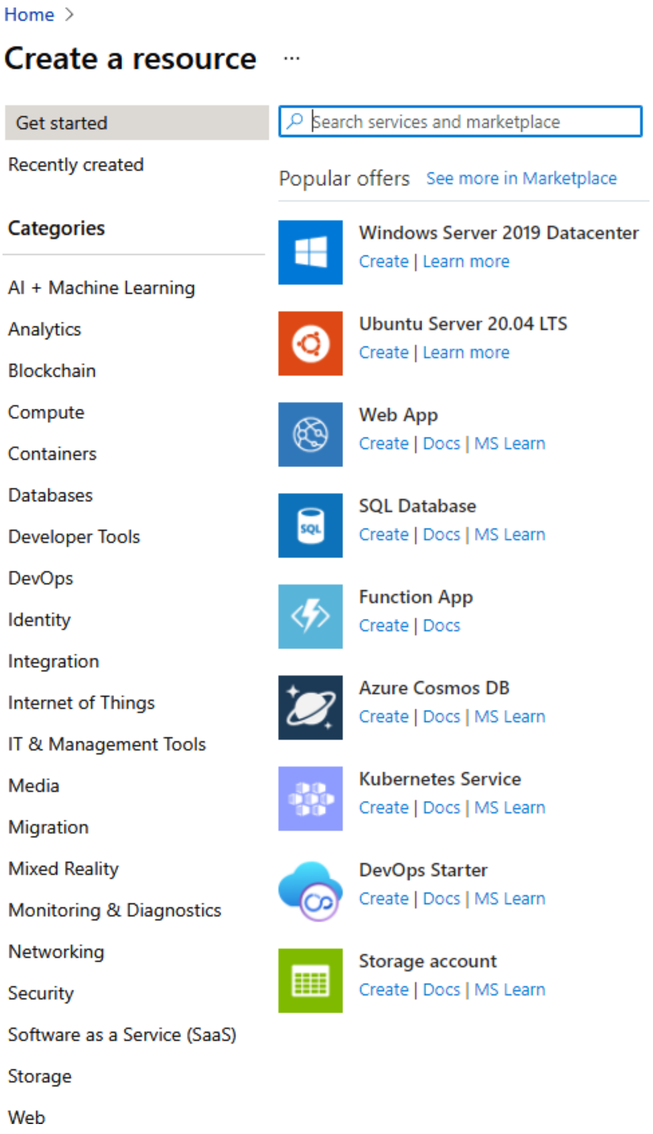
The Azure portal is a web-based, unified console that provides an alternative to command-line tools. You can manage your Azure subscription with the Azure portal and build, manage, and monitor everything from simple web apps to complex cloud deployments. It allows you to create custom dashboards for an organized view of resources and configure accessibility options for the best experience.

Dashboards provide a focused view of the resources in your subscription that matter to you the most. The default dashboard is provided to get you started. You can customize this dashboard to bring the resources you use frequently into a single view.

Any changes you make to the default view, only affect your experience. However, you can create additional dashboards for your own use or publish customized dashboards and share them with other users in your organization.

Finding and adding **services** in the Azure portal can be done in several ways. To create new services, select **+ Create a resource** on the Azure home screen, or do the same within the left-hand navigation menu, which can be expanded by selecting the hamburger icon in the upper-left corner of the Azure portal.

This opens the search box for Azure Marketplace, where you will find everything from web applications to Linux servers, as shown in *Figure 8.1*:



Home >

Create a resource ···

Get started

Recently created

Categories

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

Search services and marketplace

Popular offers See more in Marketplace

Windows Server 2019 Datacenter
Create | Learn more

Ubuntu Server 20.04 LTS
Create | Learn more

Web App
Create | Docs | MS Learn

SQL Database
Create | Docs | MS Learn

Function App
Create | Docs

Azure Cosmos DB
Create | Docs | MS Learn

Kubernetes Service
Create | Docs | MS Learn

DevOps Starter
Create | Docs | MS Learn

Storage account
Create | Docs | MS Learn

Figure 8.1: Azure Marketplace pane

The Azure Marketplace pane is pre-populated with popular services grouped into categories. If this list does not include what you are looking for, then you can use the **Search services and marketplace** box to type in a search keyword. When you find the service you want from the search results, you need to select it and a wizard will take you through its configuration and deployment.

To learn more about how to navigate the Azure portal, use Azure Quickstart Center, a guided experience in the Azure portal available to anyone who wants to improve their knowledge of Azure. For organizations new to Azure, it is the fastest way to onboard and set up your cloud environment.

Check out this quickstart tutorial and other additional resources.

TUTORIAL

Create a Linux VM in the Azure portal

Get started with Azure

Weekly webinar: Azure demo and live Q&A

Now, let's use the Azure portal to create a new VM.

Develop your first web app and extend it with Logic Apps and Cognitive Services

We will use this tutorial to develop and deploy a sample to-do list application to Azure. You will learn how to create a .NET Core app and a SQL database in Azure, connect the app with the database, and deploy it to Azure App Service. You will also learn how to update the data model and redeploy the app, stream diagnostic logs from Azure, and manage the app in the Azure portal.

To prepare for this tutorial, you will need Git v2 or higher, .NET Core, and Visual Studio Code installed on your device.

Extending applications with Logic Apps and Cognitive Services

Once you have your app and the database deployed in Azure, you can start adding additional features. A powerful feature of your application could be the ability to analyze the content of to-do items and then automatically create calendar appointments for tasks that include a specific date.

For example, if a user creates a to-do item with the text "family dinner next Friday at 7:00 PM," the application will create a calendar item for that specific Friday at 7:00 PM with the subject "family dinner."

This can be set up using the Logic Apps feature of Microsoft Azure App Service and Language Understanding (LUIS), as follows:

- The .NET Core application writes the to-do item in the SQL database.
- The logic application is triggered by every new row created in the database.
- The logic application takes the to-do item text and passes it to LUIS.
- LUIS analyzes the text and creates a calendar item in your Microsoft 365 calendar if the text contains a date and time.

You don't have to change your application to add this functionality. Logic Apps and Cognitive Services are additional services that simply analyze the data that's already there.

Creating LUIS

First, you'll create LUIS so that you can use it later in your logic application.

You can use the LUIS portal to build a language model. First, you need to add some entities, which are items in the text the service will recognize. Then, you enter utterances, which are sample texts that represent the intent you want to detect.

Here is an example of an utterance:

```
"family dinner next Friday at 7 PM"
```

QUICK START

How to create LUIS

Creating the logic application

Next, to integrate LUIS into the application, create an Azure logic app. Within the logic app, the LUIS connector can be used to add actions to integrate LUIS with your applications.

When using LUIS actions, you enter the API key for your LUIS service, so the logic app can connect to and integrate with your Language Understanding service. Then, you configure the logic app to pass the particular utterance text to LUIS and use it for recognizing utterances.

For example, a "LUIS—Get prediction" action could be added to a logic application that connects to a SQL database that's triggered when a new row is created. The logic app could pass in text retrieved from the database row to LUIS so that it can perform utterance recognition. Then, the logic app could save the result back to the database, or use it to execute additional actions to perform tasks based on those results.

Ready for production

So far, we've been pushing code from our local Git repository to Azure. This is fine if you work alone, but if you work in a team, you'll need another type of source control, such as Azure Repos, one of the services in Azure DevOps, or GitHub.

Setting up continuous delivery with GitHub

With your application running in Azure, you could use a GitHub repository to push your code and then link that to your web application so that changes are deployed automatically in a continuous delivery pipeline.

Continuous delivery can be configured using the Deployment Center feature of Web Apps through the Azure portal. This feature enables you to choose the location of your code as well as options for building and deploying it to the cloud.

Setting up staging environments

Using web apps from the Azure App Service, you can set up a staging slot to test new versions of your application through deployment slots. Deployment slots are application services with which you can test your code before you promote it to the next slot.

There are deployment slots for staging, load testing, and production, which is always the original app service—in our example, the .NET Core web application. In fact, you can have as many deployment slots as you wish without incurring additional costs.

All the deployment slots run in the same App Service plan, which is what you pay for. Having additional deployment slots in an App Service plan will consume resources such as CPU and memory, so you need to be mindful of how additional slots might impact production.

You can create new deployment slots from the Deployment Slots menu item in the web app. You need to run the web application in the Standard or Premium pricing tier because the free plan doesn't come with additional deployment slots.

In each deployment slot you create, you can configure the deployment options as we did earlier to deploy code automatically. You can even work on different source code branches for different environments and automatically deploy specific branches to specific deployment slots.

Additionally, you can test your final version in a deployment slot and then swap it with the version in the production slot. This warms up the application before it swaps, resulting in a deployment with no downtime.

Scaling the web app

When your web app is inundated with lots of traffic and user activities, you can scale up your web app to accommodate the increased traffic. Conversely, when your web app is idling, you can scale it back to reduce costs. Thanks to the automatic scaling feature of Azure App Service, you can achieve this with ease. The best part of this feature is that it only takes a matter of seconds to adjust the scale settings and you do not even need to make any changes to your code or redeploy your application.

To use this feature, you need to run the web apps in the Standard or Premium pricing tier. Alternatively, you can use the Free tier to run a single instance of a web app.

You can learn more about how to scale your application in Azure App Service here.

Using diagnostic logs

An efficient way to monitor an application is by using diagnostic logs to see live diagnostic logging from the web app. You can even pipe the logs into the console window. To do this, run the following command in Cloud Shell:

```
az webapp log tail --name <app_name>
                   --resourcegroup <myResourceGroup>
```

You'll be able to see diagnostic logs when you use the web application to generate some traffic.

Setting up monitoring and alerts

Azure Monitor Application Insights provides another powerful way to track applications. This monitoring tool provides information about your application, such as how many visitors used it, how many exceptions occurred, and where they occurred in the code. Unlike diagnostic logs, Application Insights requires a nominal fee.

Adding Secure Sockets Layer

When an application is ready for production, you need to confirm that it's secure. Besides authentication and authorization, serving the web application over HTTPS is one of the most important security measures you can take. This is because, without HTTPS, intruders could see the traffic among your resources and use this information for malicious purposes, like signing in to your application. Additionally, HTTPS is a requirement for leading-edge features such as service workers.

Serving traffic to your web application over Secure Sockets Layer (SSL) is possible by importing an SSL certificate into Web Apps and binding it to one of your custom domain names. You can either import your own SSL certificate or purchase one through Azure App Service Certificates. This service makes it easy to buy and validate certificates. After importing the certificate, couple it to one of the domain name bindings of your web app. You can do all this from the TLS/SSL settings in the web app.

Notifying users about new versions

Your business will benefit from making users aware of new production releases. By extending the continuous integration/continuous delivery (CI/CD) process in Azure builds, you can use a Logic Apps workflow to manage social media communication, such as sending out tweets or publishing posts with release notes.

An Azure pipeline could be instrumented to trigger the logic app to execute after a release pipeline has finished publishing new application changes. Alternatively, the Events feature of the App Service web application could be configured to trigger a logic app based on events emitted from Azure App Service, such as when the deployment slots are swapped.

Learn more about architecting Azure solutions in these free resources:

RESOURCE HUB

Azure for Architects

Architect great solutions in Azure learning path on Microsoft Learn

09 /

Summary and resources

In this guide, we introduced the power that Azure can bring to your applications. Using Azure, you can do incredible things with your applications, including hosting and scaling your web applications, taking advantage of containers, and using AI in your applications, while only paying for what you use.

You've learned that Azure has services for almost every scenario, so it can help you no matter which programming language you use or which platform you write applications for. Before we wrap up, we will provide you with some valuable resources to help you embark on your Azure journey.

Keep learning with Azure

With your Azure free account, you get all of this—and you won't be charged until you choose to upgrade:

- 12 months of popular free services
- USD 200 credit to explore any Azure service for 30 days
- 25+ services, always free

Azure tips and tricks

Browse a collection of useful ideas to help you become more productive with Azure.

Azure Friday

Take a look at Azure services and features with the Microsoft engineering team.

Microsoft.Source

Receive a regular digest of relevant technical content, events, and training. Learn about new technologies and find opportunities to connect with other developers online and locally.

Azure Certifications

Earn certifications that show you are keeping pace with today's technical roles and requirements.

Azure communities and meetups

Join our community-led meetups, where you'll learn from your peers about solutions to common problems, fun projects, and what's new in Microsoft Azure.

Microsoft Learn

Learn new skills and discover the power of Microsoft Azure products with step-by-step guidance.

Learn TV

Start your journey today by exploring our Azure learning paths and modules, including Learn TV, which features the latest digital content so you can always keep updated on the latest announcements, features, and products.

Free resources extravaganza

In addition to this guide, there are many other free resources related to Azure, including the following:

- Azure for Developers: A list of developer resources for app development.
- Azure for Architects: A comprehensive guide for Azure architects.
- Azure Strategy and Implementation Guide: Get a step-by-step introduction to using Azure for your cloud infrastructure and learn how to create a successful cloud adoption strategy with new innovations, capabilities, and security features from Microsoft Azure.
- Learn Azure in a Month of Lunches: A practical way to learn Azure from scratch over a month of lunches.
- Azure Proof of Concept Guide for Developers: Prove whether a concept works or not before your organization makes a significant investment. Learn how to create and execute a proof of concept for developing applications in Azure, from a well-designed plan to measurable test results.
- Azure Serverless Computing Cookbook: Find use cases, hands-on recipes, and tutorials for quickly configuring your own serverless environment.
- Get up and running with Kubernetes: With the Kubernetes collection, you'll get multiple resources that will help you gain the knowledge and hands-on experience necessary to get started with Kubernetes—all in one place.
- Cloud Analytics with Microsoft Azure: Maximize your BI impact by bringing data together from all your sources with Azure Synapse Analytics.
- Introducing Microsoft SQL Server 2019: Find out what's new in Microsoft SQL Server 2019, a platform for secure and compliant modern data management.
- Azure Networking Cookbook: Configure, manage, monitor, and troubleshoot networks more effectively with networking solutions from Azure.
- Building Intelligent Cloud Applications: Build and deploy scalable deep learning and machine learning models using serverless architectures with Azure.

Thanks to the wealth of prebuilt solutions in Azure, the days of having to write complicated plumbing are over. Free yourself up to work on the things that matter to you by taking advantage of all that Azure offers. We hope you continue to consult this guide to become better acquainted with the vast range of Azure services and determine which ones best fit your needs.

About the authors

Has Altaiar

Has is a passionate and award-winning technologist with over 15 years of professional experience in large corporate, government, and digital agencies. In his work, Has focuses on Data, IoT, AI, and DevOps. He has successfully delivered many secure, scalable, and award-winning projects for a variety of fields including medical, financial, and utilities. Has is also a Microsoft MVP (Most Valuable Professional) and a regular organizer and speaker at local and international conferences. You can follow him on LinkedIn @altaiar or on Twitter @hasaltaiar.

Ingrid Babel

Ingrid Babel is a senior technical product manager at Microsoft Azure. Her goal is to help developers use the full potential of the cloud by creating content that's accessible to anyone and at any level. You can follow her on LinkedIn at https://www.linkedin.com/in/ingridbabel/.

Jack Lee

Jack Lee is a senior Azure certified consultant and an Azure practice lead with a passion for software development, cloud, and DevOps innovations. He is an active Microsoft tech community contributor and has presented at various user groups and conferences, including the Global Azure Bootcamp at Microsoft Canada. Jack is an experienced mentor and judge at hackathons and is also the president of a user group that focuses on Azure, DevOps, and software development. He is the co-author of *Azure for Architects*, *Azure Strategy and Implementation Guide*, and *Cloud Analytics with Microsoft Azure*, published by Packt Publishing. He has been recognized as a Microsoft MVP for his contributions to the tech community. You can follow Jack on Twitter at @jlee_consulting.

Josh Garverick

Josh Garverick is a Microsoft MVP in Azure and application lifecycle management who has over 15 years of experience in software development. He has experience using DevOps best practices as well as with architecting and modernizing applications to enable the adoption of Azure. He is the author of the book *Migrating to Azure: Transforming Legacy Applications into Scalable Cloud-First Solutions*, published by Apress. You can follow Josh on Twitter at @jgarverick or on LinkedIn at https://linkedin.com/in/josh-garverick.

Mustafa Toroman

Mustafa Toroman is a solution architect focused on cloud-native applications and migrating existing systems to the cloud. He is very interested in DevOps processes and cybersecurity, and is also an Infrastructure-as-Code enthusiast and DevOps Institute Ambassador. Mustafa often speaks at international conferences about cloud technologies. He has been an MVP for Microsoft Azure since 2016 and a C# Corner MVP since 2020. Mustafa has also authored several books about Microsoft Azure and cloud computing.

Vahe Minasyan

Vahe Minasyan is a technical project manager on the Microsoft Azure product marketing team. His passion is to help developers use the power of Azure for their software development needs. You can reach Vahe on LinkedIn at https://www.linkedin.com/in/vaheminasyan2/ and you can also find him on GitHub.

Authors from previous editions

Michael Crump

Michael Crump works at Microsoft on the Azure platform and is a coder, blogger, and international speaker on various cloud development topics. He's passionate about helping developers understand the benefits of the cloud in a no-nonsense way.

You can reach Michael on Twitter, @mbcrump, and follow his live coding stream at https://www.twitch.tv/mbcrump.

Chris Pietschmann

Chris Pietschmann is a principal cloud and DevOps solution architect with Solliance, and a Microsoft MVP with Azure and IoT. He's passionate about helping individuals and teams be more productive in the cloud.

Follow his blog at https://build5nines.com.

Barry Luijbregts

Barry Luijbregts is an independent software architect and developer with a passion for the cloud and authors courses for Pluralsight.

You can reach Barry on Twitter, @AzureBarry, and through his website at https://www.azurebarry.com/.

PUBLISHED BY Microsoft Press, A division of Microsoft Corporation

One Microsoft Way, Redmond, Washington 98052-6399

This book is provided "as-is" and expresses the author's views and opinions. The views, opinions and information expressed in this book, including URL and other Internet website references, may change without notice. Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Microsoft