E-book Series



The Developer's Guide to Azure







03 /

Introduction

We're here to help

05 /

<u>Chapter 1: Getting started</u> with Azure

What can Azure do for you? Where to host your application? Azure App Service features **Azure Functions** Azure Logic Apps Azure Batch Containers What to use, and when? Making your application more performant Azure Front Door Azure Content Delivery Network Azure Cache for Redis **Azure Security** DevOps Developer tools

29 /

<u>Chapter 2: Connecting your application with data</u>

What can Azure do for your data?
Where to store your data?
Azure SQL Database
Azure Cosmos DB
Azure Database for MySQL,
PostgreSQL, and MariaDB
Azure Storage
Azure data analytics solutions

43 /

Chapter 3: Securing you application

How can Azure help to secure your application?
Encryption
Azure Security Center
Logging and monitoring

55 /

Chapter 4: Adding intelligence to your application

How can Azure integrate Al into your application?
What to use, and when?
Cognitive Services
Azure Cognitive Search
Azure Bot Service
Azure Machine Learning Studio
Developer tooling for Al
Al and mixed reality
Using events and messages in your application

75 /

Chapter 5: Connecting your business with IoT and gaining insights

How can Azure connect, secure, manage, monitor, and control your IoT devices in the cloud?
Azure IoT Hub
Azure IoT Central
Azure IoT Edge
Azure Digital Twins
Azure Sphere
Azure Maps
Windows for IoT
Azure RTOS
Learn more about Azure IoT
What to use, and when?

90 /

<u>Chapter 6: Deploying your</u> <u>services and reducing costs</u>

How can Azure deploy your services and reduce costs?
Infrastructure as Code
Azure Blueprints
Containers in Azure
Tracking your Azure usage
Creating a billing alert
How to use your billing APIs?
Where to deploy, and when?

102 /

<u>Chapter 7: Share your code,</u> <u>track work, and ship software</u>

How can Azure help you plan smarter, collaborate better, and ship your applications faster? Azure Boards Azure Repos Azure Pipelines Azure Test Plans Azure Artifacts GitHub

111 /

<u>Chapter 8: Microsoft Azure</u> in action

Navigating the Azure portal
Developing your first web application
and database on Azure
Extending applications with Logic
Apps and Cognitive Services
Ready for production

124 /

<u>Chapter 9: Summary and resources</u>

Keep learning with Azure About the authors

The Developer's Guide to Azure

This guide is designed for developers and architects who are starting their journey into Microsoft Azure. In this guide, you'll learn how to get started and which services you can use 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.

We're here to help

We can assist you in a variety of ways to suit your needs.

With our support plans, you'll 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.

You can also get help through other channels, such as:

<u>Documentation and guides</u> that give you an overview of everything in Azure and provide deep insights through the documentation of each feature.

<u>Service License Agreements (SLAs)</u>, which can inform you about our uptime guarantees and downtime credit policies.

<u>@Azure</u> on Twitter, which is the account to follow for news and updates from the Azure team and community.

<u>@AzureSupport on Twitter</u>, which is operated by skillful Azure engineers who respond quickly to issues that you tweet to them.

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

Azure Advisor, which 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, which gives you a personalized view of the health of your Azure services.

<u>Stack Overflow</u>, which provides answers to Azure questions and includes many active posts by members of the Azure engineering teams.

01/

Getting started with Azure

You've made the decision to build applications on the cloud, and now you want to get started. You don't need to do much—just sign up for an Azure free account. This includes 12 months of free services, including \$200 in credits for 30 days, enabling you to explore paid Azure services, and over 25 services that you can use for free ad infinitum.

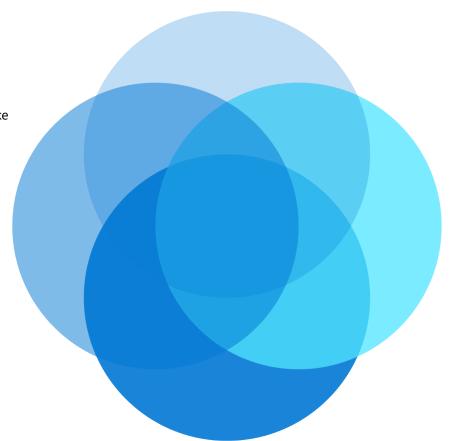
Simply choose which programming languages, tools, platforms, and frameworks you want to use, and then start running your applications on Azure.

What can Azure do for you?

Whether you're 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 and makes it easy to build for the cloud in your preferred language.

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

Let's dive in and see what you can do.



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.

Azure App Service

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

Azure App Service comprises the following:

Web Apps: 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.



Mobile Apps: Provide a back end for your mobile applications with Mobile Apps. When you host an API in Mobile Apps, your mobile applications connect with it through the cross-platform client software development kit (SDK). This is available for iOS, Android, and Windows. Mobile Apps provides features such as offline sync and push notifications to help you create a modern, performant, and secure mobile experience.

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 operating system (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 virtual machines (VMs). One or more VMs run your Azure App Service, but since Azure takes care of them, it's 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 shunt 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're 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 should be separated for security, scaling, billing, and performance. You can swap deployment slots manually through the Azure command-line interface (CLI) and the Azure Management API. 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 also can use the <u>continuous deployment (CD)</u> feature in App Service. This makes it possible for you to create a build-test-release pipeline right in App Service.

The process does the following:

- 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)
- Deploys the application in a staging environment and load-tests it
- Deploys the application to production after approval (you can indicate whether you want to use a deployment slot)

Connect to on-premises resources

You can connect external resources such as data stores to your App Services. These resources don't need to be located in Azure; they can be anywhere, such as on-premises or in your own datacenter. Depending on your requirements, you can connect to services on-premises through many mechanisms, such as Azure Hybrid Connections, Azure Virtual Networks, and Azure ExpressRoute.

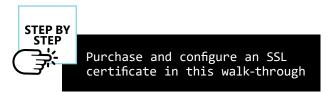
Custom domains and Azure App Service certificates

When you spin up an application in Azure
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 Services. Here's how to do that.

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 buy an Azure 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

sub-domain. They are also managed by App Service and are renewed automatically.



App Service Environment

In a multitier 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 article describes this service in more detail and shows you how to use it.

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 <u>Azure App</u>
<u>Service Environment</u>. 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're 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're using a fully managed platform, you don't manage your own infrastructure at all, but you do benefit from automatic OS and framework patching.

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 set 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 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:

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

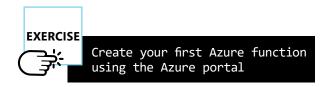
Input and output bindings connect your function code to other services, including Azure Storage, Azure Cosmos DB, Azure Service Bus, and even third-party services such as Twilio and SendGrid. Using Azure Functions, you can build small pieces of functionality quickly and host them in an elastic environment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

With Azure Functions, it's 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.



Cold start/warm start

When Azure Functions has been sitting idle, hasn't been used for a while, and is scaled to zero instances, when new events come in, a new instance needs to be specialized with your application running on it. Specializing new instances may take some time (latency) before the first event can be handled. This occurs automatically behind the scenes and applies to both Consumption and App Service Plan pricing.

With the <u>Azure Functions Premium plan</u>, you can configure the number of pre-warmed instances to eliminate cold start latency. 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 available connectors to APIs that can connect to Azure SQL Database, Salesforce, SAP, and so on.

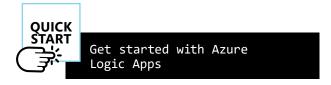
You can also expose your own APIs or Azure Functions as connectors to use in a logic app, making it possible for you to easily perform actions against external systems in your workflow or have your logic app be activated by one of them.

Just like Azure Functions, Logic Apps is serverless and scaled automatically, and you pay for them only when they're running.

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

 The logic app is activated when an email containing a shipping order arrives in Office 365.

- Using the data in the email, the logic app checks on the availability of the ordered item in SQL Server.
- 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.



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.

Logic Apps versus Power Automate

Both Microsoft Power Automate and Logic Apps offer designer-first integration services that can create workflows. Both services can integrate with various SaaS and enterprise applications; however, they are both targeted at different users. Here's a side-by-side comparison to help determine when to use Power Automate or Logic Apps for a particular integration scenario:

Table 1-1: What to use whenChoosing between Power Automate and Logic Apps

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

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 SaaS solutions that use Batch as a compute platform.



Containers

While much more lightweight, containers are similar to 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.

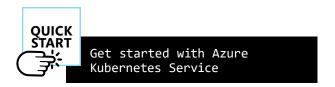
Scale and orchestrate containers with Azure Kubernetes Service

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.

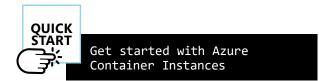
With Azure handling the management of the nodes in your AKS cluster, there are many tasks that you don't have to perform manually, such as cluster upgrades. Because Azure handles these critical maintenance tasks for you, AKS does not provide direct access (such as with SSH) to the cluster.



Host containers with Azure Container Instances

You can host your container using Azure Container Instances. Container Instances provides fast, isolated compute to meet traffic that comes in spikes, without the need to manage servers. For example, AKS can use Virtual Kubelet to provision pods inside Container Instances that start in seconds. This enables AKS to run with just enough capacity for an average workload. As you run out of capacity in your AKS cluster, you can scale out additional pods in Container Instances without any additional servers to manage. The Container

Instances service is billed per second, per virtual CPU, per gigabyte, or by memory usage.



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 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 pre-defined 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've created a container image to run your application in, you can store that container in <u>Azure Container Registry</u>, which is a highly available and secure storage service, specifically built to store container images. This is great for storing your private Docker images.

You can also use Container Registry for your existing container development and deployment pipelines. Use ACR Build 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.

Azure Service Fabric

Another way to run applications in Azure is with Azure Service Fabric. This is actually the service that runs many of the Azure services inside Microsoft, such as Azure SQL Database and Azure App Service. Run your applications in Service Fabric to achieve high availability, run at massive scale, and perform rolling upgrades.

You can use Service Fabric to run .NET microservice-based applications—solutions that consist of many small services that talk to each other and are employed by user interfaces and other components. Service Fabric is ideal for solutions such as these because it orchestrates application components together and runs them in a highly available and performant manner.

Service Fabric is unique in that you can run it anywhere. Install it on your local development computer, on-premises, or in any cloud—including Azure. You can also use Azure Service Fabric Mesh to run containers on a Service Fabric cluster that Microsoft manages for you as a service. This opens up a lot of possibilities.

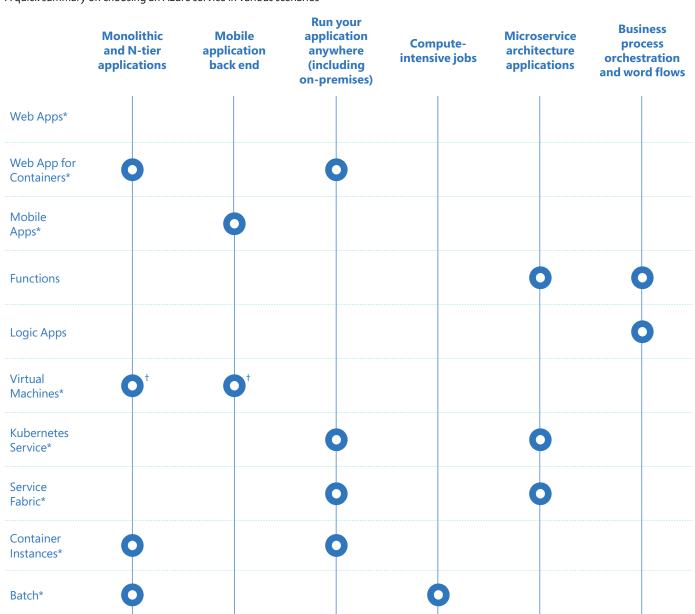
It's easy to deploy applications to Service Fabric and manage them with your favorite tools, such as Visual Studio and Azure DevOps. In addition, Service Fabric recently became open-source.

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, Table 1-2 will help you to identify which services in Azure are right for your situation:

Table 1-2: Deciding the right serviceA quick summary on choosing an Azure service in various scenarios



^{*} 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.

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

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

Including Azure 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. Azure Front Door can route to available endpoints, while avoiding endpoints that are down.

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

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

Azure 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.

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

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.

If you need help choosing between Azure Front Door and Traffic Manager, consider this guidance:

Table 1-3: Selecting the best optionChoosing between Front Door and Traffic Manager

You only need routing (performanceor geography-based) and high
availability

Azure Front
Door

You need SSL termination (also called
SSL offloading)

You need application layer features such as URL rewriting and WAF

SSL termination

With this, you can secure your traffic 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.

Azure Content Delivery Network

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



Get started with Azure Content Delivery Network

Azure Cache for Redis

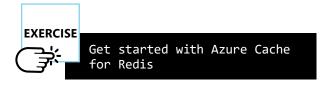
Every modern application works with data. When you retrieve data from a data store such as a database, this typically involves scanning multiple tables or documents in some distant server, weaving the results together, and then sending the result to the requesting device. This, of course, takes time and can frustrate and annoy your users.

To eliminate some of these "roundtrips," you can cache data that doesn't change often. This way, instead of querying the database every time, you can retrieve some of the data from a cache, like Azure Cache for Redis. The benefit of the cache is that it stores data in a simple format, such as keyvalue. You don't need to run a complex query to get this data—you just need to know the key to retrieve the value.

This can improve the performance of your application dramatically. Here's how this workflow operates:

- The application needs some data and attempts to retrieve it from the cache.
- If the data is not there, the application gets it from the database and stores the data in the cache.
- The next time the application searches for that piece of data, it will find it in the cache, saving a trip to the database.

Azure provides Cache-as-a-Service with Azure Cache for Redis. This is based on the open-source Redis project and is now backed by industry-leading SLAs. It is highly performant and has advanced options, such as clustering and geo-replication.



Azure Security

Security is always a concern for any application.

One of the best reasons to use Azure for your applications and services is to take advantage of its wide array of security tools and capabilities.

These tools and capabilities help to make it possible to create secure solutions on the secure Azure platform. Azure provides confidentiality, integrity, and availability of customer data, while also enabling transparent accountability.

There are encryption features supported by many Azure services that enable the encryption of data at rest and encryption in transit. Azure Disk Encryption, Azure Storage Service Encryption, and others ensure that data is kept secure when stored on disk or within a storage account. Other features, such as SSL/ TLS support, encrypt data that's transferred over HTTPS or Azure service REST API endpoints.

Here are some of the Azure security services that will help you to create secure solutions:

- <u>Azure Active Directory</u> Synchronize onpremises directories and enable single sign-on.
- <u>Key Vault</u> Safeguard and maintain control of keys and other secrets.
- Azure Sentinel Put cloud-native SIEM and intelligent security analytics to work to help protect your enterprise.
- <u>Security Center</u> Unify security management and enable advanced threat protection across hybrid cloud workloads.
- Application Gateway Build secure, scalable, and highly available web front ends in Azure.

In *Chapter 3, Securing your application*, we'll dive a little deeper into securing your applications.

DevOps

There are a few different aspects that encompass what DevOps is and what it means for every development team and organization. The word itself is a combination of "Dev" and "Ops" that refers to the removal of silos between development and operations teams, encouraging collaboration across different disciplines and the adoption of shared processes and tools. Essential DevOps practices include agile planning, continuous integration, continuous delivery, and the monitoring of applications.

"DevOps is the union of people, process, and technology to enable continuous delivery of value to our end users."

— SAM GUCKENHEIMER

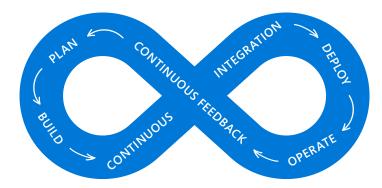


Figure 1-1: Essential DevOps practices

Azure includes a number of services that are used by development teams to implement DevOps within their teams and organizations. DevOps teams implement a similar set of workflows and processes:

- Share code, track work, and ship software.
- Plan, track, and discuss work across the team.
- Implement continuous integration and continuous delivery (CI/CD) to deliver value in short increments.

GitHub

GitHub offers a number of DevOps features, the most notable of which is the ability to create, manage, and use Git repositories. This includes additional capabilities for teams to communicate on work being done through GitHub Issues, and manage the process of merging code by using the GitHub Pull Request functionality.

<u>GitHub includes several tools</u> that support every part of your workflow:

- GitHub Actions CI/CD workflow automation to build, test, and deploy your code from GitHub.
- GitHub Advanced Security Use advanced features to secure your code by discovering

vulnerabilities with Semmle's semantic code analysis engine, <u>CodeQL</u>.

- GitHub Open Source Security Report and disclose security vulnerabilities in open-source code through security policies on projects to let communities know the best way to responsibly report vulnerabilities.
- GitHub Packages Safely publish and consume packages within your organization or across the entire world.

Azure DevOps

Azure offers several services to more easily implement DevOps practices within your team and organization. The <u>Azure DevOps Services</u> can be used individually or in combination with each other:

- Azure Pipelines Provide build and release services to support continuous integration and the delivery of your applications.
- Azure Boards Deliver a suite of Agile tools to support the planning and tracking of work, code defects, and issues using the Kanban and Scrum methods.
- Azure Test Plans Provide tools to perform manual/exploratory testing of your applications.
- Azure Artifacts Allow teams to share Maven, npm, PyPi, and NuGet packages from public

- and private sources, and integrate package sharing into your CI/CD pipelines.
- Azure Repos Provide Git repositories for source control of your code.

DevOps integration

You can also use your favorite DevOps toolchain. Azure provides integration with popular open-source and third-party tools and services. The various Azure DevOps services can also be used in multiple combinations of configurations integrated with your favorite DevOps toolchain.

You could use Azure Pipelines or deploy directly to the Azure platform from your favorite CI/CD solution, such as Jenkins.

Azure services, third-party DevOps tools, and related products all work together to help meet the most common business needs and scenarios.

Developer tools

Azure supports a comprehensive set of development tools for any developer—using any platform or language—to deliver cloud applications. Code with the language of your choice using a range of SDKs and take advantage of full-featured integrated development environments (IDEs) and editors with advanced capabilities and built-in Azure support.

Your tool of choice

You can also use the development tools you already know. This enables you to develop and deploy from both Microsoft and open-source developer tools seamlessly integrated with Azure:

- Visual Studio
- Visual Studio Code
- Azure Toolkit for Eclipse
- Azure Toolkit for IntelliJ
- Azure tools for Maven

Developers have their tools, languages, and frameworks of choice, and Azure supports a wide array of developer options.

SDKs and command-line tools

Azure includes a collection of <u>SDKs</u> 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 <u>Azure CLI</u> is a set of commands used to create and manage Azure services that is compatible with Windows, macOS, and Linux. <u>Azure PowerShell</u> is a set of cmdlets that allows for the managing of Azure resources from the PowerShell command line.

GitHub

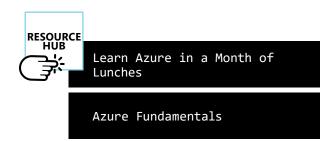
GitHub, the world's leading developer platform, can be seamlessly integrated with Azure.

GitHub Enterprise can be used to securely bring open-source code and best practices to your enterprise projects. Flexibility, security, compliance, and deployment controls make it easy for your team to use GitHub Enterprise wherever you need it.

The features of GitHub can also be easily integrated with several DevOps services within Azure DevOps. For example, you could use Azure Pipelines to build and deploy code that resides in a GitHub repository.

Further reading

If you want to learn more about using Azure Kubernetes Service, Azure Container Instances, and other Azure services to create distributed applications, download and read the following free resources:



02 /

Connecting your application with data

What can Azure do for your data?

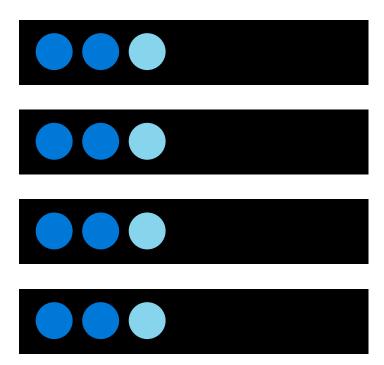
Wherever your data is, Azure will help you to unlock its potential. Support rapid growth and save more time for innovation with a portfolio of secure, enterprise-grade database services that support open-source database engines.

Azure database services are fully managed, freeing up valuable time so that you can focus on new ways to delight your users and unlock opportunities, rather than spending that time managing your database. Enterprise-grade performance with built-in high availability means you can scale quickly and reach global distribution without worrying about costly downtime.

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. On top of all of this, your investment is protected by financially backed service-level agreements (SLAs).

Whatever you build, we'll help you to get it to market quickly, distribute it widely, and manage it easily and confidently.

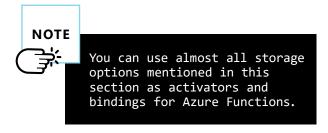
Let's dive in.



Where to store your data?

Azure provides many types of data stores that can help you to maintain and retrieve data in any scenario. Table 2-1 presents the storage options available in Azure.

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



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

Table 2-1: Which option to use when A quick summary on storage options



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

Azure SQL Database

If you want to use tables with columns and rows to store data, <u>Azure SQL Database</u> is a great choice. A relational database system similar to on-premises Microsoft SQL Server, SQL Database runs in the cloud, so it's fully managed, performant, scalable, automatically backed up, and includes many advanced features.

With SQL Database, you can do <u>almost everything</u> that you can do with on-premises SQL Server. In fact, new SQL Server features are incorporated first in Azure SQL Database and later in on-premises SQL Server.

You can use SQL Database with your favorite tools, including SQL Server Management Studio and Entity Framework. Databases in SQL Database are extremely reliable and robust and offer an <u>SLA that guarantees</u> <u>99.99 percent uptime</u>.

Here are some of the more advanced features in SQL Database:

- Geo-replication, which replicates data to other geographical regions in real time
- <u>Dynamic data masking</u>, which masks sensitive data for certain users at runtime
- Auditing, which provides a complete audit trail of all the actions that happen to the data
- Automatic database tuning, which monitors the performance of your 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 protects sensitive
 data by storing it encrypted within the database
 engine, and encrypting/decrypting it within the
 client application to separate those who manage
 the data but should not have access to it

SQL Database offers several service tiers that are geared toward specific scenarios:

- General purpose/Standard: This tier offers budget-oriented, balanced, and scalable compute and storage options. Fully managed, with performance comparable to Azure SQL VMs, this tier is the best option for most business workloads.
- Business critical/Premium: This tier offers the highest resilience to failures using several isolated replicas. With consistently high I/O, it includes a built-in availability group for high availability. This is the best option for critical online transaction processing (OLTP) (normal CRUD operations) and business applications with consistently high I/O requirements.
- Hyperscale: This tier offers very large database (VLDB) support without the headaches. With a built-for-the-cloud architecture of highly scalable storage and a multilayer cache optimized for very large and demanding workloads, it provides low latency and high throughput regardless of the size of data operations. This is the best tier for very large and demanding workloads with highly scalable storage and read-scale requirements.

Azure Cosmos DB

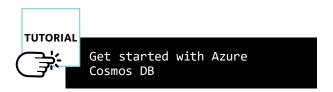
<u>Azure Cosmos DB</u> is a new kind of database made for the cloud. Its key features include:

- A 99.99 percent SLA (up to 99.999 percent for read and write operations) that includes low latencies (less than 10 ms on reads and less than 15 ms on writes).
- Geo-replication, which <u>replicates data to other</u> geographical regions in real time.
- Tunable data consistency levels, so you
 can enable a truly globally distributed data
 system. You can choose from a spectrum of
 data consistency models, including strong
 consistency, session consistency, and eventual
 consistency.
- <u>Traffic Manager</u>, which sends users to the service endpoint to which they are closest.
- Limitless global scale, so you pay only for the throughput and storage that you need.
- <u>Multi-master</u>, which allows an account to be created with multiple write regions enabled.
- Automatic indexing of data, which removes the need to maintain or tune the database.

 A free tier, which enables you to develop and test applications, offering the first 400 RU/s and 5 GB of storage for free for the lifetime of the account.

In addition to all these features, Azure Cosmos DB offers different APIs with which you can store and retrieve data, including SQL, JavaScript, Gremlin, MongoDB, Azure Table Storage, Apache Cassandra, and Etcd. Different APIs handle data in different ways. You can use documents as data as well as unstructured tables, graphs, and Blobs. You use the API that fits your needs, and Azure Cosmos DB takes care of the rest.

You benefit from cloud-grade performance, scalability, and reliability, while using the programming model you're already accustomed to.



Azure Database for MySQL, PostgreSQL, and MariaDB

Azure provides MySQL, PostgreSQL, and MariaDB databases as managed databases, which means that you just spin them up and don't have to worry about any of the underlying infrastructure. Just like Azure SQL Database and Azure Cosmos DB, these databases are universally available, scalable, highly secure, and fully managed.

Each of these databases is suited for slightly different use cases, but in general, their functionality overlaps a lot. You would use Azure databases for MySQL, PostgreSQL, and MariaDB when you've already been using one of their on-premises versions and want the advantage of having it run fully managed in the cloud.

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

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.

Azure Database for PostgreSQL

Azure Database for PostgreSQL is a relational database service in the Microsoft cloud built for developers. It is based on the community version of the open-source PostgreSQL database engine.

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 Storage

Azure Storage is one of the oldest, most reliable, and most 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 so that you can recover it 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, File, and Disk (Figure 2-1):

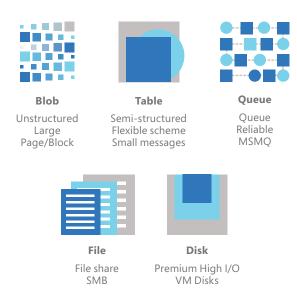


Figure 2-1: Types of storage in Azure

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 datastore, and you can use it to host a static website on Azure Storage. All you have to do in order for your website to run is upload the files of your 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). Your website will run quickly for very little cost—in fact, you only pay for the storage you use, and the static website feature doesn't cost anything extra. Additionally, when you use geo-redundancy (which is enabled by default), your website will be up and running even if your primary datacenter fails.

Blob storage

<u>Azure Blob storage</u> 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: <u>Page, Block, and</u>
<u>Append Blobs</u>. Page Blobs are optimized for random read and write operations, and are perfect for storing a VHD.

Block Blobs are optimized for efficiently uploading large amounts of data. These are perfect for storing large video files that don't change frequently.

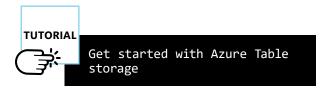
Append Blobs are optimized for append operations, such as storing operation logs that can't be updated or deleted.



Get started with Azure Blob storage

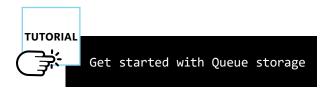
Table storage

Azure Table storage is an inexpensive, extremely fast NoSQL key-value store that you can use to store data in flexible tables. A table can contain one row describing an order and another row describing customer information. You don't need to define a data schema, making Table storage very flexible.



Queue storage

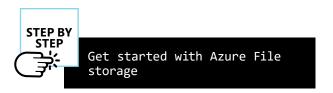
Azure Queue storage is an unusual type of storage. While it's used to store small messages of data, its main purpose is to serve as a queue. You put messages in the queue and other processes pick them up. This pattern decouples the message sender from the message processor, resulting in performance and reliability benefits. Queue storage is found in previous versions of Windows.



File storage

You can use <u>Azure File storage</u> as a drive from which to share files. It uses the Server Message Block (SMB) protocol, meaning you can use it

with Windows and Linux and access it from either the cloud or on-premises systems. Like the other services in Azure Storage, File storage is scalable and inexpensive.



Disk storage

Azure Disk storage is similar to File storage but is 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. You can 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 you can analyze it or transform and move it.

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, in which you define data schemas upfront.

You can store all of the data that you get from Internet of Things (IoT) devices (collecting temperature data, for example) in Data Lake Storage. You can leave the data in the store and then filter through it to create a view of the data per hour or per week. Storing the data in Data Lake Storage is inexpensive, so you can keep years of data there at a very low cost.



Azure Synapse Analytics

Azure Synapse Analytics (formerly Azure SQL Data Warehouse) brings together enterprise data warehousing and big data analytics. It gives you the freedom to query data on your terms, using either serverless on-demand or provisioned resources at scale.

When you need a traditional data warehousing solution that is completely managed, scalable in size, and performant and secure, <u>Azure Synapse Analytics</u> can provide the solution. Store data in predefined schemas and query it by using the familiar SQL Server dialect.

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

Build end-to-end analytics solutions with a unified experience. The Azure Synapse studio provides a unified workspace for data prep, data management, data warehousing, big data, and AI tasks. Data engineers can use a code-free visual environment for managing data pipelines, and business analysts can securely access datasets and use Power BI to build dashboards in minutes, all while using the same analytics service.



Create an Azure Synapse Analytics SQL pool

Azure data analytics solutions

Almost as important as storing data is analyzing it to get insights. Azure provides many services for data analytics scenarios, enabling you to get valuable and actionable insights from your data—no matter how large, small, or complex it is.

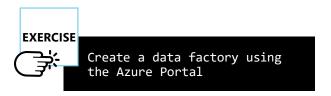
Data Factory

Moving and transforming data is not a trivial task, but <u>Data Factory</u> can help you to do just that. Within Data Factory, you can create a comprehensive pipeline that performs your complete extraction, transformation, and loading (ETL) process.

Data Factory can reliably move data from onpremises to the cloud, within the cloud, or to on-premises—it doesn't matter where your data sources are. Data Factory also provides many connectors that you can use to easily connect to your data source, including SQL Server, Azure Cosmos DB, Oracle, and many more.

When you move data, you can also filter it before you send it to an end destination, clean it up, or transform it with an activity in the pipeline, such as the <u>Apache Spark activity</u>. In addition, Data Factory allows you to schedule and monitor pipelines, as

well as <u>lift and shift your SQL Server Integration</u>
<u>Services (SSIS) packages</u> to the cloud.



Azure Analysis Services

With Azure Analysis Services, you can create a semantic model of your data that users can access directly with visualization tools such as Power BI. Built on the <u>SQL Server Analysis Services</u> tools that run on-premises with SQL Server, this service now runs managed in the cloud. This means that the service is scalable and data is stored redundantly—and when you aren't 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, including from data sources in the cloud and on-premises. You can use Azure Blob storage, Azure SQL Database, Azure SQL Data Warehouse, and many other services as data sources for the model. You can also use data sources, including onpremises Active Directory, Access databases, and Oracle databases.



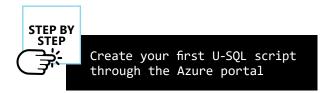
Create an Azure Analysis Services server using the Azure portal

Azure Data Lake Analytics

Another Azure service for performing data analytics tasks is <u>Azure Data Lake Analytics</u>. 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.

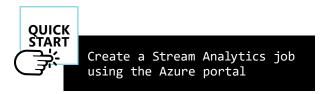
You pay for the jobs that you submit and run, and the service scales automatically depending on the power that the jobs need. Azure Data Lake Analytics is typically used for long-running analytics jobs against massive amounts of data.



Azure Stream Analytics

You can use the Azure Stream Analytics service to analyze, query, and filter real-time streaming data. For example, when you receive a stream of temperature data from an IoT device, it tells you how warm it is outside. It might provide the same temperature every second for an hour until the temperature changes, but you are only interested in the changes. Azure Stream Analytics can query the data in real time and store only the differential data in an Azure SQL Database.

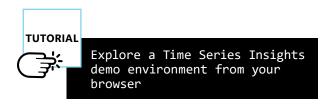
Stream Analytics can get its data from many services, including Azure Blob storage, Azure Event Hubs, and Azure IoT Hub. You can analyze the data by using a simple SQL-like language or custom code. After querying and filtering the stream of data, Stream Analytics can output the result to many Azure services, including Azure SQL Database, Azure Storage, and Azure Event Hubs.



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 your own 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. This is like Azure Analysis Services, where data lives in-memory in a model for users to query. The key differences are that Time Series Insights is optimized for IoT and time-based data, and it contains its own data visualization tool.



Azure Databricks

Azure Databricks allows you to run a managed and scalable Databricks cluster in the cloud. 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 needs.

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, including Azure Storage and Azure Data Lake Storage. Databricks also works with data from Azure SQL Data Warehouse, Azure SQL Database, and Azure Cosmos DB. Additionally, you can plug Databricks into Power BI to create and show powerful dashboards.



Run a Spark job on Azure Databricks using the Azure Portal

HDInsight

HDInsight is a platform within Azure that you can use to run open-source data analytics services. You can also use it to run specialized clusters of your favorite 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, they can scale and easily connect to one another, other Azure services, and on-premises data sources and services.

Most of the specialized open-source data analytics cluster types in HDInsight use Azure Blob storage or Azure Data Lake Storage to access or store data, as these services work with the Hadoop File System.

You can run potentially massive specialized clusters of different types, such as an **Apache Hadoop cluster**. This enables you to process and analyze data with Hadoop tools such as Hive, Pig, and Oozie.

You can also spin up an **Apache HBase cluster**, which provides a very fast NoSQL database. The data actually lives within Azure Storage or an Azure Data Lake, but HBase provides an abstraction layer on top, which has its own functionality and unique performance.

You can create an **Apache Storm cluster**, which is geared toward analyzing data streams, just like Azure Stream Analytics. In addition, you can have an **Apache Spark cluster**, which 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). This allows you to run R-based jobs to analyze data.

Finally, you can create a cluster that runs **Apache Kafka**, which is a publish-subscribe messaging system used to build applications with queueing mechanisms.

There are more cluster types, as well as tools that you can use within clusters. You can perform almost any data analytics and processing task with a combination of these clusters, and they all run managed in the cloud. Table 2-2 can help you to pick the right Azure services to analyze your data.

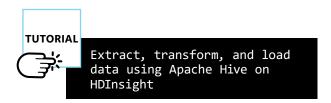
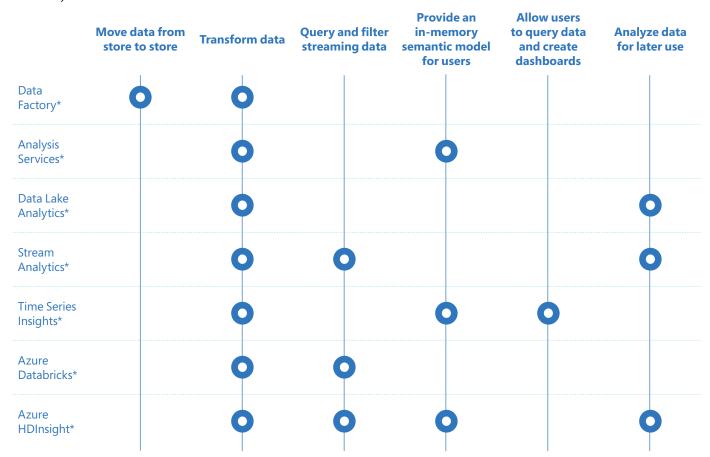


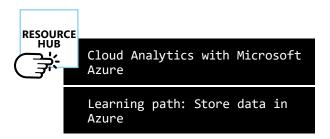
Table 2-2: Cluster optionsA summary of Azure services for different data use cases



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

Further reading

If you want to learn more about data and data analytics in Azure, you can download and read the following free resources:



03/

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 <u>Azure</u>. 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.

Let's dive in.



Azure Active Directory

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 password management, create a secure authentication handshake, and so on.

Azure Active Directory (Azure AD) provides all of these things and more out of the box. You store your user identities in Azure AD and have users authenticate against it, redirecting 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.

Since Azure AD is used by millions of applications every day—including the Azure portal, Outlook. com, and Office 365—it's able to more readily detect and act on malicious behavior. 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.

Key Vault

As part of your security architecture, you need a secure place to store and manage certificates, keys,

and other secrets. <u>Key Vault</u> provides this capability. With Key Vault, you can store the secrets that your applications use in a single secured, central location that leverages the <u>FIPS 140-2 Level 2 validated</u> <u>Hardware Security Module</u> (HSM).



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 from the configuration system. This way, administrators can control the secrets, and developers never need to deal with them. Key Vault also stores SSL and other certificates used to secure the traffic to and from your applications over HTTPS.

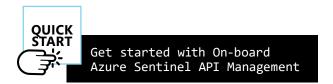
Azure Sentinel

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 Al-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 like 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.



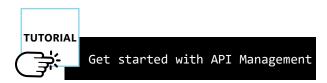
APIs should be secure. This is true for APIs you create yourself as well as those from third-party vendors. To assist in making your APIs secure, you can use <u>API Management</u>. This is basically a proxy you put in front of APIs that adds features such as caching, throttling, and authentication or 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 with an up to 99.95% SLA guarantee. The Consumption pricing tier offers the ability to have the API Management service automatically scale to handle load.

Much more than security

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



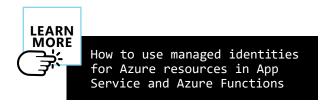
Azure AD Application Proxy

Azure AD Application Proxy provides single sign-on (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 Office 365. End users can access your on-premises applications the same way they access Office 365 and other SaaS applications integrated with Azure AD.

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 <u>managed identities for Azure resources</u> feature in Azure AD provides a solution.

You can use managed identities for a lot of services in Azure, including Azure App Service. You simply enable a managed identity with a button to inject credentials into your application at runtime and then use those credentials to access other services, including Key Vault, Azure SQL Database, and Azure Storage. All authentication between services is done on the infrastructure level, meaning your application doesn't have to deal with it and can just use other services.



Encryption

Default encryption of data

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 Store. 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 (laaS) virtual machine (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.
- <u>Transparent Data Encryption (TDE)</u> encrypts <u>SQL Server</u>, <u>Azure SQL Database</u>, and <u>Azure</u> <u>Synapse Analytics</u> 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 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 onpremises 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 onpremises 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.

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.

Azure DDoS Protection

You've heard about it on the news, and you certainly don't want it to happen to your enterprise: an application is targeted by a Distributed Denial of Service (DDoS) attack. 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 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.



Azure VPN Gateway

One of the many options for connecting Azure to your on-premises network is <u>VPN Gateway</u>. 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, secured connection to

Azure even when you're on the road.



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 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.

Web Application Firewall

You need to secure your application against many threats, including SQL injection, cross-site scripting (XSS), and others defined in the Open Web Application Security Project (OWASP). A WAF from Azure can assist with that. A feature of the Application Gateway service, a WAF provides real-time protection for your application. It 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.

Network Watcher

Network Watcher is a regional service that enables you to monitor and diagnose conditions at the network level in, to, and from Azure.

Its many diagnostic and visualization tools can help you understand and gain deeper insights into your network in Azure.

Examples include:

- <u>Topology</u>: Provides a network-level view of the resources in a virtual network, the resources associated with resources in a virtual network, and the relationship between these resources.
- Variable packet capture: Captures packet data in and out of a VM. Advanced filtering options and fine-tuned controls, such as the ability to set time and size limitations, provide versatility. The packet data can be stored in a Blob store or on the local disk in .cap format.
- IP flow verify: Checks whether a packet is allowed or denied based on 5-tuple flow information and packet parameters (destination IP, source IP, destination port, source port, and protocol). If the packet is denied by a security group, the rule and group that denied the packet are returned.

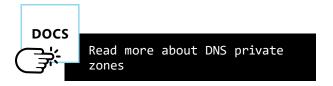
Network security groups

A <u>network security group</u> (NSG) holds a list of security rules that allow or deny network traffic to resources connected to Azure virtual networks.

NSGs can be associated with subnets, individual VMs (classic-style VMs), or individual network interface controllers (NICs) attached to VMs (Resource Manager–style VMs). When an NSG is associated with a subnet, the rules apply to all resources connected to the subnet. You can restrict traffic even further by also associating an NSG with a VM or NIC.

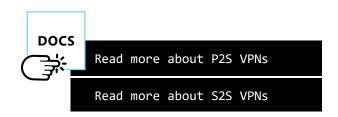
Azure DNS private zones

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.



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 that 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.



Azure ExpressRoute

Azure ExpressRoute lets you extend your onpremises 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, Office 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.



Logging and monitoring

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 trending and detailed analysis, or you can create alert rules that will proactively notify you of critical issues.

Azure Monitor Logs

Azure Monitor Logs contain different kinds of data organized into records with different sets of properties for each type. Logs in Azure Monitor are 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 <u>log</u> <u>queries in the Azure portal</u>.

Azure NSG flow logs

A feature of Network Watcher, <u>Azure NSG flow</u> <u>logs</u> allow you to view information about ingress and egress IP traffic through an 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.

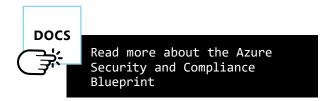
Application Insights integrates with your DevOps process and has connection points to a variety of development tools. It can monitor and analyze telemetry from mobile applications by integrating with Visual Studio App Center.

Azure Security and Compliance Blueprints

The Azure Security and Compliance Blueprints
- HIPAA/HITRUST Health Data and Al provides
tools and guidance to help deploy a platform-as-aservice (PaaS) environment for compliance with the
Health Insurance Portability and Accountability Act

(HIPAA) and the Health Information Trust Alliance (HITRUST).

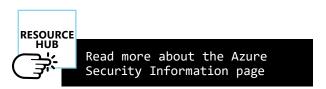
This PaaS offering supports ingesting, storing, analyzing, and interacting with personal and non-personal medical records in a secure, multi-tier cloud environment deployed as an end-to-end solution. The blueprint showcases a common reference architecture that could be applied to use cases beyond healthcare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



Azure security technical and architectural documentation

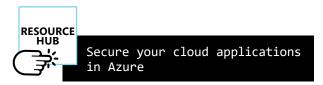
Azure maintains a large library of security technical documentation that supplements security information with individual services. White papers, best practices documents, and checklists are included on the Azure Security Information page.

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.



Further reading

Learn more about Azure security in the following free resource:



04 /

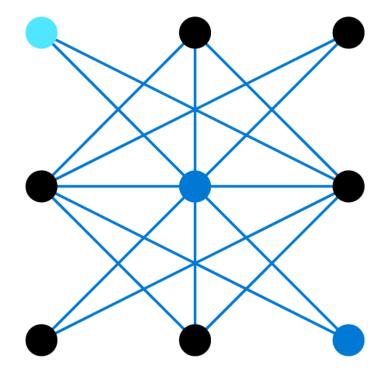
Adding intelligence to your application

How can Azure integrate Al into your application?

Al can give your application an edge over the competition. Just imagine what you can build—applications that translate speech in real time as you're speaking, or an application that helps you identify parts of a motor in mixed-reality training. The possibilities are endless. However, building an Al-powered application seems difficult. How do you create an algorithm that can understand speech, for instance?

Fortunately, you don't have to do everything yourself: Azure is here to help. It provides many Al services that you can just plug into your application, including the Speech Translation service (a service in Azure Cognitive Services), which translates speech in real time.

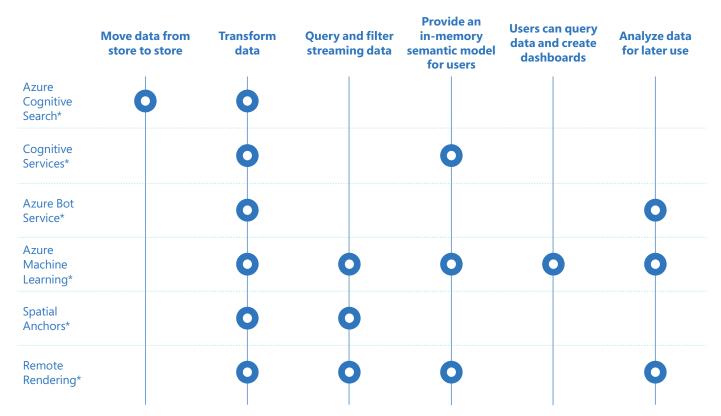
Using a service like this, you're able to simply consume AI, but you can also build your own machine learning algorithms using services such as Azure Machine Learning.



What to use, and when?

Before diving into the options for AI in Azure, let's look at Table 4-1, which summarizes which services are available and their capabilities:

Table 4-1: Picking up the right option A summary of Azure services for Al



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

Cognitive Services

<u>Cognitive Services</u> provides machine learning algorithms, created by Microsoft, and data as a service. For most services, Microsoft has also provided the data to train those algorithms.

For some services, you can use your own custom data to train the algorithms.

Cognitive Services provides an exceptionally easy way to incorporate machine learning and AI into your application—by simply calling APIs.

Table 4-2 shows which APIs are currently available. Note that this list keeps growing.

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

Each category in the table contains multiple services that you can use by calling an API. Some categories contain custom services, including <u>Custom</u>

<u>Vision</u>, <u>Language Understanding</u>, and <u>Bing</u>

<u>Custom Search</u>. These custom services provide preconfigured machine learning algorithms, just like the other services, and they also enable you to use your own data to train the model.

In addition to these services, <u>Cognitive Research</u> <u>Technologies</u> provides additional services. This contains innovative APIs and SDKs for researchers and developers with an early look at emerging cognitive capabilities. One such experimental service is <u>Project Gesture</u>, which enables you to detect gestures such as the waving of a hand and weave them into your user experience.

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

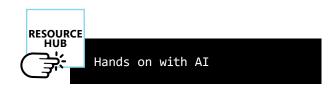


Table 4-2: A quick glance at Cognitive Services APIsServices and APIs in Cognitive Services and their categories

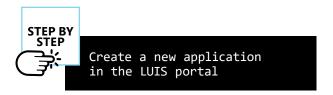
Computer Vision	Custom Vision	<u>Face</u>	Form Recognizer	<u>Ink</u> <u>Recognizer</u>	<u>Video</u> <u>Indexer</u>
Speech to Text	Text to Speech	Speech Translation	Speaker Recognition		
Immersive Reader	<u>Language</u> <u>Understanding</u>	QnA Maker	Text Analytics	<u>Translator</u> <u>Text</u>	
Anomaly Detector	Content Moderator	<u>Personalizer</u>			

Language Understanding

Use the <u>Language Understanding (LUIS)</u> service to understand what users are saying to you on social media, in chatbots, or in speech-enabled applications. For example, you can book flights or schedule meetings using LUIS.

To use the Language Understanding service, give it examples of what you want it to understand, such as "Book a flight to Seattle" or "Schedule a meeting at 1 PM with Bob," and tell it which words you're looking for. In these examples, you might be looking for the destination of the flight (Seattle) and the time and person for the meeting (1 PM and Bob).

After the Language Understanding service creates a machine learning model based on the examples that you put in, it can extract information from natural language that users put in.



Custom Vision

With the <u>Custom Vision service</u>, you can detect information in images based on your own training data. Custom Vision works in a similar manner to other Cognitive Services in that it comes with a predefined machine learning algorithm. All you have to do is feed the service with your data.

Let's say you want to create a model that can detect types of rain clouds in the sky, such as cumulus and stratus. To create this model, you upload images of different types of clouds to the Custom Vision portal and give them tags, which tells the service how to train the model. In this example, you would tag an image with "cumulus" or "stratus."

Once you've uploaded enough images, you can train your model. The more images you upload with tags and the more training you do, the more accurate your model will be.

Once you have a model that performs well, you can start using it by making calls to the Custom Vision API and feeding it new images. When you upload a new image, the service tells you whether it recognizes it based on the images already uploaded.

Figure 4-1 shows an example of what the API endpoint looks like:

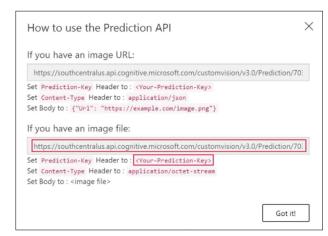
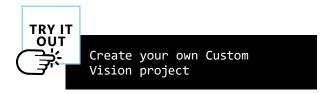


Figure 4-1: An example of API endpoint

Using the Custom Vision service to detect information based on your own model is impressive enough, but Custom Vision can do even more. The model you create when you train the Custom Vision service with your data can be deployed to the

"intelligent edge." This means that the model and API can run somewhere other than the cloud, such as on an on-premises server in a Docker container or on another device, such as your phone.

This offers great flexibility because you don't need an active internet connection to use the capabilities of the Custom Vision service; you can also run it locally, which provides great performance. In addition, the model you run on the edge isn't very large—only approximately tens of megabytes—because you deploy only the model and API, not the training data.



Video Indexer

The <u>Video Indexer service</u> analyzes the video and audio files you upload to it. This Cognitive Service is part of <u>Azure Media Services</u>. It provides a predefined machine learning algorithm and you provide the data.

In addition to <u>many others</u>, Video Indexer has the following capabilities:

- It creates a transcript of the text in a video. You can refine the transcript manually and train
 Video Indexer to recognize industry terms such as "DevOps."
- It tracks faces and identifies who is in a video and at what points. Video Indexer has the same

- capability for audio, for which it recognizes who is speaking and when.
- It recognizes visual text in a video, such as text on a slide, and makes that part of the transcript.
- It performs sentiment analysis, which identifies when something positive, negative, or neutral is said or displayed.

As the breadth of these functionalities demonstrates, Video Indexer combines many Cognitive Services, including Speech to Text and Speaker Recognition. Cumulatively, these services provide powerful capabilities that make content more discoverable, accessible, and valuable.

You can upload media files to Video Indexer using the Video Indexer portal or the API. Figure 4-2 shows the results of a video that was uploaded to the Video Indexer service:

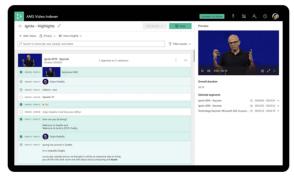


Figure 4-2: Demonstrating the results on the Video Indexer portal

As shown in the figure, Video Indexer created a transcript of the audio in the video. The transcript can be edited and even translated into other languages. You can also see that Video Indexer recognized text on the slide behind the speakers

and marked it as "OCR." You can skip to that text by clicking it. Video Indexer provides this functionality for individual applications by embedding the Cognitive Insights widget.

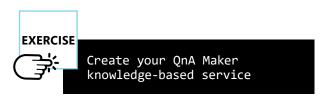


QnA Maker

QnA Maker offers an easy way to create a conversational layer over existing data, such as the frequently asked questions (FAQs) pages, support websites, and product manuals. QnA Maker helps you analyze and extract information and convert it into question-and-answer pairings that can be easily managed. Simply put, QnA Maker allows you to build applications that can provide information to your users in a conversational manner.

With QnA Maker, it's possible to create and manage knowledge bases using the easy-to-use QnA Maker Portal or REST APIs. We have simplified the bot creation process by allowing you to easily create a bot from your knowledge base—without the need for any code or setting changes. Refer to this link for further details: Create a QnA bot. Of course, you can also use QnA Maker to create a bot using the Azure Bot Service and augment your QnA bot by adding the Language Understanding service. To add personality, you can add chit-chat to your bot, and answer commonly asked small-talk scenarios out of the box.

You <u>pay only for the hosting of QnA Maker</u>, and not for how many times the resulting service gets queried by users.



Personalizer

<u>Personalizer</u> helps your application choose the best single content item to show each user. It selects the best item, from content items, based on collective real-time information you provide pertaining to content and context.

Content can be any unit of information, such as text, images, URLs, or emails that you want to select from to show to your user. This helps boost usability and user satisfaction with reinforcement learning—based capabilities.

Anomaly Detector

Anomaly Detector allows you to easily embed anomaly detection capabilities into your applications so users can quickly identify problems. This enables you to monitor and detect abnormalities in your time series data with machine learning.

Using Anomaly Detector doesn't require any prior experience in machine learning, and the RESTful API enables you to easily integrate the service into your applications and processes.

Azure Cognitive Search

Azure Cognitive Search is the only cloud search service with built-in AI capabilities that enrich all types of information to easily identify and explore relevant content at scale. Formerly known as Azure Search, it uses the same integrated Microsoft natural language stack that Bing and Office have used for more than a decade, and AI services across vision, language, and speech. You can spend more time innovating and less time maintaining a complex cloud search solution.

Azure Cognitive Search provides a lot of the plumbing to do searches. You just spin up an Azure Cognitive Search instance, create an index that helps you search, and fill it with data. This means, for example, that you could easily implement Azure Cognitive Search to help users search your product catalog in an e-commerce application.

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

 Geo-search, which lets users explore data based on the proximity of a search result to a physical location.

- Language analyzers from <u>Apache Lucene</u>, as well as <u>natural language processing (NLP)</u> <u>from Microsoft</u>, available in 56 languages, which intelligently handle linguistics, including verb tense, gender, irregular plural nouns, word decompounding, and word breaking for languages with no spaces.
- Monitoring and reporting, which provide information on what was searched for and how fast and successful the search was.
- User experience features, such as sorting and paging search results, intelligent filtering, and providing search suggestions.

Al enrichment is a capability of indexing that is used to extract text from images, blobs, and other unstructured data sources. Enrichment and extraction make your content more searchable.

For example, a team with a ton of PDFs with pictures, along with handwritten notes, can use Azure Cognitive Services to analyze the documents and connect places, domain experts, and events and then navigate all of this information to make key decisions.

Azure Cognitive Search uses Cognitive Services to analyze all this data, extract information, and correlate it—all without the need to write complicated image recognition or optical character recognition (OCR) software.



Create your first Azure Cognitive Search index in the portal

Azure Bot Service

Creating a bot—an application that automatically and autonomously interacts with users—is no simple task. You need to keep track of the context of your interaction with each user and be ready to respond to a multitude of possible interaction parameters.

The Azure Bot Service enables you to build intelligent, enterprise-grade bots and experiences that can extend your brand and keep you in control of your data. Begin with a simple Q&A bot or build a sophisticated virtual assistant. Use comprehensive open-source SDK and tools to easily connect your bot across popular channels and devices. Give your bot the ability to speak, listen, and understand your users with native integration to Cognitive Services.

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

- It provides a way to host and manage bots you've built using the <u>Microsoft Bot</u> <u>Framework</u>, with a comprehensive <u>open-source SDK and tools</u> for bot development.
- It integrates natively with Cognitive Services.
- It helps you connect your bot to where your customers are, with connectors to channels including Facebook, Slack, Microsoft Teams, Line, Telegram, and more.
- It offers all the benefits of a managed service in Azure, including massive scale and built-in continuous deployment, and you pay only for what you use.

An example of a bot you can build with Azure Bot Service is one that provides users with answers to FAQs.

You can use this with **QnA Maker**. The interface of the bot can be a chat box on your website. You could also build your own branded <u>virtual assistant</u>.



Azure Machine Learning Studio

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

First, let's talk about what machine learning is.

What is machine learning?

Machine learning is often thought to mean the same thing as AI, but they aren't actually the same. AI involves machines that can perform tasks characteristic of human intelligence. AI can also be implemented by using machine learning, in addition to other techniques.

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

So, what's involved in machine learning? Figure 4-3 shows the basic workflow of machine learning:

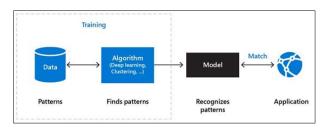


Figure 4-3: Basic workflow of machine learning

The machine learning process works as follows:

- Data contains patterns. You probably know about some patterns, such as user ordering habits. It's also likely that there are many patterns in data with which you're unfamiliar.
- The machine learning algorithm is an intelligent piece of software that can find patterns in data.
 This algorithm can be one you create using techniques such as deep learning or supervised learning.
- Finding patterns in data using a machine learning algorithm is called "training a machine learning model." The training results in a machine learning model. This contains the learnings of the machine learning algorithm.
- Applications use the model by feeding it new data and working with the results. New data is analyzed according to the patterns found in the data. For example, when you train a machine learning model to recognize dogs in images, it should identify a dog in an image that it has never seen before.

The crucial part of this process is that it is iterative. The machine learning model is constantly improved by training it with new data and adjusting the algorithm or helping it to distinguish correct results from wrong ones.

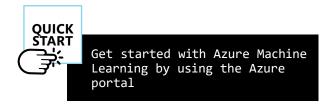
Azure Machine Learning

You can use <u>Azure Machine Learning</u> to create data analytics algorithms with open-source tools such as Python, R, and the Azure CLI. Just like with Azure Machine Learning Studio and Azure Machine Learning designer, you can create whatever algorithm you want, providing flexibility for a variety of scenarios, including predictive analytics, data recommendations, and data classification.

With Azure Machine Learning, you create custom machine learning algorithms from scratch. This is different to Machine Learning Studio and Machine Learning designer, where you visually create an algorithm by connecting predefined pieces of a machine learning algorithm. Azure Machine Learning fully supports open-source technologies such as TensorFlow, PyTorch, and scikit-learn.

Azure Machine Learning is a complete service that offers start-to-finish capabilities. You can create your algorithm, prepare your data, train the algorithm on it, test and deploy the algorithm, and track and manage it when it's running.

Azure Machine Learning works with many Azure services that can help create, train, and run your algorithm. You can, for instance, create your algorithm in Jupyter Notebook, train it using Azure Databricks, and deploy it on a Kubernetes container cluster in Azure Kubernetes Service.



Azure Machine Learning designer

You can use Azure Machine Learning designer to prep data and train, test, deploy, manage, and track machine learning models without writing any code. There is no programming required; you visually connect datasets and modules to construct your model.

To get started using Azure Machine Learning designer, you first need to create an Azure Machine Learning workspace. This workspace provides a centralized place to work with all the artifacts you create in Azure Machine Learning.

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

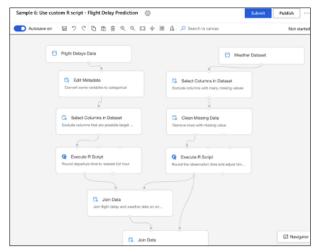


Figure 4-4: Using a custom R script for flight delay prediction

Figure 4-4 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.



AutoML

AutoML, is the process of automating the time-consuming, iterative tasks of machine learning model development. Traditional machine learning model development is resource-intensive, requiring significant domain knowledge and time to produce and compare dozens of models. With automated machine learning, you'll accelerate the time it takes to get production-ready ML models with great ease and efficiency.

During training, Azure Machine Learning creates a number of pipelines in parallel to try different algorithms and parameters for you. 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.

AutoML can save time in choosing the correct algorithm and parameters needed to produce a good model that fits your data.

Developer tooling for Al

ML.NET

ML.NET is an open-source and cross-platform machine learning framework with support for macOS, Windows, and Linux. ML.NET lets you reuse all the knowledge, skills, code, and libraries you already have as a .NET developer so that you can easily integrate machine learning into your web, mobile, desktop, gaming, and Internet of Things (IoT) applications.

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

Al Toolkit for Azure IoT Edge

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

Described earlier in the <u>Cognitive Services</u> section, <u>Custom Vision</u> already supports running on the edge. You can expect more services to run on the edge in the future.

To run machine learning models on the edge, you need tooling to help you deploy the models and web services. The AI Toolkit for Azure IoT Edge helps with this tooling by enabling you to package machine learning models in Azure IoT Edge—compatible Docker containers and to expose those models as REST APIs. The Docker containers containing the machine learning models can then be deployed as an IoT edge module to run on edge devices using Azure IoT Edge.

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

Al and mixed reality

Your applications are no longer limited to a 2D environment. The world is now your application canvas, backed by spatial intelligence from things such as IoT sensors, mixed reality, and computer vision. With Azure mixed-reality services, you can bring data to life in 3D when and where your users need it.

Azure Spatial Anchors

In the world of mixed reality, you can integrate digital information within the context of your physical environment, such as a hologram of your favorite game characters on your kitchen counter. With <u>Azure Spatial Anchors</u>, you can place digital content in a physical location and share that with 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 mixedreality experiences.
- Way-finding is a method of connecting two or more spatial anchors together, creating a relationship between them. These connected points of interest can be used to create 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.

For example, people entering a hospital often have difficulty getting to where they need to go. With Azure Spatial Anchors, the hospital can create a mobile application that shows digital information in the physical hospital to guide people to various locations. Within the application on their iOS device, people can use the directional arrows on the hospital's physical information boards to get to their destination.

Another way in which Azure Spatial Anchors is being used is in a training application for nurses.

Pearson Education has enabled nursing students and professors to practice diagnosing and treating patients in 3D before the pressure of a real case. Students and professors may use HoloLens devices or mobile phones and tablets running iOS or Android.

Azure Spatial Anchors enables you to share digital information and holograms that are positioned in the physical world. It works with applications built on Unity, ARKit, ARCore, and Universal Windows Platform (UWP), and can be used with a HoloLens device, iOS-based devices supporting ARKit, and Android-based devices supporting ARCore.

With Azure Spatial Anchors, you can easily secure your spatial data and give users access through Azure Active Directory. You can also integrate storage, Al, analytics, and IoT services in your spatial application.



Get started by sharing Azure Spatial Anchors across sessions and devices

Remote Rendering

When you use 3D models in scenarios such as design reviews and medical procedure plans, you need them to be as detailed as possible. Every detail matters.

Many businesses use complex 3D models containing hundreds of millions of polygons, and edge devices with low or medium graphics-processing power are not capable of rendering them. Traditionally, developers have tried to address this problem using a technique called "decimation." This makes the model simpler by removing polygons so that it can display on those devices.

But this loss of detail sacrifices information needed to make the right decision in many situations. With Remote Rendering, 3D models are rendered in the cloud and streamed to devices in real time—with no compromise of visual quality.

This enables you to keep the original quality of the model and interact with the content on edge devices such as headsets and mobile phones with every detail intact.

Azure Kinect DK

Azure Kinect DK is a developer kit with advanced Al 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 software development kits (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

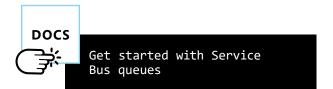
Azure Kinect DK can be used to build new possibilities, such as in the retail industry to design a shopping experience that manages inventory and tracks products faster and more accurately, and to create interactive signage that's both helpful and delightful.

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 leverage AI.

Service Bus

The core of messaging in Azure is <u>Service Bus</u>. 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 users 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 user 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.

A Service Bus queue is a simple mechanism.

Multiple applications can put messages on the queue, but a queue message can be processed by only one application at a time. A number of clever features are available for working with messages on the queue, including duplicate detection and a dead letter subqueue to which messages are moved when they fail to be processed correctly.

Service Bus topics

Just like Service Bus queues, <u>Service Bus topics</u> 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.

Comparing Service Bus queues and Azure Queue storage

Service Bus queues and Azure Queue storage basically do the same thing, but there are a number of differences, as shown in Table 4-3:

Table 4-3: Understanding the key differencesService Bus queues versus Queue storage

Use Azure Service Bus queues if you need	Use Azure Queue storage if you need	
Message lifetime > 7 days	Message lifetime < 7 days	
Guaranteed message order (first in–first out)	Queue size > 80 GB	
Duplicate detection	Transaction logs	
Message size ≤ 1 MB	Message size ≤ 64 KB	

Event Hubs

<u>Event Hubs</u> can help enterprises capture massive amounts of data to analyze it or transform and move it 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 seven 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.



Get started sending messages to Event Hubs

Azure IoT Hub

Just like Event Hubs, <u>Azure IoT Hub</u> is built for massive data ingestion. It's specifically geared toward handling the enormous volume of data messages from devices on the IoT, such as smart thermostats and sensors in cars.

It possesses many of the same properties as Event Hubs, including the ability to retain messages for up to seven days and replay them.

What makes IoT Hub unique is that it can send messages to devices. It has the ability to manage your complete IoT infrastructure—you can use it to register devices, report their state, manage them by securing and restarting them, and send data to them.



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 to 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 4-5:

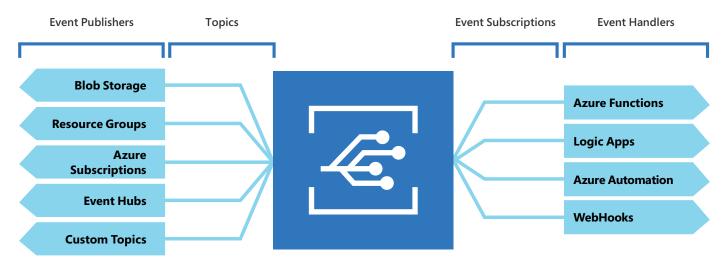
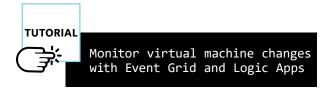


Figure 4-5: Flow of events from event publishers to event handlers

Event handlers can be Azure functions or Azure 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 Azure Logic Apps and Azure Functions, Event Grid scales automatically and doesn't need an instance of it to be deployed. You just configure it and use it, and you pay only when it's used.

You can use Event Grid if you want an email notification every time a user is added to, or removed from, your mailing list in Mailchimp. Event Grid is used to activate an application in Azure 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 a user who has been added or deleted and the action that was performed.



Azure SignalR Service

You can use <u>Azure SignalR Service</u> 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.



Create a chat room with SignalR

What to use, and when

Azure provides a myriad of options to perform messaging and to decouple applications. Which one should you use, and when? Table 4-4 summarizes the differences to help you choose.

Further reading

You can learn more about using Azure Al services in your application by clicking on the following links:



Table 4-4: What to use whenA summary of Azure services for events and messages

	Event ingestion	Device management	Messaging	Multiple consumers	Multiple senders	Use for decoupling	Use for publish/ subscribe	Max message size
<u>SignalR</u> <u>Service*</u>			0	•	0		•	64 KB
Event Grid*	•		0	0	0		•	64 KB
Event Hubs*	0		0	0	0	•		256 KB
<u>loT</u> <u>Hub*</u>	0	•	0	0	0	0		256 KB
Service Bus topics*			0	0	0	0		1 M
Service Bus queues*			0		0	0		1 M
Azure Queue storage	•	0						64 KB

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

05 /

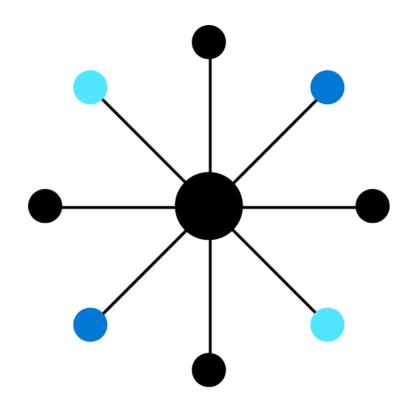
Connecting your business with IoT and gaining insights

How can Azure connect, secure, manage, monitor, and control your loT devices in the cloud?

One of the most exciting things that you get to do as a developer is impact the real world. You can do that with devices like robots, sensors, and microcontrollers. When you create applications with these devices, you can do things like predict when a machine needs maintenance before it actually requires repairs or even create a self-driving car.

Working with devices to impact the real world can be difficult if you develop all the software yourself. Fortunately, Azure provides solutions that can help make this a lot easier. You can, for instance, use Azure IoT Hub to securely ingest messages from and send commands back to devices. When you're building anything with a microcontroller (single-chip computers) in it, consider using Azure Sphere, which is a combination of hardware, OS, and cloud services with security built into the silicon.

Let's go through the services in Azure that can help you build amazing Internet of Things (IoT) applications.



Azure IoT Hub

At the core of Azure IoT is <u>Azure IoT Hub</u>, a flexible cloud PaaS that connects, monitors, and manages IoT devices in a secure and scalable manner.

You already know that you can use IoT Hub to ingest massive amounts of messages that typically come from IoT devices, like messages that contain data from temperature sensors. What's more, IoT Hub is unique because it not only receives messages but also sends commands back to devices. It establishes two-way communication with devices and even lets you execute code on devices.

IoT Hub is powerful because it allows you to manage devices in various ways, like sending them a message to reboot themselves or running a startup script. This makes IoT Hub the central service that enables a robust IoT application in Azure.

loT Hub can also connect, monitor, and manage devices that can run workloads developed in the cloud, including those that run the Azure IoT Edge runtime and modules. Once a device is connected to IoT Hub, the hub has a record of its identity. This enables IoT Hub to send messages and monitor the device; it also allows IoT Hub to secure the device and the communications between them. Devices are required to authenticate to IoT Hub using several industry best-practice security protocols, like X.509 certificates and SAS token-based authentication. You can manage the security of

each connected device and revoke privileges if you no longer want a particular device to be connected.

When devices send messages to Azure IoT Hub, you can either store the messages or route them to another service for analysis or action. It's possible, for instance, to route incoming messages using IoT Hub message routing that offers simplicity, reliability, and scalability without the complexity of building custom routing solutions.

Another option for integrating IoT events into Azure services or business applications is to use <u>Azure</u> <u>Event Grid</u>, a fully managed event routing solution that uses a publish-subscribe model. IoT Hub and Event Grid work together to integrate IoT Hub events into Azure and non-Azure services in near-real time.

You can also create bi-directional communication tunnels using device streams. <u>Azure IoT Hub</u>

<u>Device Streams</u> facilitate the creation of secure bidirectional TCP tunnels for a variety of cloud-to-device communication scenarios.

Example: Azure IoT Hub

A company that provides insights into the movement and usage of trucks is scaling out its business. Previously, the company tracked its assets by using custom code on a native phone app, which called a custom web service, and by polling GPS dongles attached to the trucks. This solution was challenging to maintain because it was difficult to provision new assets with new

devices, and the company needed to enroll a new customer that had more than 2,000 assets.

Now the company uses Azure IoT Hub for device management and communication. It can use the IoT Hub Device Provisioning Service to onboard the 2,000 new devices and hook them up to a specific IoT hub for that customer. The phone app now uses Node.js and the Azure IoT Device SDK to interact with IoT Hub. Importantly, the company now has control over the security of its devices and can detect their status and reset them as needed. In addition, the company routes the data from its GPS dongles through Azure Stream Analytics, so only the data of GPS changes is kept. This reduces the data burden because the dongles send their location every second.

Using Azure IoT Hub enabled this company to scale and mature its business by providing first-class security and device management. It also opened new opportunities to do more with devices than the company ever thought possible.

Azure IoT Hub Device Provisioning Service

The Azure IoT Hub Device Provisioning Service (DPS) provides a rich set of integrated public cloud services for all your IoT solution needs. It's a helper service for Azure IoT Hub that enables zero-touch, just-in-time provisioning to the appropriate IoT hub without intervention, allowing you to provision devices in a secure and scalable manner.

The service can help you with many device provisioning scenarios for getting devices connected and configured to Azure IoT Hub:

- Zero-touch provisioning Provision devices to a single IoT solution without hardcoding the Azure IoT Hub connection information at the factory (during initial setup)
- Geo-sharding Connect devices to the Azure IoT Hub with the lowest latency
- Rolling Key Updates Update device keys (when not using X.509 certificate authentication) used by the device to connect to Azure IoT Hub

To provision devices through DPS, an enrollment for the device(s) is added to the DPS service using one of three attestation mechanisms:

- X.509 Certificate based on the standard X.509 certificate authentication flow (similar to how TLS / SSL works)
- Trusted Platform Module (TPM) based on a nonce challenge, using the TPM standard for keys to present a signed Shared Access Signature (SAS) token.
- Symmetric Key based on Shared Access Signature (SAS) security token

The device(s) will use the configured attestation mechanism to connect to the DPS service. From there, the DPS service will determine the appropriate Azure IoT Hub for the device and provision the device with that particular IoT hub. Then the device will connect to and communicate with that Azure IoT Hub. This provides a complete zero-touch, just-in-time provisioning mechanism for connecting devices secure and scalable manner.

Azure Security Center for IoT

Billions of new IoT devices are being connected to the Internet and integrated into your daily lives and businesses. A difficult job of security operations teams is to ensure their security strategies evolve quickly enough to cover each new attack surface.

Azure Security Center for IoT provides threat prevention and analysis for every device, IoT Edge and IoT Hub, across your IoT solution. It simplifies hybrid workload protection by delivering unified visibility and control, adaptive threat prevention, and intelligent threat detection and response across workloads running on edge, on-premises, in Azure, and in other clouds.

There are a few different components that Azure Security Center for IoT is composed of:

- IoT Hub integration
- Device agents (optional)
- Send security message SDK
- Analytics pipeline

Azure IoT Central

Azure IoT Central is a Software as a Service (SaaS) offering that enables you 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 all that for you, provisioning and configuring everything you need.

There are several benefits to using Azure IoT Central:

- Lower barrier to entry into building and managing IoT solutions that can adapt to your needs and processes
- Push the limits of existing boundaries by utilizing a SaaS solution to more easily create new revenue opportunities and accelerate through the use of smart, connected products
- Utilize industry leading technology and enterprise-grade security features to enable you to stay in control of your IoT devices and data
- Enterprise-grade security features are built-in; offering you instant security benefits without building it yourself

Azure IoT Central gives you a jump-start to creating new IoT solutions by choosing from a number of industry-specific app templates for retail, energy, healthcare, and government to help you get up and running. Alternatively, you can also choose a blank, custom app template to build a fully customized solution with Azure IoT Central.

The Azure IoT Central app templates consist of:

- Sample operator dashboards
- Sample device templates
- Simulated device producing real-time data
- Pre-configured rules and jobs
- Rich documentation tutorials and how-tos

You get the same capabilities as if you had created the solution from scratch, but without the need for years of programming experience. If you do want more control over certain areas, you can always go deeper and tweak the solution to your needs.

Solutions built using Azure IoT Central still utilize Azure IoT Hub and Device Provisioning Service (DPS). In addition to the ability to hook into these services and customize your solution, you also benefit from many other capabilities these services offer.

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."

You would use edge computing when you don't want to or can't rely on your connection to the cloud, when you want to improve your application performance by eliminating roundtrips to the cloud, or when you can't communicate with the cloud from the device because of security, privacy, or regulatory reasons.

For scenarios such as these, use <u>Azure IoT Edge</u>. Azure IoT Edge is managed from IoT Hub, enabling you to move parts of your workload to the edge. This reduces time spent by devices sending messages to the cloud and allows offline scenarios as well as faster reactions to status changes.

Azure IoT Edge is composed of the following components:

- IoT Edge modules are containers that run
 Azure services, third-party services, or your own
 code. They're deployed to IoT Edge devices and
 execute locally on those devices.
- The IoT Edge runtime runs on each IoT Edge device and manages the modules deployed to each device.
- IoT Hub exposes specific interfaces to remotely monitor and manage IoT Edge devices available through the Azure portal, the Azure CLI, or the SDKs.

These three components work together on devices and in the cloud to run your workloads at the intelligent edge.

It's possible to run many Azure services at the edge to help with certain scenarios—and the list of available services keeps growing. Table 5-1 lists just some of them.

Table 5-1: Making a scenario based decisionServices available on Azure IoT Edge

If you want to	Use this on Azure IoT Edge		
Build and deploy Al models	Machine learning		
Customize computer vision models for your use case	Custom Vision Service		
Process real-time streaming data	Stream Analytics		
Process events using serverless code	<u>Functions</u>		
Deploy a SQL Server database to the edge	SQL Server databases		
Comply with Industry 4.0 interoperability standards	OPC Unified Architecture		
Build custom logic	Custom module		

Once you start using Azure IoT Edge, you'll be able to create fast applications that run machine learning algorithms locally and provide instant feedback on their findings.

Azure Digital Twins

In the world of IoT, you might work with many IoT devices and sensors that function in connection to environments and business systems. To really understand the data that IoT devices capture, you need to understand the physical environment in which the devices operate.

For instance, if you have a temperature sensor, the temperature data by itself doesn't tell you that much. However, when you know which room the sensor is in, how large the room is, which other devices are in the room and what data they capture, and how many people are in the room and how they're moving, you get a much better picture of what the temperature data means. You can model the physical environment in which your IoT devices reside with Azure Digital Twins. This service enables you to create a graph of data that includes places, people, and things—for instance, an office building that contains a room with people and sensors.

Azure Digital Twins is an IoT platform that enables enterprises to create a digital representation of any assets, environments and business systems. Leveraging their domain expertise on top of Azure Digital Twins' flexible modeling, enterprises can build contextually aware solutions to address scenarios such as workflow modeling in manufacturing, energy efficiency in smart spaces, digital twin simulation in oil and gas and many more. With Azure Digital Twins next generation IoT

solutions, customers can track the past, control the present and predict the future.

Azure Digital Twins applied to all types of environments, such as, warehouses, offices, schools, hospitals, banks, stadiums, factories, and more. There are many scenarios where Azure Digital Twins can be helpful:

- Predict maintenance needs for a factory
- Analyze real-time energy requirements for an electrical grid
- Optimize the use of available space for an office
- Track daily temperature across several states
- Monitor busy drone paths
- Identify autonomous vehicles
- Analyze occupancy levels for a building
- Find the busiest cash register in your store

Whatever the real-world business scenario, it's likely a corresponding digital instance can be provisioned through Azure Digital Twins.

Azure Digital Twins also lets you to manage permissions to data and devices in the context of the physical world. You can use Azure AD to specify that certain users are able to access data only from a certain physical location.



Get started by finding available rooms using Azure Digital Twins.

Azure Sphere

More and more devices contain microcontrollers (single-chip computers) to make them smarter. This enables scenarios such as a washing machine sending a routine maintenance notification to the owner.

However, it's complex and challenging to secure devices with embedded microcontrollers, as evidenced by the many security incidents with connected devices over the last years.

Azure Sphere provides a secured, high-level application platform with built-in communication and security features for internet-connected devices. The platform is comprised of hardware, operating system (OS), and cloud components to actively protect your devices, your business, and your customers.

These three components provide the basis for a connected, secure world using Azure Sphere:

- Azure Sphere MCU Secured, connected, crossover microcontroller unit (MCU)
- Azure Sphere OS Custom, high-level Linuxbased operating system (OS)
- Azure Sphere Security Service (AS3) Cloudbased security service that provides continuous, renewable security



You can use the Visual Studio family to develop applications that run on Azure Sphere. This opens up the world of microcontroller development to a much larger group of developers across both Windows and Linux.

Azure Sphere Security

The Azure Sphere Security Service (AS3) is a cloud service that provides remote attestation to authenticate devices and ensure they haven't been tampered with. It also provides a secure mechanism to push down Azure Sphere OS and other application updates to Azure Sphere devices.

Based on Microsoft's decades of experience with internet security, the Azure Sphere team has identified seven properties of highly secured devices. The Azure Sphere platform is designed around these seven properties to provide a more highly secured IoT platform.

Azure Sphere Development Kit

Azure Sphere applications can be developed using a Windows or Linux system using an Azure Sphere Development Kit. The development kit is a hardware board that provides everything you need to start prototyping and developing Azure Sphere applications. There are a few different options of Azure Sphere Development Kits available for purchase.

Azure Maps

Azure Maps provides developers from all industries with powerful geospatial capabilities. Those geospatial capabilities are packed with the freshest mapping data. Azure Maps is available for both web and mobile applications.

Azure Maps provides a collection of geospatial services that use fresh mapping data to provide geographic context to web and mobile applications. These services can be used to develop secure, scalable IoT and AI solutions with dynamic, location-based services.

Azure Maps services are available through SDKs. These tools help you quickly develop and scale solutions that integrate location information into Azure solutions.

- Search Create apps that provide details about nearby points of interest, render locations on a map, and geocode (or reverse geocode) addresses to get corresponding latitude / longitude coordinates.
- Maps Integrate clear, easy-to-read maps into your solutions with JavaScript Map Control or the Render API.
- Time zones Make it easy for users to see what time it is anywhere in the world.

- Spatial operations Enhance your location intelligence with a library of common geospatial mathematical calculations, including geofencing, closest point, great circle distance, and buffers.
- Traffic Develop mobility solutions that improve travel time and avoid gridlock.
- Routing Present the shortest and fastest routes available or provide specialized routs and facts for walkers, bicyclists, and commercial vehicles.
- Mobility Provide real-time location intelligence and nearby public transit services, including stops, route information, and travel time estimations.
- Geolocation Look up the country for an IP Address.
- Data Service Upload and store geospatial data for use with spatial operations or image composition.
- Weather Service Obtain current weather conditions, weather forecast, and weather along a route to enable weather-based decisions within your applications.

Windows for IoT

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

Windows 10 IoT come 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 form 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. This way, you can understand what features you'll need and what configurations you'll want when it's time to manufacture.

Windows 10 IoT Core is optimized for smaller devices with or without a display that run on both ARM and x86/x64 devices. IoT Core is a UWP (Universal Windows Platform) centric operating system, and UWP apps are the primary app type.

Universal Windows Platform (UWP) is a common app platform across all versions of Windows 10, including Windows 10 IoT Core. UWP apps are written using Visual Studio and is supported by a handful of languages; including C#, VB.NET, and JavaScript.

Windows Server IoT 2019

Windows Server IoT 2019 is a full version of Windows Server 2019 that delivers enterprise manageability and security to IoT solutions. Windows Server IoT 2019 shares all the benefits of the world-wide Windows ecosystem. It is a binary equivalent to Windows Server 2019, so you can use the same familiar development and management tools that you use on your general-purpose servers.

Azure RTOS

Azure RTOS is a small, fast, reliable, and easy-to-use real-time operating system (RTOS) for connecting deeply embedded sensors, devices, and gateways from the edge to the Internet of Things (IoT).

Azure RTOS is best suited for building applications on resource constrained devices. It provides developers a rich set of real-time multi-tasking, FAT file system, GUI, USB and networking components to quickly build applications; including out-of-the-box connectivity to Azure IoT.

Azure RTOS is comprised of a few different components:

- Azure RTOS ThreadX is an advanced real-time operating system designed specifically for deeply embedded applications
- Azure RTOS FileX is a high-performance, file allocation table (FAT)-compatible file system.
- Azure RTOS GUIX Studio provides a complete, embedded graphical user interface (GUI) application design environment, facilitating the creation and maintenance of all graphical elements in the application's GUI.

- Azure RTOS TraceX is a Windows-based analysis tool that provides a view of real-time system events.
- Azure RTOS NetX is a high-performance TCP/IP protocol implementation.
- Azure RTOS NetX Duo is an advanced, industrial-grade TCP/IP network stack designed for embedded real-time and IoT applications.
- Azure RTOS USBX is a high-performance USB host, device, and on-the-go (OTG) embedded stack.

This advanced real-time operating system (RTOS) is designed specifically for deeply embedded applications. Among the multiple benefits it provides are advanced scheduling facilities, message passing, interrupt management, and messaging services. Azure RTOS ThreadX has many advanced features, including picokernel architecture, preemption threshold, event chaining, and a rich set of system services.

The decision to use a real-time operating system (RTOS) should be made when an application requires real-time capability. An RTOS should offer consistently fast response times (in microseconds), a small memory footprint (less than 2MB), and a level of security that effectively protects users and connected systems.

Learn more about Azure IoT

Azure IoT solutions are easy to use, and there are many related resources, including:

- Azure IoT Developer Specialty Certification:
 Earn a certification that demonstrates you understand how to implement Azure IoT solutions.
- Azure IoT application page: This resource provides an overview of Azure IoT and examples of how it can be used.
- Azure IoT Hub: This resource provides an overview of Azure IoT Hub and examples of how it can be used.
- Azure IoT Edge: This resource provides an overview of Azure IoT Edge and an example of how it can be used.
- Azure IoT technical videos: Learn more about IoT on Channel 9.

What to use, and when?

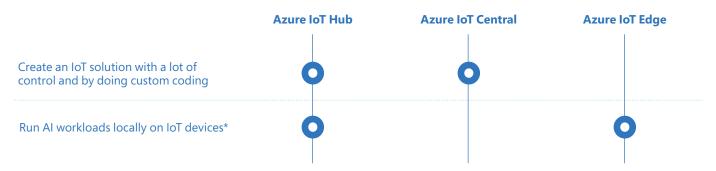
Now that you've read about the available Azure IoT services, how do you know which service to use for your scenario? Table 5-2 shows when you should use each IoT option in Azure.

Table 5-2: Making the right choice Various scenarios for using Azure IoT options

Further reading

Learn more about using Azure for your IoT solution in the following free resource:





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

^{*}Azure IoT Hub is required to manage Azure IoT Edge deployments and devices.

06/

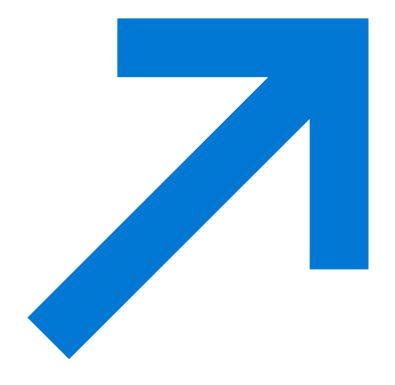
Deploying your services and reducing costs

How can Azure deploy your services and reduce costs?

Azure has an option for every type of organization, including those that need Azure in their own datacenters. You can deploy your applications either in the public Azure cloud or on-premises in Azure Stack. You choose how portable your applications should be.

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

Let's explore these options in more detail.



Infrastructure as Code

laC captures environment definitions as declarative code, such as JSON documents, for automated provisioning and configuration. This enables you to use the same versioning 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 enables the cost of development and test environments to be reduced by creating them on demand.

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 as 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.

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

Terraform with Azure

Hashicorp <u>Terraform</u> 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. 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.



Get started by defining and assigning an Azure Blueprint in the Azure portal.

Containers in Azure

Containers are a standard for packaging up code and its dependencies for running quickly and reliably across different compute environments. A container is basically a lightweight VM that starts and stops much faster than a traditional VM and is therefore more useful for development, testing, and running applications in production.

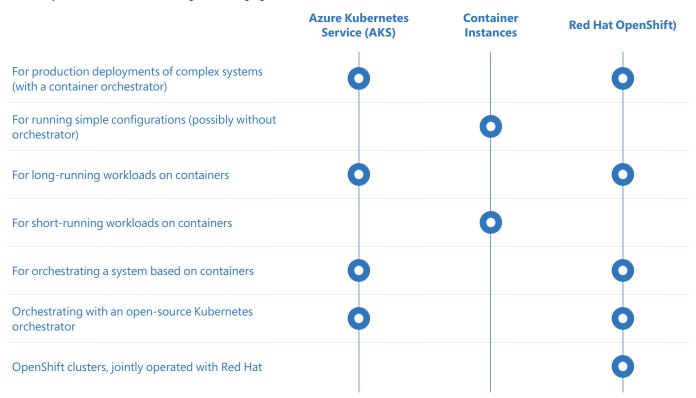
The major benefit of containers is that an individual container is always the same, or immutable. You run a container locally when you develop your app, and then use the same container configuration in the cloud and everywhere else. Your entire team

Table 6-1: What to use whenA summary of Azure services for running and managing containers

uses the exact same container configuration, so you know that the infrastructure is the same for everybody as it is in production. With containers, the age-old developer's fallback statement—"it works on my machine"—now means that it will also work in production.

Container instances are disposable and can be quickly created or destroyed when scaling or running a workload on-demand. This provides flexibility to run containers when needed, or even move them to a different node or cluster easily, too.

There are many technologies for running containers, including <u>Docker</u>. Azure can run and manage containers with <u>Azure Container Instances</u>, <u>Azure Kubernetes Service</u>, and <u>Azure Red Hat OpenShift</u>. Table 6-1 shows which service you might choose for various scenarios when using containers:

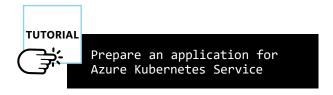


Azure Kubernetes Service

Azure Kubernetes Service (AKS) makes it simple to deploy a managed Kubernetes cluster in Azure. AKS reduces the complexity and operational overhead of managing Kubernetes by offloading much of that responsibility to Azure.

As a hosted Kubernetes service, Azure handles critical tasks, such as health monitoring and maintenance, for you. When deploying an AKS cluster, the Kubernetes master and all nodes are deployed and configured for you.

As a managed Kubernetes service, AKS is free—you only pay for the agent nodes within your clusters, and not for the masters.



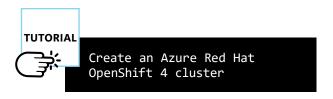
Azure Container Instances

Azure Container Instances provides a fast, isolated compute environment to easily run containers on Azure without managing servers. You can develop applications quickly while running your application in a container or in the cloud. It allows you to focus on designing and building your applications instead of managing the infrastructure that runs them.

Azure Red Hat OpenShift

Azure Red Hat OpenShift provides a flexible, self-service deployment of fully managed OpenShift clusters. Maintain regulatory compliance and focus on your application development while your master, infrastructure, and application nodes are patched, updated, and monitored by both Microsoft and Red Hat.

Azure Red Hat OpenShift extends Kubernetes. It is jointly engineered, operated, and supported by Red Hat and Microsoft to provide an integrated support experience.



It's important to choose the deployment and hosting option for your applications. It's also important to keep track of the cost of hosting your applications, and your overall spend on Azure services. Next, let's take a look at tracking your Azure usage.

Tracking your Azure usage

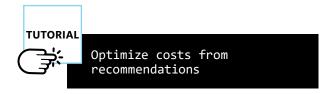
With Azure products and services, you only pay for what you create and the Azure resources you use. It's important to keep track of what you are using and the costs involved.

Azure Cost Management

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

Reports in Azure Cost Management show the usage-based costs consumed by Azure services and third-party Azure Marketplace offerings. Costs are based on negotiated prices and factor in reservation 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.



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.



Figure 6-1: Azure Advisor showing recommendations for under-utilized VMs

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 at 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.

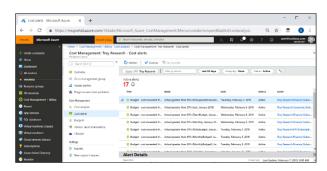


Figure 6-2: Cost alerts displayed in Azure Cost Management

Possible status on 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 our 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 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 for getting better insights into your cloud spend during the month, your estimated consumption, and a few other things.

Where to deploy, and when?

There are several service options for deploying and hosting applications in Azure. Each service option includes its own capabilities and features that make it the best fit for a certain application.

Let's look at the reasons why you might choose platform as a service (PaaS) or infrastructure as a service (laaS) for hosting different application types.

Platform as a service

There are many benefits to hosting applications with PaaS services. PaaS services offer managed VMs, where you no longer need to perform updates to the OS or the platform stack installed on the machine. This enables you to spend more time on writing the application, managing your data, and delivering business value.

There are several different types of applications, each with its own unique requirements. These applications will fit best being hosted using different Azure PaaS services.

Here are some example application types, and the Azure services to consider for hosting:

 Microservices: Azure Kubernetes Service, Azure Red Hat OpenShift, Azure Container Instances

- Large N-tier application: Azure Kubernetes
 Service, Azure Red Hat OpenShift
- Small N-tier application: Azure App Service
- Serverless and event-driven: Azure Functions
- Traditional monolithic application: Azure App Service

There is some capability overlap between the different services that may be used to host a particular application. This helps give flexibility in choosing which Azure service is best suited for hosting your application based on your custom business needs and scenario.

It may not be the best option to choose a container-based hosting option, such as AKS, when deploying an existing enterprise application to Azure. The Azure App Service, or even laaS, may be viable alternatives. However, some existing applications may fit nicely in a container, or could be redesigned as such with minimal work.

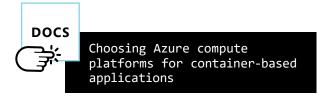
For new application development, the first choices to look at are containers and serverless. These are both more modern application hosting and design options. When building new applications, it can be easier to choose the hosting and design upfront, and then build the application with the necessary technologies.

Infrastructure as a service

laaS offers greater flexibility since you can install and configure various software components on the VMs as required. This can be especially useful for existing applications that may require some custom configuration or dependency to be set up on the server that isn't allowed on other PaaS services.

While choosing PaaS services is the best path to start with, there are plenty of scenarios where your application may require the full flexibility and capabilities of using laaS. This includes the ability to install any software and/or database on the VM that you require for your application.

Keep in mind that laaS brings with it the need for you to manage the VM, OS, and software updates yourself.



Further reading

Learn more about deploying your applications to Azure and reducing costs in these free resources:



Control Azure spending and manage bills with Azure Cost Management + Billing

Deploy a website to Azure with Azure App Service

07/

Share your code, track work, and ship software

How can Azure help you plan smarter, collaborate better, and ship your applications faster?

If you've spent weekends or nights deploying new versions of your applications, you've likely also spent a lot of time trying to fix the bugs that keep users away from those new versions. There's a better way.

Azure DevOps is a set of solutions that can help automate your builds and deployments and automatically test your code and applications before launch.

To help you build, deploy, test, and track your code and applications, Azure DevOps includes:

Azure Boards

Use Azure Boards to plan, track, and discuss work across teams.

Azure Repos

Use Azure Repos to collaborate on code development with Git public and private repositories, pull requests, and code reviews.

Azure Pipelines

Use Azure Pipelines to create build and release pipelines that automate builds and deployments.

Azure Test Plans

Use Azure Test Plans to improve your overall code quality with manual and exploratory testing services for your applications.

Azure Artifacts

Use Azure Artifacts to share code packages (such as the npm, NuGet, and Maven packages) across your organization.

<u>GitHub</u> also offers Git repositories and GitHub Advanced Security, which can be used to help secure your code.

Let's explore the Azure DevOps and GitHub services in more detail.

Azure Boards

Planning your work and tracking your progress are important tasks—and <u>Azure Boards</u> can help you complete them:



Figure 7-1: Viewing backlogs with Azure Boards

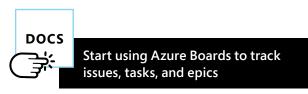
In Azure Boards, you can create a complete backlog of work items (such as user stories) and plan them in sprints so your team can work iteratively to finish the tasks.

The whole planning system is optimized for working in an agile way. It even includes Kanban boards for managing your work (Figure 7-1).

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.

Try customizing your boards and creating charts (such as burndown charts or task lists) that show the

information you need. You can query work items and progress, and then use this information to customize your boards, charts, and lists. From there, you can share them or pin them on your Azure DevOps dashboard for everyone to see.



Azure Repos

Version control is essential for working together and ensuring that your most important asset—your code—is safely stored. Azure Repos is a set of version control tools for storing your code and sharing it with your team. This is useful both for teams and individual developers. Version control keeps a history of your development so you can review or even roll back to any version of your code.

Git

This is a widely used version control system among developers and is also the basis for GitHub. Git is a distributed version control system, meaning the complete source code (all versions of all files) is on your machine—which makes it easy to work offline. With Git, the source of truth is essentially on everyone's machine and is synchronized when developers push their code to the Git server (in this case, Azure Repos).

Azure Repos uses standard Git. This means that you can use it with any Git tool and IDE, including <u>Visual Studio</u> and <u>Visual Studio Code</u>, as well as Git for <u>Windows</u>, <u>macOS</u>, Linux, <u>Eclipse</u>, and <u>IntelliJ</u>.

When you follow the <u>Git workflow</u>, 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.



Get started by learning how to code with Git

Azure Pipelines

Once your code is in a repository such as Azure Repos, you can start to automate your build and release processes with Azure Pipelines.

Azure Pipelines provides a lot of value in a small amount of time. It enables continuous integration (CI) for compiling and testing code when changes come in, as well as continuous deployment (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 or 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

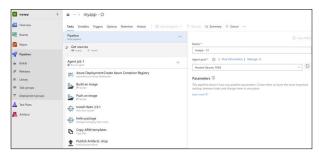


Figure 7-2: Azure Pipelines: build pipeline

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.

There are two ways of working with Azure Pipelines. You can <u>create pipelines</u> using the visual designer in the Azure DevOps portal, or you can use the more advanced YAML-based approach. In this approach, you create a YAML code file, which contains all the steps of the pipeline, and commit that to source control.

The easiest way to get started is to use the visual designer. Let's take a look at an example of a build and release pipeline:

Build pipeline: The build pipeline (or CI pipeline) in Figure 7-2 shows a list of tasks that will be executed when this pipeline runs. The pipeline is configured to run as soon as new changes are committed to Azure Repos. It will take the code in Azure Repos (which is a Node.js app), build a Docker container image from it, and push that to Azure Container

Registry. From there, Helm can use it to compile the image into a package that can be deployed on Azure Kubernetes Service.

Note that you can configure which hosts run your pipelines for you. On the right side of the figure, you can see that this particular pipeline will run on a hosted pool of Ubuntu Linux machines. There are also Windows hosts available, and even a hosted macOS that you can use to build your macOS and iOS applications. This is just one example of a build pipeline. It's possible to create one for every imaginable application. You can also integrate tests, including unit tests and static code tests, into the build pipeline.

Release pipeline: The release pipeline (or CD pipeline) executes as soon as the build pipeline runs successfully, though you can also configure it to be triggered manually. The release pipeline in Figure 7-3 contains nine tasks that first create an Azure Kubernetes Service cluster and then deploy the Helm package that was produced in the build pipeline to the cluster.

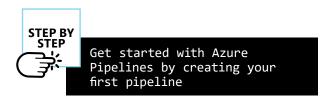
Figure 7-3 shows the steps in the development stage. Stages are like environments. You can configure a stage for your development, test, and production environments, and so on. You can also configure things such as pre-deployment approvals, which require someone to approve the release of an application into a specific environment (such as the production environment):



Figure 7-3: Steps in the development stage of the release pipeline

This means you can automate everything and leave the decision to release to production up to a manager based on test results for previous steps in the pipeline.

Make your pipelines as simple or 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 <u>Visual Studio Marketplace</u>.



Azure Test Plans

To improve the quality of your applications, use <u>Azure Test Plans</u> to define test plans and then create and execute manual and exploratory tests. Azure Test Plans provides the tooling to author tests, execute them, record feedback, and track the test results.

For automated testing as part of your CI/CD workflow, consider leveraging Azure Pipelines. It provides mechanisms for continuous building, testing, and deployment to any platform and cloud.

In Azure Test Plans, start by creating a test plan. This contains multiple test suites and test cases. A test case can be an exploratory test, in which the application is explored to see whether it works as expected; a guided manual test, in which test steps and expected outcomes are described in detail; or an automated test. You can even record test steps by recording clicks in an application and letting Azure Test Plans automate those clicks into a test. You can also incorporate stress and load tests into your build and release pipelines. Test cases are work items, just like user stories and tasks, and can therefore be scheduled within an iteration.

Once you've created a test (Figure 7-4), a tester can run it. In a manual test, for example, the tester uses the test tool to run through the test steps and records findings, including the screen, the tester's voice, screenshots, and attachments. The tester passes or fails each step of the test.

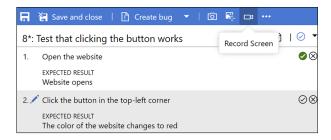
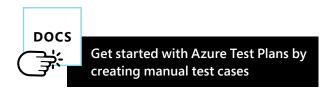


Figure 7-4: Creating a test

From the context of the test, the tester can also create a bug that needs to be solved.

To ensure stakeholders' expectations are in line with your plan, Azure Test Plans also enables you to request feedback for work items, such as user stories. This enables stakeholders to take a look at what you propose and provide feedback in the form of text, attachments, video, or voice.

Perhaps most importantly, Azure Test Plans provides dashboards and charts on the progress and status of the tests in your project. You can use these to see what the quality of your application is and how it progresses over time. This can help you identify features that aren't ready to be deployed.



Azure Artifacts

Because packages offer functionality that you don't have to build yourself, you probably use a lot of them in your applications. And you likely access them from just as many sources: NuGet, npm, Maven, and others. But what if your team creates packages that you want to only use internally? Where do you host them securely, and how do you share them? Azure Artifacts provides this capability. Azure Artifacts is a package feed that allows you to host packages that you create and secure them for your organization.

You can host <u>all sorts of packages</u> 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. <u>Publish</u> your package to the feed.
- Consume the feed in your favorite IDE, such as Visual Studio.



Get started with Python packages in Azure Artifacts

GitHub

<u>GitHub</u> offers Git repositories along with additional capabilities centered around team collaboration and communication on projects. GitHub also offers features that can help secure your code.

Repositories on GitHub

GitHub provides Git source code control with some additional features that help to increase teams' ability to collaborate. The features include the option to create and track issues and pull requests within a GitHub repository. Issues are used to help monitor the discussion around bugs and other actionable items that need to be performed. Pull requests enable contributors to collaborate on a project by submitting code change proposals in a way that allows discussion and approval before changes are merged into the repository.

GitHub Advanced Security

GitHub Advanced Security is used to secure your code by providing a semantic code analysis engine that allows developers to write queries that identify code patterns in large codebases and search for vulnerabilities and their variants.

Simple declarative queries can be used by security researchers to quickly find vulnerabilities in code. These queries can then be shared with the open

source community to be used for improving the safety of code in other codebases.

No single organization can find every vulnerability or secure the open source supply chain behind everyone's code. Semmle's community-driven approach helps more easily identify and prevent security vulnerabilities in code.

GitHub Dependabot

GitHub Dependabot creates pull requests to keep your dependencies secure and up to date. Dependabot creates pull requests that include release notes, changelogs, commit links, and vulnerability details whenever available.

Dependabot monitors security advisories for Ruby, Python, JavaScript, Java, .NET, PHP, Elixir, and Rust. Pull requests are created immediately in response to new advisories.

Further reading

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:



08/

Microsoft Azure in action

Navigating the Azure portal

The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription using a graphical user interface.

Dashboards in the Azure portal

Dashboards provide a focused view of the resources in your subscription that matter most to you. 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 affect your experience only. However, you can create additional dashboards for your own use or publish customized dashboards and share them with other users in your organization.

Adding services

You can find and add services in the Azure portal 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'll find everything from web applications to Linux servers, as shown in Figure 8-1:

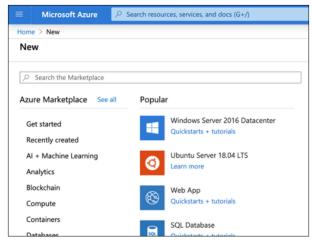


Figure 8-1: Azure Marketplace pane

The Azure Marketplace pane is pre-populated with popular services grouped into categories. If this list doesn't include what you're looking for, then you can use the Search the Marketplace box to type in a search.

When you find the service you want from the search results, as shown in Figure 8-2, a wizard takes you through its configuration and deployment:

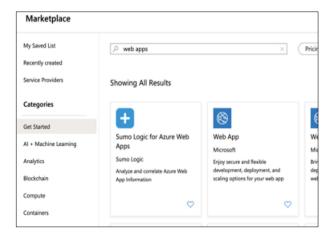


Figure 8-2: Searching for a service in Azure Marketplace

You can use the search box at the top of the portal to search through all your resources and go directly to them (Figure 8-3).

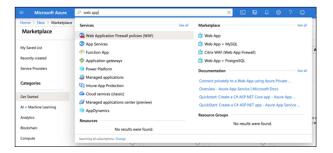


Figure 8-3: Navigating to a resource using the search box

This search drop-down menu displays a list of Azure services that can be created, your existing resources and resource groups, Marketplace items, and documentation links that match your search. It's there to help you discover what you're looking for.

Understanding blades

Pages in Azure are also called blades, and you can pin them to your dashboards. When you open a web application, you first see the Overview blade, as shown in Figure 8-4.

For example, the App Service blade provides the ability to stop, start, and restart the web application and display tiles showing its metrics, such as the number of requests and errors:

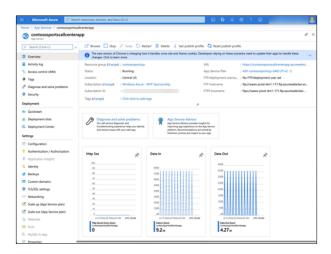


Figure 8-4: The Overview blade

Creating a new VM

Let's use the Azure portal to create a new VM. Once we've done so, we'll shut it down and remove it so that you don't continue to pay for it.

A word about resource groups

The VM will be deployed in a resource group, a logical container that holds resources. All Azure resources reside inside resource groups. You can manage the security of a resource group as well as seeing what the resources in the group cost. It's common practice to bundle related services in a resource group so that they're easier to secure:

- In the Azure portal, go to the home screen and select + Create a resource.
- Search for and select Windows Server, and then choose the Windows Server 2019 Datacenter option.
- 3. The Create a virtual machine wizard opens.
- 4. Choose a resource group to add the virtual machine to. Selecting Create new can be used to create a new resource group at this time.
- 5. Choose a Virtual machine name.
- 6. Choose the Region of the VM.
- 7. Notice that the Size of the VM is already set to a default pricing tier. You can select **Change size** to select a different size if desired. There are many sizing options for VMs. VM performance determines the cost. Use the wizard to select how many cores and how much memory you

want, and choose options based on that. In addition, there are other features that come with size options, such as:

- The type of hard drive (SSD or normal HDD).
- The number of max input/output operations per second (IOPS). This determines the performance of the VM in a significant way, especially if your applications read and write extensively from and to the hard drive.
- The amount of data drives that can be installed in the VM.
- The ability to perform load balancing.
- The graphics card installed in the VM.
 This is useful if you need to execute substantial graphics rendering or a heavy computational workload.
- 8. Type a **Username** for the administrator account for the virtual machine.
- 9. Type a **Password** and confirm.
- 10. Notice that the Public inbound ports option is set to Allow selected ports, and the Remote Desktop Protocol (RDP) port of 3389 is selected. This will enable you to remote into the VM once created.
- 11. Select Next: Disks >
- **12.** You can configure additional settings such as the OS disk type, any additional data disks to

attach to the VM, as well as the virtual network, IP address, and extensions on the machine. For now, leave everything as is and select **Review + create**.

13. Review the summary, agree to the terms, and then select **Create**.

It usually takes just a few minutes for the VM to be deployed. When you navigate to the VM in the Azure portal, you can configure it further and log in using Remote Desktop Protocol (RDP):



Figure 8-5: Using RDP for VM configuration

To log in to the VM using RDP, click Connect in the VM's Overview blade in the Azure portal (Figure 8-5). From there, you can download an RDP file you can use to connect to the VM using Microsoft Remote Desktop.

Cleaning up the walkthrough resources

When you're finished with the VM, shut it down and remove it by deleting the resource group that we created when we generated the VM. This contains the VM and all other resources that are automatically created. Once the resource group is deleted, you no longer pay for any of the resources that you've used in this walkthrough.

Developing your first web application and database on Azure

In this walkthrough, we'll deploy a simple .NET Core application that connects with a SQL database. Then, we'll host using the Web Apps feature of Azure App Service.

To follow along, you'll need <u>Git v2 or higher, .NET</u> <u>Core</u>, and <u>Visual Studio Code</u> installed on your device. We'll also use a sample ASP.NET Core MVC application to manage a to-do list.

Creating a web application and database using the Azure portal

To host the .NET Core application, we'll create a new web application in the Azure portal.

- 1. In the Azure portal, select + Create a resource.
- Search for Web App. The Web App blade opens. Select Create. The Web App Create blade opens.
- Create a new Resource Group by giving it a name.
- 4. Type a Name for the web app.

- Leave the Publish selection as Code, and then select the .NET Core 3.1 (LTS) value for Runtime stack.
- 6. Leave the OS selection as **Windows**.
- 7. Select or create an **App Service Plan**.
- Select Review + create, and then select Create.

Services such as Web Apps run on Azure App Service plans. App Service plans are an abstraction of resources and features, like CPU and memory, and are represented in pricing tiers.

App Service plans are also bound to a specific geographic region that you choose. You can, for instance, run your Web Apps application in an App Service plan of pricing tier S1, which has 1 core and 1.75 GB RAM, as shown in Figure 8-6:



Figure 8-6: Choosing an App Service plan

You can run as many application services on an App Service plan as you wish, but note that you need to share resources among all the application services.

To host the database, we'll create a SQL database. This works the same as a local SQL Server database and now runs fully managed in Azure:

1. In the Azure portal, click + Create a resource.

- Search for SQL Database and click it to open the SQL Database blade. Click Create. The Create SQL Database blade opens.
- Select the resource group that you created for the web app.
- 4. Type a database name.
- 5. Click **Server** to create a new SQL database server:
 - a. Type a name for the server.
 - b. Type the server admin login. This is the username for the server.
 - c. Type the password that you'll use to log on to the server.
 - d. Confirm the password.
 - e. Choose a location. Choose the same location that you selected for the App Service plan.
 - f. Click **OK** to submit the new server configuration.
- On Compute + storage, select Configure
 database, and then select a pricing tier. For
 development and test purposes, the Basic tier is
 sufficient.
- Click Review + create, and then Create. The database will now be created.
- Navigate to the SQL database and click Show Database Connection String.

Make a note of the connection string because you'll need it later in this tutorial.

Running the .NET Core application locally

Let's run the application locally before we run it in Azure. The application can run locally because, by default, it uses a SQLite database, which is a selfcontained SQL database engine:

- Open a Command Prompt or terminal window, and then navigate to a directory you want to use as your source code directory for this project.
- 2. Run the following commands to get the source code and navigate to the project folder:

```
git clone https://github.com/azure-samples/
dotnetcore-sqldb-tutorial
cd dotnetcore-sqldb-tutorial
```

3. The project uses Entity Framework Core to populate its database. To ensure that the database is up to date and to run the application locally, execute the following commands:

```
dotnet restore

dotnet ef database update

dotnet run
```

 The application should now be running, and the URL to the application (such as http:// localhost:5000) should be in the output in the Command Prompt window.

- Navigate to that URL in a browser. This will load the application, which will look like that shown in Figure 8-7. Now you can create new to-do items by selecting the Create New link.
- 6. Close the application by closing the Command Prompt window or by pressing Ctrl+C:

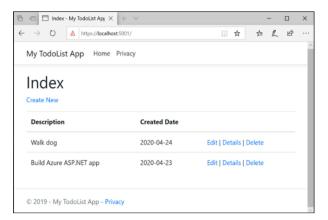


Figure 8-7: The application running locally in a browser

Connecting the local web application to the database running in Azure

You now have a working application running locally. Before we deploy it to Azure, we'll change the source code so that it can connect to the SQL database:

 In your local source code repository, find the Startup.cs file and locate the following code:

```
servicesAddDbContext<MyDatabaseContext>
(options => options.UseSqlite("Data
Source=localdatabase.db"));
```

Replace the code with the following code, which will connect to the Azure SQL database:

```
//Use SQL Database if in Azure, otherwise,
use SQLite
```

```
if(Environment.
GetEnvironmentVariable("ASPNETCORE_
ENVIRONMENT ") == "Production") services.
AddDbContext<MyDatabaseContext>(options =>

options.UseSqlServer(Configuration.
GetConnectionString("MyDbConnection")));
else

services.
AddDbContext<MyDatabaseContext>(options =>

options.
UseSqlite("DataSource=localdatabase.db"));

// Automatically perform database
migration services.BuildServiceProvider().
GetService<MyDatabaseContext>().Database.
Migrate();
```

This code looks at the environment in which it's running and changes its database connection based on that information. When running in the production environment (Azure, in this case), the code will get the connection string for the database from the MyDbConnection variable, which we'll configure in Azure.

The code also runs the Database.Migrate() method, which executes the Entity Framework Core migrations that we previously ran manually.

Save your changes and run the following commands to commit the changes to your local Git repository:

```
git add .
git commit -m "connect to SQLDB in Azure"
```

Now, we'll configure the connection string variable in Azure.

4. In the Azure portal, navigate to the web application that we created earlier.

- 5. Navigate to **Configuration** under Settings.
- 6. Create a new connection string named MyDbConnection. The value should be the connection string to the SQL database (including the username and password) you saved earlier when you created the database and set Type to SQLAzure.
- Click Save. The application settings in the Azure portal should look like those shown in Figure 8-8:



Figure 8-8: Application settings in the Azure portal

Deploying the web application to Azure

We'll use Git to push the application to Azure. To connect the local Git repository to Azure, you must have a deployment user configured on the server (Web Apps) to authenticate your deployment. The deployment user is account level and is different from your Azure subscription account. You need to configure this deployment user only once:

- 1. In the Azure portal, navigate to the **web app**.
- On the web app in the Azure portal, select Deployment Center.

- Select Local Git, then App Service build service, and then click **Finish**.
- Copy the **Git Clone Uri** that has been created for the web app. This will be used to push the application code to the web application using Git:



Figure 8-9: Git Clone Uri being displayed

- Select the **Deployment Credentials** button to set the credentials that will be used to authenticate Git when deploying the app.
- On the Deployment Credentials pane, select the User Credentials scope under the Local Git tab.
- 7. Enter a Username and Password, and then click Save Credentials. This needs to be set up the first time so that you have credentials to authenticate against the Git endpoint that will be used.
- 8. Open the command prompt on your local machine.
- Add a remote to your local Git repository by using the remote Git URL:
 - a. Replace <deploymentLocalGitUrl> with the Git Clone Uri from the Azure web app Deployment Center.

b. Use the URL to run the following command:

git remote add azure
<deploymentLocalGitUrl>

10. Once the remote target is added to the Git repository, you can push your code to it by running the following command. You'll need to enter credentials to be able to push code to Azure. Use the username and password you used to create the deployment user:

git push azure master

When prompted, enter the Username and Password that were created previously using Azure Cloud Shell.

Pushing the source code to Azure might take a few minutes the first time. Once complete, navigate to the URL of your Azure web app, which will look like this: http://<app_name>. azurewebsites.net.

11. Add some to-do items in the application to test its connection to the database.

Now you have a working application running in Azure.

Extending applications with Logic Apps and Cognitive Services

A powerful feature of our 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 <u>Logic Apps</u> feature of Microsoft Azure App Service and <u>Language</u> <u>Understanding (LUIS)</u>, 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 Office 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 we can use it later in our 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"

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 send to LUIS to use 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

With your application running, you can now use Azure to make it more robust and easier to update.

Setting up continuous delivery with GitHub

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.

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 to the cloud.

Setting up staging environments

Using Azure App Service Web Apps, 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.

The deployment slots all run in the same App Service plan, which is what you pay for. Keep in mind that having additional deployment slots in an App Service plan will consume resources such as CPU and memory, so be mindful of how additional slots might impact production.

You 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 you have many users, you need web apps to scale up to accommodate increased traffic. When they're not busy, you need them to scale back to reduce costs. You can do this with the Automatic Scaling feature of App Service. You need to run the web apps in the Standard or Premium pricing tier to use this feature.

Alternatively, the Free tier could be used to run a single instance of a web app.

Web Apps has a menu item called Scale Out, as shown in Figure 8-10. You can use this to scale out manually or automatically. Scaling out means you add more instances of your application to handle the load:



Figure 8-10: The Scale Out menu item

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 see logging when you use the application in 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 in the code they happened. 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.

Further reading

Learn more about architecting Azure solutions in these free resources:



Microsoft Learn

in Azure learning Path on

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, using AI in your applications, and scaling as much as you want—and you only pay 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. We hope you continue to consult this e-book to become better acquainted with the vast range of Azure services and determine which ones best fit your needs.

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.

Keep learning with Azure

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

- 12 months of popular free services.
- \$200 credit to explore any Azure service for 30 days.
- 25+ services, always free.
- Get started with Azure: Watch these short
 tutorials on how to use Azure and start building
 projects right away. You can also join our
 weekly webinar, which provides demos of
 Azure basics and provides ongoing access to
 experts.
- Microsoft Learn: Learn new skills and discover the power of Microsoft Azure products with step by step guidance. 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.

- 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!

Free resources extravaganza

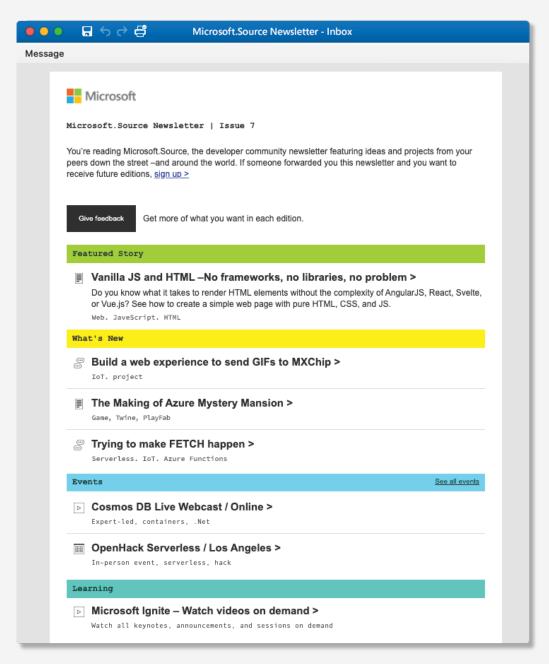
In addition to this guide, there are many other free resources about Azure, including:

- Learn Azure in a Month of Lunches: A practical way to learn Azure from scratch over a month of lunches.
- Azure Serverless Computing Cookbook:
 Find use cases, hands-on recipes, and tutorials for quickly configuring your own serverless environment.
- <u>Designing Distributed Systems</u>: An e-book on building containerized applications, with hands-on labs on Azure Kubernetes Service.
- Containerize Your Apps with Docker and Kubernetes: A practical guide for moving applications to the cloud with Docker and Kubernetes.
- Effective DevOps: A useful guide for improving collaboration across teams, promoting the efficient use of tools, and using the concepts of DevOps.
- <u>Kubernetes Up & Running</u>: Improve the agility, reliability, and efficiency of your distributed systems with Kubernetes.
- <u>Building Intelligent Cloud Applications</u>: Build and deploy scalable deep learning and machine learning models using serverless architectures with Azure.

- <u>Developers Guide to IoT</u>: An e-book that provides an overview of Azure IoT services and gets you started.
- Azure for Architects: A comprehensive guide for Azure architects.
- Thoughtful Machine Learning with Python:
 Explore machine learning fundamentals with
 Python, techniques for improving models, and developing viable frameworks.
- 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.
- Hands-On Linux Administration on Azure: Find everything you need to know about deploying and managing Linux workloads on Azure.
- Making the Most of the Cloud Everywhere:
 Unified development and modernization practices in hybrid environments.
- Enterprise Cloud Strategy: What you need to know now about moving your enterprise to a cloud computing strategy.







By developers, for developers

Microsoft.Source newsletter

Get technical articles, sample code, and information on upcoming events in Microsoft. Source, the curated monthly developer community newsletter.

- Keep up on the latest technologies
- Connect with your peers at community events
- Learn with hands-on resources



About the authors

Michael, Chris, and Vahe are passionate about Azure and encourage you to reach out to them on Twitter for questions regarding this book.



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, ombcrump, follow his live coding stream at https://www.twitch.tv/mbcrump, or catch up on a recent post in the Azure Tips and Tricks series.



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.

You can reach Chris on Twitter, <u>@crpietschmann</u>, and follow his blog at https://build5nines.com.



Vahe Minasyan is a technical project manager on the Microsoft Azure Product Marketing team. You can reach Vahe on Twitter, <u>@vaheminasyan2</u>, and find him on <u>GitHub</u>.

PUBLISHED BY Microsoft Press, A division of Microsoft Corporation

One Microsoft Way, Redmond, Washington 98052-6399

Copyright © 2020 by Microsoft Corporation. All rights reserved. No part of the contents of this book may be or transmitted in any form or by any means without the written permission of the publisher.

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 and the trademarks listed at www.microsoft.com on the "Trademarks" webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

