(1) Physical / Colocan data centers

(i) Instead of ~~big bought~~ bldg costly capital intensive data, IT companies (bunch of em) come together and rent spaces in shared facilities

(ii) Frees up capital for more flexib. uses as compared to real estate

(2) Virtualized data center

(i) Comps. match the parts of a physical data center; servers, disks etc.

(ii) But now, there are virtual devices separately managable from underlyin hardware

(iii) Resources used more → efficiently / flexibly

(iv) You can still buy, house maintain infra but you still gotta guess how much hardware you'll need & when, settin it up and keepin it runnin

(3) Cloud (~~serverless~~ GCP provides this in a serverless manner)

(i) In internal dould, services auto-provision and config the infra used to run familiar Google apps.

(ii) Uses ~~cont~~ container (cntr) based architecture

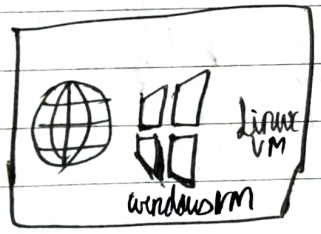
(iii) Provides wide variety of services (pre-configed) to manage & get data at large scale.

Cloud Computing

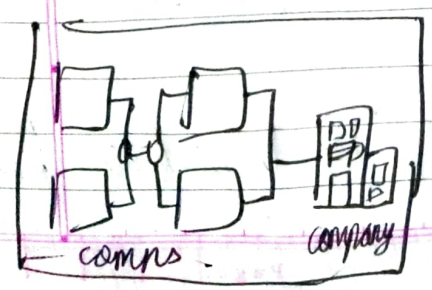
Delivery Model for services like

1. Storage → unstructured or structured data in files or DBs
   Cloud delivers user 1000s of services to do so.
   → Also stores all the services and tools read to migrate your data to the cloud.

2. Compute Power → web apps
   → Windows, Linux VMs
   → Any of 100s of services available in the cloud cuz cloud cuz cloud is apt creating apps → web
   → AI
   → ML
   → reporting
   → etc.



3. Networking → Connect your components to each other securely
   → cloud delivers a set of facilities to create secure networking b/w these comps, company

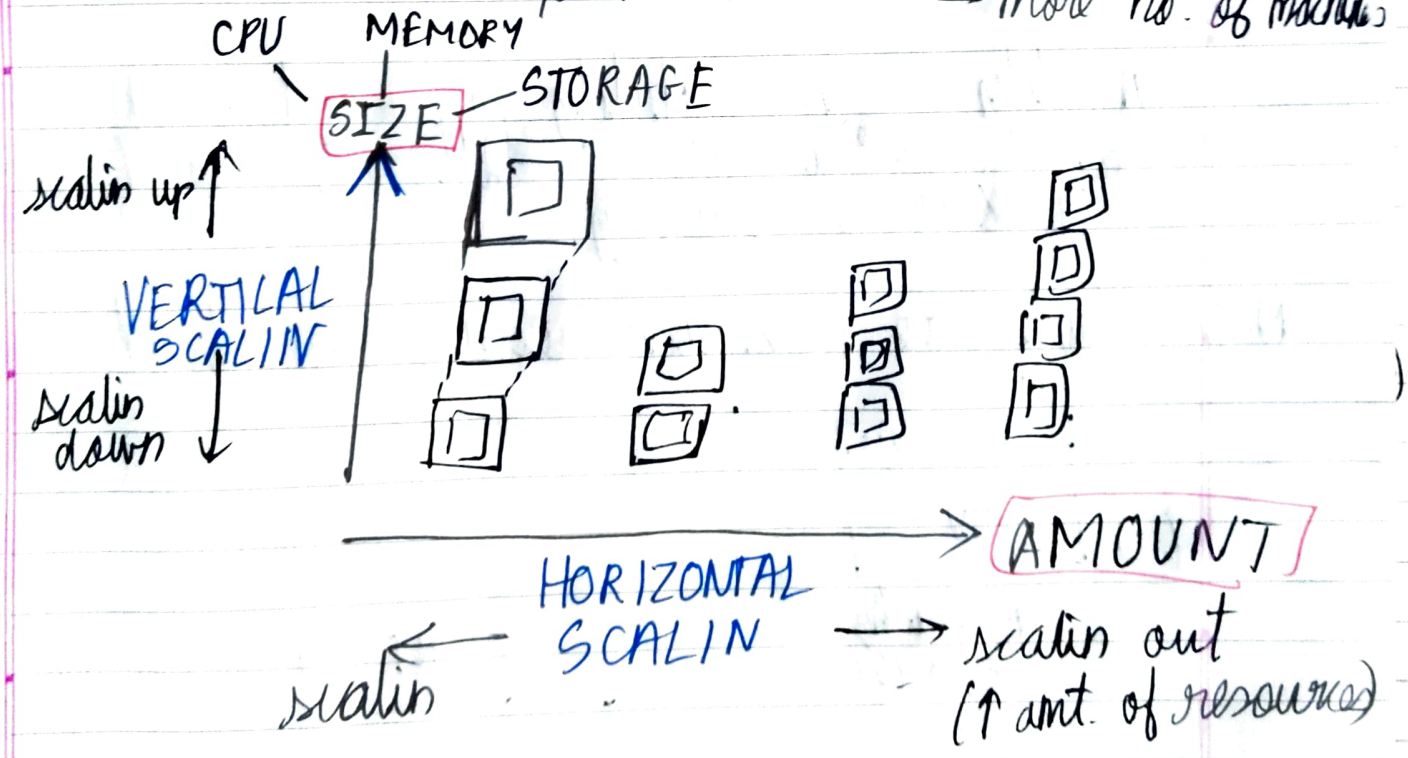


④ Analytics → to derive performance, telemetry data for your servers so you can perform read. op^n

These are the 4 MAIN services delivered by cloud computing over the internet

other characteristics Cloud Computin fulfils.

① Scalability → addin more power via
- more CPU, mem, faster storage
- more no. of machines



Scalin → process of addin or removin resources
- size → vertical
- amt. of instances of the resource → hori.

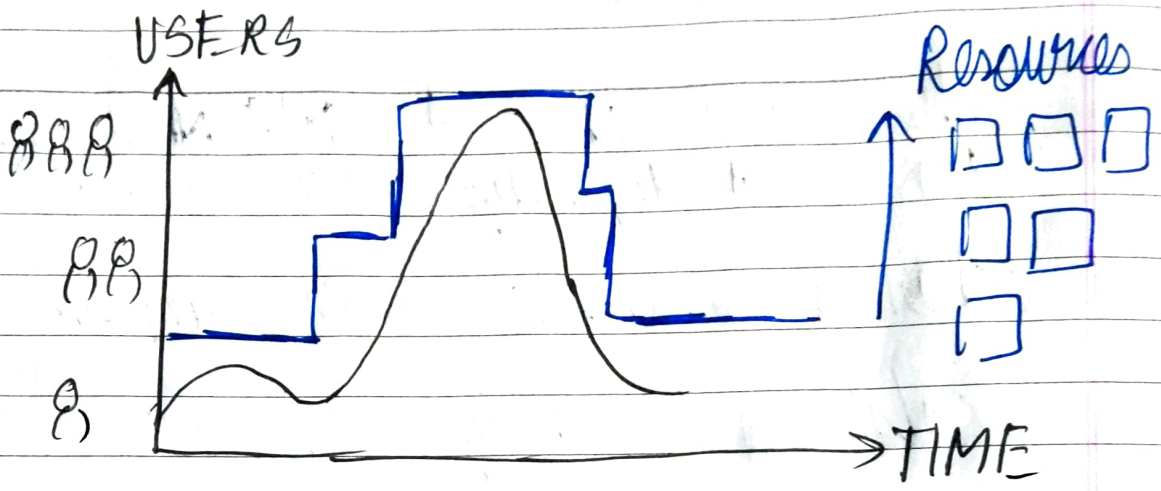
eg:- ↑ RAM to 64GB (verti)

↑ no. of 8GB RAM devices to 8 (hori)

(2) Elasticity :- dynamic alloca"/removal of resources as reqd.

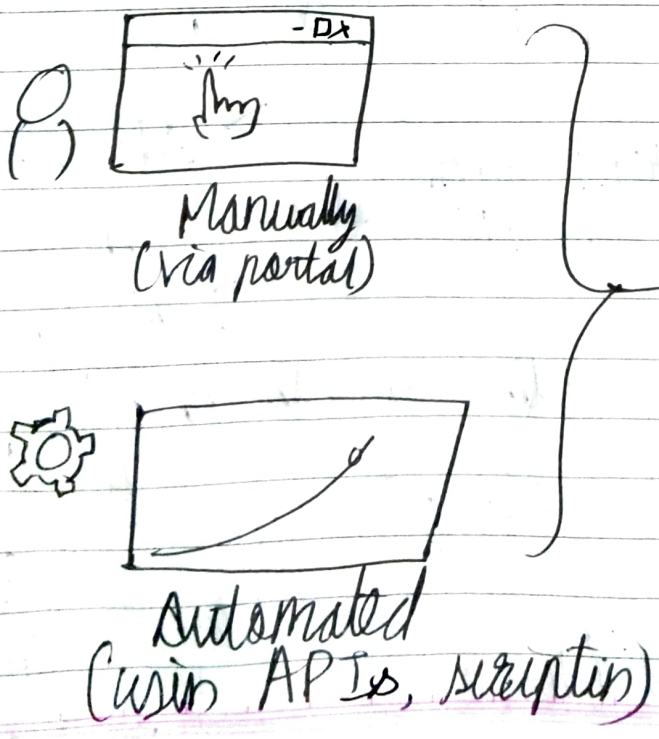
↳ aka auto-scaling

eg:- addin/removin resources as user workflow for that day changes



(3) Agility → The ability to react quickly i.e. ability to allocate & deallocate (scale) resources quickly

NOTE: 2 ways to provision resources in cloud



b/w req. is and getting back res.

Cloud vs On-Premise

Cloud
sec
min
hr

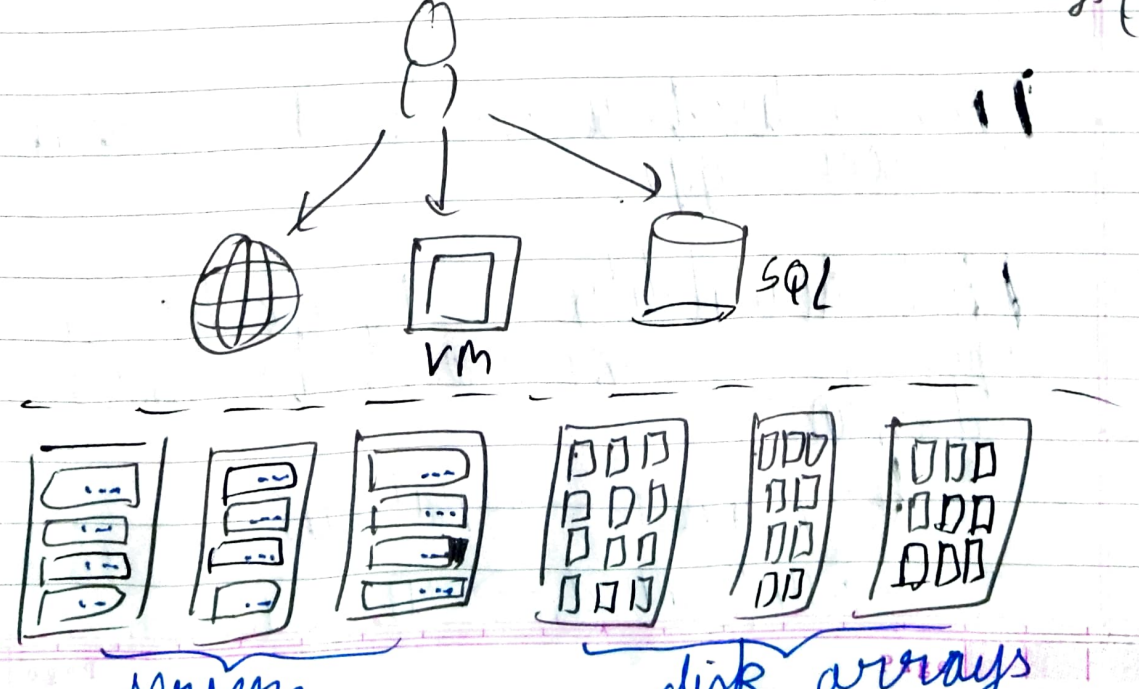
On-Premise
day
week
month

Cloud responds with the resource you req. ed very fast
∴ It is agile

(4) Fault Tolerance → Whener there is any service or comp failures on servers or when your data in disk array crashes → you get immediately assigned another one. s.t. no data is lost.

Fault tolerance → ability of sys to remain up & running durin comp & service failures.

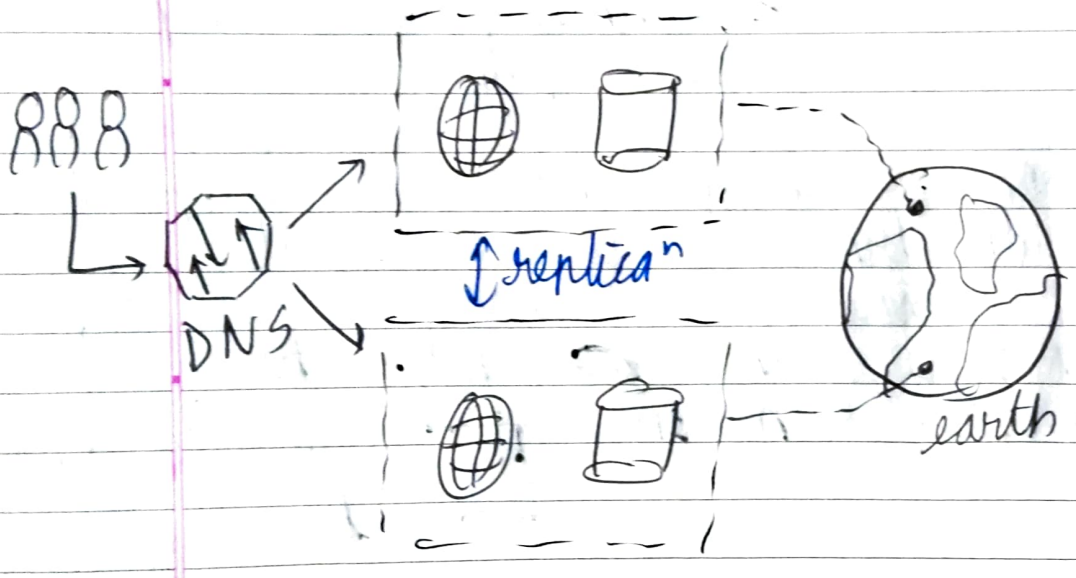
NOTE : Whener you purchase a web app, VM or SQL dB frm a service, regardless of the service you choose or interface you work with, under the hood services run on → servers (do computation)
→ disk arrays (store data)



Localized failures do not interrupt your service as most of the time, in the cloud all the services hv inbuilt fault-tolerance sys (immune to localized failures)

(5) Disaster Recovery → failure of a v. big scale
↳ disrupⁿ of services caused by natural or human-induced causes
eg:- power grid failures, storms.

Setup disaster recovery by 'replicaⁿ'
↓
enables sys to work properly even after a disaster



DNS redirects users to the workin version of your app.

(6) High Availability → Metric that measures uptime (sys being accessible to users) vs downtime
↳ planned → patchin
↳ unplanned → sys. failures

(allt)

availability → calc of uptime w.r.t a specific (time period (eg:- 1 yr, lifetime of srvce)

$$\frac{\text{uptime}}{\text{uptime} + \text{downtime}}$$

Depending upon client you calc allt per yr / month / day.

Allt	Year
99%	3.65 days
99.9%	8.77 hrs
99.99%	52.6 mins

Allt → measure of uptime for users / services

High allt → ability of sys to keep services running for extended periods of time with v. lil downtime.

(7) **Global Reach** → Ability to reach audiences around the globe.
↓
more customer base for your app.
→ Cloud services can have a presence in various regions across the globe, which & your customers can access, giving you a presence in those regions even tho you may not have any infra in that region

(8) **Custer latency capabs.** → Cloud services hv the ability to deploy resources in datacenters around the globe, which addresses any customer latency issues.

Cloud services

~~Cloud~~ gives amazin latencies to services to react to cust dc"s ~~then~~ when the service and local to the cust. eg
Modern optic fibres + local service → faster than cloud.

(9) Predictive cost considera"s → The ability for users to predict costs that they'll incur for a particular cloud service.

i) Cost for indiv services are made avail. and tools are provided to predict cost

ii) Can perform analysis based on planned growth.

(10) Technical skill reqmts & considera"s → A user can be an expert in the app they wanna run w/o requirin skills to build & maintain underlyin hardware & software for workloads (as cloud provides it all) infrd

(11) ↑ed Productivity →

i) On prem. data centers require →
- hardware setup aka "racking & stacking"
- software patchin
- othr time-consumin IT management chores

ii) Cloud Computin (CC) elims. need for many of these tasks, allowin IT teams to ~~spend~~ spend time focussin on achievin more imp business goals

(12) Security → Cloud providers (CPs) offer a broad set of policies, techs, controls, expert tech skills → provide better security than most orgs.
∴ We get strengthened security to protect
1) data
2) apps
3) infra
} → protected against potential threats

Principles of economics of scale

eg:- your own delivery company

Scale	X 3	X 300
Car	Indiv purchase : 10k	Bulk purchase : 9k
Maintenance	Indiv : 100	Contract : 90
Insurance	Indiv : 500	Bulk purchase : 400
Other services	Indiv : 100	Shared services : 80
Customers	8 x 5	8 x 5000
Price per unit (PPU)	10	9

Basically as your company grows (scales up), your PPU falls due to bulk purchases, contracts, shared services

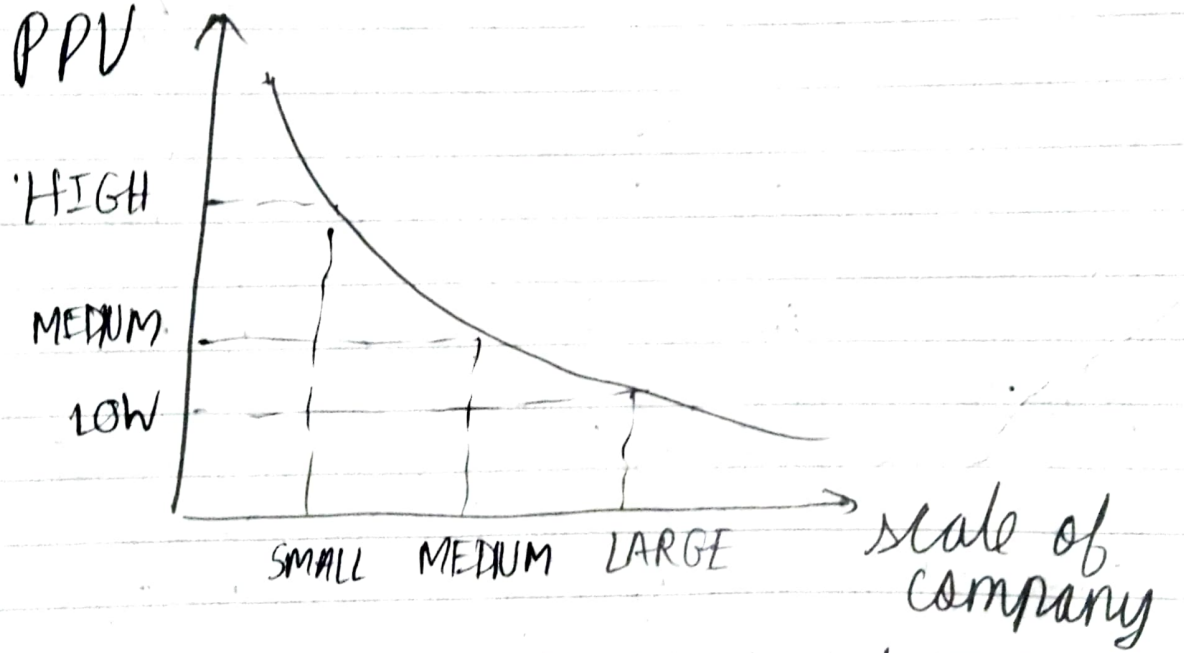
employ some 3rd party staff for maintenance etc.

Large companies → scale is large → can handle more cust-ers w/o sacrificing quality of service

Large companies ← lower PPV ←

get a lotta internal benefits which it shares with cust-ers

→ makes company more competitive to grow in market



This principle perfectly describes what's happening with the cloud ~~right~~ right now.

As the cloud grows & has more cust-ers, the prices of services go down

Reason → Big companies like Microsoft will get more efficient at what it does → building data centers
→ getting better ~~hard~~, advanced hardware

Thus,

Economies of scale → the ability to reduce costs and gain efficiency when operating at a larger scale.

Storage costs in the past decade have ↓ed significantly due to the CSPs' ability to ... → purchase larger amts of storage at significant discounts

→ then, they use that storage more efficiently and pass on those benefits to end users at low prices

Cloud Computing is the delivery of computing services—servers, storage, databases, networking, software, analytics, intelligence and more—over the internet (the cloud), enabling faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change.

The company providing these services is referred to as a cloud provider. Some example providers are Microsoft Azure, Amazon Web Services (AWS), and the Google Cloud Platform (GCP). The cloud provider is responsible for the physical hardware required to execute your work, in addition to keeping it up to date. Every business is unique and has different needs. To meet those needs, cloud computing providers offer a wide range of services. Typically, these services include:

- **Compute power** - such as Linux servers or web applications.
- **Storage** - such as files and databases.
- **Networking** - such as secure connections between the cloud provider and your company.
- **Analytics** - such as visualizing telemetry and performance data.

Cloud computing services

The goal of cloud computing is to make running a business easier and more efficient, whether it's a small start-up or a large enterprise. Every business is unique and has different needs. To meet those needs, cloud computing providers offer a wide range of services.

You need to have a basic understanding of some of the services it provides. Let's briefly discuss the two most common services that all cloud providers offer – *compute power* and *storage*.

Compute power

When you send an email, book a reservation on the Internet, pay a bill online, or even take this Microsoft Learn module you're interacting with cloud-based servers that are processing each request and returning a response. As a consumer, we're all dependent on the computing services provided by the various cloud providers that make up the Internet.

When you build solutions using cloud computing, you can choose how you want work to be done based on your resources and needs. For example, if you want to have more control and responsibility over maintenance, you could create a *virtual machine* (VM). A VM is an emulation of a computer - just like your desktop or laptop you're using now. Each VM includes an operating system and hardware that appears to the user like a physical computer running Windows or Linux. You can then install whatever software you need to do the tasks you want to run in the cloud.

The difference is that you don't have to buy any of the hardware or install the OS. The cloud provider runs your virtual machine on a physical server in one of their datacenters - often sharing that server with other VMs (isolated and secure). With the cloud, you can have a VM ready to go in minutes at less cost than a physical computer.

VMs aren't the only computing choice - there are two other popular options: *containers* and *serverless computing*.

What are containers?

Containers provide a consistent, isolated execution environment for applications. They're similar to VMs except they don't require a guest operating system. Instead, the application and all its dependencies is packaged into a "container" and then a standard runtime environment is used to execute the app. This allows the container to start up in just a few seconds, because there's no OS to boot and initialize. You only need the app to launch.

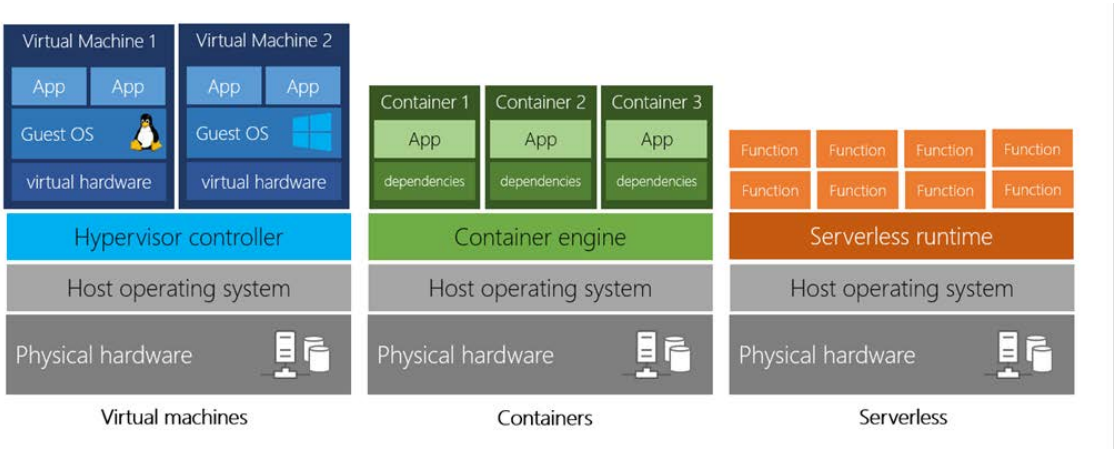
The open-source project, Docker, is one of the leading platforms for managing containers. Docker containers provide an efficient, lightweight approach to application deployment because they allow different components of the application to be deployed independently into different containers. Multiple containers can be run on a single machine, and containers can be moved between machines. The portability of the container makes it easy for applications to be deployed in multiple environments, either on-premises or in the cloud, often with no changes to the application.

What is serverless computing?

Serverless computing lets you run application code without creating, configuring, or maintaining a server. The core idea is that your application is broken into separate *functions* that run when triggered by some action. This is ideal for automated tasks - for example, you can build a serverless process that automatically sends an email confirmation after a customer makes an online purchase.

The serverless model differs from VMs and containers in which you only pay for the processing time used by each function as it executes. VMs and containers are charged while they're running - even if the applications on them are idle. This architecture doesn't work for every app - but when the app logic can be separated to independent units, you can test them separately, update them separately, and launch them in microseconds, making this approach the fastest option for deployment.

Here's a diagram comparing the three compute approaches we've covered.



Storage

Most devices and applications read and/or write data. Here are some examples:

- Buying a movie ticket online
- Looking up the price of an online item
- Taking a picture
- Sending an email
- Leaving a voicemail

In all of these cases, data is either *read* (looking up a price) or *written* (taking a picture). The type of data and how it's stored can be different in each of these cases.

Cloud providers typically offer services that can handle all of these types of data. For example, if you wanted to store text or a movie clip, you could use a file on disk. If you had a set of relationships such as an address book, you could take a more structured approach like using a database.

The advantage to using cloud-based data storage is you can scale to meet your needs. If you find that you need more space to store your movie clips, you can pay a little more and add to your available space. In some cases, the storage can even expand and contract automatically - so you pay for exactly what you need at any given point in time.

✓ Every business has different needs and requirements, and cloud computing is flexible and cost-efficient. The goal of cloud computing is to make running a business easier and more efficient, whether it's a small start-up or a large enterprise.

CapEx VS OpEx

In the past, companies needed to acquire physical premises and infrastructure to start their business. There was a substantial up-front cost in hardware and infrastructure to start or grow a business. Cloud computing provides services to customers without significant upfront costs or equipment setup time.

These two approaches to investment are referred to as:

- **Capital Expenditure (CapEx)**: CapEx is the spending of money on physical infrastructure up front, and then deducting that expense from your tax bill over time. CapEx is an upfront cost, which has a value that reduces over time.
- **Operational Expenditure (OpEx)**: OpEx is spending money on services or products now and being billed for them now. You can deduct this expense from your tax bill in the same year. There's no upfront cost. You pay for a service or product as you use it.

CapEx computing costs

A typical on-premises datacenter includes costs such as:

Server costs

This area includes all hardware components and the cost of supporting them. When purchasing servers, make sure to design fault tolerance and redundancy, such as server clustering, redundant power supplies, and uninterruptible power supplies. When a server needs to be replaced or added to a datacenter, you need to pay for the computer. This can affect your immediate cash flow because you must pay for the server up front.

Storage costs

This area includes all storage hardware components and the cost of supporting it. Based on the application and level of fault tolerance, centralized storage can be expensive. For larger organizations, you can create tiers of storage where more expensive fault-tolerant storage is used for critical applications and lower expense storage is used for lower priority data.

Network costs

Networking costs include all on-premises hardware components, including cabling, switches, access points, and routers. This also includes wide area network (WAN) and Internet connections.

Backup and archive costs

This is the cost to back up, copy, or archive data. Options might include setting up a backup to or from the cloud. There's an upfront cost for the hardware and additional costs for backup maintenance and consumables like tapes.

Organization continuity and disaster recovery costs

Along with server fault tolerance and redundancy, you need to plan for how to recover from a disaster and continue operating. Your plan should consist of creating a disaster recovery site. It could also include backup

generators. Most of these are upfront costs, especially if you build a disaster recovery site, but there's an additional ongoing cost for the infrastructure and its maintenance.

Datacenter infrastructure costs

These are costs for construction and building equipment, as well as future renovation and remodeling costs that may arise as demands grow. Additionally, this infrastructure incurs operational expenses for electricity, floor space, cooling, and building maintenance.

Technical personnel

While not a capital expenditure, the personnel required to work on your infrastructure are specific to on-premises datacenters. You will need the technical expertise and workforce to install, deploy, and manage the systems in the datacenter and at the disaster recovery site.

OpEx cloud computing costs

With cloud computing, many of the costs associated with an on-premises datacenter are shifted to the service provider. Instead of thinking about physical hardware and datacenter costs, cloud computing has a different set of costs. For accounting purposes, all these costs are operational expenses:

Leasing software and customized features

Using a pay-per-use model requires actively managing your subscriptions to ensure users do not misuse the services, and that provisioned accounts are being utilized and not wasted. As soon as the provider provisions resources, billing starts. It is your responsibility to de-provision the resources when they aren't in use so that you can minimize costs.

Scaling charges based on usage/demand instead of fixed hardware or capacity.

Cloud computing can bill in various ways, such as the number of users or CPU usage time. However, billing categories can also include allocated RAM, I/O operations per second (IOPS), and storage space. Plan for backup traffic and disaster recovery traffic to determine the bandwidth needed.

Billing at the user or organization level.

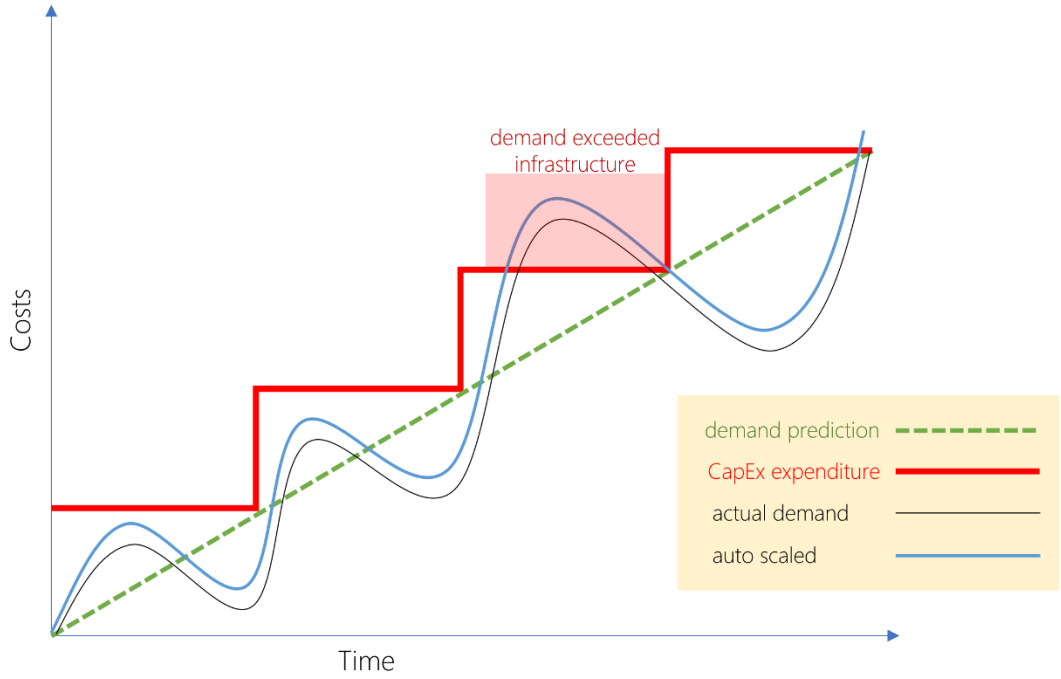
The subscription (pay-per-use) model is a computing billing method that is designed for both organizations and users. The organization or user is billed for the services used, typically on a recurring basis. You can scale, customize, and provision computing resources, including software, storage, and development platforms. For example, when using a dedicated cloud service, you could pay based on server hardware and usage.

Benefits of CapEx

With capital expenditures, you plan your expenses at the start of a project or budget period. Your costs are fixed, meaning you know exactly how much is being spent. This is appealing when you need to predict the expenses before a project starts due to a limited budget.

Benefits of OpEx

Demand and growth can be unpredictable and can outpace expectation, which is a challenge for the CapEx model as shown in the following graph.



Copy

Graph shows costs versus time, with time on the horizontal axis. Lines are plotted for demand prediction, actual demand costs, capital expenditure costs, and auto scaled costs. The demand prediction goes up linearly over time. Actual costs form an increasing sine wave style plotting. Capital expenditure costs go up in a stairstep shape as infrustructure is added to meet exceeded actual demand. Auto scaled costs nearly align to the sine wave style curve of the actual demand.

With the OpEx model, companies wanting to try a new product or service don't need to invest in equipment. Instead, they pay as much or as little for the infrastructure as required.

OpEx is particularly appealing if the demand fluctuates or is unknown. Cloud services are often said to be *agile*. Cloud agility is the ability to rapidly change an IT infrastructure to adapt to the evolving needs of the business. For example, if your service peaks one month, you can scale to demand and pay a larger bill for the month. If the following month the demand drops, you can reduce the used resources and be charged less. This agility lets you manage your costs dynamically, optimizing spending as requirements change.

Infrastructure as a service (IaaS)

Infrastructure as a Service is the most flexible category of cloud services. It aims to give you the most control over the provided hardware that runs your application (IT infrastructure servers and virtual machines (VMs), storage, and operating systems). Instead of buying hardware, with IaaS, you rent it. It's an instant computing infrastructure, provisioned and managed over the internet.

Note

When using IaaS, ensuring that a service is up and running is a shared responsibility: the cloud provider is responsible for ensuring the cloud infrastructure is functioning correctly; the cloud customer is responsible for ensuring the service they

are using is configured correctly, is up to date, and is available to their customers. This is referred to as the **shared responsibility model**.

IaaS is commonly used in the following scenarios:

- **Migrating workloads.** Typically, IaaS facilities are managed in a similar way as on-premises infrastructure and provide an easy migration path for moving existing applications to the cloud.
- **Test and development.** Teams can quickly set up and dismantle test and development environments, bringing new applications to market faster. IaaS makes scaling development and testing environments, fast and economical.
- **Storage, backup, and recovery.** Organizations avoid the capital outlay and complexity of storage management, which typically requires skilled staff to manage data and meet legal and compliance requirements. IaaS is useful for managing unpredictable demand and steadily growing storage needs. IaaS can also simplify the planning and management of backup and recovery systems.

Platform as a service (PaaS)

PaaS provides an environment for building, testing, and deploying software applications. The goal of PaaS is to help you create an application quickly without managing the underlying infrastructure. For example, when deploying a web application using PaaS, you don't have to install an operating system, web server, or even system updates.

PaaS is a complete development and deployment environment in the cloud, with resources that enable organizations to deliver everything from simple cloud-based apps to sophisticated cloud-enabled enterprise applications. Resources are purchased from a cloud service provider on a pay-as-you-go basis and accessed over a secure Internet connection.

PaaS is commonly used in the following scenarios:

- **Development framework.** PaaS provides a framework that developers can build upon to develop or customize cloud-based applications. Just like Microsoft Excel macro, PaaS lets developers create applications using built-in software components. Cloud features such as scalability, high-availability, and multi-tenant capability are included, reducing the amount of coding that developers must do.
- **Analytics or business intelligence.** Tools provided as a service with PaaS allow organizations to analyze and mine their data. They can find insights and patterns, and predict outcomes to improve business decisions such as forecasting, product design, and investment returns.

Software as a service (SaaS)

SaaS is software that is centrally hosted and managed for the end customer. It is usually based on an architecture where one version of the application is used for all customers, and licensed through a monthly or annual subscription. Microsoft 365, Skype, and Dynamics CRM Online are perfect examples of SaaS software.

Cost and Ownership

TABLE 1

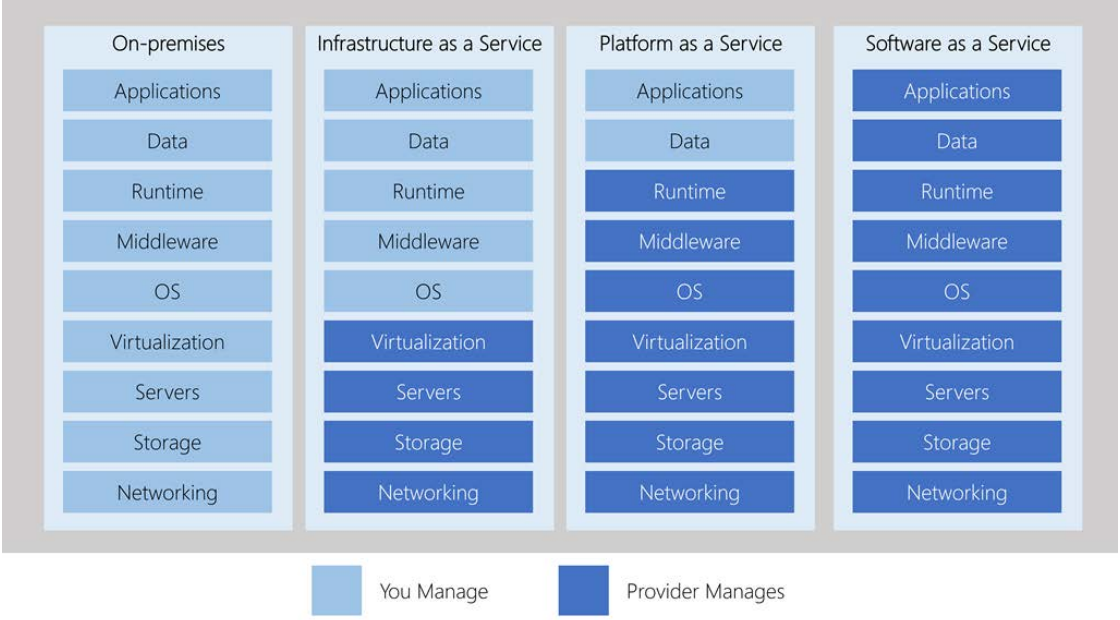
	IaaS	PaaS	SaaS
Upfront costs	There are no upfront costs. Users pay only for what they consume.	There are no upfront costs. Users pay only for what they consume.	Users have no upfront costs; they pay a subscription, typically on a monthly or annual basis.
User ownership	The user is responsible for the purchase, installation, configuration, and management of their own software, operating systems, middleware, and applications.	The user is responsible for the development of their own applications. However, they are not responsible for managing the server or infrastructure. This allows the user to focus on the application or workload they want to run.	Users just use the application software; they are not responsible for any maintenance or management of that software.

TABLE 1

	IaaS	PaaS	SaaS
Cloud provider ownership	The cloud provider is responsible for ensuring that the underlying cloud infrastructure (such as virtual machines, storage, and networking) is available for the user.	The cloud provider is responsible for operating system management, network, and service configuration. Cloud providers are typically responsible for everything apart from the application that a user wants to run. They provide a complete managed platform on which to run the application.	The cloud provider is responsible for the provision, management, and maintenance of the application software.

Management responsibilities

One thing to understand is that these categories are layers on top of each other. For example, PaaS adds a layer on top of IaaS by providing a level of abstraction. The abstraction has the benefit of hiding the details that you may not care about, so that you can get to coding quicker. However, one aspect of the abstraction is that you have less control over the underlying hardware. The following illustration shows a list of resources that you manage and that your service provider manages in each cloud service category.



Copy

First column, on-premises, shows all elements managed by you. Second, infrastructure as a service, moves virtualization, servers, storage, and networking to the cloud provider. Third, platform as a service, moves runtime, middleware, and OS to the cloud provider. And fourth, software as a service, moves all elements to the cloud provider, with applications and data being the last elements moving.

- IaaS requires the most user management of all the cloud services. The user is responsible for managing the operating systems, data, and applications.
- PaaS requires less user management. The cloud provider manages the operating systems, and the user is responsible for the applications and data they run and store.
- SaaS requires the least amount of management. The cloud provider is responsible for managing everything, and the end user just uses the software.

Combine cloud services to fit your needs : IaaS, PaaS, and SaaS each contain different levels of managed services. You may easily use a combination of these types of infrastructure. You could use Microsoft 365 on your company's computers (SaaS), and in Azure, you could host your VMs (IaaS) and use Azure SQL Database (PaaS) to store your data. With the cloud's flexibility, you can use any combination that provides you with the maximum result.

Compliance terms and requirements

When selecting a cloud provider to host your solutions, you should understand how that provider can help you comply with regulations and standards. Some questions to ask about a potential provider include:

- How compliant is the cloud provider when it comes to handling sensitive data?
- How compliant are the services offered by the cloud provider?
- How can I deploy my own cloud-based solutions to scenarios that have accreditation or compliance requirements?
- What terms are part of the privacy statement for the provider?

Compliance Offerings

The following list provides details about *some* of the compliance offerings available.

- **Criminal Justice Information Services (CJIS)**. Any US state or local agency that wants to access the FBI's CJIS database is required to adhere to the CJIS Security Policy. Azure is the only major cloud provider that contractually commits to conformance with the CJIS Security Policy, which commits Microsoft to adhering to the same requirements that law enforcement and public safety entities must meet.
- **Cloud Security Alliance (CSA) STAR Certification**. Azure, Intune, and Microsoft Power BI have obtained STAR Certification, which involves a rigorous independent third-party assessment of a cloud provider's security posture. This STAR certification is based on achieving ISO/IEC 27001 certification and meeting criteria specified in the Cloud Controls Matrix (CCM). This certification demonstrates that a cloud service provider:
  - Conforms to the applicable requirements of ISO/IEC 27001.
  - Has addressed issues critical to cloud security as outlined in the CCM.
  - Has been assessed against the STAR Capability Maturity Model for the management of activities in CCM control areas.
- **General Data Protection Regulation (GDPR)**. As of May 25, 2018, a European privacy law — GDPR — is in effect. GDPR imposes new rules on companies, government agencies, non-profits, and other organizations that offer goods and services to people in the European Union (EU), or that collect and analyze data tied to EU residents. The GDPR applies no matter where you are located.
- **EU Model Clauses**. Microsoft offers customers EU Standard Contractual Clauses that provide contractual guarantees around transfers of personal data outside of the EU. Microsoft is the first company to receive joint approval from the EU's Article 29 Working Party that the contractual privacy protections Azure delivers to its enterprise cloud customers meet current EU standards for international transfers of data. This ensures that Azure customers can use Microsoft services to move data freely through Microsoft's cloud from Europe to the rest of the world.
- **Health Insurance Portability and Accountability Act (HIPAA)**. HIPAA is a US federal law that regulates patient Protected Health Information (PHI). Azure offers customers a HIPAA Business Associate Agreement (BAA), stipulating adherence to certain security and privacy provisions in HIPAA and the **Health Information Technology for Economic and Clinical Health (HITECH)** Act. To assist customers in their individual compliance efforts, Microsoft offers a BAA to Azure customers as a contract addendum.
- **International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) 27018**. Microsoft is the first cloud provider to have adopted the ISO/IEC 27018 code of practice, covering the processing of personal information by cloud service providers.
- **Multi-Tier Cloud Security (MTCS) Singapore**. After rigorous assessments conducted by the MTCS Certification Body, Microsoft cloud services received MTCS 584:2013 certification across all three service classifications:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

  Microsoft was the first global cloud solution provider (CSP) to receive this certification across all three classifications.

- **Service Organization Controls (SOC) 1, 2, and 3**. Microsoft-covered cloud services are audited at least annually against the SOC report framework by independent third-party auditors. The Microsoft cloud services audit covers controls for data security, availability, processing integrity, and confidentiality as applicable to in-scope trust principles for each service.
- **National Institute of Standards and Technology (NIST) Cybersecurity Framework (CSF)**. NIST CSF is a voluntary Framework that consists of standards, guidelines, and best practices to manage cybersecurity-related risks. Microsoft cloud services have undergone independent, third-party Federal Risk and Authorization Management Program (FedRAMP) Moderate and High Baseline audits, and are certified according to the FedRAMP standards. Additionally, through a validated assessment performed by the Health Information Trust Alliance (HITRUST), a leading security and privacy standards development and accreditation organization, Microsoft 365 is certified to the objectives specified in the NIST CSF.
- **UK Government G-Cloud**. The UK Government G-Cloud is a cloud computing certification for services used by government entities in the United Kingdom. Azure has received official accreditation from the UK Government Pan Government Accreditor.

Understand Availability Zones in Azure

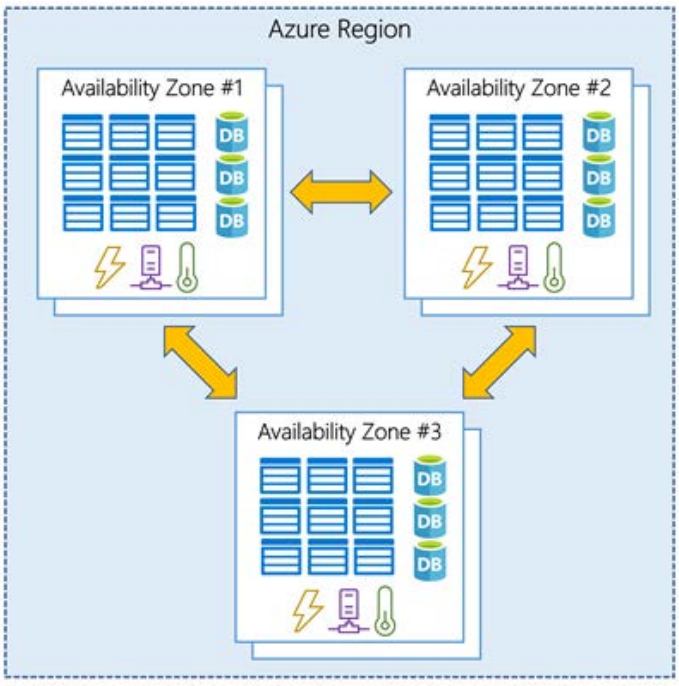
- 5 minutes

You want to ensure your services and data are redundant so you can protect your information in case of failure. When you are hosting your infrastructure, this requires creating duplicate hardware environments. Azure can help make your app highly available through *Availability Zones*.

What is an Availability Zone?

Availability Zones are physically separate datacenters within an Azure region.

Each Availability Zone is made up of one or more datacenters equipped with independent power, cooling, and networking. It is set up to be an *isolation boundary*. If one zone goes down, the other continues working. Availability Zones are connected through high-speed, private fiber-optic networks.



Supported regions

Not every region has support for Availability Zones. The following regions have a minimum of three separate zones to ensure resiliency.

- Central US
- East US 2
- West US 2
- West Europe
- France Central
- North Europe
- Southeast Asia

Tip

The list of supported regions is expanding - check the documentation for the latest information.

Using Availability Zones in your apps

You can use Availability Zones to run mission-critical applications and build high-availability into your application architecture by co-locating your compute, storage, networking, and data resources within a zone and replicating in other zones. Keep in mind that there could be a cost to duplicating your services and transferring data between zones.

Availability Zones are primarily for VMs, managed disks, load balancers, and SQL databases. Azure services that support Availability Zones fall into two categories:

- **Zonal services** – you pin the resource to a specific zone (for example, virtual machines, managed disks, IP addresses)
- **Zone-redundant services** – platform replicates automatically across zones (for example, zone-redundant storage, SQL Database).

Check the documentation to determine which elements of your architecture you can associate with an Availability Zone.

Understand Region Pairs in Azure

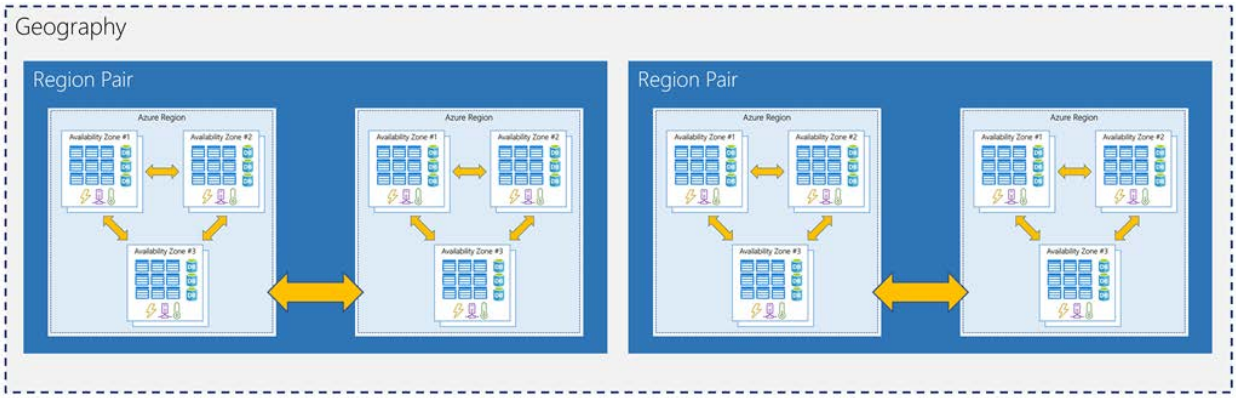
- 5 minutes

Availability zones are created using one or more datacenters, and there is a minimum of three zones within a single region. However, it's possible that a large enough disaster could cause an outage large enough to affect even two datacenters. That's why Azure also creates *region pairs*.

What is a region pair?

Each Azure region is always paired with another region within the same geography (such as US, Europe, or Asia) at least **300 miles away**. This approach allows for the replication of resources (such as virtual machine storage) across a geography that helps reduce the likelihood of interruptions due to events such as natural disasters, civil unrest, power outages, or physical network outages affecting both regions at once. If a region in a pair was affected by a natural disaster, for instance, services would automatically fail over to the other region in its region pair.

Examples of region pairs in Azure are West US paired with East US, and SouthEast Asia paired with East Asia.



Since the pair of regions is directly connected and far enough apart to be isolated from regional disasters, you can use them to provide reliable services and data redundancy. Some services offer automatic geo-redundant storage using region pairs.

Additional advantages of region pairs include:

- If there's an extensive Azure outage, one region out of every pair is prioritized to make sure at least one is restored as quick as possible for applications hosted in that region pair.
- Planned Azure updates are rolled out to paired regions one region at a time to minimize downtime and risk of application outage.
- Data continues to reside within the same geography as its pair (except for Brazil South) for tax and law enforcement jurisdiction purposes.

Having a broadly distributed set of datacenters allows Azure to provide a high guarantee of availability. Let's explore what that means.

What are resource groups?

Resource groups are a fundamental element of the Azure platform. A resource group is a logical container for resources deployed on Azure. These resources are anything you create in an Azure subscription like virtual machines, Application Gateways, and CosmosDB instances. All resources must be in a resource group and a resource can only be a member of a single resource group. Many resources can be moved between resource groups with some services having specific limitations or requirements to move. Resource groups can't be nested. Before any resource can be provisioned, you need a resource group for it to be placed in.

Logical grouping

Resource groups exist to help manage and organize your Azure resources. By placing resources of similar usage, type, or location, you can provide some order and organization to resources you create in Azure. Logical grouping is the aspect that you're most interested in here, since there's a lot of disorder among our resources.

Life cycle

If you delete a resource group, all resources contained within are also deleted. Organizing resources by life cycle can be useful in non-production environments, where you might try an experiment, but then dispose of it when done. Resource groups make it easy to remove a set of resources at once.

Authorization

Resource groups are also a scope for applying role-based access control (RBAC) permissions. By applying RBAC permissions to a resource group, you can ease administration and limit access to allow only what is needed.

Create a Resource Group

Resource groups can be created by using the following methods:

- Azure portal
- Azure PowerShell
- Azure CLI
- Templates
- Azure SDKs (like .NET, Java)

Let's walk through the steps you'd take to create a resource group in the Azure portal. If you'd like to follow along in your own subscription, you may.

1. Open a web browser and sign into the Azure portal .

   Important

   Make sure to use your *own* subscription. When you are in the free sandbox environment, it will not allow you to create resource groups. You can tell which subscription you are using by looking at the tenant name under your profile picture. You can switch tenants by selecting your profile picture and selecting **Switch Directory** from the options menu.

2. On the Azure portal menu or from the **Home** page, select **Create a resource**.
3. Type **Resource group** in the search box and hit Enter. If this doesn't go immediately to the resource group creation, select **Resource group** from the search results and select the **Create** button.
4. Select the subscription it should be in, and select the region for the resource group.
5. Enter your resource group name, let's use **msftlearn-core-infrastructure-rg**.
6. Select **Review + Create** and then, once it is validated, select **Create** to create the resource group.

That's it, you've created a resource group that you can now use when you deploy Azure resources. Let's take a closer look at this resource group and some important things to consider.

Explore a resource group and add a resource

On the Azure portal menu or from the **Home** page, select **Resource groups**, and select your newly created resource group. Note that you may also see a resource group called **NetworkWatcherRG**. You can ignore this resource group, it's created automatically to enable Network Watcher in Azure virtual networks.

On the Overview panel, there's the basic information about the resource group like the subscription it's in, the subscription ID, any tags that are applied, and a history of the deployments to this resource group. You'll cover tags in the next unit. The deployments link takes you to a new panel with the history of all deployments to this resource group. Anytime you create a resource, it's a deployment, and you see that history for the resource group here.

Across the top you can add more resources, change the columns in the list, move the resource group to another subscription, or delete it entirely.

On the left menu, there are a number of options

You don't have any resources in this resource group yet, so the list at the bottom is empty. Let's create a couple resources inside the resource group.

1. Select **+ Add** at the top or select the **Create resources**; either will work.
2. Search for **Virtual Network**. The first result should be the virtual network resource. Select it, and on the next screen select **Create**.
3. Name the virtual network `msftlearn-vnet1`. For the **Resource group** drop-down, select the resource group that you created earlier.
4. Select **Review + create** and then select **Create** to add the virtual network to your resource group.
5. Repeat the virtual network creation steps again to create one more virtual network. Name the network `msftlearn-vnet2` and make sure to place the virtual network in the resource group that you created earlier.
6. Go back to your resource group, and on the **Overview** panel you should see the two virtual networks you created.

Our resource group now contains two virtual network resources because you specified in our deployment (when you created the resources) which resource group you wanted the virtual network to be placed in. You could create additional resources inside this resource group, or you could create additional resource groups in the subscription to deploy resources into.

When creating resources, you usually have the option to create a new resource group as an alternative to using an existing resource group. This simplifies the process a bit, but as you see in your new organization, can lead to resources spread across resource groups with little thought as to how to organize them.

Use resource groups for organization

So how can you use resource groups to your advantage in your new organization? There are some guidelines and best practices that can help with the organization.

Consistent naming convention

You can start with using an understandable naming convention. You named our resource group **msftlearn-core-infrastructure-rg**. You've given some indication of what it's used for (**msftlearn**), the types of resources contained within (**core-infrastructure**), and the type of resource it is itself (**rg**). This descriptive name gives us a better idea of what it is. If you had named it **my-resource-group** or **rg1**, you have no idea on a glance of what the usage may be. In this case, you can deduce that there are probably core pieces of infrastructure contained within. If you created additional virtual networks, storage accounts, or other resources the company may consider *core infrastructure*, you could place them here as well, to improve the organization of our resources. Naming conventions can vary widely between and even within companies, but some planning can help.

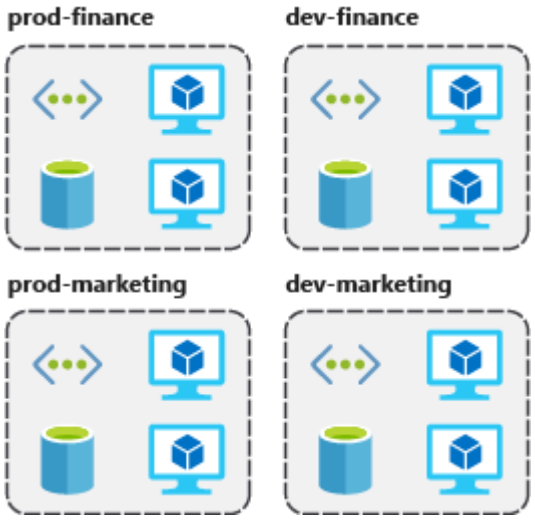
Organizing principles

Resource groups can be organized in a number of ways, let's take a look at a few examples. You might put all resources that are *core infrastructure* into this resource group. But you could also organize them strictly by resource type. For example, put all virtual networks in one resource group, all virtual machines in another resource group, and all Azure Cosmos DB instances in yet another resource group.

You could organize them by environment (prod, qa, dev). In this case, all production resources are in one resource group, all test resources are in another resource group, and so on.

You could organize them by department (marketing, finance, human resources). Marketing resources go in one resource group, finance in another resource group, and HR in a third resource group.

You could even use a combination of these strategies and organize by environment and department. Put production finance resources in one resource group, dev finance resources in another, and the same for the marketing resources.



There are a few factors that can play into the strategy you use to organize resources: authorization, resource life cycle, and billing.

Organizing for authorization

Since resource groups are a scope of RBAC, you can organize resources by *who* needs to administer them. If your database administration team is responsible for managing all of your Azure SQL Database instances, putting them in the same resource group would simplify administration. You could give them the proper permissions at the resource group level to administer the databases within the resource group. Similarly, the database administration team could be denied access to the resource group with virtual networks, so they don't inadvertently make changes to resources outside the scope of their responsibility.

Organizing for life cycle

We mentioned earlier that resource groups serve as the life cycle for the resources within it. If you delete a resource group, you delete all the resources in it. Use this to your advantage, especially in areas where resources are more disposable, like non-production environments. If you deploy 10 servers for a project that you know will only last a couple of months, you might put them all in a single resource group. One resource group is easier to clean up than 10 or more resource groups.

Organizing for billing

Lastly, placing resources in the same resource group is a way to group them for usage in billing reports. If you're trying to understand how your costs are distributed in your Azure environment, grouping them by resource group is one way to filter and sort the data to better understand where costs are allocated.

Summary

The bottom line is that you have flexibility in how to organize resources in your resource groups. Put some thought into it so that you have a coherent approach to how you use resource groups in your Azure environment.

Azure Resource

- Object **used to manage services** in Azure
- Represents **service lifecycle**
- Saved as **JSON definition**

Resource Groups

- **Grouping** of resources
- Holds **logically related** resources
- Typically organizing by
    - **Type**
    - **Lifecycle** (app, environment)
    - **Department**
    - **Billing**,
    - **Location** or
    - **combination of those**

Additional Info

- Each **resource must** be in one, and **only one resource group**
- Resource **groups have their own location** assigned
- Resources in the resource groups **can reside in a different locations**
- Resources **can be moved** between the resource groups
- Resource **groups can’t be nested**
- Organize based on your organization needs but consider

- Billing
- Security and access management
- Application Lifecycle

Use tagging to organize resources

- 8 minutes

You've gone through your resources and moved them into resource groups that are more organized than before. But what if resources have multiple uses? How do you better search, filter, and organize these resources? Tags can be helpful as you look to improve organization of your Azure resources.

What are tags?

Tags are name/value pairs of text data that you can apply to resources and resource groups. Tags allow you to associate custom details about your resource, in addition to the standard Azure properties a resource has the following properties:

- department (like finance, marketing, and more)
- environment (prod, test, dev)
- cost center
- life cycle and automation (like shutdown and startup of virtual machines)

A resource can have up to 50 tags. The name is limited to 512 characters for all types of resources except storage accounts, which have a limit of 128 characters. The tag value is limited to 256 characters for all types of resources. Tags aren't inherited from parent resources. Not all resource types support tags, and tags can't be applied to classic resources.

Tags can be added and manipulated through the Azure portal, Azure CLI, Azure PowerShell, Resource Manager templates, and through the REST API. For example, to add a resource tag to a virtual network using the Azure CLI, you could use the following command:

Azure CLICopy

```
az resource tag --tags Department=Finance \
    --resource-group msftlearn-core-infrastructure-rg \
    --name msftlearn-vnet1 \
    --resource-type "Microsoft.Network/virtualNetworks"
```

You can use Azure Policy to automatically add or enforce tags for resources your organization creates based on policy conditions that you define. For example, you could require that a value for the Department tag is entered when someone in your organization creates a virtual network in a specific resource group.

Apply tags to resources

Let's apply some tags to the resources you created. Recall that you created a resource group **msftlearn-core-infrastructure-rg** and two virtual networks inside that resource group, **msftlearn-vnet1** and **msftlearn-vnet2**. The names of the virtual networks are relatively generic, so you'd like to associate the virtual networks with services from different departments.

1. Open the Azure portal , and navigate to your **msftlearn-core-infrastructure-rg** resource group.
2. On the **Overview** tab of your resource group, you should see your two virtual networks listed. The default view doesn't display the tags column, so you'll add that to the display. Select **Edit columns** at the top. In the **Available columns** list, select **Tags** and click **->** to add it to the **Selected columns** list. Click **Apply** to apply your changes.

You should now see the tags column, but it will be empty since you haven't added any tags yet. You'll add the tags directly here.

3. You can also add tags to any resource that supports it on the resource's **Tags** panel. In the list of resources, you should see an ellipsis menu (**...**). Select the **...** for the **msftlearn-vnet1** resource, then select **Edit tags** to display the **Edit tags** dialog.
4. You'll add a couple tags to this virtual network. In the **Name** box type `Department`, and in the **Value** box type `Finance`. Click **Save** to save your changes.
5. Do the same steps for the **msftlearn-vnet2** virtual network. For this virtual network, add a `Department` tag to the resource with value `Marketing`.

   You should now see your tags applied to each resource.

6. Add tags to both of these resources in bulk. Select the checkbox on the left for each of the virtual networks and click **Assign tags** in the top menu. (The option may be contained inside an **...** menu.) By selecting multiple resources, you can add a tag to them in bulk, making it easy if you have multiple resources you want to apply the same tag to.

   Add the `Environment`:`Training` tag to the resources. You should see in the dialog that the tag will be applied to each of the virtual networks.

   Back in the resource list you'll now see the tags column with multiple values. If your window width is limited, you may see an ellipsis indicating more tags are applied to each resource that are not shown.

7. Take a look at how you can use tags to filter your resources. On the Azure portal menu or from the **Home** page, select **All resources**.
8. Select **Add filter**. In the **Tags**, select **Environment**, then select **Training**. You should see only your two virtual networks displayed, since you tagged those resources with the **Environment** tag set to **Training**.
9. You can further filter these resources by additionally filtering on a **Department** tag with a value of **Finance** or **Marketing**.

Use tags for organization

The above example is just one example of where you can use tags to organize your resources. With their flexibility, there are several ways you can use tags to your advantage.

You can use tags to group your billing data. For example, if you're running multiple VMs for different organizations, use the tags to group usage by cost center. You can also use tags to categorize costs by runtime environment, such as the billing usage for VMs running in the production environment. When exporting billing data or accessing it through billing APIs, tags are included in that data and can be used to further slice your data from a cost perspective.

You can retrieve all the resources in your subscription with a specific tag name or value. Tags enable you to retrieve related resources from different resource groups. This approach is helpful when you need to organize resources for billing or management.

Tagging resources can also help in monitoring to track down impacted resources. Monitoring systems could include tag data with alerts, giving you the ability to know exactly who is impacted. In our example above, you applied the **Department** tag with a value of **Finance** to the **msftlearn-vnet1** resource. If an alarm was thrown on **msftlearn-vnet1** and the alarm included the tag, you'd know that the finance department may be impacted by the condition that triggered the alarm. This contextual information can be valuable if an issue occurs.

It's also common for tags to be used in automation. If you want to automate the shutdown and startup of virtual machines in development environments during off-hours to save costs, you can use tags to assist in this automation. Add a **shutdown:6PM** and **startup:7AM** tag to the virtual machines, then create an automation job

that looks for these tags, and shuts them down or starts them up based on the tag value. There are several solutions in the Azure Automation Runbooks Gallery that use tags in a similar manner to accomplish this result.

Essential Azure compute concepts

Your research team has collected massive amounts of image data that might lead to a discovery on Mars. They need to perform computationally intense data processing but don't have the equipment to do the work. Let's see why Azure is a good choice to do the data analysis.

What is Azure compute?

Azure compute is an on-demand computing service for running cloud-based applications. It provides computing resources like multi-core processors and supercomputers via virtual machines and containers. It also provides serverless computing to run apps without requiring infrastructure setup or configuration. The resources are available on-demand and can typically be created in minutes or even seconds. You pay only for the resources you use and only for as long as you're using them.

There are four common techniques for performing compute in Azure:

- Virtual machines
- Containers
- Azure App Service
- Serverless computing

What are virtual machines?

Virtual machines, or VMs, are software emulations of physical computers. They include a virtual processor, memory, storage, and networking resources. They host an operating system (OS), and you're able to install and run software just like a physical computer. And by using a remote desktop client, you can use and control the virtual machine as if you were sitting in front of it.

What are containers?

Containers are a virtualization environment for running applications. Just like virtual machines, containers run on top of a host operating system. But unlike VMs, containers don't include an operating system for the apps running *inside* the container. Instead, containers bundle the libraries and components needed to run the application and use the existing host OS running the container. For example, if five containers are running on a server with a specific Linux kernel, all five containers and the apps within them share that same Linux kernel.

What is Azure App Service?

Azure App Service is a platform-as-a-service (PaaS) offering in Azure that is designed to host enterprise-grade web-oriented applications. You can meet rigorous performance, scalability, security, and compliance requirements while using a fully managed platform to perform infrastructure maintenance.

What is Serverless Computing?

Serverless computing is a cloud-hosted execution environment that runs your code but completely abstracts the underlying hosting environment. You create an instance of the service, and you add your code; no infrastructure configuration or maintenance is required, or even allowed.

Which computing strategy is right for me?

You don't need to take an "all or nothing" approach when choosing a cloud computing strategy. Virtual machines, containers, App Service, and serverless computing each provide benefits as well as tradeoffs against other options.

For example, although serverless computing removes the need for you to manage infrastructure, serverless computing expects work to be completed quickly; usually within seconds or less. Therefore, you might run your core application on a virtual machine or container but offload some of the data processing onto a serverless app.

Let's look at each option more closely to help you decide when to use each service.

Explore Azure Virtual Machines

Azure Virtual Machines (VMs) let you create and use virtual machines in the cloud. They provide infrastructure as a service (IaaS) in the form of a virtualized server and can be used in many ways. Just like a physical computer, you can customize all of the software running on the VM. VMs are an ideal choice when you need:

- Total control over the operating system (OS)
- The ability to run custom software, or
- To use custom hosting configurations

An Azure VM gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs the VM. However, you still need to maintain the VM—that is, configure, update, and maintain the software that runs on the VM.

You can create and provision a VM in minutes when you select a pre-configured VM image. Selecting an image is one of the most important decisions you'll make when creating a VM. An image is a template used to create a VM. These templates already include an OS and often other software, like development tools or web hosting environments.

Examples of when to use virtual machines

- During testing and development. VMs provide a quick and easy way to create different OS and application configurations. Test and development personnel can then easily delete the VMs when they no longer need them.
- When running applications in the cloud. The ability to run certain applications in the public cloud as opposed to creating a traditional infrastructure to run them can provide substantial economic benefits. For example, if an application needs to handle fluctuations in demand, being able to shut down VMs when you don't need them or quickly start them up to meet a suddenly increased demand means you pay only for the resources you use.
- When extending your datacenter to the cloud. An organization can extend the capabilities of its own on-premises network by creating a virtual network in Azure and adding VMs to that virtual network. Applications like SharePoint can then run on an Azure VM instead of running locally, making it easier or less expensive to deploy than in an on-premises environment.
- During disaster recovery. As with running certain types of applications in the cloud and extending an on-premises network to the cloud, you can get significant costs savings by using an IaaS-based approach to disaster recovery. If a primary datacenter fails, you can create VMs running on Azure to run your critical applications and then shut them down when the primary datacenter becomes operational again.

Moving to the cloud with VMs

VMs are also an excellent choice when moving from a physical server to the cloud ("lift and shift"). You can create an image of the physical server and host it within a VM with little or no changes. Just like a physical on-premises server, you must maintain the VM. You update the installed OS and the software it runs.

Scaling VMs in Azure

You can run single VMs for testing, development, or minor tasks; or you can group VMs together to provide high availability, scalability, and redundancy. Azure has several features such that, no matter what your uptime requirements are, Azure can meet them. These features include:

- Availability sets
- Virtual Machine Scale Sets
- Azure Batch

What are availability sets?

An **availability set** is a logical grouping of two or more VMs that help keep your application available during planned or unplanned maintenance.

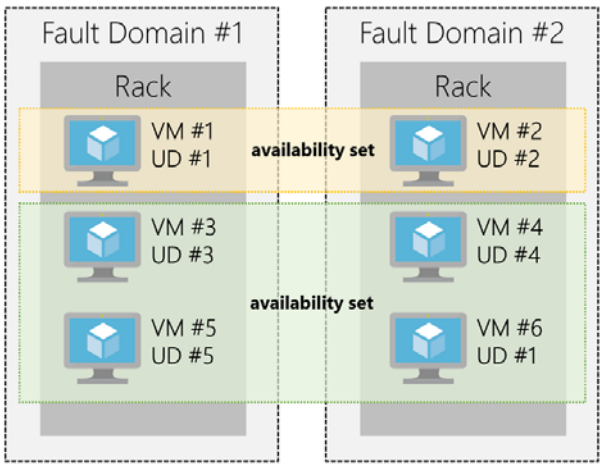
A *planned maintenance event* is when the underlying Azure fabric that hosts VMs is updated by Microsoft. A planned maintenance event is done to patch security vulnerabilities, improve performance, and add or update features. Most of the time these updates are done without any impact to the guest VMs. But sometimes VMs require a reboot to complete an update. When the VM is part of an availability set, the Azure fabric updates are sequenced so not all of the associated VMs are rebooted at the same time. VMs are put into different *update domains*. Update domains indicate groups of VMs and underlying physical hardware that can be rebooted at the same time. Update domains are a logical part of each data center and are implemented with software and logic.

Unplanned maintenance events involve a hardware failure in the data center, such as a server power outage or disk failure. VMs that are part of an availability set automatically switch to a working physical server so the VM continues to run. The group of virtual machines that share common hardware are in the same *fault domain*. A fault domain is essentially a rack of servers. It provides the physical separation of your workload across different power, cooling, and network hardware that support the physical servers in the data center server racks. In the event the hardware that supports a server rack becomes unavailable, only that rack of servers is affected by the outage.

With an availability set, you get:

- Up to three fault domains that each have a server rack with dedicated power and network resources
- Five logical update domains which then can be increased to a maximum of 20

Your VMs are then sequentially placed across the fault and update domains. The following diagram shows an example where you have six VMs in two availability sets distributed across the two fault domains and five update domains.



Two outlines surround fault domain 1 and fault domain 2. Fault domain 1 contains a rack with virtual machine 1 inside update domain 1, virtual machine 3 inside update domain 3, and virtual machine 5 inside update domain 5. Fault domain 2 contains a rack with virtual machine 2 inside update domain 2, virtual machine 4 inside update domain 4, and virtual machine 6 as part of update domain 1. Virtual machine 1 from fault domain 1 and virtual machine 2 from fault domain 2 are part of an availability set. Virtual machine 3 and 5 from fault domain 1 and virtual machine 4 and 6 from fault domain 2 are part of a separate availability set.

There's no cost for an availability set. You only pay for the VMs within the availability set. We highly recommend that you place each workload in an availability set to avoid having a single point of failure in your VM architecture.

What are virtual machine scale sets?

Azure Virtual Machine Scale Sets let you create and manage a group of identical, load balanced VMs. Imagine you're running a website that enables scientists to upload astronomy images that need to be processed. If you duplicated the VM, you'd normally need to configure an additional service to route requests between multiple instances of the website. Virtual Machine Scale Sets could do that work for you.

Scale sets allow you to centrally manage, configure, and update a large number of VMs in minutes to provide highly available applications. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. With Virtual Machine Scale Sets, you can build large-scale services for areas such as compute, big data, and container workloads.

What is Azure Batch?

Azure Batch enables large-scale job scheduling and compute management with the ability to scale to tens, hundreds, or thousands of VMs.

When you're ready to run a job, Batch does the following:

- Starts a pool of compute VMs for you
- Installs applications and staging data
- Runs jobs with as many tasks as you have
- Identifies failures
- Requeues work
- Scales down the pool as work completes

There may be situations in which you need raw computing power or supercomputer level compute power. Azure provides these capabilities.

Explore Containers in Azure

If you wish to run multiple instances of an application on a single host machine, containers are an excellent choice. The container orchestrator can start, stop, and scale out application instances as needed.

A container is a modified runtime environment built on top of a host OS that executes your application. A container doesn't use virtualization, so it doesn't waste resources simulating virtual hardware with a redundant OS. This environment typically makes containers more lightweight than VMs. This design allows you to respond quickly to changes in demand or failure. Another benefit of containers is you can run multiple isolated applications on a single container host. Since containers are secured and isolated, you don't need separate servers for each app.

VMs versus containers

Containers in Azure

Azure supports Docker containers (a standardized container model), and there are several ways to manage containers in Azure.

- Azure Container Instances (ACI)
- Azure Kubernetes Service (AKS)

Azure Container Instances

Azure Container Instances (ACI) offers the fastest and simplest way to run a container in Azure. You don't have to manage any virtual machines or configure any additional services. It is a PaaS offering that allows you to upload your containers and execute them directly with automatic elastic scale.

Azure Kubernetes Service

The task of automating, managing, and interacting with a large number of containers is known as orchestration. Azure Kubernetes Service (AKS) is a complete orchestration service for containers with distributed architectures with multiple containers.

What is Kubernetes?

Using containers in your solutions

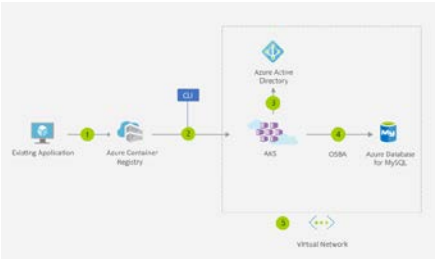
Containers are often used to create solutions using a *microservice architecture*. This architecture is where you break solutions into smaller, independent pieces. For example, you may split a website into a container hosting your front end, another hosting your back end, and a third for storage. This split allows you to separate portions of your app into logical sections that can be maintained, scaled, or updated independently.

What is a microservice?

Imagine your website backend has reached capacity but the front end and storage aren't being stressed. You could scale the back end separately to improve performance, or you could decide to use a different storage service. Or you could even replace the storage container without affecting the rest of the application.

Migrating apps to containers

You can move existing applications to containers and run them within AKS. You can control access via integration with Azure Active Directory (Azure AD) and access Service Level Agreement (SLA)–backed Azure services, such as Azure Database for MySQL for any data needs, via Open Service Broker for Azure (OSBA).



Step one is between an existing application and the Azure Container Registry. Step two is the CLI between Azure Container Registry and AKS. Step three is between AKS and Azure Active Directory. Step four is between AKS and Azure Database for MySQL, labeled OSBA. Step five is the dotted containing virtual network box around AKS, Azure Active Directory, and Azure Database for MySQL.

The preceding figure depicts this process as follows:

1. You convert an existing application to one or more containers and then publish one or more container images to the Azure Container Registry.
2. By using the Azure portal or the command line, you deploy the containers to an AKS cluster.
3. Azure AD controls access to AKS resources.
4. You access SLA-backed Azure services, such as Azure Database for MySQL, via OSBA.
5. Optionally, AKS is deployed with a virtual network.

Explore Azure App Service

Azure App Service enables you to build and host web apps, background jobs, mobile backends, and RESTful APIs in the programming language of your choice without managing infrastructure. It offers automatic scaling and high availability. App Service supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo to support a continuous deployment model.

This platform as a service (PaaS) allows you to focus on the website and API logic while Azure handles the infrastructure to run and scale your web applications.

App Service costs

You pay for the Azure compute resources your app uses while it processes requests based on the App Service Plan you choose. The App Service plan determines how much hardware is devoted to your host - for example, whether it's dedicated or shared hardware, and how much memory is reserved for it. There is even a *free* tier you can use to host small, low-traffic sites.

Types of app services

With Azure App Service, you can host most common app service styles, including:

- Web Apps
- API Apps
- WebJobs
- Mobile Apps

Azure App Service handles most of the infrastructure decisions you deal with in hosting web-accessible apps: deployment and management are integrated into the platform, endpoints can be secured, sites can be scaled quickly to handle high traffic loads, and the built-in load balancing and traffic manager provide high availability. All of these app styles are hosted in the same infrastructure and share these benefits. This flexibility makes App Service the ideal choice to host web-oriented applications.

Web apps

App Service includes full support for hosting web apps using ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can choose either Windows or Linux as the host operating system.

API apps

Much like hosting a website, you can build REST-based Web APIs using your choice of language and framework. You get full Swagger support, and the ability to package and publish your API in the Azure Marketplace. The produced apps can be consumed from any HTTP(S)-based client.

Web jobs

WebJobs allows you to run a program (.exe, Java, PHP, Python, or Node.js) or script (.cmd, .bat, PowerShell, or Bash) in the same context as a web app, API app, or mobile app. They can be scheduled, or run by a trigger. WebJobs are often used to run background tasks as part of your application logic.

Mobile app back-ends

Use the Mobile Apps feature of Azure App Service to quickly build a back-end for iOS and Android apps. With just a few clicks in the Azure portal you can:

- Store mobile app data in a cloud-based SQL database
- Authenticate customers against common social providers such as MSA, Google, Twitter, and Facebook
- Send push notifications
- Execute custom back-end logic in C# or Node.js

On the mobile app side, there is SDK support for native iOS & Android, Xamarin, and React native apps.

Explore Serverless computing in Azure

Serverless computing is the abstraction of servers, infrastructure, and OSs. With *serverless* computing, Azure takes care of managing the server infrastructure and allocation/deallocation of resources based on demand. Infrastructure isn't your responsibility. Scaling and performance are handled automatically, and you are billed only for the exact resources you use. There's no need to even reserve capacity.

Serverless computing encompasses three ideas: the abstraction of servers, an event-driven scale, and micro-billing:

1. **Abstraction of servers**: Serverless computing abstracts the servers you run on. You never explicitly reserve server instances; the platform manages that for you. Each function execution can run on a different compute instance, and this execution context is transparent to the code. With serverless architecture, you simply deploy your code, which then runs with high availability.
2. **Event-driven scale**: Serverless computing is an excellent fit for workloads that respond to incoming events. Events include triggers by timers (for example, if a function needs to run every day at 10:00 AM UTC), HTTP (API and webhook scenarios), queues (for example, with order processing), and much more. Instead of writing an entire application, the developer authors a function, which contains both code and metadata about its triggers and bindings. The platform automatically schedules the function to run and scales the number of compute instances based on the rate of incoming events. Triggers define how a function is invoked and bindings provide a declarative way to connect to services from within the code.
3. **Micro-billing**: Traditional computing has the notion of per-second billing, but often, that's not as useful as it seems. Even if a customer's website gets only one hit a day, they still pay for a full day's worth of availability. With serverless computing, they pay only for the time their code runs. If no active function executions occur, they're not charged. For example, if the code runs once a day for two minutes, they're charged for one execution and two minutes of computing time.

Serverless computing in Azure

Azure has two implementations of serverless compute:

- **Azure Functions**, which can execute code in almost any modern language.
- **Azure Logic Apps**, which are designed in a web-based designer and can execute logic triggered by Azure services without writing any code.

Azure Functions

When you're concerned only about the code running your service, and not the underlying platform or infrastructure, Azure Functions are ideal. They're commonly used when you need to perform work in response to an event, often via a REST request, timer, or message from another Azure service and when that work can be completed quickly, within seconds or less.

Azure Functions scale automatically based on demand, so they're a solid choice when demand is variable. For example, you may be receiving messages from an IoT solution used to monitor a fleet of delivery vehicles. You'll likely have more data arriving during business hours.

Using a VM-based approach, you'd incur costs even when the VM is idle. With functions, Azure runs your code when it's triggered and automatically deallocates resources when the function is finished. In this model, you're only charged for the CPU time used while your function runs.

Furthermore, Azure Functions can be either stateless (the default), where they behave as if they're restarted every time they respond to an event, or stateful (called "Durable Functions"), where a context is passed through the function to track prior activity.

Functions are a key component of serverless computing, but they're also a general compute platform for running any type of code. If the needs of the developer's app change, you can deploy the project in an environment that isn't serverless, which provides the flexibility to manage scaling, run on virtual networks, and even completely isolate the functions.

Azure Logic Apps

Azure Logic Apps are similar to Functions - both enable you to trigger logic based on an event. Where Functions execute code, Logic Apps execute *workflows* designed to automate business scenarios and built from predefined logic blocks. Every logic app workflow starts with a trigger, which fires when a specific event happens or when newly available data meets specific criteria. Many triggers include basic scheduling capabilities, so developers can specify how regularly their workloads will run. Each time the trigger fires, the Logic Apps engine creates a logic app instance that runs the actions in the workflow. These actions can also include data conversions and flow controls, such as conditional statements, switch statements, loops, and branching.

You create Logic App workflows using a visual designer on the Azure portal or in Visual Studio. The workflows are persisted as a JSON file with a known workflow schema.

Azure provides over 200 different connectors and processing blocks to interact with different services - including most popular enterprise apps. You can also build custom connectors and workflow steps if the service you need to interact with isn't covered. You then use the visual designer to link connectors and blocks together, passing data through the workflow to do custom processing - often all without writing any code.

As an example, let's say a ticket arrives in ZenDesk. You could:

1. Detect the intent of the message with cognitive services
2. Create an item in SharePoint to track the issue
3. If the customer isn't in your database, add them to your Dynamics 365 CRM system
4. Send a follow-up email to acknowledge their request

All of that could be designed in a visual designer making it easy to see the logic flow, which is ideal for a business analyst role.

Functions vs. Logic Apps

Functions and Logic Apps can both create complex orchestrations. An orchestration is a collection of functions or steps, that are executed to accomplish a complex task. With Azure Functions, you write code to complete each step, with Logic Apps, you use a GUI to define the actions and how they relate to one another.

You can mix and match services when you build an orchestration, calling functions from logic apps and calling logic apps from functions. Here are some common differences between the two.

FUNCTIONS VS. LOGIC APPS

-	**Functions**	**Logic Apps**
State	Normally stateless, but Durable Functions provide state	Stateful
Development	Code-first (imperative)	Designer-first (declarative)
Connectivity	About a dozen built-in binding types, write code for custom bindings	Large collection of connectors, Enterprise Integration Pack for B2B scenarios, build custom connectors
Actions	Each activity is an Azure function; write code for activity functions	Large collection of ready-made actions
Monitoring	Azure Application Insights	Azure portal, Log Analytics
Management	REST API, Visual Studio	Azure portal, REST API, PowerShell, Visual Studio
Execution context	Can run locally or in the cloud	Runs only in the cloud.

Virtualization

- Emulation of physical machines
- Different virtual hardware configuration per machine/app
- Different operating systems per machine/app
- Total separation of environments
    - file systems,
    - services,
    - ports,
    - middleware,
    - configuration

Virtual Machines

- Infrastructure as a Service (IaaS)
- Total control over the operating system and the software
- Supports marketplace and custom images
- Best suited for
    - Custom software requiring custom system configuration
    - Lift-and-shift scenarios

- Can run any application/scenario
    - web apps & web services,
    - databases,
    - desktop applications,
    - jumpboxes,
    - gateways, etc.

Virtual Machine Scale Sets

- Infrastructure as a Service (IaaS)
- Set of identical virtual machines
- Built-in auto scaling features
- Designed for manual and auto-scaled workloads like web services,* batch processing, etc.

Containers

- Use host's operating system
- Emulate operating system (VMs emulate hardware)
- Lightweight (no O/S)
    - Development Effort
    - Maintenance
    - Compute & storage requirements
- Respond quicker to demand changes
- Designed for almost any scenario

Azure Container Instances

- Simplest and fastest way to run a container in Azure
- Platform as a Service
- Serverless Containers
- Designed for
    - Small and simple web apps/services
    - Background jobs
    - Scheduled scripts

Azure Kubernetes Service (AKS)

- Open-source container orchestration platform
- Platform as a Service
- Highly scalable and customizable
- Designed for high scale container deployments (anything really!)

App Service

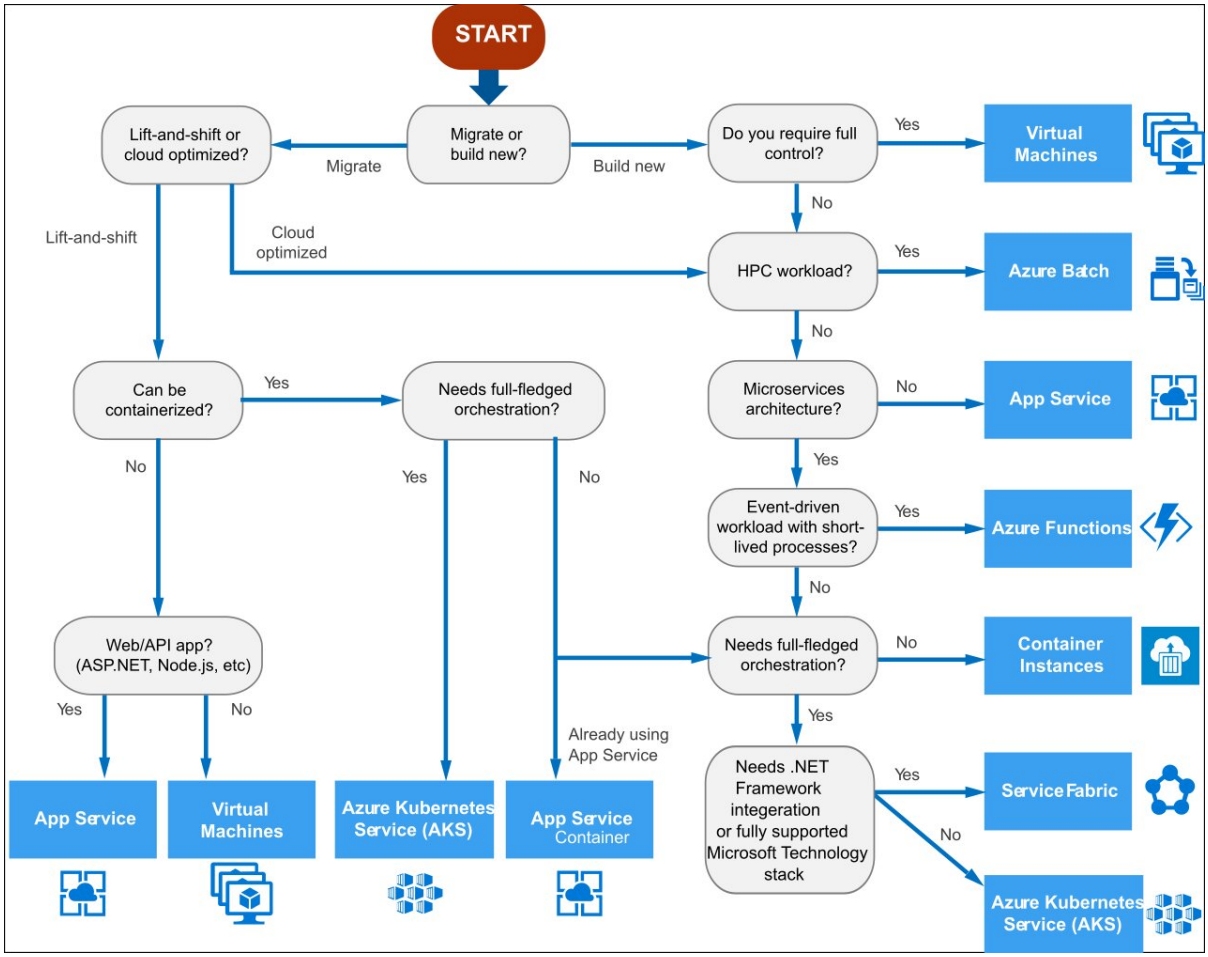
- Designed as enterprise grade web application service
- Platform as a Service
- Supports multiple programming languages and containers

Azure Functions (Function Apps)

- Platform as a Service
- Serverless
- Two hosting/pricing models
    - Consumption-based plan
    - Dedicated plan
- Designed for micro/nano-services

Summary

- Virtual Machines (IaaS) - Custom software, custom requirements, very specialized, high degree of control
- VM Scale Sets (IaaS) - Auto-scaled workloads for VMs
- Container Instances (PaaS) - Simple container hosting, easy to start
- Kubernetes Service (PaaS) - Highly scalable and customizable * container hosting platform
- App Services (PaaS) - Web applications, a lot of enterprise web * hosting features, easy to start
- Functions (PaaS) (Function as a Service) (Serverless) - micro/nano-services, excellent consumption-based pricing, easy to start



Scale with Azure Load Balancer

You now have your site up and running on Azure. But how can you help ensure your site is running 24/7?

For instance, what happens when you need to do weekly maintenance? Your service will still be unavailable during your maintenance window. And because your site reaches users all over the world, there's no good time to take down your systems for maintenance. You may also run into performance issues if too many users connect at the same time.

What are availability and high availability?

Availability refers to how long your service is up and running without interruption. *High availability*, or *highly available*, refers to a service that's up and running for a long period of time.

You know how frustrating it is when you can't access the information you need. Think of a social media or news site that you visit daily. Can you always access the site, or do you often see error messages like "503 Service Unavailable"?

You may have heard terms like "five nines availability." Five nines availability means that the service is guaranteed to be running 99.999 percent of the time. Although it's difficult to achieve 100 percent availability, many teams strive for at least five nines.

What is resiliency?

Resiliency refers to a system's ability to stay operational during abnormal conditions.

These conditions include:

- Natural disasters
- System maintenance, both planned and unplanned, including software updates and security patches.
- Spikes in traffic to your site
- Threats made by malicious parties, such as distributed denial of service (DDoS) attacks

Imagine your marketing team wants to have a flash sale to promote a new line of vitamin supplements. You might expect a huge spike in traffic during this time. This spike could overwhelm your processing system, causing it to slow down or halt, disappointing your users. You may have experienced this disappointment for yourself. Have you ever tried to access an online sale only to find the website wasn't responding?

What is a load balancer?

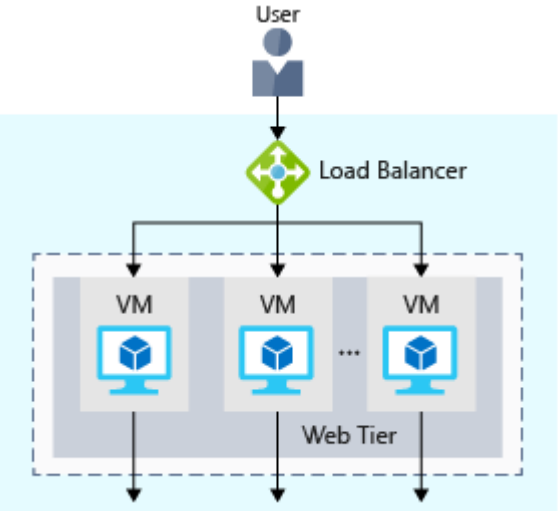
A *load balancer* distributes traffic evenly among each system in a pool. A load balancer can help you achieve both high availability and resiliency.

Say you start by adding additional VMs, each configured identically, to each tier. The idea is to have additional systems ready, in case one goes down, or is serving too many users at the same time.

The problem here is that each VM would have its own IP address. Plus, you don't have a way to distribute traffic in case one system goes down or is busy. How do you connect your VMs so that they appear to the user as one system?

The answer is to use a *load balancer* to distribute traffic. The load balancer becomes the entry point to the user. The user doesn't know (or need to know) which system the load balancer chooses to receive the request.

The following illustration shows the role of a load balancer.



The load balancer receives the user's request and directs the request to one of the VMs in the web tier. If a VM is unavailable or stops responding, the load balancer stops sending traffic to it. The load balancer then directs traffic to one of the responsive servers.

Load balancing enables you to run maintenance tasks without interrupting service. For example, you can stagger the maintenance window for each VM. During the maintenance window, the load balancer detects that the VM is unresponsive, and directs traffic to other VMs in the pool.

For your e-commerce site, the app and data tiers can also have a load balancer. It all depends on what your service requires.

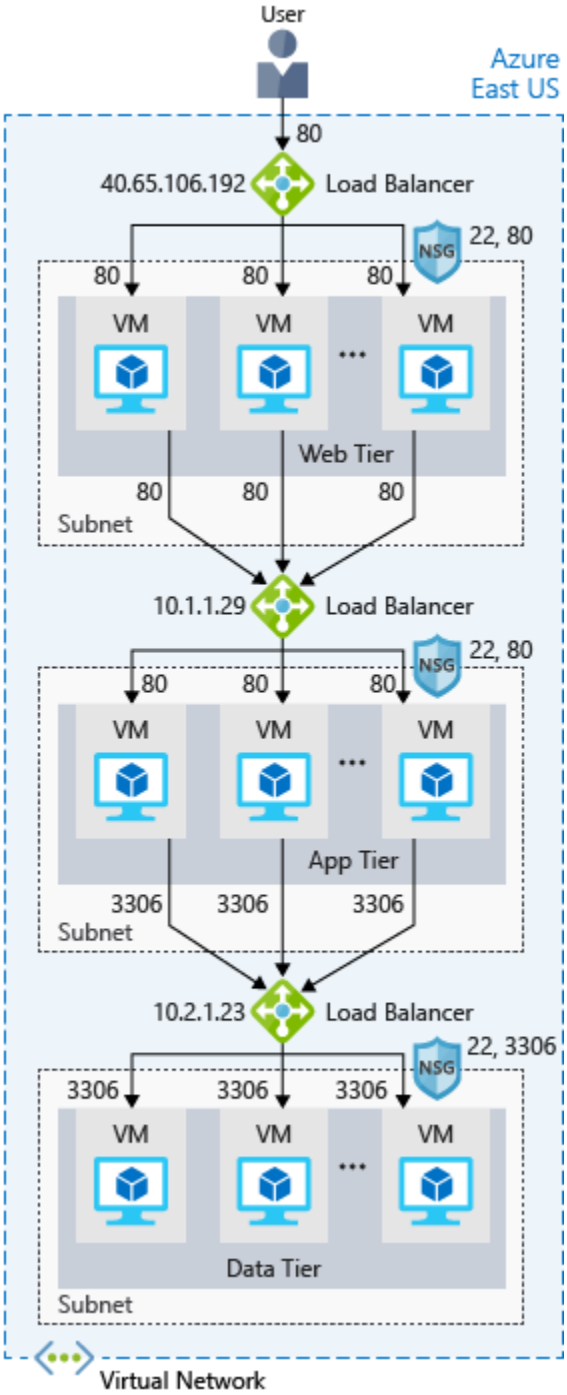
What is Azure Load Balancer?

Azure Load Balancer is a load balancer service that Microsoft provides that helps take care of the maintenance for you. Load Balancer supports inbound and outbound scenarios, provides low latency and high throughput, and scales up to millions of flows for all Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications. You can use Load Balancer with incoming internet traffic, internal traffic across Azure services, port forwarding for specific traffic, or outbound connectivity for VMs in your virtual network.

When you manually configure typical load balancer software on a virtual machine, there's a downside: you now have an additional system that you need to maintain. If your load balancer goes down or needs routine maintenance, you're back to your original problem.

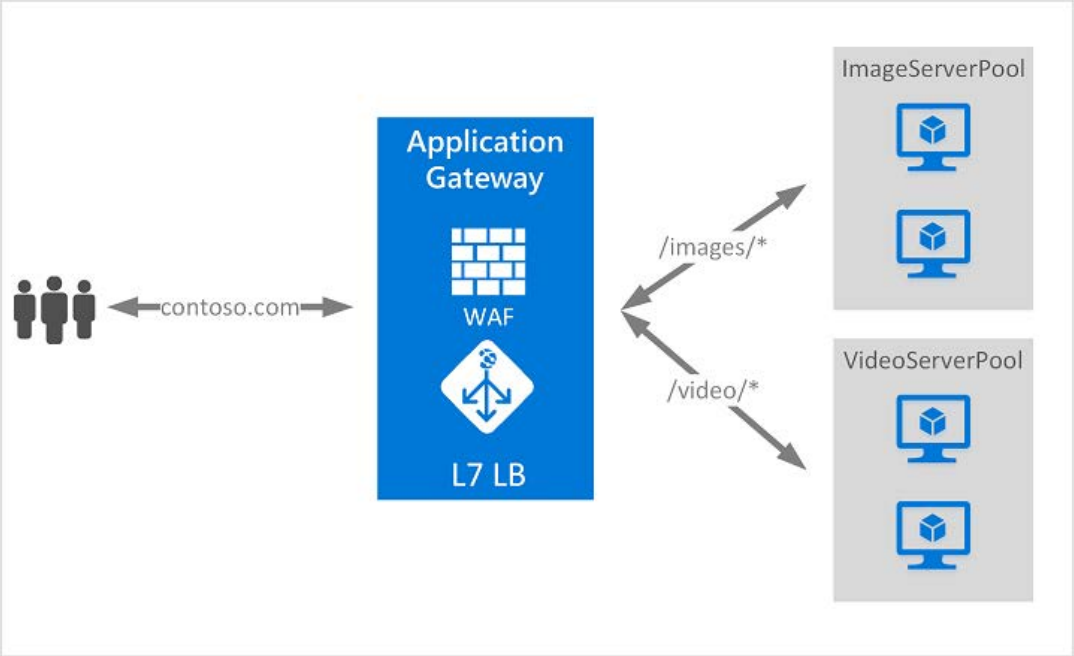
If instead, however, you use Azure Load Balancer, there's no infrastructure or software for you to maintain. You define the forwarding rules based on the source IP and port to a set of destination IP/ports.

The following illustration shows the role of Azure load balancers in a multi-tier architecture.



Azure Application Gateway

If all your traffic is HTTP, a potentially better option is to use Azure Application Gateway. Application Gateway is a load balancer designed for web applications. It uses Azure Load Balancer at the transport level (TCP) and applies sophisticated URL-based routing rules to support several advanced scenarios.



This type of routing is known as application layer (OSI layer 7) load balancing since it understands the structure of the HTTP message.

Here are some of the benefits of using Azure Application Gateway over a simple load balancer:

- **Cookie affinity**. Useful when you want to keep a user session on the same backend server.
- **SSL termination**. Application Gateway can manage your SSL certificates and pass unencrypted traffic to the backend servers to avoid encryption/decryption overhead. It also supports full end-to-end encryption for applications that require that.
- **Web application firewall**. Application gateway supports a sophisticated firewall (WAF) with detailed monitoring and logging to detect malicious attacks against your network infrastructure.
- **URL rule-based routes**. Application Gateway allows you to route traffic based on URL patterns, source IP address and port to destination IP address and port. This is helpful when setting up a *content delivery network*.
- **Rewrite HTTP headers**. You can add or remove information from the inbound and outbound HTTP headers of each request to enable important security scenarios, or scrub sensitive information such as server names.

What is a Content Delivery Network?

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users. It is a way to get content to users in their local region to minimize latency. CDN can be hosted in Azure or any other location. You can cache content at strategically placed physical nodes across the world and provide better performance to end users. Typical usage scenarios include web applications containing multimedia content, a product launch event in a particular region, or any event where you expect a high-bandwidth requirement in a region.

What about DNS?

DNS, or Domain Name System, is a way to map user-friendly names to their IP addresses. You can think of DNS as the phonebook of the internet.

For example, your domain name, contoso.com, might map to the IP address of the load balancer at the web tier, 40.65.106.192.

You can bring your own DNS server or use Azure DNS, a hosting service for DNS domains that runs on Azure infrastructure.

The following illustration shows Azure DNS. When the user navigates to **contoso.com**, Azure DNS routes traffic to the load balancer.

Summary

With load balancing in place, your e-commerce site is now more highly available and resilient. When you perform maintenance or receive an uptick in traffic, your load balancer can distribute traffic to another available system.

Although you can configure your own load balancer on a VM, Azure Load Balancer reduces upkeep because there's no infrastructure or software to maintain.

DNS maps user-friendly names to their IP addresses, much like how a phonebook maps names of people or businesses to phone numbers. You can bring your own DNS server, or use Azure DNS.

Reduce latency with Azure Traffic Manager

- 7 minutes

Previously, you saw how **Azure Load Balancer** helps you achieve high availability and minimize downtime.

Although your e-commerce site is more highly available, it doesn't solve the issue of latency or create resiliency across geographic regions.

How can you make your site, which is located in the United States, load faster for users located in Europe or Asia?

What is network latency?

Latency refers to the time it takes for data to travel over the network. Latency is typically measured in milliseconds.

Compare latency to bandwidth. Bandwidth refers to the amount of data that can fit on the connection. Latency refers to the time it takes for that data to reach its destination.

Factors such as the type of connection you use and how your application is designed can affect latency. But perhaps the biggest factor is distance.

Think about your e-commerce site on Azure, which is in the East US region. It would typically take less time to transfer data to Atlanta (a distance of around 400 miles) than to transfer data to London (a distance of around 4,000 miles).

Your e-commerce site delivers standard HTML, CSS, JavaScript, and images. The network latency for many files can add up. How can you reduce latency for users located far away geographically?

Scale out to different regions

Recall that Azure provides data centers in regions across the globe.

Think about the cost of building a data center. Equipment costs aren't the only factor. You need to provide the power, cooling, and personnel to keep your systems running at each location. It might be prohibitively expensive to replicate your entire data center. But doing so with Azure can cost much less, because Azure already has the equipment and personnel in place.

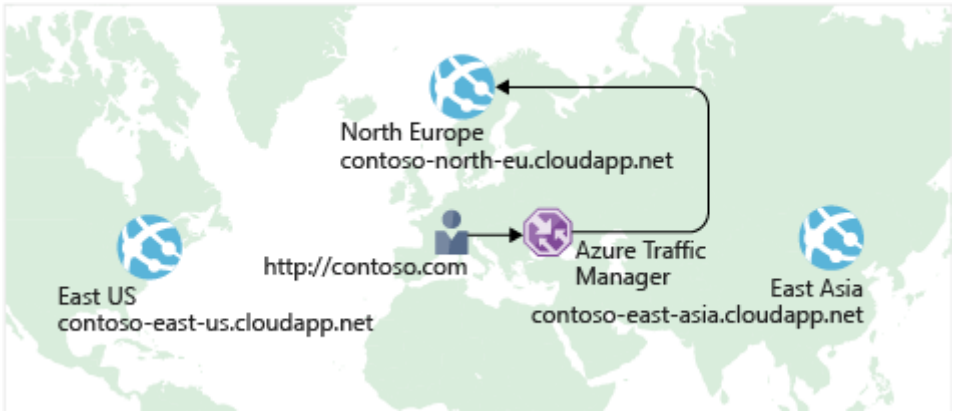
One way to reduce latency is to provide exact copies of your service in more than one region. The following illustration shows an example of global deployment.

The diagram shows your e-commerce site running in three Azure regions: East US, North Europe, and East Asia. Notice the DNS name for each. How can you connect users to the service that's closest geographically, but under the contoso.com domain?

Use Traffic Manager to route users to the closest endpoint

One answer is **Azure Traffic Manager**. Traffic Manager uses the DNS server that's closest to the user to direct user traffic to a globally distributed endpoint.

The following illustration shows the role of the Traffic Manager.



Traffic Manager doesn't see the traffic that's passed between the client and server. Rather, it directs the client web browser to a preferred endpoint. Traffic Manager can route traffic in a few different ways, such as to the endpoint with the lowest latency.

Although not shown here, this setup could also include your on-premises deployment running in California. You can connect Traffic Manager to your own on-premises networks, enabling you to maintain your existing data center investments. Or you can move your application entirely to the cloud. The choice is yours.

Compare Load Balancer to Traffic Manager

Azure Load Balancer distributes traffic within the same region to make your services more highly available and resilient. Traffic Manager works at the DNS level, and directs the client to a preferred endpoint. This endpoint can be to the region that's closest to your user.

Load Balancer and Traffic Manager both help make your services more resilient, but in slightly different ways. When Load Balancer detects an unresponsive VM, it directs traffic to other VMs in the pool. Traffic Manager monitors the health of your endpoints. When Traffic Manager finds an unresponsive endpoint, it directs traffic to the next closest endpoint that is responsive.

Summary

Geographic distance is one of the biggest factors that contributes to latency. With Traffic Manager in place, you can host exact copies of your service in multiple geographic regions. That way, users in the United States, Europe, and Asia will all have a good experience using your e-commerce site.

Azure Networking

- Connect cloud and on-premises
- On-premise networking functionality

Azure Virtual Network

- Logically isolated networking components
- Segmented into one or more subnets
- Subnets are discrete sections
- Enable communication of resources with each-other, internet and on-premises
- Scoped to a single region
- VNet peering allow cross region communication
- Isolation, Segmentation, Communication, Filtering, Routing

Azure Load Balancer

- Even traffic distribution
- Supports both inbound and outbound scenarios
- High-availability scenarios

- Both TCP (transmission control protocol) and UDP (user datagram protocol) applications
- Internal and External traffic
- Port Forwarding
- High scale with up to millions of flows

VPN Gateway

- Specific type of virtual network gateway for on-premises to azure traffic over the public internet

Application Gateway

- Web traffic load balancer
- Web application firewall
- Redirection
- Session affinity
- URL Routing
- SSL termination

Content Delivery Network

- Define content
- Minimize latency
- POP (points of presence) with many locations

Benefits of using Azure to store data

To address the storage problem for your online learning portal, you're considering storing your data in the cloud. But you're concerned about security, backup, and disaster recovery. On top of those issues, you're worried about how difficult it could be to manage cloud-hosted data. So, here's what you need to know.

The Azure data storage options are cloud-based, secure, and scalable. Its features address the key challenges of cloud storage and provide you with a reliable and durable storage solution.

Benefits of using Azure to store data

Here are some of the important benefits of Azure data storage:

- **Automated backup and recovery**: mitigates the risk of losing your data if there is any unforeseen failure or interruption.
- **Replication across the globe**: copies your data to protect it against any planned or unplanned events, such as scheduled maintenance or hardware failures. You can choose to replicate your data at multiple locations across the globe.

- **Support for data analytics**: supports performing analytics on your data consumption.
- **Encryption capabilities**: data is encrypted to make it highly secure; you also have tight control over who can access the data.
- **Multiple data types**: Azure can store almost any type of data you need. It can handle video files, text files, and even large binary files like virtual hard disks. It also has many options for your relational and NoSQL data.
- **Data storage in virtual disks**: Azure also has the capability of storing up to 32 TB of data in its virtual disks. This capability is significant when you're storing heavy data such as videos and simulations.
- **Storage tiers**: storage tiers to prioritize access to data based on frequently used versus rarely used information.

Types of data

There are three primary types of data that Azure Storage is designed to hold.

1. **Structured data**. Structured data is data that adheres to a schema, so all of the data has the same fields or properties. Structured data can be stored in a database table with rows and columns. Structured data relies on keys to indicate how one row in a table relates to data in another row of another table. Structured data is also referred to as *relational data*, as the data's schema defines the table of data, the fields in the table, and the clear relationship between the two. Structured data is straightforward in that it's easy to enter, query, and analyze. All of the data follows the same format. Examples of structured data include sensor data or financial data.
2. **Semi-structured data**. Semi-structured data doesn't fit neatly into tables, rows, and columns. Instead, semi-structured data uses *tags* or *keys* that organize and provide a hierarchy for the data. Semi-structured data is also referred to as *non-relational* or *NoSQL* data.
3. **Unstructured data**. Unstructured data encompasses data that has no designated structure to it. This lack of structure also means that there are no restrictions on the kinds of data it can hold. For example, a blob can hold a PDF document, a JPG image, a JSON file, video content, etc. As such, unstructured data is becoming more prominent as businesses try to tap into new data sources.

How Azure data storage can meet your business storage needs

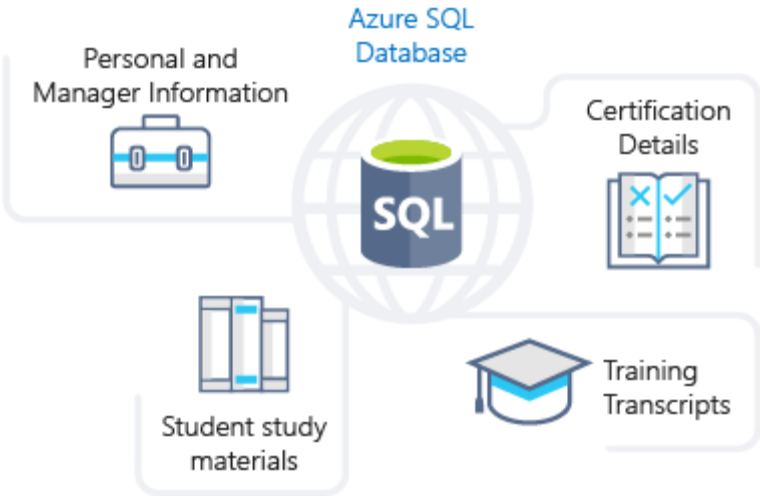
Azure provides several storage options that accommodate specific types of data storage needs.

Azure SQL Database

Azure SQL Database is a relational database as a service (DaaS) based on the latest stable version of the Microsoft SQL Server database engine. SQL Database is a high-performance, reliable, fully managed and secure database. You can use it to build data-driven applications and websites in the programming language of your choice without needing to manage infrastructure.

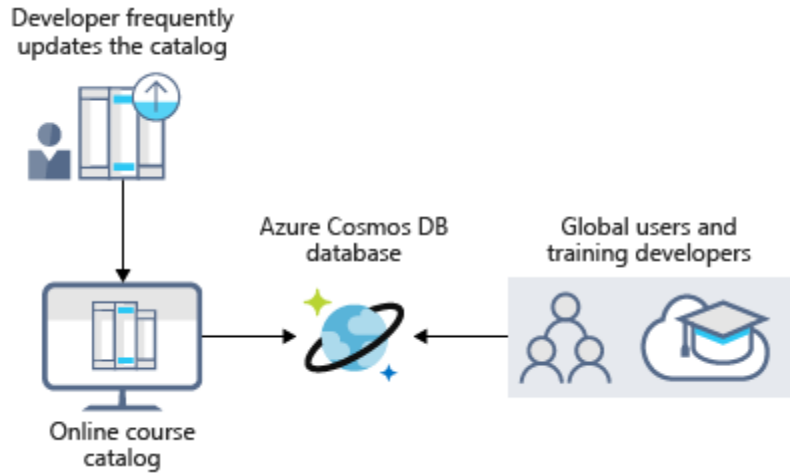
You can migrate your existing SQL Server databases with minimal downtime using the Azure Database Migration Service. The service uses the *Microsoft Data Migration Assistant* to generate assessment reports that provide recommendations to help guide you through required changes prior to performing a migration. Once you assess and perform any remediation required, you're ready to begin the migration process. The Azure Database Migration Service performs all of the required steps. You just change the connection string in your apps.

The following illustration shows the types of data from the online learning portal scenario that would be stored in an Azure SQL database.



Azure Cosmos DB

Azure Cosmos DB is a globally distributed database service. It supports schema-less data that lets you build highly responsive and **Always On** applications to support constantly changing data. You can use this feature to store data that is updated and maintained by users around the world. The following illustration shows a sample Azure Cosmos DB database that's used to store data that's accessed by people located across the globe.

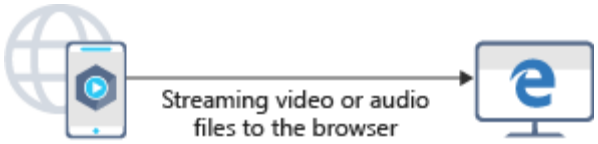


Azure Blob storage

Azure Blob Storage is *unstructured*, meaning that there are no restrictions on the kinds of data it can hold. Blobs are highly scalable and apps work with blobs in much the same way as they would work with files on a disk, such as reading and writing data. Blob Storage can manage thousands of simultaneous uploads, massive amounts of video data, constantly growing log files, and can be reached from anywhere with an internet connection.

Blobs aren't limited to common file formats. A blob could contain gigabytes of binary data streamed from a scientific instrument, an encrypted message for another application, or data in a custom format for an app you're developing.

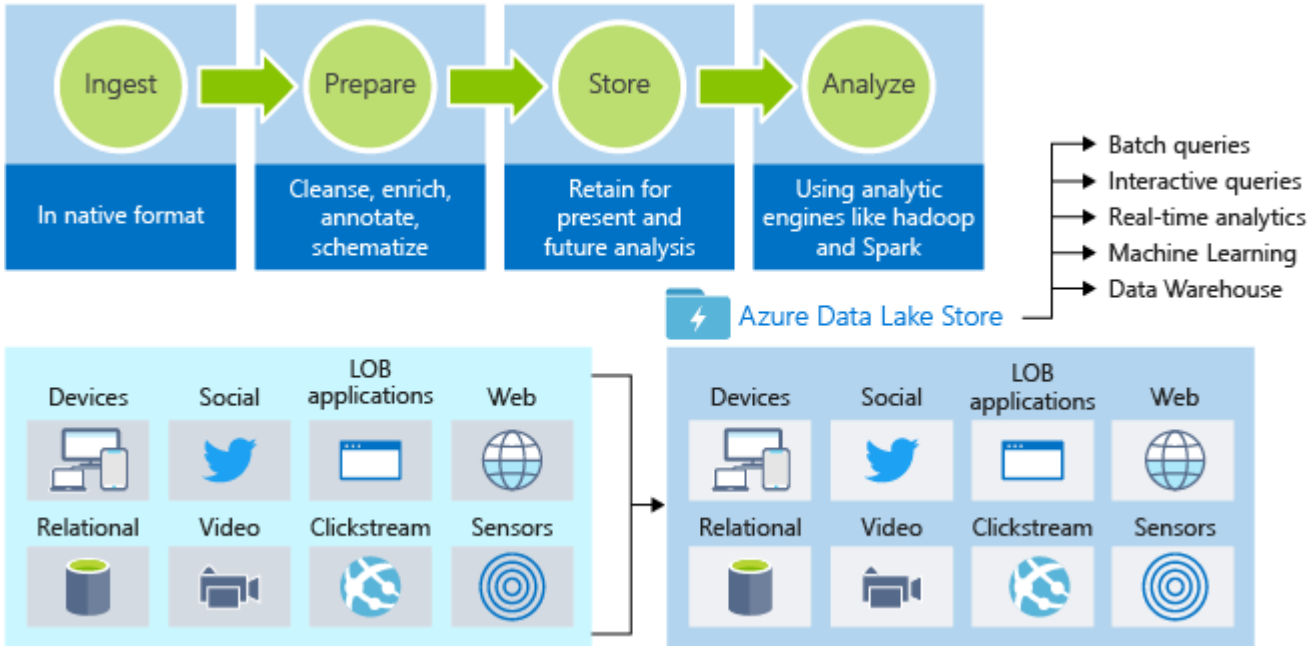
Azure Blob storage lets you stream large video or audio files directly to the user's browser from anywhere in the world. Blob storage is also used to store data for backup, disaster recovery, and archiving. It has the ability to store up to 8 TB of data for virtual machines. The following illustration shows an example usage of Azure blob storage.



Azure Data Lake Storage

The Data Lake feature allows you to perform analytics on your data usage and prepare reports. Data Lake is a large repository that stores both structured and unstructured data.

Azure Data Lake Storage combines the scalability and cost benefits of object storage with the reliability and performance of the Big Data file system capabilities. The following illustration shows how Azure Data Lake stores all your business data and makes it available for analysis.

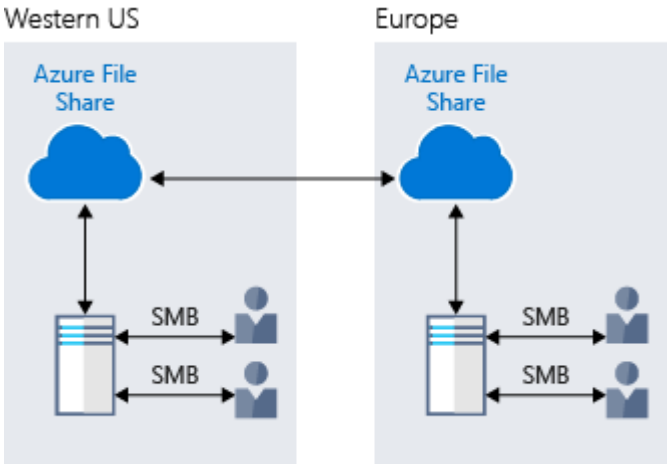


Data progresses through a flow diagram from ingest in its native format; prepare, where data is cleansed, enriched, annotated, and schematized; store, where data is retained for present and future analysis; then to analyze, where analytics engines like Hadoop and Spark are used on the data. Data is shown ingested to Azure Data Lake Store from devices, social media, LOB applications, web sites, relational databases, video, Clickstream, and sensors. From there, it can be accessed with batch queries, interactive queries, real-time analytics, machine learning, and data warehouse.

Azure Files

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) protocol. Azure file shares can be mounted concurrently by cloud or on-premises deployments of Windows, Linux, and macOS. Applications running in Azure virtual machines or cloud services can mount a file storage share to access file data, just as a desktop application would mount a typical SMB share. Any number of Azure virtual machines or roles can mount and access the file storage share simultaneously. Typical usage scenarios would be to share files anywhere in the world, diagnostic data, or application data sharing.

The following illustration shows Azure Files being used to share data between two geographical locations. Azure Files uses the Server Message Block (SMB) protocol that ensures the data is encrypted at rest and in transit.

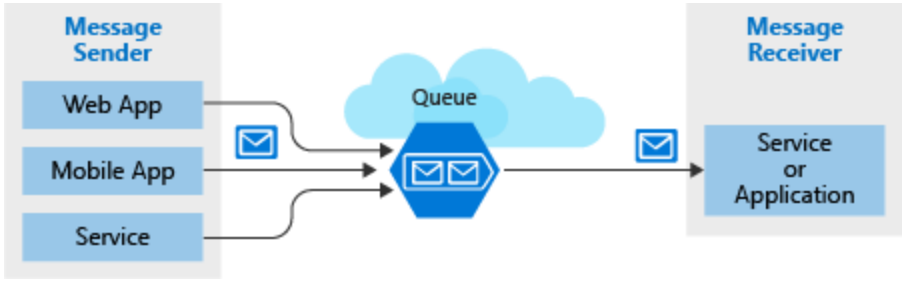


Azure Queue

Azure Queue storage is a service for storing large numbers of messages that can be accessed from anywhere in the world.

Azure Queue Storage can be used to help build flexible applications and separate functions for better durability across large workloads. When application components are decoupled, they can scale independently. Queue storage provides asynchronous message queueing for communication between application components, whether they are running in the cloud, on the desktop, on-premises, or on mobile devices.

Typically, there are one or more sender components and one or more receiver components. Sender components add messages to the queue, while receiver components retrieve messages from the front of the queue for processing. The following illustration shows multiple sender applications adding messages to the Azure Queue and one receiver application retrieving the messages.



You can use queue storage to:

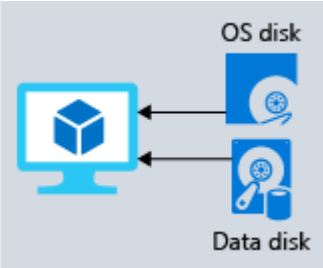
- Create a backlog of work and to pass messages between different Azure web servers.
- Distribute load among different web servers/infrastructure and to manage bursts of traffic.
- Build resilience against component failure when multiple users access your data at the same time.

Disk Storage

Disk storage provides disks for virtual machines, applications, and other services to access and use as they need, similar to how they would in on-premises scenarios. Disk storage allows data to be persistently stored and accessed from an attached virtual hard disk. The disks can be managed or unmanaged by Azure, and therefore managed and configured by the user. Typical scenarios for using disk storage are if you want to lift and shift applications that read and write data to persistent disks, or if you are storing data that is not required to be accessed from outside the virtual machine to which the disk is attached.

Disks come in many different sizes and performance levels, from solid-state drives (SSDs) to traditional spinning hard disk drives (HDDs), with varying performance abilities.

When working with VMs, you can use standard SSD and HDD disks for less critical workloads, and premium SSD disks for mission-critical production applications. Azure Disks have consistently delivered enterprise-grade durability, with an industry-leading ZERO% annualized failure rate. The following illustration shows an Azure virtual machine using separate disks to store different data.



Storage tiers

Azure offers three storage tiers for blob object storage:

1. **Hot storage tier**: optimized for storing data that is accessed frequently.
2. **Cool storage tier**: optimized for data that are infrequently accessed and stored for at least 30 days.
3. **Archive storage tier**: for data that are rarely accessed and stored for at least 180 days with flexible latency requirements.

Encryption and replication

Azure provides security and high availability to your data through encryption and replication features.

Encryption for storage services

The following encryption types are available for your resources:

1. **Azure Storage Service Encryption (SSE)** for data at rest helps you secure your data to meet the organization's security and regulatory compliance. It encrypts the data before storing it and decrypts the data before returning it. The encryption and decryption are transparent to the user.
2. **Client-side encryption** is where the data is already encrypted by the client libraries. Azure stores the data in the encrypted state at rest, which is then decrypted during retrieval.

Replication for storage availability

A replication type is set up when you create a storage account. The replication feature ensures that your data is durable and always available. Azure provides regional and geographic replications to protect your data against natural disasters and other local disasters like fire or flooding.

Comparison between Azure data storage and on-premises storage

- 4 minutes

Now that you know about the benefits and features of Azure data storage, let's see how it differs from on-premises storage.

The term "on-premises" refers to the storage and maintenance of data on local hardware and servers. There are several factors to consider when comparing on-premises to Azure data storage.

Cost effectiveness

An on-premises storage solution requires dedicated hardware that needs to be purchased, installed, configured, and maintained. This requirement can be a significant up-front expense (or capital cost). Change in requirements can require investment in new hardware. Your hardware needs to be capable of handling peak demand, which means it may sit idle or be under-utilized in off-peak times.

Azure data storage provides a pay-as-you-go pricing model, which is often appealing to businesses as an operating expense instead of an upfront capital cost. It's also scalable, allowing you to scale up or scale out as demand dictates and scale back when demand is low. You are charged for data services only as you need them.

Reliability

On-premises storage requires data backup, load balancing, and disaster recovery strategies. These requirements can be challenging and expensive as they often each need dedicated servers requiring a significant investment in both hardware and IT resources.

Azure data storage provides data backup, load balancing, disaster recovery, and data replication as services to ensure data safety and high availability.

Storage types

Sometimes multiple different storage types are required for a solution, such as file and database storage. An on-premises approach often requires numerous servers and administrative tools for each storage type.

Azure data storage provides a variety of different storage options including distributed access and tiered storage. This variety makes it possible to integrate a combination of storage technologies providing the best storage choice for each part of your solution.

Agility

Requirements and technologies change. For an on-premises deployment, these changes may mean provisioning and deploying new servers and infrastructure pieces, which are a time consuming and expensive activity.

Azure data storage gives you the flexibility to create new services in minutes. This flexibility allows you to change storage back-ends quickly without needing a significant hardware investment.

Compare on-premises storage to Azure data storage

The following table describes the differences between on-premises storage and Azure data storage.

COMPARE ON-PREMISES STORAGE TO AZURE DATA STORAGE

Needs	On-premises	Azure data storage
Compliance and security	Dedicated servers required for privacy and security	Client-side encryption and encryption at rest
Store structured and unstructured data	Additional IT resources with dedicated servers required	Azure Data Lake and portal analyzes and manages all types of data
Replication and high availability	More resources, licensing, and servers required	Built-in replication and redundancy features available
Application sharing and access to shared resources	File sharing requires additional administration resources	File sharing options available without additional license

COMPARE ON-PREMISES STORAGE TO AZURE DATA STORAGE

Needs	On-premises	Azure data storage
Relational data storage	Needs a database server with database admin role	Offers database-as-a-service options
Distributed storage and data access	Expensive storage, networking, and compute resources needed	Azure Cosmos DB provides distributed access
Messaging and load balancing	Hardware redundancy impacts budget and resources	Azure Queue provides effective load balancing
Tiered storage	Management of tiered storage needs technology and labor skill set	Azure offers automated tiered storage of data

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Core storage services offer a massively scalable object store for data objects, disk storage for Azure virtual machines (VMs), a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store. The services are:

- **Durable and highly available.** Redundancy ensures that your data is safe in the event of transient hardware failures. You can also opt to replicate data across datacenters or geographical regions for additional protection from local catastrophe or natural disaster. Data replicated in this way remains highly available in the event of an unexpected outage.
- **Secure.** All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- **Scalable.** Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
- **Managed.** Azure handles hardware maintenance, updates, and critical issues for you.
- **Accessible.** Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides client libraries for Azure Storage in a variety of languages, including .NET, Java, Node.js, Python, PHP, Ruby, Go, and others, as well as a mature REST API. Azure Storage supports scripting in Azure PowerShell or Azure CLI. And the Azure portal and Azure Storage Explorer offer easy visual solutions for working with your data.

Core storage services

The Azure Storage platform includes the following data services:

- Azure Blobs: A massively scalable object store for text and binary data. Also includes support for big data analytics through Data Lake Storage Gen2.
- Azure Files: Managed file shares for cloud or on-premises deployments.
- Azure Queues: A messaging store for reliable messaging between application components.
- Azure Tables: A NoSQL store for schemaless storage of structured data.
- Azure Disks: Block-level storage volumes for Azure VMs.

Each service is accessed through a storage account. To get started, see Create a storage account.

Example scenarios

The following table compares Files, Blobs, Disks, Queues, and Tables, and shows example scenarios for each.

EXAMPLE SCENARIOS

Feature	Description	When to use
Azure Files	Offers fully managed cloud file shares that you can access from anywhere via the industry standard Server Message Block (SMB) protocol. You can mount Azure file shares from cloud or on-premises deployments of Windows, Linux, and macOS.	You want to "lift and shift" an application to the cloud that already uses the native file system APIs to share data between it and other applications running in Azure. You want to replace or supplement on-premises file servers or NAS devices. You want to store development and debugging tools that need to be accessed from many virtual machines.
Azure Blobs	Allows unstructured data to be stored and accessed at a massive scale in block blobs. Also supports Azure Data Lake Storage Gen2 for enterprise big data analytics solutions.	You want your application to support streaming and random access scenarios. You want to be able to access application data from anywhere. You want to build an enterprise data lake on Azure and perform big data analytics.
Azure Disks	Allows data to be persistently stored and accessed from an attached virtual hard disk.	You want to "lift and shift" applications that use native file system APIs to read and write data to persistent disks. You want to store data that is not required to be accessed from outside the virtual machine to which the disk is attached.
Azure Queues	Allows for asynchronous message queueing between application components.	You want to decouple application components and use asynchronous messaging to communicate between them. For guidance around when to use Queue storage versus Service Bus queues, see Storage queues and Service Bus queues - compared and contrasted.
Azure Tables	Allow you to store structured NoSQL data in the cloud, providing a key/attribute store with a schemaless design.	You want to store flexible datasets like user data for web applications, address books, device information, or other types of metadata your service requires. For guidance around when to use Table storage versus the Azure Cosmos DB Table API, see Developing with Azure Cosmos DB Table API and Azure Table storage.

Blob storage

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data, such as text or binary data.

Blob storage is ideal for:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Storing data for backup and restore, disaster recovery, and archiving.
- Storing data for analysis by an on-premises or Azure-hosted service.

Objects in Blob storage can be accessed from anywhere in the world via HTTP or HTTPS. Users or client applications can access blobs via URLs, the Azure Storage REST API, Azure PowerShell, Azure CLI, or an Azure Storage client library. The storage client libraries are available for multiple languages, including .NET, Java, Node.js, Python, PHP, and Ruby.

For more information about Blob storage, see Introduction to Blob storage.

Azure Files

Azure Files enables you to set up highly available network file shares that can be accessed by using the standard Server Message Block (SMB) protocol. That means that multiple VMs can share the same files with both read and write access. You can also read the files using the REST interface or the storage client libraries.

One thing that distinguishes Azure Files from files on a corporate file share is that you can access the files from anywhere in the world using a URL that points to the file and includes a shared access signature (SAS) token. You can generate SAS tokens; they allow specific access to a private asset for a specific amount of time.

File shares can be used for many common scenarios:

- Many on-premises applications use file shares. This feature makes it easier to migrate those applications that share data to Azure. If you mount the file share to the same drive letter that the on-premises application uses, the part of your application that accesses the file share should work with minimal, if any, changes.
- Configuration files can be stored on a file share and accessed from multiple VMs. Tools and utilities used by multiple developers in a group can be stored on a file share, ensuring that everybody can find them, and that they use the same version.
- Resource logs, metrics, and crash dumps are just three examples of data that can be written to a file share and processed or analyzed later.

For more information about Azure Files, see Introduction to Azure Files.

Some SMB features are not applicable to the cloud. For more information, see Features not supported by the Azure File service.

Queue storage

The Azure Queue service is used to store and retrieve messages. Queue messages can be up to 64 KB in size, and a queue can contain millions of messages. Queues are generally used to store lists of messages to be processed asynchronously.

For example, say you want your customers to be able to upload pictures, and you want to create thumbnails for each picture. You could have your customer wait for you to create the thumbnails while uploading the pictures. An alternative would be to use a queue. When the customer finishes their upload, write a message to the queue. Then have an Azure Function retrieve the message from the queue and create the thumbnails. Each of the parts of this processing can be scaled separately, giving you more control when tuning it for your usage.

For more information about Azure Queues, see Introduction to Queues.

Table storage

Azure Table storage is now part of Azure Cosmos DB. To see Azure Table storage documentation, see the Azure Table Storage Overview. In addition to the existing Azure Table storage service, there is a new Azure Cosmos DB Table API offering that provides throughput-optimized tables, global distribution, and automatic secondary indexes. To learn more and try out the new premium experience, see Azure Cosmos DB Table API.

For more information about Table storage, see Overview of Azure Table storage.

Disk storage

An Azure managed disk is a virtual hard disk (VHD). You can think of it like a physical disk in an on-premises server but, virtualized. Azure-managed disks are stored as page blobs, which are a random IO storage object in Azure. We call a managed disk 'managed' because it is an abstraction over page blobs, blob containers, and Azure storage accounts. With managed disks, all you have to do is provision the disk, and Azure takes care of the rest.

For more information about managed disks, see Introduction to Azure managed disks.

Types of storage accounts

Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model. For more information about storage account types, see Azure storage account overview.

Secure access to storage accounts

Every request to Azure Storage must be authorized. Azure Storage supports the following authorization methods:

- **Azure Active Directory (Azure AD) integration for blob and queue data.** Azure Storage supports authentication and authorization with Azure AD for the Blob and Queue services via Azure role-based access control (Azure RBAC). Authorizing requests with Azure AD is recommended for superior security and ease of use. For more information, see Authorize access to Azure blobs and queues using Azure Active Directory.
- **Azure AD authorization over SMB for Azure Files.** Azure Files supports identity-based authorization over SMB (Server Message Block) through either Azure Active Directory Domain Services (Azure AD DS) or on-premises Active Directory Domain Services (preview). Your domain-joined Windows VMs can access Azure file shares using Azure AD credentials. For more information, see Overview of Azure Files identity-based authentication support for SMB access and Planning for an Azure Files deployment.
- **Authorization with Shared Key.** The Azure Storage Blob, Files, Queue, and Table services support authorization with Shared Key. A client using Shared Key authorization passes a header with every request that is signed using the storage account access key. For more information, see Authorize with Shared Key.
- **Authorization using shared access signatures (SAS).** A shared access signature (SAS) is a string containing a security token that can be appended to the URI for a storage resource. The security token encapsulates constraints such as permissions and the interval of access. For more information, see Using Shared Access Signatures (SAS).
- **Anonymous access to containers and blobs.** A container and its blobs may be publicly available. When you specify that a container or blob is public, anyone can read it anonymously; no authentication is required. For more information, see Manage anonymous read access to containers and blobs.

Encryption

There are two basic kinds of encryption available for the core storage services. For more information about security and encryption, see the Azure Storage security guide.

Encryption at rest

Azure Storage encryption protects and safeguards your data to meet your organizational security and compliance commitments. Azure Storage automatically encrypts all data prior to persisting to the storage account and decrypts it prior to retrieval. The encryption, decryption, and key management processes are transparent to users. Customers can also choose to manage their own keys using Azure Key Vault. For more information, see Azure Storage encryption for data at rest.

Client-side encryption

The Azure Storage client libraries provide methods for encrypting data from the client library before sending it across the wire and decrypting the response. Data encrypted via client-side encryption is also encrypted at rest by Azure Storage. For more information about client-side encryption, see Client-side encryption with .NET for Azure Storage.

Redundancy

To ensure that your data is durable, Azure Storage stores multiple copies of your data. When you set up your storage account, you select a redundancy option. For more information, see Azure Storage redundancy.

Transfer data to and from Azure Storage

You have several options for moving data into or out of Azure Storage. Which option you choose depends on the size of your dataset and your network bandwidth. For more information, see Choose an Azure solution for data transfer.

Pricing

When making decisions about how your data is stored and accessed, you should also consider the costs involved. For more information, see Azure Storage pricing.

Storage APIs, libraries, and tools

You can access resources in a storage account by any language that can make HTTP/HTTPS requests. Additionally, the core Azure Storage services offer programming libraries for several popular languages. These libraries simplify many aspects of working with Azure Storage by handling details such as synchronous and asynchronous invocation, batching of operations, exception management, automatic retries, operational behavior, and so forth. Libraries are currently available for the following languages and platforms, with others in the pipeline:

Storage Account

- Group of services which include
    - blob storage,
    - queue storage,
    - table storage, and
    - file storage
- Used to store
    - files,
    - messages, and
    - semi-structured data
- Highly scalable (up to petabytes of data)
- Highly durable (99.999999999% - 11 nines, up to 16 nines)
- Cheapest per GB storage

Blob Storage

- BLOB – binary large object – file
- Designed for storage of files of any kind
- Three storage tiers
    - Hot – frequently accessed data
    - Cool – infrequently accessed data (lower availability, high durability)
    - Archive – rarely (if-ever) accessed data

Queue Storage

- Storage for small pieces of data (messages)
- Designed for scalable asynchronous processing

Table Storage

- Storage for semi-structured data (NoSQL)
    - No need for foreign joins, foreign keys, relationships or strict schema
    - Designed for fast access
- Many programming interfaces and SDKs

File Storage

- Storage for files accessed via shared drive protocols
- Designed to extend on-premise file shares or implement lift-and-shift scenarios

Disk Storage

- Disk emulation in the cloud
- Persistent storage for Virtual Machines
- Different
    - sizes,
    - types (SSD, HDD)
    - performance tiers
- Disk can be unmanaged or managed

What is the commercial marketplace?

After meeting with your company's executive team, you start to compile some information about the commercial marketplace to report back to the group during your next meeting. So far, you've learned that the commercial marketplace is a cloud-based, on-demand market that lets Microsoft partners to publish their solutions into Microsoft's online product catalog.

Built for customers

For your current and future customers, the commercial marketplace is a powerful tool to find, try, buy, and deploy best-in-class solutions that they can use to accelerate digital transformation and innovate in the cloud. For your customers that are already Microsoft customers, purchasing solutions from the commercial marketplace has an added benefit—they can include commercial marketplace purchases in their existing Microsoft purchase agreements and receive a consolidated invoice from Microsoft.

Built for partners

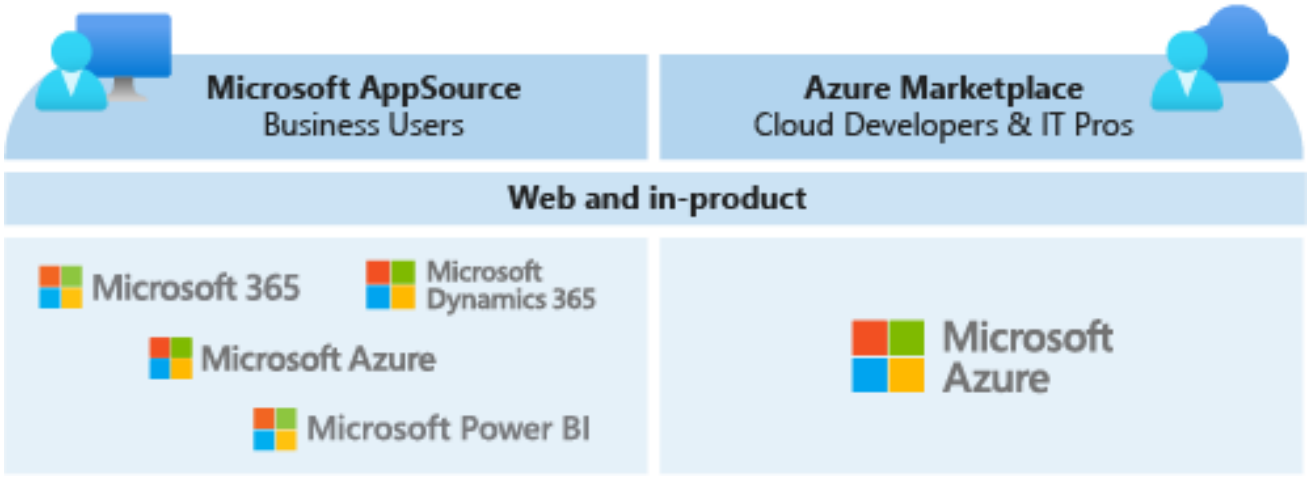
The commercial marketplace is free to join. You can use the commercial marketplace as a dynamic sales and marketing channel to access new markets, customer segments, and Microsoft cloud users with your software and services. You can also leverage it as a means to find best-in-class solutions you can deploy for use within your own cloud infrastructure to build your own solutions.

Multiple access points

When you list your company's solutions in the commercial marketplace, customers will be able to find them in two online stores, through the Microsoft network of resellers, and through in-product experiences in products like Microsoft 365, Dynamics 365, the Power Platform, and Azure.

The commercial marketplace has two online stores, **Microsoft AppSource** and **Azure Marketplace**, which are designed to serve different buyer roles within your customers' organizations.

Between the two online stores and the in-product experiences, the commercial marketplace draws millions of active users per month, with each unique user representing a potential customer for your company. Because Microsoft products and services are used widely by organizations around the world, these monthly users represent companies of all sizes in every industry.



Commercial marketplace online stores

When Microsoft partners add their solutions to the commercial marketplace, one way that customers can find their products and services is by visiting the online stores. You go to the commercial marketplace documentation to learn more.

Azure Marketplace

As you read the documentation, you learn that Azure Marketplace is the Microsoft online store for IT solutions. It contains technical software and services that are certified to run on Microsoft Azure.

Azure Marketplace has thousands of solutions that are made to extend directly into users' Azure infrastructure. Products in this storefront include Azure building blocks, digital cloud services, finished software solutions, and consulting and managed services. Products and services are organized by category and easy to filter, making it easy for customers to find the right app or service.

Your customers can access the Azure Marketplace online store by visiting https://azuremarketplace.microsoft.com.

Because Azure Marketplace is a part of Azure, they can also access these solutions through the Azure portal at https://portal.azure.com. Any time Azure users click "Create a resource" on the Azure portal homepage, they're accessing solutions from Azure Marketplace.

Both the online store and the Azure portal experiences help customers quickly find, try, buy, and deploy solutions to Azure online in minutes.

In addition to the cloud products and services available through both the online store and the portal, the Azure Marketplace online store also lists consulting and managed services offered by Microsoft partners to help customers in myriad areas including analytics, DevOps, migration, and security.

Microsoft AppSource

The Microsoft AppSource contains business software that extends directly into Azure, Dynamics 365, Microsoft 365, and the Power Platform. Microsoft AppSource also contains web applications, which are hosted on the cloud and accessed over the internet. Solutions are organized by category and industry and are easy to filter, making it simple for customers to find the right app or service.

Your customers can access the Microsoft AppSource online store by visiting https://appsource.microsoft.com.

AppSource apps are also available within Microsoft products by clicking “Get more apps” from the homepage in Dynamics 365, “Get apps” in Power BI, “Apps” in Teams, and “Insert>Get Add-ins” in Microsoft 365 products (excluding Teams).

In addition to the cloud products that are available through the online store and in-product experiences, the Microsoft AppSource also lists consulting services provided by Microsoft partners. AppSource consulting services are organized by industry, service type, location, and Microsoft product type.

Summary

Now that you have a better understanding of the online stores and the applications and services they contain, you're convinced that your future customers will be able to find your solutions when you choose to join the commercial marketplace. Now you turn your attention to learning more about the different ways you can list your products in Microsoft AppSource and Azure Marketplace.

Go to market with Microsoft

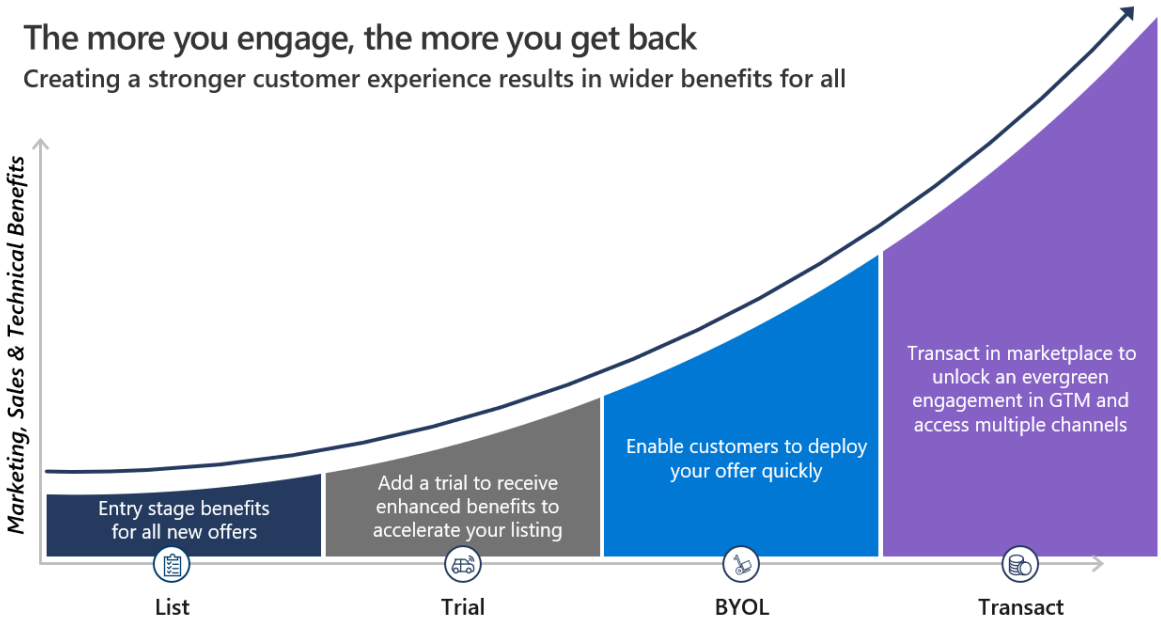
Now that you understand the different ways that customers can find, try, buy, and deploy your solutions through the Microsoft commercial marketplace, you start to read about Microsoft programs that will help accelerate your commercial marketplace offer’s growth. You find that your participation in the commercial marketplace unlocks go-to-market engagement and investment by Microsoft through a program called **Marketplace Rewards**.

What is Marketplace Rewards?

When you publish your offer to the commercial marketplace, Microsoft provides you access to sales, technical and marketing benefits. As a publisher, one of the ways your benefits are differentiated is based on the listing option you choose.

If you’re engaging customers through a “Contact Me” listing, Microsoft’s Marketplace Rewards team will reach out to provide guidance on how to increase the awareness of your solution.

If you’re engaging customers by offering a **free trial** listing, enabling deployments through a **BYOL** offer, or making sales through a **transact** listing, Microsoft will support your growth with increasing demand generation and sales activities.



You can also earn higher tiers of Marketplace Rewards benefits when your commercial marketplace offer hits certain revenue milestones, which are reached when customers buy your solution through Microsoft AppSource, Azure Marketplace, our network of resellers, and the myriad in-product experiences. Hitting these revenue tiers requires that you have a **transact** offer in the commercial marketplace.

The Marketplace Rewards program is designed to support you at your business' stage of growth, starting with awareness activities to help you get your first customers and shifting to more advanced sales and technical benefits as you sell more through Microsoft. These benefits are available to all partners with active commercial marketplace offers.

You can learn more about the benefits available through this commercial marketplace program by referencing the Marketplace Rewards program slide deck.

Azure Marketplace

- Think of it like an "Azure Shop" where you purchase services and solutions for the Azure platform
- Each product is a template which contains one or multiple services
- Products are delivered by first and third-party vendors
- Solutions can leverage all service categories like IaaS, PaaS and SaaS

Introduction to the Internet of Things

For many businesses, the Internet of Things (IoT) is new territory opened up by the cloud. It's a technology that enables a mass of devices, with *some* computational power, to connect to a single cloud process with a mass of computational power.

The amount of data that IoT devices can capture creates a significant challenge for IT operations. Businesses expect IT administrators to help them understand what data to collect, and to help them realize the full potential of that data.

Flatten the IoT learning curve

A "thing" in the IoT world is often a sensor device. The device has some processing power but isn't a computer that can compare with a smartphone, laptop, PC, or workstation. The sensor device typically takes some measurements (including temperature, velocity, acceleration, and humidity) at a specified time interval. The device then transmits the values for processing to the cloud. These values are called *telemetry*.

In another example, an IoT system might capture and run predictive analytics on data from health monitors. Doctors can then make informed treatment plans for their patients. And it's all handled remotely.

In addition to transmitting data, IoT devices can get some instructions from the cloud. For example, the time interval might be changed, or additional sensors might be enabled or disabled. For a more complex device, one that both records telemetry and changes settings, the device can receive *desired* values from the cloud. The device will then adjust its settings to try to match the desired values it has received. These values might be temperature, humidity, velocity, and related telemetry.

In the world of IoT devices, *recorded* values and *desired* values are key concepts. In this module, we'll test only recorded values.

Imagine a scenario where there are many sensor devices. For example, devices record weather data, devices record the temperature of the contents of refrigeration units, and devices record the vibration in a series of conveyor belts. All the telemetry data is transmitted to a single source for processing. That source is an Azure IoT hub. And you've been assigned the job as administrator of that hub!

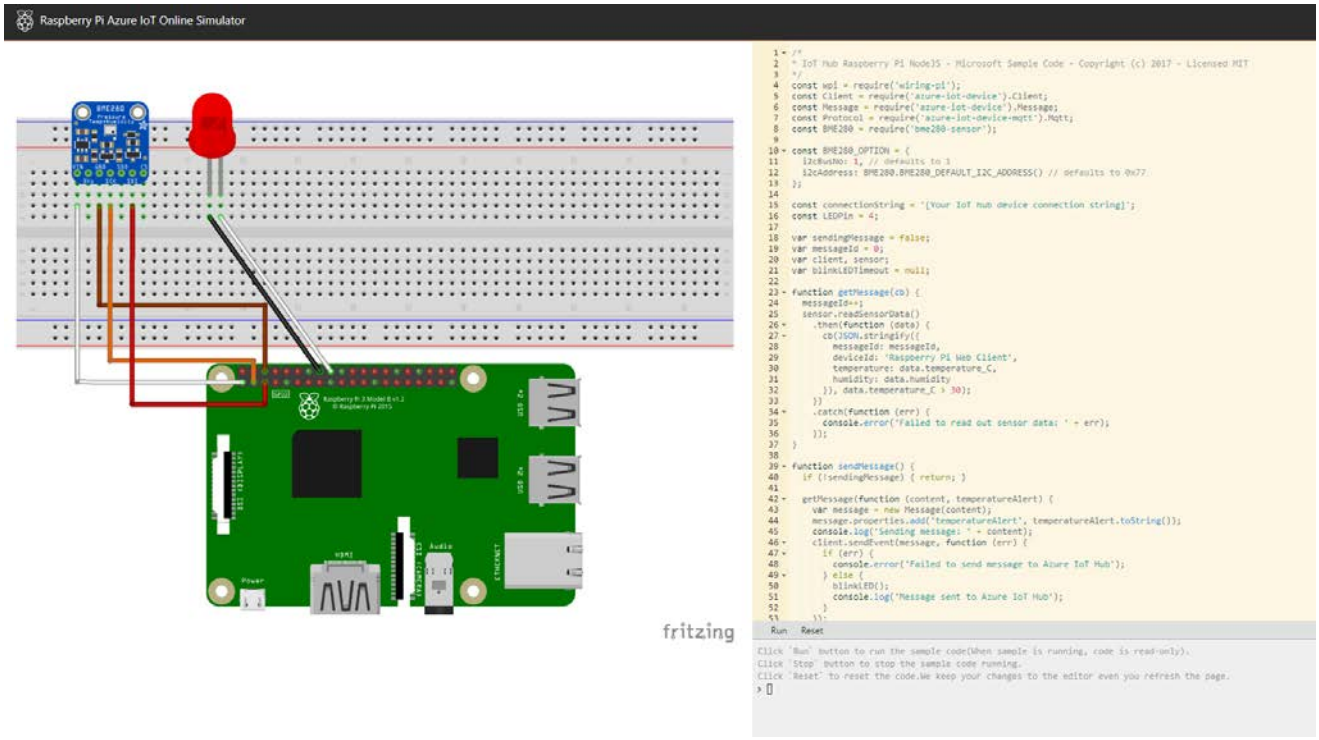
A popular device for getting experience with IoT is the Raspberry Pi device. This device is a small circuit board with some LED lights, some sensors (including for temperature), and the ability to communicate with an IoT hub. Some developers will want to physically connect the device and gain experience that way. A simpler alternative is to use Microsoft's Raspberry Pi simulator and do everything in software on your computer. This process is the approach we take in this module: you'll set up an IoT hub, define a single device, and then use the simulator to send telemetry to the hub.

Introduction to the Raspberry Pi simulator

Raspberry Pi boards are popular IoT devices for education, for testing ideas, and even for building some worthwhile systems. Although the cost of a board isn't prohibitive, we can test the Raspberry Pi functionality before investing in actual hardware.

Microsoft has built an online Raspberry Pi Azure IoT simulator . Users control the emulated hardware via code. The simulator shows a graphic of Raspberry Pi connected to a temperature, humidity, and pressure sensor, and a red LED. A breadboard allows circuits to be wired together. The displayed side panel enables users to enter Node.js code to control the LED, and to collect dummy data from the simulated sensor.

By default, the simulator operates a sample temperature-capture program, which is displayed via the command line. The same sample application can also be run on a real Pi device, because the simulator is designed to allow people to test code before transferring it to a real device.



The web simulator has three screen areas:

1. **Assembly area**. This area is where you can see your device status. By default, it shows a Pi device connecting with a BME280 sensor and an LED light.
2. **Coding area**. This area is an online code editor where you can make an app on Raspberry Pi by using Node.js. The default sample application helps collect sensor data from the BME280 sensor, and then sends it to your Azure IoT hub.
3. **Integrated console window**. This window is where you can see the output of your app. Within the console, there are three functions:
    - **Run** runs the sample code. When the sample is running, code is read-only.
    - **Stop** stops the sample code from running. You can select **Run** again to restart where you left off.
    - **Reset** resets the simulator to its starting state.

Now that you have an overview of the Raspberry Pi simulator, we'll explore the IoT hub in Azure where you'll create a new resource to capture data from the simulator.

Set up your IoT hub, then register and run a device

This module requires a sandbox to complete. A **sandbox** gives you access to Azure resources. Your Azure subscription will not be charged. The sandbox may only be used to complete training on Microsoft Learn. Use for any other reason is prohibited, and may result in permanent loss of access to the sandbox.

Due to the impact of the global health pandemic, Azure resources are being prioritized towards health and safety organizations. You may experience some issues when you deploy resources used in the exercises. Please try again or choose a different region. For more information, see Azure blog post - **Update #3: Business continuity with Azure**.

Activate sandbox

In this unit, you'll interact with the online Raspberry Pi simulator that you learned about in the previous unit.

Although this exercise is being conducted in a simulated environment, the app running on the simulated device is similar to the code that would be running on a real device. In the IoT world, device simulation is a valuable step in building a production-grade solution.

Administer your IoT hub

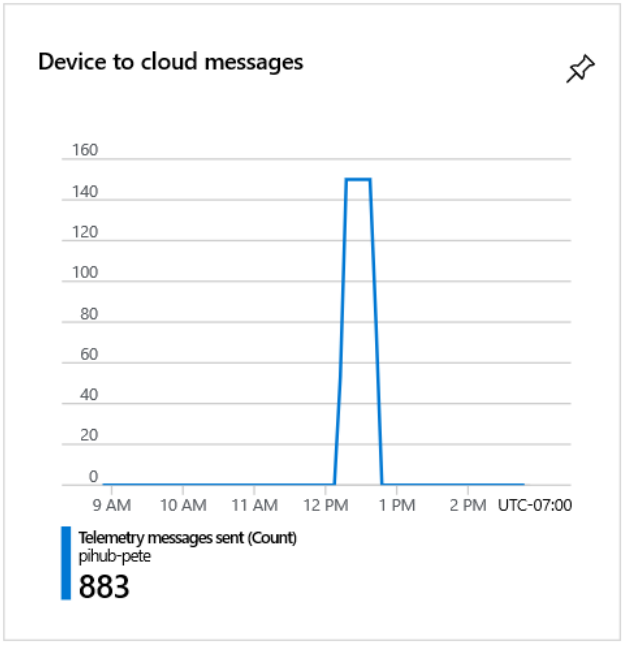
As an IoT admin, you need to get familiar with the left menu for your IoT hub. In this unit, we examine the most commonly used of these menu items. We don't examine them all, because some won't make sense until you have specific resources that will show up when you select these entries.

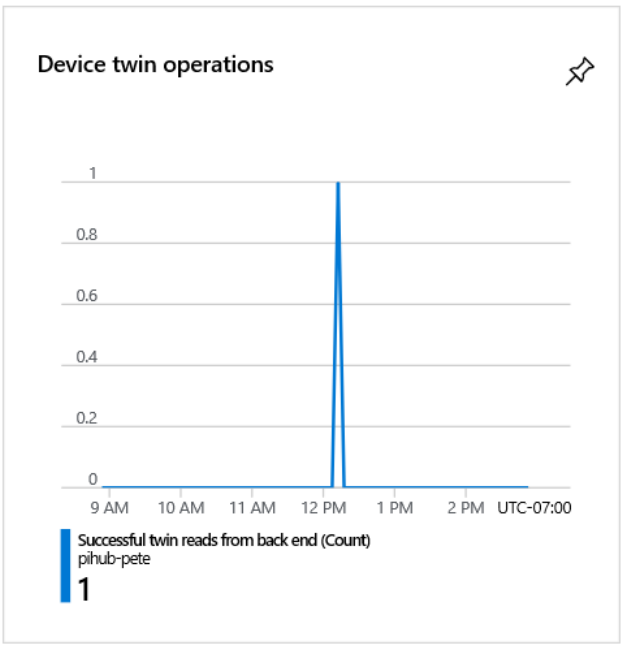
There's no need to try to learn everything at once. The purpose of this unit is to identify the menu items that are most valuable to you when you begin your adventure in IoT hub administration.

Explore the IoT hub menu

Select - Overview

The overview contains some useful data. If you scroll to the bottom of the page, you'll see two charts. By default, the **Telemetry messages sent** and **Successful twin reads from back end** metrics are displayed here. These charts are a good place to go to when you're verifying that the telemetry from a device is being received.





The overview, however, isn't the most useful of the menu entries, as we'll see as we explore the menu. You can select **Activity log**, though you might not understand why the entries are there. Logs often contain too much information! Also, select **Diagnose and solve problems**. This page is a resource to go to if problems arise. There's no guarantee that it will have all the information you need, but it's a good place to start.

Select - Settings

The important entry here is **Shared access policies**. You have already used this entry to extract the primary key for your hub. This entry is used to get the keys for all the policies that have been defined for the hub. You can create custom policies if needed.

The default list of policies is shown in the following image. Each policy has its own set of keys.

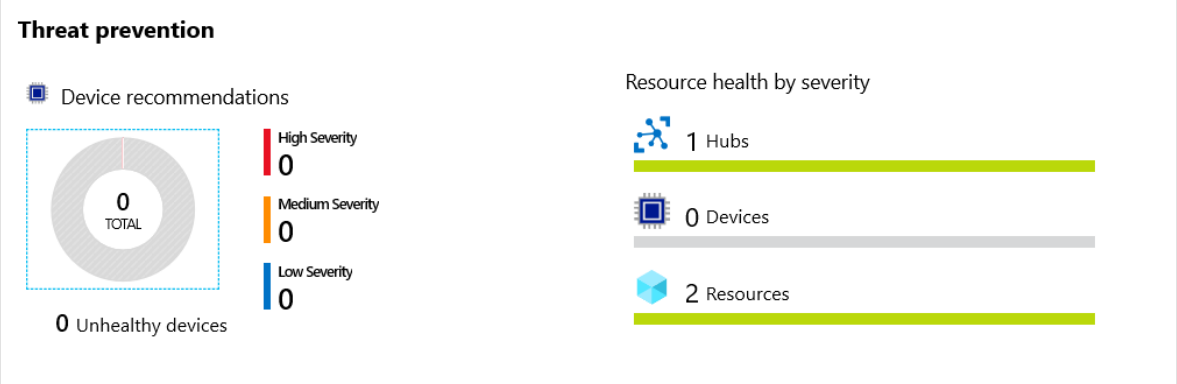
Another useful entry is **Properties**. Select this entry to bring up a range of strings that apply to your hub, including name, region, and subscription. This entry is a good point of reference. It's something to check if you get an unexpected error, for example. Also note that you can copy any of the strings by using the icon to the right.

Select - Explorers

The **IoT devices** entry, which you've already used, is one of the most important entries on the left menu. When you start adding multiple devices to a hub, you'll see how helpful this entry is in administering remote devices. One of the main uses of this entry is to locate the *device* connection strings and keys. Device keys are distinct from the *service* keys for the hub itself.

Select - Security

Select the **Overview** menu entry for this section. There will be nothing dramatic in this example, but you can see where to go for security recommendations when you have a production hub running.



Select - Monitoring

You have already used **Metrics**, in the previous unit. Another useful entry is **Alerts**. Alerts fire when certain conditions, usually errors or warnings, are met.

To get detailed experience with metrics, and to set up some alerts, add the following module to your to-do list: Manage your Azure IoT hub with alerts and metrics.

Summary

Azure IoT Hub provides the features, and an extensibility model, that enable developers to build robust remote device-management solutions. Devices range from constrained sensors and single-purpose microcontrollers to powerful gateways that route communications for groups of devices. Also, the use cases and requirements for IoT operators vary significantly across industries. Despite this variation, device management with IoT Hub provides the capabilities to cater to a diverse set of devices, users, and objectives.

Clean up

The sandbox automatically cleans up your resources when you're finished with this module.

When you're working in your own subscription, it's a good idea at the end of a project to identify whether you still need the resources you created. Resources left running can cost you money. You can delete resources individually or delete the resource group to delete the entire set of resources.

What is Big Data?

Big Data is a field of technology that helps with the **extraction**, **processing** and **analysis** of information that is **too large or complex** to be dealt with by traditional software.

The three V's rule

Big data typically has one of the following characteristics

- **Velocity** - how fast the data is coming in or how fast we are processing it
  - Batch
  - Periodic
  - Near Real Time
  - Real Time
- **Volume** - how much data we are processing
  - Megabytes
  - Gigabyte
  - Terabytes
  - Gigabytes
- **Variety** - how structured/complex the data is
  - Tables
  - Databases
  - Photo, Audio
  - Video, Social Media

What is Azure Synapse Analytics (formerly SQL DW)?

Azure Synapse is an analytics service that 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. Azure Synapse brings these two worlds together with a unified experience to ingest, prepare, manage, and serve data for immediate BI and machine learning needs.

Azure Synapse has four components:

- Synapse SQL: Complete T-SQL based analytics – Generally Available
  - SQL pool (pay per DWU provisioned)
  - SQL on-demand (pay per TB processed) (preview)
- Spark: Deeply integrated Apache Spark (preview)
- Synapse Pipelines: Hybrid data integration (preview)
- Studio: Unified user experience. (preview)

Synapse SQL pool in Azure Synapse

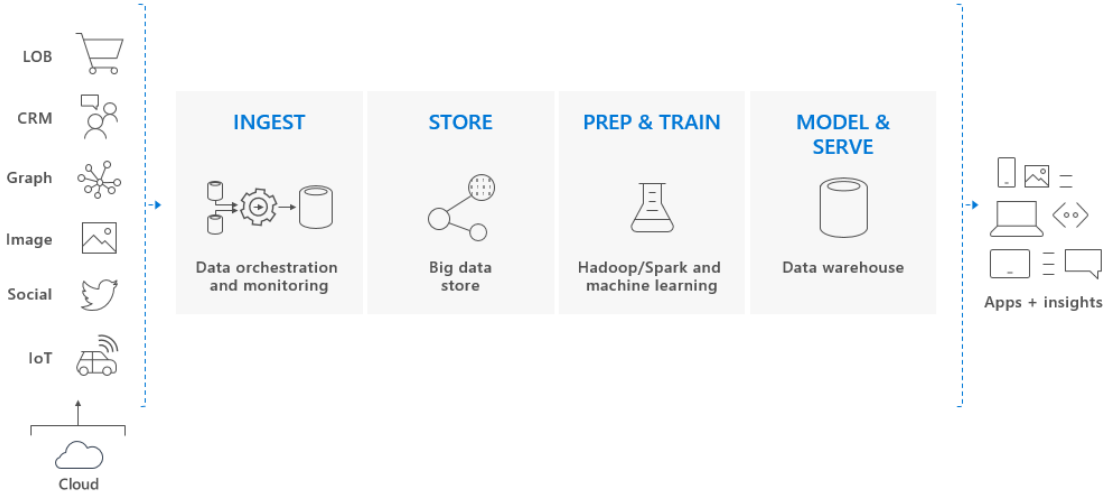
Synapse SQL pool refers to the enterprise data warehousing features that are generally available in Azure Synapse.

SQL pool represents a collection of analytic resources that are being provisioned when using Synapse SQL. The size of SQL pool is determined by Data Warehousing Units (DWU).

Import big data with simple PolyBase T-SQL queries, and then use the power of the distributed query engine to run high-performance analytics. As you integrate and analyze the data, Synapse SQL will become the single version of truth your business can count on for faster and more robust insights.

Key component of a big data solution

Data warehousing is a key component of a cloud-based, end-to-end big data solution.



In a cloud data solution, data is ingested into big data stores from a variety of sources. Once in a big data store, Hadoop, Spark, and machine learning algorithms prepare and train the data. When the data is ready for complex analysis, Synapse SQL pool uses PolyBase to query the big data stores. PolyBase uses standard T-SQL queries to bring the data into Synapse SQL pool tables.

Synapse SQL pool stores data in relational tables with columnar storage. This format significantly reduces the data storage costs, and improves query performance. Once data is stored, you can run analytics at massive scale. Compared to traditional database systems, analysis queries finish in seconds instead of minutes, or hours instead of days.

The analysis results can go to worldwide reporting databases or applications. Business analysts can then gain insights to make well-informed business decisions.

Azure Synapse Analytics

- Big data analytics platform (PaaS)
- Multiple components
    - Spark
    - Synapse SQL
        - SQL pools (dedicated – pay for provisioned performance)
        - SQL on-demand (ad-hoc – pay for TB processed)
    - Synapse Pipelines (Data Factory – ETL)
    - Studio (unified experience)

Azure HDInsight

- Flexible multi-purpose big data platform (PaaS)
- Multiple technologies supported (Hadoop, Spark, Kafka, HBase, Hive, Storm, Machine Learning)

Azure Databricks

- Big data collaboration platform (PaaS)
- Unified workspace for notebook, cluster, data, access management and collaboration
- Based on Apache Spark
- Integrates very well with common Azure data services

Azure Machine Learning studio = the web portal for data scientist developers in Azure Machine Learning. The studio combines no-code and code-first experiences for an inclusive data science platform.

Author machine learning projects

The studio offers multiple authoring experiences depending on the type project and the level of user experience.

- **Notebooks**

  Write and run your own code in managed Jupyter Notebook servers that are directly integrated in the studio.

- **Azure Machine Learning designer**

  Use the designer to train and deploy machine learning models without writing any code. Drag and drop datasets and modules to create ML pipelines. Try out the designer tutorial.

- **Automated machine learning UI**

  Learn how to create automated ML experiments with an easy-to-use interface.

- **Data labeling**

  Use Azure Machine Learning data labeling to efficiently coordinate data labeling projects.

Manage assets and resources

Manage your machine learning assets directly in your browser. Assets are shared in the same workspace between the SDK and the studio for a seamless experience. Use the studio to manage:

- Models
- Datasets
- Datastores
- Compute resources
- Notebooks
- Experiments
- Run logs
- Pipelines
- Pipeline endpoints

Even if you're an experienced developer, the studio can simplify how you manage workspace resources.

What is machine learning?

Machine learning is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends. By using machine learning, computers learn without being explicitly programmed.

Forecasts or predictions from machine learning can make apps and devices smarter. For example, when you shop online, machine learning helps recommend other products you might want based on what you've bought. Or when your credit card is swiped, machine learning compares the transaction to a database of transactions and helps detect fraud. And when your robot vacuum cleaner vacuums a room, machine learning helps it decide whether the job is done.

Machine learning tools to fit each task

Azure Machine Learning provides all the tools developers and data scientists need for their machine learning workflows, including:

- The Azure Machine Learning designer: drag-n-drop modules to build your experiments and then deploy pipelines.
- Jupyter notebooks: use our example notebooks or create your own notebooks to leverage our SDK for Python samples for your machine learning.
- R scripts or notebooks in which you use the SDK for R to write your own code, or use the R modules in the designer.
    - The Many Models Solution Accelerator (preview) builds on Azure Machine Learning and enables you to train, operate, and manage hundreds or even thousands of machine learning models.
- Visual Studio Code extension
- Machine learning CLI
- Open-source frameworks such as PyTorch, TensorFlow, and scikit-learn and many more
- Reinforcement learning with Ray RLlib

You can even use MLflow to track metrics and deploy models or Kubeflow to build end-to-end workflow pipelines.

Build ML models in Python or R

Start training on your local machine using the Azure Machine Learning Python SDK or R SDK. Then, you can scale out to the cloud.

With many available compute targets, like Azure Machine Learning Compute and Azure Databricks, and with advanced hyperparameter tuning services, you can build better models faster by using the power of the cloud.

You can also automate model training and tuning using the SDK.

Build ML models in the studio

Azure Machine Learning studio is a web portal in Azure Machine Learning for low-code and no-code options for model training, deployment, and asset management. The studio integrates with the Azure Machine Learning SDK for a seamless experience. For more information, see What is Azure Machine Learning studio.

- **Azure Machine Learning designer**

  Use the designer to train and deploy machine learning models without writing any code. Try the designer tutorial to get started.

- **Track experiments**

  Learn how to track and visualize data science experiments in the studio.

MLOps: Deploy & lifecycle management

When you have the right model, you can easily use it in a web service, on an IoT device, or from Power BI. For more information, see the article on how to deploy and where.

Then you can manage your deployed models by using the Azure Machine Learning SDK for Python, Azure Machine Learning studio, or the machine learning CLI.

These models can be consumed and return predictions in real time or asynchronously on large quantities of data.

And with advanced machine learning pipelines, you can collaborate on each step from data preparation, model training and evaluation, through deployment. Pipelines allow you to:

- Automate the end-to-end machine learning process in the cloud
- Reuse components and only rerun steps when needed
- Use different compute resources in each step
- Run batch scoring tasks

If you want to use scripts to automate your machine learning workflow, the machine learning CLI provides command-line tools that perform common tasks, such as submitting a training run or deploying a model.

Integration with other services

Azure Machine Learning works with other services on the Azure platform, and also integrates with open source tools such as Git and MLFlow.

- Compute targets such as **Azure Kubernetes Service**, **Azure Container Instances**, **Azure Databricks**, **Azure Data Lake Analytics**, and **Azure HDInsight**.
- **Azure Event Grid**.
- **Azure Monitor**.
- Data stores such as **Azure Storage accounts**, **Azure Data Lake Storage**, **Azure SQL Database**, **Azure Database for PostgreSQL**, and **Azure Open Datasets**.
- **Azure Virtual Networks**.
- **Azure Pipelines**.
- **Git repository logs**.
- **MLFlow**.
- **Kubeflow**.

Secure communications

Your Azure Storage account, compute targets, and other resources can be used securely inside a virtual network to train models and perform inference.

What is Serverless?

Serverless computing is cloud-hosted execution environment that allows customers to **run their applications** in the cloud while **completely abstracting underlying infrastructure**.

Azure Functions

- Serverless coding platform (Functions as a Service, FaaS)
- Designed for nano-service architectures and event-based applications
- Scales up and down very quickly
- Highly scalable
- Supports popular languages and frameworks (.NET & .NET Core, Java, Node.js, Python, PowerShell, etc.)

Azure Logic Apps

- Serverless enterprise integration service (PaaS)
- 200+ connectors for popular services
- Designed for orchestration of
    - business processes,
    - integration workflows for applications, data, systems and services
- No-code solution

Azure Event Grid

- Fully managed serverless event routing service
- Uses publish-subscribe model
- Designed for event-based and near-real time applications
- Supports dozen of built-in events from most common Azure services

What is DevOps?

DevOps is a set of practices that combine both development (**Dev**) and operations (**Ops**).

DevOps aims to **shorten the development life cycle** by providing **continuous integration** and **delivery** (CI/CD) capabilities while **ensuring high quality** of deliverables.

Azure DevOps

- **Collection of services** for building solutions using DevOps practices
- Services included
    - **Boards** – tracking work
    - **Pipelines** – building CI/CD workflows (build, test and deploy apps)
    - **Repos** – code collaboration and versioning with Git
    - **Test Plans** – manual and exploratory testing
    - **Artifacts** – manage project deliverables
- Extensible with **Marketplace** – over 1000 of available apps
- Evolved from **TFS** (Team Foundation Server), through **VSTS** (Visual Studio Team Services)

Azure DevTest Labs

- Service for creation of **sandbox environments** for developers/testers (PaaS)
- Quick setup of **self-managed virtual machines**
- **Preconfigured templates** for VMs
- Plenty of additional **artifacts** (tools, apps, custom actions)
- Lab **policies** (quotas, sizes, auto-shutdowns)
- **Share** and **automate** labs via custom images
- Premade plugins/API/tools for **CI/CD pipeline automation**

Azure management options

You can configure and manage Azure using a broad range of tools and platforms. There are tools available for the command line, language-specific Software Development Kits (SDKs), developer tools, tools for migration, and many others.

Tools that are commonly used for day-to-day management and interaction include:

- **Azure portal** for interacting with Azure via a Graphical User Interface (GUI)
- **Azure PowerShell** and **Azure Command-Line Interface** (CLI) for command line and automation-based interactions with Azure
- **Azure Cloud Shell** for a web-based command-line interface
- **Azure mobile app** for monitoring and managing your resources from your mobile device

Azure portal

The Azure portal is a public website that you can access with any web browser. Once you sign in with your Azure account, you can create, manage, and monitor any available Azure services. You can identify a service you're looking for, get links for help on a topic, and deploy, manage, and delete resources. It also guides you through complex administrative tasks using wizards and tooltips.

The dashboard view provides high-level details about your Azure environment. You can customize the dashboard by moving and resizing tiles, and displaying services you're interested in.

The portal doesn't provide any way to automate repetitive tasks. For example, to set up multiple VMs, you would need to create them one at a time by completing the wizard for each VM. This process makes the portal approach time-consuming and error-prone for complex tasks.

Azure PowerShell

Azure PowerShell is a module that you can install for Windows PowerShell or PowerShell Core, which is a cross-platform version of PowerShell that runs on Windows, Linux, or macOS. Azure PowerShell enables you to connect to your Azure subscription and manage resources. Windows PowerShell and PowerShell Core provide services such as the shell window and command parsing. Azure PowerShell then adds the Azure-specific commands.

For example, Azure PowerShell provides the `New-AzVM` command that creates a virtual machine for you inside your Azure subscription. To use it, you would launch PowerShell, install the Azure PowerShell module, sign in to your Azure account using the command `Connect-AzAccount`, and then issue a command such as:

PowerShellCopy

```
New-AzVM `
    -ResourceGroupName "MyResourceGroup" `
    -Name "TestVm" `
    -Image "UbuntuLTS" `
    ...
```

Creating administration scripts and using automation tools is a powerful way to optimize your workflow. You can automate repetitive tasks. Once a script is verified, it runs consistently, which can reduce errors. Another scripting environment is the Azure CLI.

Azure CLI

Azure CLI is a cross-platform command-line program that connects to Azure and executes administrative commands on Azure resources. *Cross-platform* means that it can be run on Windows, Linux, or macOS. For example, to create a VM, you would open a command prompt window, sign in to Azure using the command `az login`, create a resource group, then use a command such as:

Azure CLICopy

```
az vm create \
  --resource-group MyResourceGroup \
  --name TestVm \
  --image UbuntuLTS \
  --generate-ssh-keys \
  ...
```

Azure Cloud Shell

Azure Cloud Shell is an interactive, authenticated, browser-accessible shell for managing Azure resources. It provides the flexibility of choosing the shell experience that best suits the way you work, either Bash or PowerShell.

You can switch between the two shells, and both support the Azure CLI and Azure PowerShell module. Bash defaults to the Azure CLI (with the `az` command pre-installed), but you can switch to PowerShell Core within Linux by typing `pwsh`. The PowerShell environment has both CLI tools pre-installed. In addition to these administrative tools, the Cloud Shell has a suite of developer tools, text editors, and other tools available, including:

Developer Tools :

- .NET Core
- Python
- Java
- Node.js
- Go

Editors : code (Cloud Shell Editor), vim, nano, emacs

Other tools : git, maven, make,. npm

You can create, build, and deploy apps right from this browser-based environment. It's all persistent as well - you're prompted to create an Azure Storage Account when you access the Azure Cloud Shell. This storage area is used as your $HOME folder and any scripts or data you place here is kept across sessions. Each subscription has a unique storage account associated with it, so you can keep the data and tools you need specific to each account you manage.

We'll use the Cloud Shell in Microsoft Learn for many of the interactive exercises to try out Azure features.

Azure mobile app

The Microsoft Azure mobile app allows you to access, manage, and monitor all your Azure accounts and resources from your iOS or Android phone or tablet. Once installed, you can:

- Check the current status and important metrics of your services
- Stay informed with notifications and alerts about important health issues
- Quickly diagnose and fix issues anytime, anywhere
- Review the latest Azure alerts
- Start, stop, and restart virtual machines or web apps
- Connect to your virtual machines
- Manage permissions with role-based access control (RBAC)
- Use the Azure Cloud Shell to run saved scripts or perform ad hoc administrative tasks
- and more...

Other options

There are also Azure SDKs for a range of languages and frameworks, and REST APIs that you can use to manage and control Azure resources programmatically. For a full list of tools available, see the Downloads page.

When starting with Azure, you'll most often use the Azure portal. Let's take a closer look at the portal approach.

Navigate the portal

With an Azure account, we can sign into the **Azure portal**. The portal is a web-based administration site that lets you interact with all of your subscriptions and resources you have created. Almost everything you do with Azure can be done through this web interface.

Azure portal layout

The Azure portal is the primary graphical user interface (GUI) for controlling Microsoft Azure. You can carry out the majority of management actions in the portal, and it is typically the best interface for carrying out single tasks or where you want to look at the configuration options in detail.

Resource panel

In the left-hand sidebar of the portal is the resource panel, which lists the main resource types. Note that Azure has more resource types than just those shown. The resources listed are part of your *favorites*.

You can customize this with the specific resource types you tend to create or administer most often.

The remainder of the portal view is for the specific elements you are working with. The default (main) page is **Home** but you can change your default view to the customizable **Dashboard** from **Settings**. We'll cover settings later in this unit.

Configuring settings in the Azure portal

The Azure portal displays several configuration options, mostly in the status bar at the top-right of the screen.

If you are viewing the Azure portal on a screen with reduced horizontal space, the following icons may be made available through an ellipsis (**...**) menu.

Cloud Shell

If you select the **Cloud Shell** icon (>_), you create a new Azure Cloud Shell session. Recall that Azure Cloud Shell is an interactive, browser-accessible shell for managing Azure resources. It provides the flexibility of choosing the shell experience that best suits the way you work. Linux users can opt for a Bash experience, while Windows users can opt for PowerShell. This browser-based terminal lets you control and administer all of your Azure resources in the current subscription through a command-line interface built right into the portal.

Directory and subscription

Select the **Book and Filter** icon to show the **Directory + subscription** pane.

Azure allows you to have more than one subscription associated with one directory. On the **Directory + subscription** pane, you can change between subscriptions. Here, you can change your subscription or change to another directory.

Notifications

Selecting the bell icon displays the **Notifications** pane. This pane lists the last actions that have been carried out, along with their status.

Settings

Select the **gear** icon to change the Azure portal settings. These settings include:

- Inactivity sign out delay
- Default view when you first sign in
- Flyout or docked option for the portal menu
- Color and contrast themes
- Toast notifications (to a mobile device)
- Language and regional format

Profile settings

If you select on your name in the top right-hand corner, a menu opens with a few options:

- Sign in with another account, or sign out entirely
- View your account profile, where you can change your password

Select the "..." button on the right-hand side for options to:

- Check your permissions
- View your bill
- Update your contact information

If you select "..." and then **View my bill**, Azure takes you to the **Cost Management + Billing - Invoices** page, which helps you analyze where Azure is generating costs.

Azure is a large product, and the Azure portal user interface (UI) reflects this scope. The sliding pane approach allows you to navigate back and forth through the various administrative tasks with ease. Let's experiment a bit with this UI so you get some practice.

Azure Advisor

Finally, the Azure Advisor is a free service built into Azure that provides recommendations on high availability, security, performance, operational excellence, and cost. Advisor analyzes your deployed services and looks for ways to improve your environment across those areas. You can view recommendations in the portal or download them in PDF or CSV format.

With Azure Advisor, you can:

- Get proactive, actionable, and personalized best practices recommendations.
- Improve the performance, security, and high availability of your resources as you identify opportunities to reduce your overall Azure costs.
- Get recommendations with proposed actions inline.

You can access Azure Advisor by selecting **Advisor** from the navigation menu, or search for it in the **All Services** menu.

View resources

1. Under Azure Marketplace, select **Compute** to show more compute options on the right side of the pane, such as Red Hat Enterprise, Reserved VM Instances, Web app for Containers, and so on.
2. To the right of **Featured**, select **See all**. The full list of available VM services now appears.
3. Select **Windows Server** under the **Operating Systems** section. On screens with limited horizontal space for the pane, you may have to scroll right and select the **See More** link to find the **Windows Server** option.
4. Select the drop-down list to see all of the Windows Server images available.
5. Select the **X** at the top right-hand corner to close the **Windows Server** window.
6. Select the **X** on the previous **Marketplace** window. You should now see the **New** pane again.

Filter results

Another way to locate services is to refine the list with filters and search terms. On the **New** pane, you may have noticed the search box at the top. Searching is the quickest way to filter what services you see.

This search defaults to checking every Azure service category to get its results. Next, you'll filter after selecting a category.

1. Type `virtual machine` into the search box and select `Enter`.
2. Select **Compute**. You see a filtered list of Compute services related to virtual machine images.
3. Select any of the results that interest you to learn more about that service, including how to get started. Select the **X** in the corner to explore a different service. When you're done, move to the next step.
4. Select the **X** in the right-hand end of the search box. The **X** button will erase your search term but does not reset any of the drop-down filters you've set. You can either reset those filters manually, or close the **Marketplace** pane with the **X** icon in the upper right corner and reopen it. When you are finished trying out the search and filtering options, move on to the next step.
5. Select the **X** at the top right-hand corner to close the **Marketplace** pane. Now you will see the **New** pane once again.
6. Select the **X** at the top right-hand corner to close the **New** pane.

Many of the principles you have learned in this exercise apply throughout the Azure portal UI experience. In the next unit, you continue your journey in the Azure portal and configure additional settings in Azure.

The Azure portal has several features and services available; let's look at some of the more common areas you'll tend to use. First, take a moment to hover your mouse pointer over each of the icons in the top menu bar for a few seconds each. You should see a tooltip label pop-up for each one. This label is the name of the menu item. You will use these icons later.

All services

1. On the top left-hand side of the Azure portal, select **Show portal menu**.
2. Select **All services**. Take a couple of minutes to look through the list to see how many services Azure offers.
3. You can search for services through the *Search All* box.
4. Select **Virtual machines**. If you don't see it, use the search box. The **Virtual Machines** pane appears. You haven't created any virtual machines so there are no results.
5. Select **+ Add** > **Virtual machine**. The **Create a virtual machine** pane appears.
6. Select the **X** in the top right-hand corner to close the **Create a virtual machine** pane.
7. Select the **X** in the top right-hand corner to close the **Virtual machines** pane.
8. Select on **Microsoft Azure** on the top left-hand side to get back to the home page.

Azure Cloud Shell

The Azure Cloud Shell allows you to use a command-line interface (CLI) to execute commands in your Azure subscription. You can access it by selecting the (>_) icon in the toolbar. You can also navigate to https://shell.azure.com to launch a Cloud Shell in the browser independent of the portal.

The Azure Cloud Shell is available in the Sandbox environment, but the Sandbox version of the shell has reduced functionality. To use all of the Azure Cloud Shell features, use your own Azure subscription.

When you launch the shell, you see a Welcome window. You can choose either a **Bash** or **PowerShell** environment, depending on your personal preferences. You can also change the shell at any time through the language drop-down on the left side of the shell.

Finally, there are a variety of management and programming tools included in the created environment.

- Azure command-line tools (Azure CLI, AzCopy, etc.)
- Languages / Frameworks including .NET Core, Python, and Java
- Container management support for Docker, Kubernetes, etc.
- Code editors such as vim, emacs, code, and nano
- Build tools (make, maven, npm, etc.)
- Database query tools such as `sqlcmd`

Directory and subscription

1. Select the **Directory + Subscription** (book and filter) icon to show the **Directory + subscription** pane.

   This is where you can switch between multiple subscriptions or directories. You should see that you are in the Concierge Subscription of the Microsoft Learn Sandbox directory here. If you have other Azure directories tied to the same email address, those subscriptions will be available as well.

2. Select the **X** in the top right-hand corner to close the **Directory + subscription** pane.

Notifications pane

1. On the icon bar menu bar, select the **Notifications** (bell) icon. This window lists any pending notifications.
2. If any notifications appear, hover your mouse over one of them. Select the **X** that appears in that notification to dismiss it.
3. Select **Dismiss all**. You should have no notifications showing.
4. Select the **X** in the top right-hand corner to close the **Notifications** pane.

Settings

1. Select the **Settings** (cog) icon to open the **Portal settings** pane, showing the **General** settings by default.
2. Drop down the **Sign me out when inactive** setting, and select **After one hour**.
3. Under **Choose a theme**, select the different colored themes and observe the changes to the portal UI. Leave it set to the one you like the best.
4. Under **High contrast theme**, try the three different options.
5. Select **Enable pop-up notifications**. When this option is checked, notifications will appear as pop-up "toast"-style notifications. They will still show up in the Notifications (bell) icon as well.
6. Select the **Language & region** tab in the settings. Select **Language** and pick **Español**, and then select the **Apply** button. If a **Translate this page** dialog box appears, close the box. The whole portal is now in Spanish.
7. To revert back to English, select the **Settings** (cog) icon in the top menu bar and switch to the **Idioma y región** settings. Select **Idioma** and pick **English**. Select the **Aplicar** button. The portal returns to English.

Profile settings

1. Select on your name in the top right-hand corner of the portal. Options include:
   - Sign in with another account, or sign out entirely
   - View your account profile, where you can change your password
   - Submit an idea
   - Check your permissions
   - View your bill
   - Update your contact information

   Some of these items do not appear unless you select the "..." icon.

2. Select "..." then **View my bill** to navigate to the **Cost Management + Billing - Invoices** page, which helps you analyze where Azure is generating costs.
3. If you're using your own account and not sandbox, you can select a subscription from the drop-down list.
4. Select a billing period.
5. Note the service costs and check them against what you expect for your current subscription.
6. Select the **X** in the top right-hand corner to close the **Costs by service** pane.
7. Select the **X** in the top right-hand corner to close the **Cost Management + Billing - Invoices** page.
8. You should now be back on the home page.

Now that we've explored all the main areas of the Azure portal, let's look at one of the most useful features - Dashboards.

Azure Portal dashboards

Let's look at how to create and modify dashboards using the Azure portal, and by editing the underlying JSON file directly. In this unit, you'll learn to navigate around the portal. And in the next unit, you will try out the things you've learned.

What is a dashboard?

A *dashboard* is a customizable collection of UI tiles displayed in the Azure portal. You add, remove, and position tiles to create the exact view you want, and then save that view as a dashboard. Multiple dashboards are supported, and you can switch between them as needed. You can even share your dashboards with other team members.

Dashboards give you considerable flexibility regarding how you manage Azure. For example, you can create dashboards for specific roles within the organization, and then use role-based access control (RBAC) to control who can access that dashboard. Hence, your database administrator would have a dashboard that contains views of the SQL database service, whereas your Azure Active Directory administrator would have views of the users and groups within Azure AD. You can even customize the portal between your production and development environments within the portal - creating a specific dashboard for each environment you are managing.

Dashboards are stored as JavaScript Object Notation (JSON) files. This format means they can be uploaded and downloaded to other computers, or shared with members of the Azure directory. Azure stores dashboards within resource groups, just like virtual machines or storage accounts that you can manage within the portal.

Because dashboards are JSON files, you can also customize them programmatically, making them compelling administrative tools. Also, some tile types can be query-based, so they update automatically when the source data changes.

Explore the default dashboard

The default dashboard is named "Dashboard". When you log into the portal for the first time and select **Dashboard** from the portal menu, you are presented with this dashboard containing five tiles.

These default web parts are

1. Dashboard controls
2. All resources tile
3. Quickstarts + tutorials tile
4. Service Health tile
5. Marketplace tile

Creating and managing dashboards

At the top of the dashboard are the controls that enable you to create, upload, download, edit, and share a dashboard. You can also switch a dashboard to full screen, clone it, or delete it.

Select dashboard

To the far left of the toolbar is the **Select Dashboard** drop-down control. Clicking this control enables you to select from dashboards that you have already defined for your account. This control makes it simple for you to define multiple dashboards for different purposes and then switch from one to another and back again, depending on what you are trying to do at the time.

Dashboards that you create will initially be private; that is, only you can see them. To make a dashboard available across your enterprise, you need to share it. We'll look at that option shortly.

Create a new dashboard

To create a new dashboard, click **New dashboard**. The dashboard workspace appears, with no tiles present. You can then add, remove, and adjust tiles however you like. When you are finished customizing the dashboard, click **Done customizing** to save and switch to that dashboard.

Upload and Download

The **Upload** and **Download** buttons enable you to download your current dashboard as a JSON file, customize it, and then distribute it and upload it or have someone else upload that file back to the Azure portal, thereby replacing their current dashboard.

If you click **Download**, the current dashboard downloads the JSON code as a file you can edit locally. You can then upload it back to Azure by clicking the **Upload** button. Downloading and uploading dashboards is discussed further below.

Edit a dashboard using the portal

Although you can edit a dashboard by downloading the JSON file, changing values in the file, and uploading the file back to Azure, you may prefer a graphical approach to designing the user interface. To use the GUI to configure your current dashboard, you can switch to editing mode in several ways:

- Click the **Edit** (pencil icon) button.
- Right-click on the dashboard background area and select **Edit**.
- Right-click on a tile and a menu will appear with edit options.
- Hover over a tile on the dashboard - a `...` menu will appear on the top/right corner with a **Customize** option.

The dashboard will switch to edit mode.

On the left-hand side appears the **Tile Gallery**, with several possible tiles. You can filter the Tile Gallery by category and resource type.

Adding tiles is as easy as selecting the tile from the list on the left and then dragging it to the work area. You can then move each tile about, resize it, or change the data that it displays.

Tip

One cool feature is that you can take elements on child panes and put them on your dashboard. Just hover over the item and look for the `...` tile edit menu - this will have a "Pin to Dashboard" option which lets you quickly grab a tile from a service and put it onto the dashboard.

The work area in edit mode is divided into squares. Each tile must occupy at least one square, and tiles will snap to the nearest largest set of tile dividers. Any overlapping tiles are moved out of the way. When you make a tile smaller, the surrounding tiles will move back up against it.

Change tile sizes

Some tiles have a set size, and you can edit their size only programmatically. However, you can edit tiles with a gray bottom right-hand corner by dragging the corner indicator.Alternatively, right-click into the contextual menu and specify the size you want.To create your dashboard, pull tiles from the Tile Gallery onto the workspace and then rearrange them.

Change tile settings

Some tiles have editable settings. For example, with the clock tile, when you drag it onto the workspace, it opens the **Edit clock** tile. You can then set the time zone, which it displays, and also set whether it displays in 12- or 24-hour format.For multi-national or transcontinental companies, you can add several clocks, each in a different time zone.

Accepting your edits

When you have arranged the tiles as you want them, either click **Done customizing** or right-click and then click **Done customizing**.

Edit a dashboard by changing the JSON file

You can also edit a dashboard by changing the JSON file. This approach provides more options for changing settings, but you cannot see the changes until you upload the file back into Azure. The easiest starting point is to download the dashboard JSON as previously described and edit that file.

As an example, in the JSON shown above, to change the size of the tile you would edit the **colSpan** and **rowSpan** variables, then save the file and upload it back to Azure.

Tip : You can also distribute the dashboard JSON file to other users.

Reset a dashboard

You can reset any dashboard to the default style. In edit mode, right-click the dashboard background and select **Reset to default state**. A dialog box will ask you to confirm that you want to reset that dashboard.

Share or unshare a dashboard

When you define a new dashboard, it is private and visible only to your account. To make it visible to others, you need to share a dashboard. However, as with any other Azure resource, you need to specify a new resource group (or use an existing resource group) in which to store shared dashboards. If you do not have an existing resource group, Azure will create a *dashboards* resource group in whichever location you specify. If you have existing resource groups, you can specify that resource group to store the dashboards.

When you have shared the template, you will see a second **Sharing + access control** pane.

You can then click **Manage users** to specify the users who have access to that dashboard.

Switching to a shared dashboard

To switch to a shared dashboard, you click on the list of dashboards, and then click **Browse all dashboards**.

You will now see the **All dashboards** pane, with the names of any shared dashboards displayed. Just click on a dashboard to apply it to the Azure portal.

Display a dashboard as a full screen

If you want the largest dashboard real estate, click the **Full screen** button to display your current dashboard without any browser menus. If you have any tiles outside the boundaries of your screen display, slider bars will appear at the right and bottom of your screen.

When you have finished working in full-screen mode, press the ESC key or click **Exit Full Screen** next to the Dashboard name at the top of the screen.

Clone a dashboard

Cloning a dashboard creates an instant copy called "Clone of <dashboard name>" and switches to that copy as the current dashboard. Cloning is also an easy way to create dashboards before sharing them. For example, if you have a dashboard that is almost as you want it, clone it, make the changes that you need, and then share it.

Delete a dashboard

Deleting a dashboard removes it from your list of available dashboards. You are prompted to confirm that you want to delete the dashboard, but there is no facility to recover a dashboard that has been deleted.

Let's try out some of these options by creating a new dashboard.

Dashboards are a flexible tool for managing different aspects of Azure services through the Portal. They make it convenient to monitor the state of your services. Because they are shareable, they help ensure that everyone on your team sees the same data and stays aware of the state of your critical components. Let's create a new dashboard and add some tiles to it.

Create a new dashboard

1. In the Azure portal , from the top left-hand side, select **Show portal menu** > **Dashboard**.
2. Select the **New Dashboard**.
3. In the center pane, change **My Dashboard** to `Customer Dashboard`.

Add and configure the Clock Tile

1. In the tile gallery, drag the clock onto the workspace. Place it on the top right of the available space.
2. On the **Edit clock** pane, change the Location to **Pacific Time (US & Canada)**.
3. Under **Time format**, select **24 hour**.
4. Select **Done**.
5. Repeat the preceding four steps, except select **Eastern Time (US & Canada)**. You should now have two clocks, one showing the time on the West Coast, the other on the East Coast.

Resize a tile

1. Under **Tile Gallery**, drag an **All resources** tile and drop it onto the top left-hand side of the new dashboard workspace.
2. Hover over the new **All resources** tile and select the ellipsis icon (**...**); then select the **6x6** size.
3. Select the gray corner on the bottom right-hand side of the tile, and resize the tile to 3.5 squares vertically by six horizontally. When you finish resizing, the tile adjusts to 4x6.
4. In the Tile Gallery, drag the **Resource Groups** tile onto the workspace. Place it underneath the **All resources** tile.
5. In the Tile Gallery, select the **Metrics chart** tile, and drag it onto the workspace. Place it to the right of the **All resources** tile.
6. Continue to add the following tiles, rearranging them to fit:
    - Help + support
    - Quick Tasks

    - Marketplace
7. When you have added these tiles, select **Done customizing**. The **Customer Dashboard** dashboard should appear.

Clone a dashboard

You now want to create a similar dashboard for some other customers.

1. Select the **Clone** button.
2. Rename the dashboard from **Clone of Customer Dashboard** to **Azure AD Admin Dashboard**.
3. On the **Resource Groups** tile, select the **Remove from dashboard** trash can icon to delete this tile.
4. From the Tile Gallery, add the following tiles:
    - Organization Identity
    - Users and Groups
    - User Activity Summary
5. Reposition the tiles as necessary, and then select **Done customizing**.

Share a dashboard

You now want to make this dashboard available to other users. In the sandbox environment, you won't be able to publish a shared dashboard. But you can see how you'd share a dashboard by completing the following steps.

1. From the Azure AD Admin dashboard, select the **Share** button at the top. The **Sharing and access control** panel that appears.
2. To publish to a specific resource group, uncheck the **Publish to the 'dashboards' resource group** checkbox.
3. Select the resource group [sandbox resource group name] from the **Resource group** dropdown.
4. Select **Publish**.
5. At this point in the sandbox environment, you'll receive an error. That's ok.
6. Close the **Sharing + access control** pane.

Edit a dashboard.json file

To show how you can download and edit a dashboard file, carry out the following steps:

1. Select **Download**.
2. Open a file explorer on your computer and navigate to where your web browser downloaded the dashboard, typically a **Downloads** folder.
3. Find the *Customer Dashboard.json* file and open it in a text editor.
4. In your editor, look for the text *ClockPart*.
5. On the first occurrence of ClockPart, change the previous **position** > **rowSpan** value to 1.
6. On the second occurrence of ClockPart, also change the previous **position** > **rowSpan** value to 1.
7. On the second occurrence of ClockPart, change the **position** > **y** value from 2 to 1.
8. Save the *Customer Dashboard.json* file and close your code editor.
9. On the Azure dashboard, select **Upload**.
10. In the **Open** dialog box, browse to the Downloads folder, and double-click *Customer Dashboard.json*.

    The clocks have resized to one row high, and the bottom clock has moved up one row.

Delete a dashboard

1. Ensure that the **Azure AD Admin** dashboard is selected.
2. Select the **Delete** button.

3. In the **Confirmation** message box, select the checkbox to confirm that this dashboard will no longer be visible, and then select **OK**.

Reset a dashboard

1. Ensure that **Customer Dashboard** is selected.
2. Select **Edit**.
3. Right-click on the workspace, and select **Reset to default state**.
4. In the **Reset dashboard to default state** message box, select **Yes**.

   The Customer Dashboard has reset to its default tiles.

5. Select **Done customizing**.
6. Select your name at the top right of the portal.
7. Select **Sign out**.
8. Close your browser.

Congratulations! You have now created and edited dashboards, shared them, altered them as **JSON** files, and finally, reset them to the default state. You should now be able to see what powerful tools dashboards can be and how you can use them to create efficient interfaces for differing roles within an organization.

Access public and private preview features

Microsoft offers previews of Azure features for evaluation purposes. With *Azure Preview Features*, you can test beta and other pre-release features, products, services, software, and regions.

Some of the common areas you will see previews for include:

- New storage types
- New Azure services, such as Machine Learning enhancements
- New or enhanced integration with other platforms
- New APIs for services

Azure feature previews are available under certain terms and conditions that are specific to each particular Azure preview. Also, some previews are not covered by customer support.

Once a feature has been evaluated and tested successfully, it might be released to customers as part of Azure's default product set. This release is referred to as **General Availability** (GA).

Feature preview categories

There are two types of previews available:

- **Private Preview**. An Azure feature marked "private preview" is available to *specific* Azure customers for evaluation purposes. This is typically by invite only and issued directly by the product team responsible for the feature or service.
- **Public Preview**. An Azure feature marked "public preview" is available to *all* Azure customers for evaluation purposes. These previews can be turned on through the preview features page as detailed below.

Finding preview features

You can learn about preview features through the preview features page . This page lists the preview features that are available for evaluation. To access a preview feature, select its entry on this page and learn more about how to evaluate it. You can also use the **RSS Feed** button on this page to subscribe to notifications and stay informed.

You can also find Azure preview features in the portal as follows:

- Sign in to Azure portal.
- Select **Create a resource** in the resources panel to open the **New** pane.
- Enter the word *preview* into the search box at the top of the **New** pane.
- A list of available preview features is displayed, with the word **(preview)** next to each one.

Typical portal preview features provide performance, navigation, and accessibility improvements to the Azure portal interface. It will be branded with **Microsoft Azure (Preview)** in the top bar, so you will know you are in the preview portal.

Azure PowerShell

- PowerShell and module
- Designed for automation
- Multi-platform with PowerShell Core
- Simple to use
    - Connect-AzAccount – log into Azure
    - Get-AzResourceGroup – list resource groups
    - New-AzResourceGroup – create new resource group
    - New-AzVm – create virtual machine

Azure CLI

- Command Line Interface for Azure
- Designed for automation
- Multi-platform (Python)
- Simple to use
    - az login – log into Azure
    - az group list – list resource groups
    - az group create – create new resource group
    - az vm create – create virtual machine
- Native OS terminal scripting

Azure Cloud Shell

- Cloud-based scripting environment
- Completely free
- Supports both Azure PowerShell and Azure CLI
- Dozen of additional tools
- Multiple client interfaces
    - Azure Portal integration (portal.azure.com)
    - Shell Portal (shell.azure.com)
    - Visual Studio Code Extension
    - Windows Terminal
    - Azure Mobile App
    - Microsoft Docs integration

What is Azure Advisor?

Azure Advisor is a free service built into Azure that provides recommendations on high availability, security, performance, operational excellence, and cost. Advisor analyzes your deployed services and looks for ways to improve your environment across each of these areas. We'll focus on the cost recommendations, but you'll want to take some time to review the other recommendations as well.

Advisor makes cost recommendations in the following areas:

1. **Reduce costs by eliminating unprovisioned Azure ExpressRoute circuits.** This recommendation identifies ExpressRoute circuits that have been in the provider status of *Not Provisioned* for more than one month. Advisor recommends deleting the circuit if you aren't planning to provision the circuit with your connectivity provider.
2. **Buy reserved instances to save money over pay-as-you-go.** Advisor will review your virtual machine usage over the last 30 days and determine if you could save money in the future by purchasing reserved instances. Advisor will show you the regions and sizes where you potentially have the most savings and will show you the estimated savings you might achieve from purchasing reserved instances.
3. **Right-size or shutdown underutilized virtual machines.** This analysis monitors your virtual machine usage for 14 days and then identifies underutilized virtual machines. Virtual machines whose average CPU utilization is 5 percent or less and network usage is 7 MB or less for four or more days are considered underutilized virtual machines. The average CPU utilization threshold is adjustable up to 20 percent. By identifying these virtual machines, you can decide to resize them to a smaller instance type, reducing your costs.

Note

If you don't have an Azure subscription, create a **free account** before you begin.

Let's take a look at where you can find Azure Advisor in the portal.

1. Sign in to the Azure portal using your Microsoft account.
2. Expand the left-hand navigation from the top-left menu and click on **All Services**.
3. Click on the **Management + governance** category and find **Advisor**. You can also type Advisor in the services filter box to filter on just that name.
4. Click on Advisor, and you'll be taken to the Advisor recommendations dashboard where you can see all the recommendations for your subscription. You'll see a box for each category of recommendations.

Note

You might not have any recommendations on cost in Advisor. Assessments may not have completed yet or Advisor may not have recommendations.

Clicking on the **Cost** box will take you to detailed recommendations where you can see the recommendations that Advisor has.

Clicking on any recommendation will take you to the details for that specific recommendation. Then you'll be able to take a specific action, such as resizing virtual machines to reduce spending.

These recommendations are all places where you might be inefficiently spending money. They're a great place to start and continue to revisit when looking for places to reduce costs. In our example,

there's an opportunity for us to save around $700 per month if we take these recommendations. This savings adds up, so be sure to review these recommendations periodically across all areas.

Azure Cost Management

Azure Cost Management is another free, built-in Azure tool that can be used to gain greater insights into where your cloud money is going. You can see historical breakdowns of what services you are spending your money on and how it is tracking against budgets that you have set. You can set budgets, schedule reports, and analyze your cost areas.

As you can see, Azure offers tools at no additional cost that you can use to track and predict your cloud spend and identify where your environment may be inefficient from a cost perspective. You'll want to make sure you make it a regular practice to review the reports and recommendations that these tools make available, so you can unlock savings across your cloud footprint.

Azure Advisor

- **Personalized consultant** service
- Designed to provide **recommendations** and **best practices** for
    - **Cost** (SKU sizes, idle services, reserved instances, etc.)
    - **Security** (MFA settings, vulnerability settings, agent installations, etc.)
    - **Reliability** (redundancy settings, soft delete on blobs, etc.)
    - **Performance** (SKU sizes, SDK versions, IO throttling, etc.)
    - **Operational** Excellence (service health, subscription limits, etc.)
- **Actionable** recommendations
- **Free**!

Deploy your site to Azure

Your first step will likely be to re-create your on-premises configuration in the cloud.

This basic configuration will give you a sense of how networks are configured, and how network traffic moves in and out of Azure.

Your e-commerce site at a glance

Larger enterprise systems are often composed of multiple inter-connected applications and services that work together. You might have a front-end web system that displays inventory and allows customers to create an order. That might talk to a variety of web services to provide the inventory data, manage user profiles, process credit cards, and request fulfillment of processed orders.

There are several strategies and patterns employed by software architects and designers to make these complex systems easier to design, build, manage, and maintain. Let's look at a few of them, starting with *loosely coupled architectures*.

Benefits of Loosely Coupled Architectures

Using an N-tier architecture

An architectural pattern that can be used to build loosely coupled systems is *N-tier*.

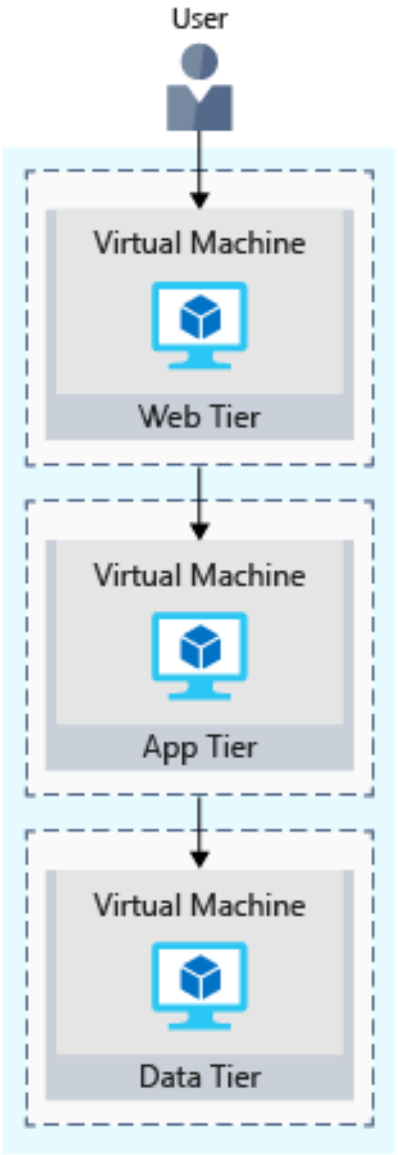
An N-tier architecture divides an application into two or more logical tiers. Architecturally, a higher tier can access services from a lower tier, but a lower tier should never access a higher tier.

Tiers help separate concerns and are ideally designed to be reusable. Using a tiered architecture also simplifies maintenance. Tiers can be updated or replaced independently, and new tiers can be inserted if needed.

Three-tier refers to an n-tier application that has three tiers. Your e-commerce web application follows this three-tier architecture:

- The **web tier** provides the web interface to your users through a browser.
- The **application tier** runs business logic.
- The **data tier** includes databases and other storage that hold product information and customer orders.

The following illustration shows the flow of a request from the user to the data tier.



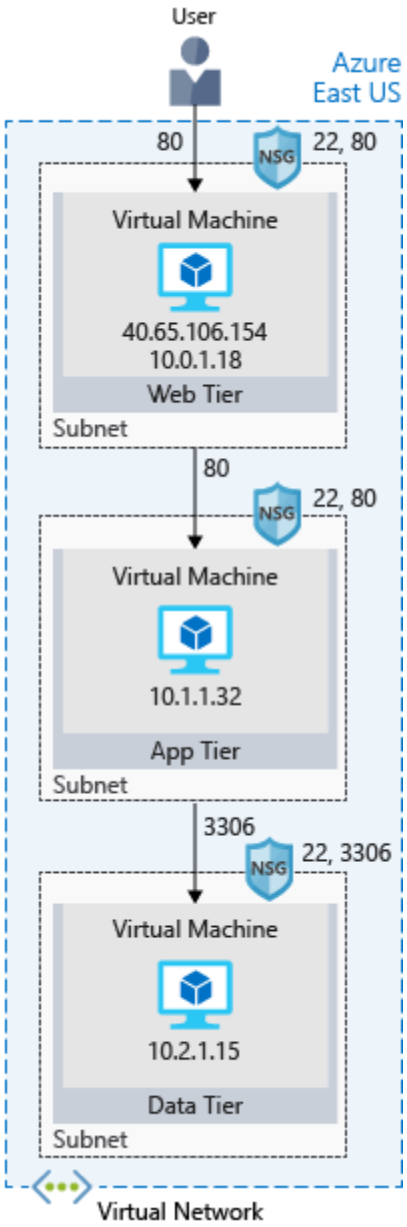
When the user clicks the button to place the order, the request is sent to the web tier, along with the user's address and payment information. The web tier passes this information to the application tier, which would

validate payment information and check inventory. The application tier might then store the order in the data tier, to be picked up later for fulfillment.

Your e-commerce site running on Azure

Azure provides many different ways to host your web applications, from fully pre-configured environments that host your code, to virtual machines that you configure, customize, and manage.

Let's say you choose to run your e-commerce site on virtual machines. Here's what that might look like in your test environment running on Azure. The following illustration shows a three-tier architecture running on virtual machines with security features enabled to restrict inbound requests.



What's an Azure region?

A *region* is one or more Azure data centers within a specific geographic location. East US, West US, and North Europe are examples of regions. In this instance, you see that the application is running in the East US region.

What's a virtual network?

A *virtual network* is a logically isolated network on Azure. Azure virtual networks will be familiar to you if you've set up networks on Hyper-V, VMware, or even on other public clouds. A virtual network allows Azure resources to securely communicate with each other, the internet, and on-premises networks. A virtual network is scoped to a single region; however, multiple virtual networks from different regions can be connected together using virtual network peering.

Virtual networks can be segmented into one or more *subnets*. Subnets help you organize and secure your resources in discrete sections. The web, application, and data tiers each have a single VM. All three VMs are in the same virtual network but are in separate subnets.

Users interact with the web tier directly, so that VM has a public IP address along with a private IP address. Users don't interact with the application or data tiers, so these VMs each have a private IP address only.

You can also keep your service or data tiers in your on-premises network, placing your web tier into the cloud, but keeping tight control over other aspects of your application. A *VPN gateway* (or virtual network gateway), enables this scenario. It can provide a secure connection between an Azure Virtual Network and an on-premises location over the internet.

Azure manages the physical hardware for you. You configure virtual networks and gateways through software, which enables you to treat a virtual network just like your own network. You choose which networks your virtual network can reach, whether that's the public internet or other networks in the private IP address space.

What's a network security group?

A *network security group*, or NSG, allows or denies inbound network traffic to your Azure resources. Think of a network security group as a cloud-level firewall for your network.

For example, notice that the VM in the web tier allows inbound traffic on ports 22 (SSH) and 80 (HTTP). This VM's network security group allows inbound traffic over these ports from all sources. You can configure a network security group to accept traffic only from known sources, such as IP addresses that you trust.

Note

Port 22 enables you to connect directly to Linux systems over SSH. Here we show port 22 open for learning purposes. In practice, you might configure VPN access to your virtual network to increase security.

Summary

Your three-tier application is now running on Azure in the East US region. A *region* is one or more Azure data centers within a specific geographic location.

Each tier can access services only from a lower tier. The VM running in the web tier has a public IP address because it receives traffic from the internet. The VMs in the lower tiers, the application and data tiers, each have private IP addresses because they don't communicate directly over the internet.

Virtual networks enable you to group and isolate related systems. You define *network security groups* to control what traffic can flow through a virtual network.

The configuration you saw here is a good start. But when you deploy your e-commerce site to production in the cloud, you'll likely run into the same problems as you did in your on-premises deployment.

Network security groups

You can use an Azure network security group to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains security rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

This article describes properties of a network security group rule, the default security rules that are applied, and the rule properties that you can modify to create an augmented security rule.

Security rules

A network security group contains zero, or as many rules as desired, within Azure subscription limits. Each rule specifies the following properties:

SECURITY RULES

Property	Explanation
Name	A unique name within the network security group.
Priority	A number between 100 and 4096. Rules are processed in priority order, with lower numbers processed before higher numbers, because lower numbers have higher priority. Once traffic matches a rule, processing stops. As a result, any rules that exist with lower priorities (higher numbers) that have the same attributes as rules with higher priorities are not processed.
Source or destination	Any, or an individual IP address, classless inter-domain routing (CIDR) block (10.0.0.0/24, for example), service tag, or application security group. If you specify an address for an Azure resource, specify the private IP address assigned to the resource. Network security groups are processed after Azure translates a public IP address to a private IP address for inbound traffic, and before Azure translates a private IP address to a public IP address for outbound traffic. . Specifying a range, a service tag, or application security group, enables you to create fewer security rules. The ability to specify multiple individual IP addresses and ranges (you cannot specify multiple service tags or application groups) in a rule is referred to as augmented security rules. Augmented security rules can only be created in network security groups created through the Resource Manager deployment model. You cannot specify multiple IP addresses and IP address ranges in network security groups created through the classic deployment model.
Protocol	TCP, UDP, ICMP or Any.
Direction	Whether the rule applies to inbound, or outbound traffic.
Port range	You can specify an individual or range of ports. For example, you could specify 80 or 10000-10005. Specifying ranges enables you to create fewer security rules. Augmented security rules can only be created in network security groups created through the Resource Manager deployment model. You cannot specify multiple ports or port ranges in the same security rule in network security groups created through the classic deployment model.
Action	Allow or deny

Network security group security rules are evaluated by priority using the 5-tuple information (source, source port, destination, destination port, and protocol) to allow or deny the traffic. You may not create two security rules with the same priority and direction. A flow record is created for existing connections. Communication is allowed or

denied based on the connection state of the flow record. The flow record allows a network security group to be stateful. If you specify an outbound security rule to any address over port 80, for example, it's not necessary to specify an inbound security rule for the response to the outbound traffic. You only need to specify an inbound security rule if communication is initiated externally. The opposite is also true. If inbound traffic is allowed over a port, it's not necessary to specify an outbound security rule to respond to traffic over the port.

Existing connections may not be interrupted when you remove a security rule that enabled the flow. Traffic flows are interrupted when connections are stopped and no traffic is flowing in either direction, for at least a few minutes.

There are limits to the number of security rules you can create in a network security group. For details, see Azure limits.

Default security rules

Azure creates the following default rules in each network security group that you create:

Inbound

AllowVNetInBound

ALLOWVNETINBOUND

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
65000	VirtualNetwork	0-65535	VirtualNetwork	0-65535	Any	Allow

AllowAzureLoadBalancerInBound

ALLOWAZURELOADBALANCERINBOUND

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
65001	AzureLoadBalancer	0-65535	0.0.0.0/0	0-65535	Any	Allow

DenyAllInbound

DENYALLINBOUND

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	Any	Deny

Outbound

AllowVnetOutBound

ALLOWVNETOUTBOUND

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
65000	VirtualNetwork	0-65535	VirtualNetwork	0-65535	Any	Allow

AllowInternetOutBound

ALLOWINTERNETOUTBOUND

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
65001	0.0.0.0/0	0-65535	Internet	0-65535	Any	Allow

DenyAllOutBound

DENYALLOUTBOUND

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
65500	0.0.0.0/0	0-65535	0.0.0.0/0	0-65535	Any	Deny

In the **Source** and **Destination** columns, *VirtualNetwork*, *AzureLoadBalancer*, and *Internet* are service tags, rather than IP addresses. In the protocol column, **Any** encompasses TCP, UDP, and ICMP. When creating a rule, you can specify TCP, UDP, ICMP or Any. *0.0.0.0/0* in the **Source** and **Destination** columns represents all addresses. Clients like Azure portal, Azure CLI, or PowerShell can use * or any for this expression.

You cannot remove the default rules, but you can override them by creating rules with higher priorities.

Augmented security rules

Augmented security rules simplify security definition for virtual networks, allowing you to define larger and complex network security policies, with fewer rules. You can combine multiple ports and multiple explicit IP addresses and ranges into a single, easily understood security rule. Use augmented rules in the source, destination, and port fields of a rule. To simplify maintenance of your security rule definition, combine augmented security rules with service tags or application security groups. There are limits to the number of addresses, ranges, and ports that you can specify in a rule. For details, see Azure limits.

Service tags

A service tag represents a group of IP address prefixes from a given Azure service. It helps to minimize the complexity of frequent updates on network security rules.

For more information, see Azure service tags. For an example on how to use the Storage service tag to restrict network access, see Restrict network access to PaaS resources.

Application security groups

Application security groups enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. You can reuse your security policy at scale without manual maintenance of explicit IP addresses. To learn more, see Application security groups.

Azure platform considerations

- **Virtual IP of the host node**: Basic infrastructure services like DHCP, DNS, IMDS, and health monitoring are provided through the virtualized host IP addresses 168.63.129.16 and 169.254.169.254. These IP addresses belong to Microsoft and are the only virtualized IP addresses used in all regions for this purpose. Effective security rules and effective routes will not include these platform rules. To override this basic infrastructure communication, you can create a security rule to deny traffic by using the following service tags on your Network Security Group rules: AzurePlatformDNS, AzurePlatformIMDS, AzurePlatformLKM. Learn how to diagnose network traffic filtering and diagnose network routing.

- **Licensing (Key Management Service)**: Windows images running in virtual machines must be licensed. To ensure licensing, a request is sent to the Key Management Service host servers that handle such queries. The request is made outbound through port 1688. For deployments using default route 0.0.0.0/0 configuration, this platform rule will be disabled.
- **Virtual machines in load-balanced pools**: The source port and address range applied are from the originating computer, not the load balancer. The destination port and address range are for the destination computer, not the load balancer.
- **Azure service instances**: Instances of several Azure services, such as HDInsight, Application Service Environments, and Virtual Machine Scale Sets are deployed in virtual network subnets. For a complete list of services you can deploy into virtual networks, see Virtual network for Azure services. Ensure you familiarize yourself with the port requirements for each service before applying a network security group to the subnet the resource is deployed in. If you deny ports required by the service, the service doesn't function properly.
- **Sending outbound email**: Microsoft recommends that you utilize authenticated SMTP relay services (typically connected via TCP port 587, but often others, as well) to send email from Azure Virtual Machines. SMTP relay services specialize in sender reputation, to minimize the possibility that third-party email providers reject messages. Such SMTP relay services include, but are not limited to, Exchange Online Protection and SendGrid. Use of SMTP relay services is in no way restricted in Azure, regardless of your subscription type.

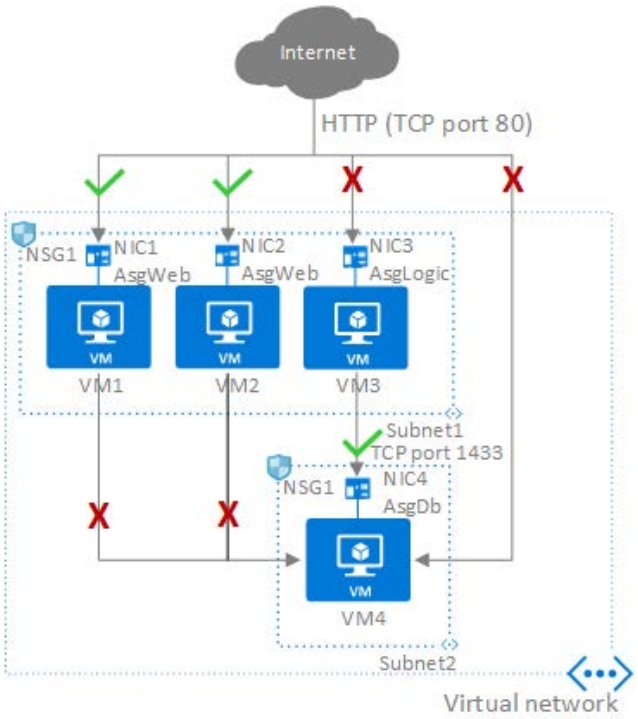
  If you created your Azure subscription prior to November 15, 2017, in addition to being able to use SMTP relay services, you can send email directly over TCP port 25. If you created your subscription after November 15, 2017, you may not be able to send email directly over port 25. The behavior of outbound communication over port 25 depends on the type of subscription you have, as follows:

  - **Enterprise Agreement**: Outbound port 25 communication is allowed. You are able to send an outbound email directly from virtual machines to external email providers, with no restrictions from the Azure platform.
  - **Pay-as-you-go:** Outbound port 25 communication is blocked from all resources. If you need to send email from a virtual machine directly to external email providers (not using an authenticated SMTP relay), you can make a request to remove the restriction. Requests are reviewed and approved at Microsoft's discretion and are only granted after anti-fraud checks are performed. To make a request, open a support case with the issue type *Technical*, *Virtual Network Connectivity*, *Cannot send e-mail (SMTP/Port 25)*. In your support case, include details about why your subscription needs to send email directly to mail providers, instead of going through an authenticated SMTP relay. If your subscription is exempted, only virtual machines created after the exemption date are able to communicate outbound over port 25.
  - **MSDN, Azure Pass, Azure in Open, Education, BizSpark, and Free trial**: Outbound port 25 communication is blocked from all resources. No requests to remove the restriction can be made, because requests are not granted. If you need to send email from your virtual machine, you have to use an SMTP relay service.
  - **Cloud service provider**: Customers that are consuming Azure resources via a cloud service provider can create a support case with their cloud service provider, and request that the provider create an unblock case on their behalf, if a secure SMTP relay cannot be used.

  If Azure allows you to send email over port 25, Microsoft cannot guarantee email providers will accept inbound email from your virtual machine. If a specific provider rejects mail from your virtual machine, work directly with the provider to resolve any message delivery or spam filtering issues, or use an authenticated SMTP relay service.

Application security groups

Application security groups enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. You can reuse your security policy at scale without manual maintenance of explicit IP addresses. The platform handles the complexity of explicit IP addresses and multiple rule sets, allowing you to focus on your business logic. To better understand application security groups, consider the following example:



In the previous picture, *NIC1* and *NIC2* are members of the *AsgWeb* application security group. *NIC3* is a member of the *AsgLogic* application security group. *NIC4* is a member of the *AsgDb* application security group. Though each network interface in this example is a member of only one network security group, a network interface can be a member of multiple application security groups, up to the Azure limits. None of the network interfaces have an associated network security group. *NSG1* is associated to both subnets and contains the following rules:

Allow-HTTP-Inbound-Internet

This rule is needed to allow traffic from the internet to the web servers. Because inbound traffic from the internet is denied by the **DenyAllInbound** default security rule, no additional rule is needed for the *AsgLogic* or *AsgDb* application security groups.

ALLOW-HTTP-INBOUND-INTERNET

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
100	Internet	*	AsgWeb	80	TCP	Allow

Deny-Database-All

Because the **AllowVNetInBound** default security rule allows all communication between resources in the same virtual network, this rule is needed to deny traffic from all resources.

DENY-DATABASE-ALL

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
120	*	*	AsgDb	1433	Any	Deny

Allow-Database-BusinessLogic

This rule allows traffic from the *AsgLogic* application security group to the *AsgDb* application security group. The priority for this rule is higher than the priority for the *Deny-Database-All* rule. As a result, this rule is processed before the *Deny-Database-All* rule, so traffic from the *AsgLogic* application security group is allowed, whereas all other traffic is blocked.

ALLOW-DATABASE-BUSINESSLOGIC

Priority	Source	Source ports	Destination	Destination ports	Protocol	Access
110	AsgLogic	*	AsgDb	1433	TCP	Allow

The rules that specify an application security group as the source or destination are only applied to the network interfaces that are members of the application security group. If the network interface is not a member of an application security group, the rule is not applied to the network interface, even though the network security group is associated to the subnet.

Application security groups have the following constraints:

- There are limits to the number of application security groups you can have in a subscription, as well as other limits related to application security groups. For details, see Azure limits.
- You can specify one application security group as the source and destination in a security rule. You cannot specify multiple application security groups in the source or destination.
- All network interfaces assigned to an application security group have to exist in the same virtual network that the first network interface assigned to the application security group is in. For example, if the first network interface assigned to an application security group named *AsgWeb* is in the virtual network named *VNet1*, then all subsequent network interfaces assigned to *ASGWeb* must exist in *VNet1*. You cannot add network interfaces from different virtual networks to the same application security group.
- If you specify an application security group as the source and destination in a security rule, the network interfaces in both application security groups must exist in the same virtual network. For example, if *AsgLogic* contained network interfaces from *VNet1*, and *AsgDb* contained network interfaces from *VNet2*, you could not assign *AsgLogic* as the source and *AsgDb* as the destination in a rule. All network interfaces for both the source and destination application security groups need to exist in the same virtual network.

Tip

To minimize the number of security rules you need, and the need to change the rules, plan out the application security groups you need and create rules using service tags or application security groups, rather than individual IP addresses, or ranges of IP addresses, whenever possible.

System routes

Azure automatically creates system routes and assigns the routes to each subnet in a virtual network. You can't create system routes, nor can you remove system routes, but you can override some system routes with custom routes. Azure creates default system routes for each subnet, and adds additional optional default routes to specific subnets, or every subnet, when you use specific Azure capabilities.

Default

Each route contains an address prefix and next hop type. When traffic leaving a subnet is sent to an IP address within the address prefix of a route, the route that contains the prefix is the route Azure uses. Learn more about how Azure selects a route when multiple routes contain the same prefixes, or overlapping prefixes.

Whenever a virtual network is created, Azure automatically creates the following default system routes for each subnet within the virtual network:

DEFAULT

Source	Address prefixes	Next hop type
Default	Unique to the virtual network	Virtual network
Default	0.0.0.0/0	Internet
Default	10.0.0.0/8	None
Default	192.168.0.0/16	None
Default	100.64.0.0/10	None

The next hop types listed in the previous table represent how Azure routes traffic destined for the address prefix listed. Explanations for the next hop types follow:

- **Virtual network**: Routes traffic between address ranges within the address space of a virtual network. Azure creates a route with an address prefix that corresponds to each address range defined within the address space of a virtual network. If the virtual network address space has multiple address ranges defined, Azure creates an individual route for each address range. Azure automatically routes traffic between subnets using the routes created for each address range. You don't need to define gateways for Azure to route traffic between subnets. Though a virtual network contains subnets, and each subnet has a defined address range, Azure does *not* create default routes for subnet address ranges, because each subnet address range is within an address range of the address space of a virtual network.
- **Internet**: Routes traffic specified by the address prefix to the Internet. The system default route specifies the 0.0.0.0/0 address prefix. If you don't override Azure's default routes, Azure routes traffic for any address not specified by an address range within a virtual network, to the Internet, with one exception. If the destination address is for one of Azure's services, Azure routes the traffic directly to the service over Azure's backbone network, rather than routing the traffic to the Internet. Traffic between Azure services does not traverse the Internet, regardless of which Azure region the virtual network exists in, or which Azure region an instance of the Azure service is deployed in. You can override Azure's default system route for the 0.0.0.0/0 address prefix with a custom route.
- **None**: Traffic routed to the **None** next hop type is dropped, rather than routed outside the subnet. Azure automatically creates default routes for the following address prefixes:
  - **10.0.0.0/8 and 192.168.0.0/16**: Reserved for private use in RFC 1918.
  - **100.64.0.0/10**: Reserved in RFC 6598.

  If you assign any of the previous address ranges within the address space of a virtual network, Azure automatically changes the next hop type for the route from **None** to **Virtual network**. If you assign an address range to the address space of a virtual network that includes, but isn't the same as, one of the four reserved address prefixes, Azure removes the route for the prefix and adds a route for the address prefix you added, with **Virtual network** as the next hop type.

Optional default routes

Azure adds additional default system routes for different Azure capabilities, but only if you enable the capabilities. Depending on the capability, Azure adds optional default routes to either specific subnets within the virtual network, or to all subnets within a virtual network. The additional system routes and next hop types that Azure may add when you enable different capabilities are:

Source	Address prefixes	Next hop type	Subnet within virtual network that route is added to
Default	Unique to the virtual network, for example: 10.1.0.0/16	VNet peering	All
Virtual network gateway	Prefixes advertised from on-premises via BGP, or configured in the local network gateway	Virtual network gateway	All
Default	Multiple	VirtualNetworkServiceEndpoint	Only the subnet a service endpoint is enabled for.

- **Virtual network (VNet) peering**: When you create a virtual network peering between two virtual networks, a route is added for each address range within the address space of each virtual network a peering is created for. Learn more about virtual network peering.
- **Virtual network gateway**: One or more routes with *Virtual network gateway* listed as the next hop type are added when a virtual network gateway is added to a virtual network. The source is also *virtual network gateway*, because the gateway adds the routes to the subnet. If your on-premises network gateway exchanges border gateway protocol (BGP) routes with an Azure virtual network gateway, a route is added for each route propagated from the on-premises network gateway. It's recommended that you summarize on-premises routes to the largest address ranges possible, so the fewest number of routes are propagated to an Azure virtual network gateway. There are limits to the number of routes you can propagate to an Azure virtual network gateway. For details, see Azure limits.
- **VirtualNetworkServiceEndpoint**: The public IP addresses for certain services are added to the route table by Azure when you enable a service endpoint to the service. Service endpoints are enabled for individual subnets within a virtual network, so the route is only added to the route table of a subnet a service endpoint is enabled for. The public IP addresses of Azure services change periodically. Azure manages the addresses in the route table automatically when the addresses change. Learn more about virtual network service endpoints, and the services you can create service endpoints for.

  Note

  The **VNet peering** and **VirtualNetworkServiceEndpoint** next hop types are only added to route tables of subnets within virtual networks created through the Azure Resource Manager deployment model. The next hop types are not added to route tables that are associated to virtual network subnets created through the classic deployment model. Learn more about Azure **deployment models**.

Custom routes

You create custom routes by either creating user-defined routes, or by exchanging border gateway protocol (BGP) routes between your on-premises network gateway and an Azure virtual network gateway.

User-defined

You can create custom, or user-defined(static), routes in Azure to override Azure's default system routes, or to add additional routes to a subnet's route table. In Azure, you create a route table, then associate the route table to zero or more virtual network subnets. Each subnet can have zero or one route table associated to it. To learn about the maximum number of routes you can add to a route table and the maximum number of user-defined route tables you can create per Azure subscription, see Azure limits. If you create a route table and associate it to a subnet, the routes within it are combined with, or override, the default routes Azure adds to a subnet by default.

You can specify the following next hop types when creating a user-defined route:

- **Virtual appliance**: A virtual appliance is a virtual machine that typically runs a network application, such as a firewall. To learn about a variety of pre-configured network virtual appliances you can deploy in a virtual network, see the Azure Marketplace. When you create a route with the **virtual appliance** hop type, you also specify a next hop IP address. The IP address can be:
  - The private IP address of a network interface attached to a virtual machine. Any network interface attached to a virtual machine that forwards network traffic to an address other than its own must have the Azure *Enable IP forwarding* option enabled for it. The setting disables Azure's check of the source and destination for a network interface. Learn more about how to enable IP forwarding for a network interface. Though *Enable IP forwarding* is an Azure setting, you may also need to enable IP forwarding within the virtual machine's operating system for the appliance to forward traffic between private IP addresses assigned to Azure network interfaces. If the appliance must route traffic to a public IP address, it must either proxy the traffic, or network address translate the private IP address of the source's private IP address to its own private IP address, which Azure then network address translates to a public IP address, before sending the traffic to the Internet. To determine required settings within the virtual machine, see the documentation for your operating system or network application. To understand outbound connections in Azure, see Understanding outbound connections.

    Note

    Deploy a virtual appliance into a different subnet than the resources that route through the virtual appliance are deployed in. Deploying the virtual appliance to the same subnet, then applying a route table to the subnet that routes traffic through the virtual appliance, can result in routing loops, where traffic never leaves the subnet.

  - The private IP address of an Azure internal load balancer. A load balancer is often used as part of a high availability strategy for network virtual appliances.

  You can define a route with 0.0.0.0/0 as the address prefix and a next hop type of virtual appliance, enabling the appliance to inspect the traffic and determine whether to forward or drop the traffic. If you intend to create a user-defined route that contains the 0.0.0.0/0 address prefix, read 0.0.0.0/0 address prefix first.

- **Virtual network gateway**: Specify when you want traffic destined for specific address prefixes routed to a virtual network gateway. The virtual network gateway must be created with type **VPN**. You cannot specify a virtual network gateway created as type **ExpressRoute** in a user-defined route because with ExpressRoute, you must use BGP for custom routes. You can define a route that directs traffic destined for the 0.0.0.0/0 address prefix to a route-based virtual network gateway. On your premises, you might have a device that inspects the traffic and determines whether to forward or drop the traffic. If you intend to create a user-defined route for the 0.0.0.0/0 address prefix, read 0.0.0.0/0 address prefix first. Instead of configuring a user-defined route for the 0.0.0.0/0 address prefix, you can advertise a route with the 0.0.0.0/0 prefix via BGP, if you've enabled BGP for a VPN virtual network gateway.
- **None**: Specify when you want to drop traffic to an address prefix, rather than forwarding the traffic to a destination. If you haven't fully configured a capability, Azure may list *None* for some of the optional system routes. For example, if you see *None* listed as the **Next hop IP address** with a **Next hop type** of *Virtual network gateway* or *Virtual appliance*, it may be because the device isn't running, or isn't fully configured. Azure creates system default routes for reserved address prefixes with **None** as the next hop type.
- **Virtual network**: Specify when you want to override the default routing within a virtual network. See Routing example, for an example of why you might create a route with the **Virtual network** hop type.
- **Internet**: Specify when you want to explicitly route traffic destined to an address prefix to the Internet, or if you want traffic destined for Azure services with public IP addresses kept within the Azure backbone network.

You cannot specify **VNet peering** or **VirtualNetworkServiceEndpoint** as the next hop type in user-defined routes. Routes with the **VNet peering** or **VirtualNetworkServiceEndpoint** next hop types are only created by Azure, when you configure a virtual network peering, or a service endpoint.

Protect virtual networks by using Azure Firewall

A *firewall* is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules. You can create firewall rules that specify ranges of IP addresses. Only clients granted IP addresses from within those ranges are allowed to access the destination server. Firewall rules can also include specific network protocol and port information.

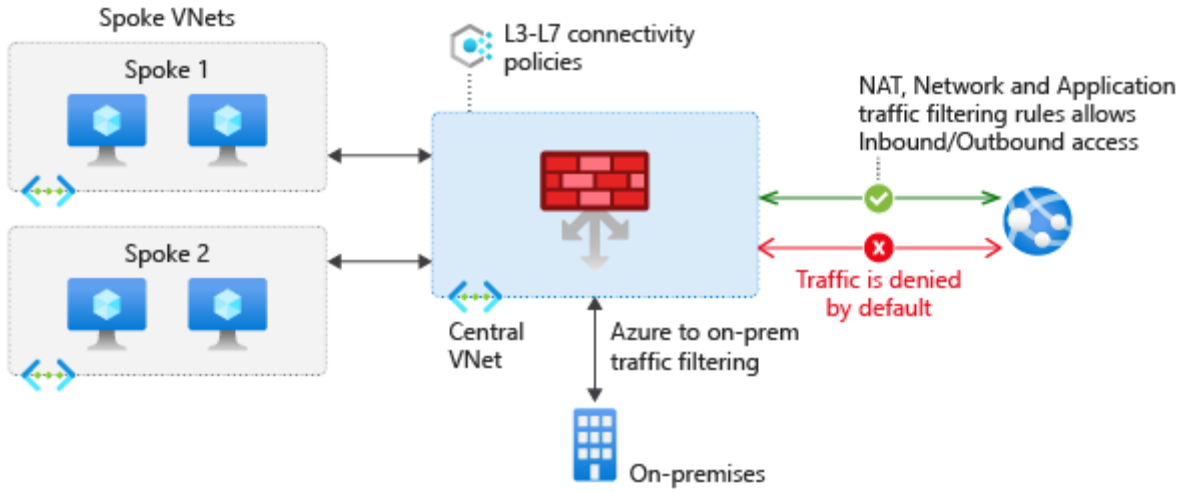
Tailwind Traders currently runs firewall appliances, which combine hardware and software, to protect its on-premises network. These firewall appliances require a monthly licensing fee to operate, and they require IT staff to perform routine maintenance. As Tailwind Traders moves to the cloud, the IT manager wants to know what Azure services can protect both the company's cloud networks and its on-premises networks.

In this part, you explore Azure Firewall.

What's Azure Firewall?

Azure Firewall is a managed, cloud-based network security service that helps protect resources in your Azure virtual networks. A virtual network is similar to a traditional network that you'd operate in your own datacenter. It's a fundamental building block for your private network that enables virtual machines and other compute resources to securely communicate with each other, the internet, and on-premises networks.

Here's a diagram that shows a basic Azure Firewall implementation:



Azure Firewall is a *stateful* firewall. A stateful firewall analyzes the complete context of a network connection, not just an individual packet of network traffic. Azure Firewall features high availability and unrestricted cloud scalability.

Azure Firewall provides a central location to create, enforce, and log application and network connectivity policies across subscriptions and virtual networks. Azure Firewall uses a static (unchanging) public IP address for your virtual network resources, which enables outside firewalls to identify traffic coming from your virtual network. The service is integrated with Azure Monitor to enable logging and analytics.

Azure Firewall provides many features, including:

- Built-in high availability.
- Unrestricted cloud scalability.
- Inbound and outbound filtering rules.
- Inbound Destination Network Address Translation (DNAT) support.
- Azure Monitor logging.

You typically deploy Azure Firewall on a central virtual network to control general network access.

What can I configure with Azure Firewall?

With Azure Firewall, you can configure:

- Application rules that define fully qualified domain names (FQDNs) that can be accessed from a subnet.
- Network rules that define source address, protocol, destination port, and destination address.
- Network Address Translation (NAT) rules that define destination IP addresses and ports to translate inbound requests.

Azure Application Gateway also provides a firewall that's called the *web application firewall* (WAF). WAF provides centralized, inbound protection for your web applications against common exploits and vulnerabilities. Azure Front Door and Azure Content Delivery Network also provide WAF services.

Protect from DDoS attacks by using Azure DDoS Protection

- 3 minutes

Any large company can be the target of a large-scale network attack. Tailwind Traders is no exception. Attackers might flood your network to make a statement or simply for the challenge. As Tailwind Traders moves to the cloud, it wants to understand how Azure can help prevent distributed denial of service (DDoS) and other attacks.

In this part, you learn how Azure DDoS Protection (Standard service tier) helps protect your Azure resources from DDoS attacks. First, let's define what a DDoS attack is.

What are DDoS attacks?

A distributed denial of service attack attempts to overwhelm and exhaust an application's resources, making the application slow or unresponsive to legitimate users. DDoS attacks can target any resource that's publicly reachable through the internet, including websites.

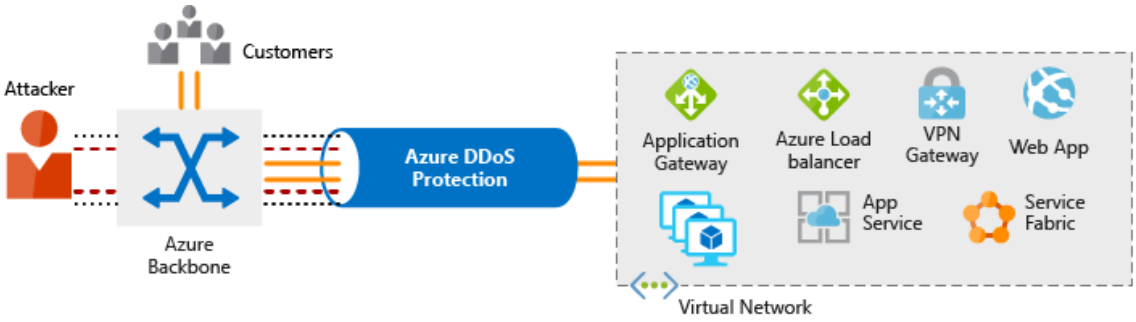
What is Azure DDoS Protection?

Azure DDoS Protection (Standard) helps protect your Azure resources from DDoS attacks.

When you combine DDoS Protection with recommended application design practices, you help provide a defense against DDoS attacks. DDoS Protection uses the scale and elasticity of Microsoft's global network to bring DDoS mitigation capacity to every Azure region. The DDoS Protection service helps protect your Azure

applications by analyzing and discarding DDoS traffic at the Azure network edge, before it can affect your service's availability.

This diagram shows network traffic flowing into Azure from both customers and an attacker:



DDoS Protection identifies the attacker's attempt to overwhelm the network and blocks further traffic from them, ensuring that traffic never reaches Azure resources. Legitimate traffic from customers still flows into Azure without any interruption of service.

DDoS Protection can also help you manage your cloud consumption. When you run on-premises, you have a fixed number of compute resources. But in the cloud, elastic computing means that you can automatically scale out your deployment to meet demand. A cleverly designed DDoS attack can cause you to increase your resource allocation, which incurs unneeded expense. DDoS Protection Standard helps ensure that the network load you process reflects customer usage. You can also receive credit for any costs accrued for scaled-out resources during a DDoS attack.

What service tiers are available to DDoS Protection?

DDoS Protection provides these service tiers:

- **Basic**

  The Basic service tier is automatically enabled for free as part of your Azure subscription.

  Always-on traffic monitoring and real-time mitigation of common network-level attacks provide the same defenses that Microsoft's online services use. The Basic service tier ensures that Azure infrastructure itself is not affected during a large-scale DDoS attack.

  The Azure global network is used to distribute and mitigate attack traffic across Azure regions.

- **Standard**

  The Standard service tier provides additional mitigation capabilities that are tuned specifically to Azure Virtual Network resources. DDoS Protection Standard is relatively easy to enable and requires no changes to your applications.

  The Standard tier provides always-on traffic monitoring and real-time mitigation of common network-level attacks. It provides the same defenses that Microsoft's online services use.

  Protection policies are tuned through dedicated traffic monitoring and machine learning algorithms. Policies are applied to public IP addresses, which are associated with resources deployed in virtual networks such as Azure Load Balancer and Application Gateway.

The Azure global network is used to distribute and mitigate attack traffic across Azure regions.

What kinds of attacks can DDoS Protection help prevent?

The Standard service tier can help prevent:

- **Volumetric attacks**

  The goal of this attack is to flood the network layer with a substantial amount of seemingly legitimate traffic.

- **Protocol attacks**

  These attacks render a target inaccessible by exploiting a weakness in the layer 3 and layer 4 protocol stack.

- **Resource-layer (application-layer) attacks (only with web application firewall)**

  These attacks target web application packets to disrupt the transmission of data between hosts. You need a web application firewall (WAF) to protect against L7 attacks. DDoS Protection Standard protects the WAF from volumetric and protocol attacks.

Identity and access

Network perimeters, firewalls, and physical access controls used to be the primary protection for corporate data. But network perimeters have become increasingly porous with the explosion of bring your own device (BYOD), mobile apps, and cloud applications.

Identity has become the new primary security boundary. Therefore, proper authentication and assignment of privileges is critical to maintaining control of your data.

Your company, Contoso Shipping, is focused on addressing these concerns right away. Your team's new hybrid cloud solution needs to account for mobile apps that have access to secret data when an authorized user is signed in — in addition to having shipping vehicles constantly send a stream of telemetry data that is critical to optimizing the company's business.

Authentication and authorization

Two fundamental concepts that need to be understood when talking about identity and access control are authentication and authorization. They underpin everything else that happens and occur sequentially in any identity and access process:

- *Authentication* is the process of establishing the identity of a person or service looking to access a resource. It involves the act of challenging a party for legitimate credentials, and provides the basis for creating a security principal for identity and access control use. It establishes if they are who they say they are.
- *Authorization* is the process of establishing what level of access an authenticated person or service has. It specifies what data they're allowed to access and what they can do with it.

Note

Authentication is sometimes shortened to *AuthN*, and authorization is sometimes shortened to *AuthZ*.

Azure provides services to manage both authentication and authorization through Azure Active Directory (Azure AD).

What is Azure Active Directory?

Azure AD is a cloud-based identity service. It has built in support for synchronizing with your existing on-premises Active Directory or can be used stand-alone. This means that all your applications, whether on-premises, in the cloud (including Microsoft 365), or even mobile can share the same credentials. Administrators and developers can control access to internal and external data and applications using centralized rules and policies configured in Azure AD.

Azure AD provides services such as:

- **Authentication.** This includes verifying identity to access applications and resources, and providing functionality such as self-service password reset, multi-factor authentication (MFA), a custom banned password list, and smart lockout services.
- **Single-Sign-On (SSO).** SSO enables users to remember only one ID and one password to access multiple applications. A single identity is tied to a user, simplifying the security model. As users change roles or leave an organization, access modifications are tied to that identity, greatly reducing the effort needed to change or disable accounts.
- **Application management.** You can manage your cloud and on-premises apps using Azure AD Application Proxy, SSO, the My apps portal (also referred to as Access panel), and SaaS apps.
- **Business to business (B2B) identity services.** Manage your guest users and external partners while maintaining control over your own corporate data
- **Business-to-Customer (B2C) identity services.** Customize and control how users sign up, sign in, and manage their profiles when using your apps with services.
- **Device Management.** Manage how your cloud or on-premises devices access your corporate data.

Let's explore a few of these in more detail.

Single sign-on

The more identities a user has to manage, the greater the risk of a credential-related security incident. More identities mean more passwords to remember and change. Password policies can vary between applications and, as complexity requirements increase, it becomes increasingly difficult for users to remember them.

Now, consider the logistics of managing all those identities. Additional strain is placed on help desks as they deal with account lockouts and password reset requests. If a user leaves an organization, tracking down all those identities and ensuring they are disabled can be challenging. If an identity is overlooked, this could allow access when it should have been eliminated.

With single sign-on (SSO), users need to remember only one ID and one password. Access across applications is granted to a single identity tied to a user, simplifying the security model. As users change roles or leave an organization, access modifications are tied to the single identity, greatly reducing the effort needed to change or disable accounts. Using single sign-on for accounts will make it easier for users to manage their identities and will increase the security capabilities in your environment.

SSO with Azure Active Directory

By leveraging Azure AD for SSO you'll also have the ability to combine multiple data sources into an intelligent security graph. This security graph enables the ability to provide threat analysis and real-time identity protection to all accounts in Azure AD, including accounts that are synchronized from your on-premises AD. By using a centralized identity provider, you'll have centralized the security controls, reporting, alerting, and administration of your identity infrastructure.

As Contoso Shipping integrates its existing Active Directory instance with Azure AD, you will make controlling access consistent across the organization. Doing so will also greatly simplify the ability to sign into email and Microsoft 365 documents without having to reauthenticate.

Multi-factor authentication

Multi-factor authentication (MFA) provides additional security for your identities by requiring two or more elements for full authentication. These elements fall into three categories:

- *Something you know*
- *Something you possess*
- *Something you are*

Something you know would be a password or the answer to a security question. **Something you possess** could be a mobile app that receives a notification or a token-generating device. **Something you are** is typically some sort of biometric property, such as a fingerprint or face scan used on many mobile devices.

Using MFA increases security of your identity by limiting the impact of credential exposure. An attacker who has a user's password would also need to have possession of their phone or their security token generator in order to fully authenticate. Authentication with only a single factor verified is insufficient, and the attacker would be unable to use only those credentials to authenticate. The benefits this brings to security are huge, and we can't emphasize enough the importance of enabling MFA wherever possible.

Azure AD has MFA capabilities built in and will integrate with other third-party MFA providers. MFA should be used for users in the Global Administrator role in Azure AD, because these are highly sensitive accounts. All other accounts can also have MFA enabled.

For Contoso Shipping, you decide to enable MFA any time a user is signing in from a non-domain-connected computer — which includes the mobile apps your drivers use.

Providing identities to services

It's usually valuable for services to have identities. Often, and against best practices, credential information is embedded in configuration files. With no security around these configuration files, anyone with access to the systems or repositories can access these credentials and risk exposure.

Azure AD addresses this problem through two methods: service principals and managed identities for Azure services.

Service principals

To understand service principals, it's useful to first understand the words **identity** and **principal**, because of how they are used in the identity management world.

An **identity** is just a thing that can be authenticated. Obviously, this includes users with a user name and password, but it can also include applications or other servers, which might authenticate with secret keys or certificates.

A **principal** is an identity acting with certain roles or claims. Usually, it is not useful to consider identity and principal separately, but think of using 'sudo' on a Bash prompt in Linux or on Windows using "run as Administrator." In both those cases, you are still logged in as the same identity as before, but you've changed the role under which you are executing. Groups are often also considered principals because they can have rights assigned.

A **service principal** is an identity that is used by a service or application. And like other identities, it can be assigned roles.

Managed identities for Azure services

The creation of service principals can be a tedious process, and there are a lot of touch points that can make maintaining them difficult. Managed identities for Azure services are much easier and will do most of the work for you.

A managed identity can be instantly created for any Azure service that supports it—and the list is constantly growing. When you create a managed identity for a service, you are creating an account on your organization's Active Directory (a specific organization's Active Directory instance is known as an "Active Directory Tenant"). The Azure infrastructure will automatically take care of authenticating the service and managing the account. You can then use that account like any other Azure AD account, including allowing the authenticated service secure access of other Azure resources.

Role-based access control

Roles are sets of permissions, like "Read-only" or "Contributor", that users can be granted to access an Azure service instance.

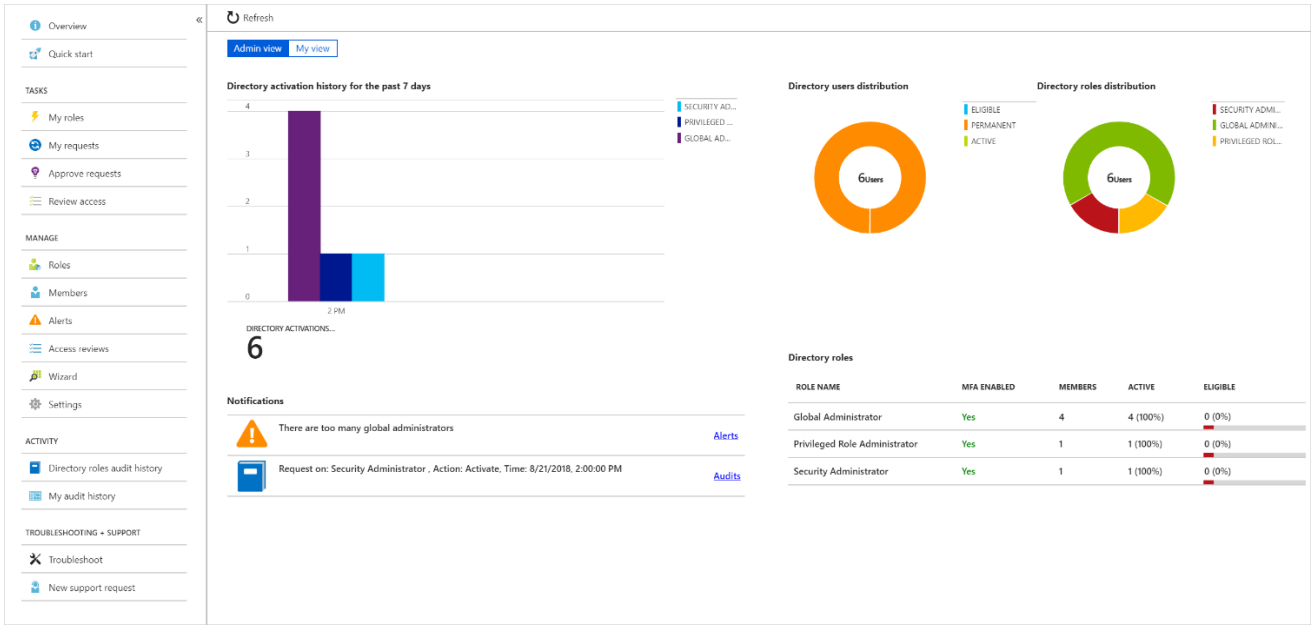
Identities are mapped to roles directly or through group membership. Separating security principals, access permissions, and resources provides simple access management and fine-grained control. Administrators are able to ensure the minimum necessary permissions are granted.

Roles can be granted at the individual service instance level, but they also flow down the Azure Resource Manager hierarchy.

Here's a diagram that shows this relationship. Roles assigned at a higher scope, like an entire subscription, are inherited by child scopes, like service instances.

Privileged Identity Management

In addition to managing Azure resource access with role-based access control (RBAC), a comprehensive approach to infrastructure protection should consider including the ongoing auditing of role members as their organization changes and evolves. Azure AD Privileged Identity Management (PIM) is an additional, paid-for offering that provides oversight of role assignments, self-service, and just-in-time role activation and Azure AD and Azure resource access reviews.



Summary

Identity allows us to maintain a security perimeter, even outside our physical control. With single sign-on and appropriate role-based access configuration, we can always be sure who has the ability to see and manipulate our data and infrastructure.

Identity

- A user with a username and password.
- Also applications or other servers with secret keys or certificates.
- The fact of being something or someone.

Authentication

The process of **verification/assertion of identity**

Authorization

The process of **ensuring** that only **authenticated identities** get **access to the resources** for which they have been granted access.

Access Management

The process of **controlling**, **verifying**, **tracking** and **managing access** to authorized users and applications.

Azure Active Directory

- Identity and Access Management service in Azure
- Identities management – users, groups, applications
- Access management – subscriptions, resource groups, roles, role assignments, authentication & authorization settings, etc.
- Used by multiple Microsoft cloud platforms
    - Azure
    - Microsoft 365
    - Office 365
    - Live.com services (Skype, OneDrive, etc.)

Multi-factor Authentication (MFA)

- Process of authentication using more than one factor (evidence) to prove identity
- Factor types
    - Knowledge Factor – “Something you know”, ex. password, pin
    - Possession Factor – “Something you have”, ex. phone, token, card, key
    - Physical Characteristic Factor – “Something you are”, ex. fingerprint, voice, face, eye iris
    - Location Factor – “Somewhere you are”, ex. GPS location
- Supported by Azure AD by default (simple on-off switch)

Free tier

The free tier is automatically enabled on all Azure subscriptions and provides security policy, continuous security assessment, and actionable security recommendations to help you protect your Azure resources. It monitors the most common app resources in Azure including:

- Compute resources such as VMs, Azure Functions and App Service
- Network access and endpoint security
- Data storage including Azure Storage, Redis cache for Azure, and Azure SQL
- Identity and access including Azure Key Vault
- IoT Hubs and resources

Standard tier

The standard tier extends the capabilities of the free tier to workloads running in private and other public clouds to provide unified security management and threat protection across all your hybrid cloud workloads.

The Standard tier adds advanced threat detection capabilities, using analytics and machine learning to identify attacks and zero-day exploits, access and application controls to reduce exposure to network attacks and malware, and more.

Migrating your Security Center subscription from the free tier to the standard tier enables the following features:

- **Security event collection**. Security Center collects logs in a central place so you can search and analyze them to identify important security events that may require your attention.
- **Network Map**. This feature allows you to visualize the topology of your Azure network infrastructure and the traffic to your Azure VMs. It also allows you to create filters by the severity level and recommendations.
- **Just-in-time VM access**. This allows admins to grant access to a VM for a defined period of time. Limiting access helps reduce exposure to outside attacks. This feature is especially useful if you're working with an outside agency that needs to access your VM.
- **Adaptive application controls (application whitelisting)**. Adaptive application controls uses artificial intelligence to recommend applications to allow. This helps protect VMs by preventing malware and unauthorized software from being installed.
- **Regulatory compliance reports**. In the Regulatory compliance dashboard, you have a clear view of the status of all standard regulatory assessments within your environment.
- **File integrity monitoring**. This feature examines files and registries of operating system, application software, and others in Windows and Linux (computers and VMs) for changes that might indicate an attack.
- **Adaptive Network Hardening**. Adaptive Network Hardening provides recommendations to harden applied NSG rules. It uses machine learning algorithms that factors in actual traffic, known trusted configuration, threat intelligence, and other indicators of compromise, and then provides recommendations to allow traffic only from specific IP/port tuples.
- **Security alerts**. Security Center supports a variety of security alerts such as detection of potential distributed denial-of-service (DDOS) attacks. Just-in-time alerts gives you the chance to investigate evolving issues before they result in a service failure.
- **Threat intelligence**. This feature can help determine the nature of an attack, the attack point of origin, and more.
- **Workflow Automation**. Workflow automation is a collection of procedures that can be executed from Security Center once a certain playbook is triggered from selected alert. Workflow automation can help to automate and orchestrate your response to a specific security alert detected by Security Center.

Switch to the Standard tier

You can try the Standard tier for free for 30 days. This allows you to evaluate the additional features, see how your current environment will benefit from them, and decide whether they're worth the investment.

You can enable Security Center on a per-subscription basis. Each subscription can choose what elements you want to enroll. Selecting the **Coverage** item under **POLICY & COMPLIANCE** will list all your available subscriptions (Not covered, Partially covered through the Free tier or partial plan, and Fully covered on the Standard tier).

Selecting a subscription allows you to control what areas you want Security Center to monitor as shown in the following screenshot.

Identity

- **Centralized/unified** infrastructure and platform **security management service**
- **Natively embedded** in Azure services
- **Integrated** with **Azure Advisor**
- Two tiers
    - **Free** (Azure Defender OFF) – included in all Azure services, provides continuous assessments, security score, and actionable security recommendations
    - **Paid** (Azure Defender ON) – hybrid security, threat protection alerts, vulnerability scanning, just in time (JIT) VM access, etc.

What is Azure Key Vault?

- 5 minutes

Azure Key Vault is a *secret store*: a centralized cloud service for storing application secrets - configuration values like passwords and connection strings that must remain secure at all times. Key Vault helps you control your applications' secrets by keeping them in a single central location and providing secure access, permissions control, and access logging.

The main benefits of using Key Vault are:

- Separation of sensitive application information from other configuration and code, reducing risk of accidental leaks
- Restricted secret access with access policies tailored to the applications and individuals that need them
- Centralized secret storage, allowing required changes to happen in only one place
- Access logging and monitoring to help you understand how and when secrets are accessed

Secrets are stored in individual *vaults*, which are Azure resources used to group secrets together. Secret access and vault management is accomplished via a REST API, which is also supported by all of the Azure management tools as well as client libraries available for many popular languages. Every vault has a unique URL where its API is hosted.

Important

Key Vault is designed to store configuration secrets for server applications. It's not intended for storing data belonging to your app's users, and it shouldn't be used in the client-side part of an app. This is reflected in its performance characteristics, API, and cost model.

User data should be stored elsewhere, such as in an Azure SQL database with Transparent Data Encryption, or a storage account with Storage Service Encryption. Secrets used by your application to access those data stores can be kept in Key Vault.

What is a secret in Key Vault?

In Key Vault, a secret is a name-value pair of strings. Secret names must be 1-127 characters long, contain only alphanumeric characters and dashes, and must be unique within a vault. A secret value can be any UTF-8 string up to 25 KB in size.

Tip

Secret names don't need to be considered especially secret themselves. You can store them in your app's configuration if your implementation calls for it. The same is true of vault names and URLs.

Note

Key Vault supports two additional kinds of secrets beyond strings — *keys* and *certificates* — and provides useful functionality specific to their use cases. This module does not cover these features and concentrates on secret strings like passwords and connection strings.

Vault authentication and permissions

Azure Key Vault's API uses Azure Active Directory to authenticate users and applications. Vault access policies are based on *actions*, and are applied across an entire vault. For example, an application with **Get** (read secret values), **List** (list names of all secrets), and **Set** (create or update secret values) permissions to a vault is able to create secrets, list all secret names, and get and set all secret values in that vault.

All actions performed on a vault require authentication and authorization — there is no way to grant any kind of anonymous access.

Tip

When granting vault access to developers and apps, grant only the minimum set of permissions needed. Permissions restrictions help avoid accidents caused by code bugs and reduce the impact of stolen credentials or malicious code injected into your app.

Developers will usually only need **Get** and **List** permissions to a development-environment vault. Some engineers will need full permissions to change and add secrets when necessary.

For apps, often only **Get** permissions are required. Some apps may require **List** depending on the way the app is implemented. The app we'll implement in this module's exercise requires the **List** permission because of the technique it uses to read secrets from the vault.

Vault authentication with managed identities for Azure resources

Azure Key Vault uses **Azure Active Directory** to authenticate users and applications that try to access a vault. To grant our web application access to the vault, we first need to register our app with Azure Active Directory. Registering creates an identity for the app. Once the app has an identity, we can assign vault permissions to it.

Apps and users authenticate to Key Vault using an Azure Active Directory authentication token. Getting a token from Azure Active Directory requires a secret or certificate, because anyone with a token could use the application identity to access all of the secrets in the vault.

Our application secrets are secure in the vault, but we still need to keep a secret or certificate outside of the vault in order to access them! This problem is called the *bootstrapping problem*, and Azure has a solution for it.

Managed identities for Azure resources

Managed identities for Azure resources is an Azure feature that your app can use to access Key Vault and other Azure services without having to manage even a single secret outside of the vault. Using a managed identity is a simple and secure way to take advantage of Key Vault from your web app.

When you enable managed identity on your web app, Azure activates a separate token-granting REST service specifically for use by your app. Your app will request tokens from this service instead of directly from Azure Active Directory. Your app needs to use a secret to access this service, but that secret is injected into your app's environment variables by App Service when it starts up. You don't need to manage or store this secret value anywhere, and nothing outside of your app can access this secret or the managed identity token service endpoint.

Managed identities for Azure resources also registers your app in Azure Active Directory for you, and will delete the registration if you delete the web app or disable its managed identity.

Managed identities are available in all editions of Azure Active Directory, including the Free edition included with an Azure subscription. Using it in App Service has no extra cost and requires no configuration, and it can be enabled or disabled on an app at any time.

Enabling a managed identity for a web app requires only a single Azure CLI command with no configuration. We'll do it later on when we set up an App Service app and deploy to Azure. Before that, though, we're going to apply our knowledge of managed identities to write the code for our app.

Explore the Azure Security Center service tiers

Azure Security Center has two available tiers: free and standard. Both provide security policy, assessment, and recommendations and connection with partner solutions.

Free tier

The free tier is automatically enabled on all Azure subscriptions and provides security policy, continuous security assessment, and actionable security recommendations to help you protect your Azure resources. It monitors the most common app resources in Azure including:

- Compute resources such as VMs, Azure Functions and App Service
- Network access and endpoint security
- Data storage including Azure Storage, Redis cache for Azure, and Azure SQL
- Identity and access including Azure Key Vault
- IoT Hubs and resources

Standard tier

The standard tier extends the capabilities of the free tier to workloads running in private and other public clouds to provide unified security management and threat protection across all your hybrid cloud workloads.

The Standard tier adds advanced threat detection capabilities, using analytics and machine learning to identify attacks and zero-day exploits, access and application controls to reduce exposure to network attacks and malware, and more.

Migrating your Security Center subscription from the free tier to the standard tier enables the following features:

- **Security event collection**. Security Center collects logs in a central place so you can search and analyze them to identify important security events that may require your attention.
- **Network Map**. This feature allows you to visualize the topology of your Azure network infrastructure and the traffic to your Azure VMs. It also allows you to create filters by the severity level and recommendations.
- **Just-in-time VM access**. This allows admins to grant access to a VM for a defined period of time. Limiting access helps reduce exposure to outside attacks. This feature is especially useful if you're working with an outside agency that needs to access your VM.
- **Adaptive application controls (application whitelisting)**. Adaptive application controls uses artificial intelligence to recommend applications to allow. This helps protect VMs by preventing malware and unauthorized software from being installed.
- **Regulatory compliance reports**. In the Regulatory compliance dashboard, you have a clear view of the status of all standard regulatory assessments within your environment.
- **File integrity monitoring**. This feature examines files and registries of operating system, application software, and others in Windows and Linux (computers and VMs) for changes that might indicate an attack.
- **Adaptive Network Hardening**. Adaptive Network Hardening provides recommendations to harden applied NSG rules. It uses machine learning algorithms that factors in actual traffic, known trusted configuration, threat intelligence, and other indicators of compromise, and then provides recommendations to allow traffic only from specific IP/port tuples.
- **Security alerts**. Security Center supports a variety of security alerts such as detection of potential distributed denial-of-service (DDOS) attacks. Just-in-time alerts gives you the chance to investigate evolving issues before they result in a service failure.
- **Threat intelligence**. This feature can help determine the nature of an attack, the attack point of origin, and more.
- **Workflow Automation**. Workflow automation is a collection of procedures that can be executed from Security Center once a certain playbook is triggered from selected alert. Workflow automation can help to automate and orchestrate your response to a specific security alert detected by Security Center.

Switch to the Standard tier

You can try the Standard tier for free for 30 days. This allows you to evaluate the additional features, see how your current environment will benefit from them, and decide whether they're worth the investment.

You can enable Security Center on a per-subscription basis. Each subscription can choose what elements you want to enroll. Selecting the **Coverage** item under **POLICY & COMPLIANCE** will list all your available subscriptions (Not covered, Partially covered through the Free tier or partial plan, and Fully covered on the Standard tier).

Selecting a subscription allows you to control what areas you want Security Center to monitor as shown in the following screenshot.

Customize Azure Security Center options

You can customize various global Azure Security Center settings using the **Pricing & settings** option on the Security Center menu. These settings are established on a per-subscription basis so you have complete control over what is monitored, what data is collected, and where it's stored.

There are four areas you can influence.

- **Pricing tier**. Information about the available pricing tiers. This is the same information found on the **Coverage** page.
- **Threat detection**. This lets you control how Security Center integrates with other Microsoft security services such as Windows Defender.
- **Data Collection**. You can enable *auto-provisioning* to install a monitoring agent on all VMs in your subscription so Security Center can collect security information from Windows and Linux VMs.
- **Email notifications**. Security contact details and email notifications for high security alerts.

Here's a screenshot of the **Pricing & settings** screen with the **Threat detection** area selected.

Data Collection is particularly interesting. Each VM can store audit logs based on configured settings established during the VM creation process. You can collect two log sources from every VM in the subscription:

1. **Boot-time diagnostics**. This includes console output and screenshots of the virtual machine running on a host to help diagnose startup issues.
2. **OS guest diagnostics**. Get metrics every minute for your virtual machine. You can use them to create alerts and stay informed on your applications.

You can also activate these options when you create new VMs. Here's a screenshot of the **Management** tab while creating a new Windows-based VM with the Azure portal that shows the Azure Security Center options being set:

By default, a Storage Account will be selected (or created) to hold the logs, but you can customize that on a per-VM basis as needed.

With this collected data, Azure Security Center can start making observations about how each of your configured workloads match up to your security policy.

Centralized policy management with Azure Security Center

Policy-based management can streamline IT operations and help to protect the organization by enforcing well-designed policies. Azure Policy lets you define requirements for your Azure subscriptions and tailor them to your type of workload or the sensitivity of your data.

Azure Security Center is fully integrated with Azure Policy. Security Center can monitor policy compliance across all of your subscriptions using a default set of *security policies*. A security policy defines the set of controls that are recommended for resources within the specified subscription or resource group. These security policies define the *desired* configuration of your workloads and help to ensure compliance with company or regulatory security requirements. These defaults can be customized and defined to match your specific organizational needs.

Here are a few of the built-in security policies that Security Center monitors:

- Secure transfer to storage accounts should be enabled
- Azure AD administrator for SQL server should be provisioned

- Client authentication should use Azure Active Directory
- Diagnostics logs in Key Vault should be enabled
- System updates should be installed on your machines
- Audit missing blob encryption for storage accounts
- Just-In-Time network access control should be applied on virtual machines

By default, all security policies are turned on for each monitored subscription. Security policies and recommendations are tied to each other. If you enable a security policy, such as OS vulnerabilities, that enables recommendations for that policy. In Security Center, you define policies for your Azure subscriptions or resource groups according to your company's security needs and the types of applications or sensitivity of data in each subscription.

For example, resources used for development or testing might have different security requirements than resources used for production applications. Likewise, applications that use regulated data, like personally identifiable information (PII), might require a higher level of security. Security policies that are enabled in Azure Security Center drive security recommendations and monitoring to help you identify potential vulnerabilities and mitigate threats.

Policies are inherited from the subscription down to the resource groups. However, you can control the security policies individually at the resource group level.

Note

To modify a security policy at the subscription level or resource group level, you need to be an Owner or Contributor for that subscription.

Working with security policies

You can view the active security policies through the Security Center dashboard through the Security policy view:

Two organizational groups are shown in the image above: management groups and subscriptions. These are taken directly from Azure policy. Selecting one of these elements allows you to drill into the details for that group or subscription.

In the above image, you can see that **System updates** is set to **AuditIfNotExists**. In this subscription, under the free tier, that means all virtual machines (VMs) will be audited to ensure they have the latest security updates applied. Any VMs that fail this check will generate an audit event.

You can collapse each group to see other policy areas.

In the above image, you can see that the security policy **Virtual machines should be associated with a Network Security Group** is **Disabled**.

Changing Azure policy

Owners and security administrators can edit the default security policy for each of the shown Azure subscriptions and management groups through Azure Policy. The Azure portal is the easiest way to make changes to policy, but you can also leverage a command-line interface (Azure CLI or Azure PowerShell) or the programmatic REST API.

Let's examine some of the recommendations Azure Security Center makes about your resources using these policy definitions

Monitor your security status with Security Center recommendations

The assigned security policies create security recommendations. These recommendations can help to identify the current security state of your created workloads in Azure. Security Center reviews your security recommendations across all workloads, uses algorithms to determine how critical each recommendation is, and calculates a **Secure Score** which is displayed on the Overview page. The recommendation Secure Score is a calculation based on the ratio between your healthy resources and your total resources.

You can select the **Review your secure score >** link to get more information on each subscription and the recommendations to improve your score.

Here you can identify the severity of the issue, and get help on correcting each violation. In some cases, Security Center can even fix the issue for you through the **1-Click Fix** tag as shown above.

Viewing recommendations by category

Under the **RESOURCE SECURITY HYGIENE** header in Security Center you can examine specific recommendations based on category. For example, the **Compute & apps** section of Azure Security Center provides recommendations for Azure VMs, non-Azure computers (standard-tier), App Services, Containers, and VM scale sets.

As in the secure score screen, some recommendations can be fixed directly from the Security Center dashboard while other issues require you to perform some steps on the resource. For example, in the above image, the **System updates should be installed on your machines** will only give you the list of computers that need updates. To address this issue you would use a solution such as Windows Update Services (WSUS).

Each recommendation can be selected to get more details. For issues which need manual remediation, you will get a list of steps to perform. For example, selecting the **Virtual machines should be migrated to new Azure Resource Manager resources** will show the following screen:

VMs are particularly important to protect as they often have a broader surface attack than other compute resources. Azure Security Center helps you safeguard your virtual machines in Azure by providing visibility *into* the security settings on each VM. As shown earlier, ASC can examine OS-level settings through the use of a *monitor* service that it installs into each Windows and Linux VM. With this feature enabled, Security Center can provide several safeguards including:

- OS security settings with the recommended configuration rules
- System security updates and critical updates that are missing
- Endpoint protection recommendations
- Disk encryption validation
- Vulnerability assessment and remediation
- Threat detection

Other categories

Security Center lists similar sections for Networking, IoT Hubs, Data & storage, Identity & access, and other security products such as the Next Generation Firewall and Web Application Firewall.

Try selecting each item in the **RESOURCE SECURITY HYGIENE** section to see examples of recommendations Security Center makes for each area.

Disabling security recommendations

It's recommended to leave all the security policies enabled, however sometimes a recommendation will be generated that isn't relevant to your environment. You can turn it off by disabling the security policy that is sending the recommendation.

1. In the **Policy & Compliance** section, select **Security policy**.
2. Select the subscription or management group that shouldn't show the recommendation.

   Note

   Remember that a management group applies its policies to its subscriptions. Therefore, if you disable a subscription's policy, and the subscription belongs to a management group that still uses the same policy, then you will continue to receive the policy recommendations. The policy will still be applied from the management level and the recommendations will still be generated.

3. Select the assigned policy:
4. In the **PARAMETERS** section, locate the policy that sends the recommendation you want to disable, and from the dropdown list, select **Disabled**.
5. Select **Save** to persist your changes. The change can take up to 12 hours to replicate through the Azure infrastructure.