Azure Notes

What are containers?

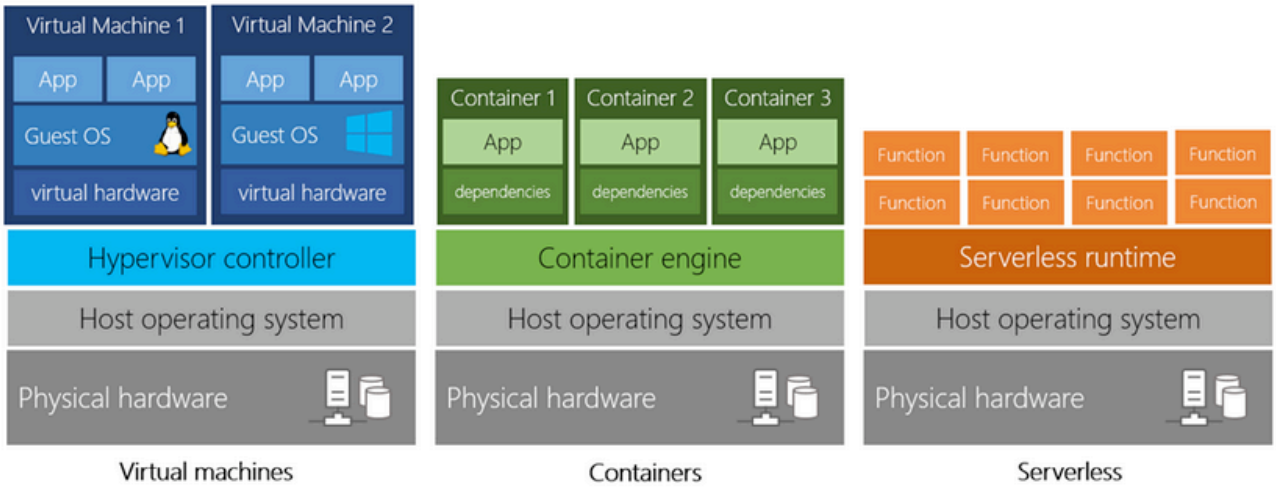
Containers provide a consistent, isolated execution environment for applications. They're similar to VMs except they don't require a guest operating system. Instead, the application and all its dependencies is packaged into a "container" and then a standard runtime environment is used to execute the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



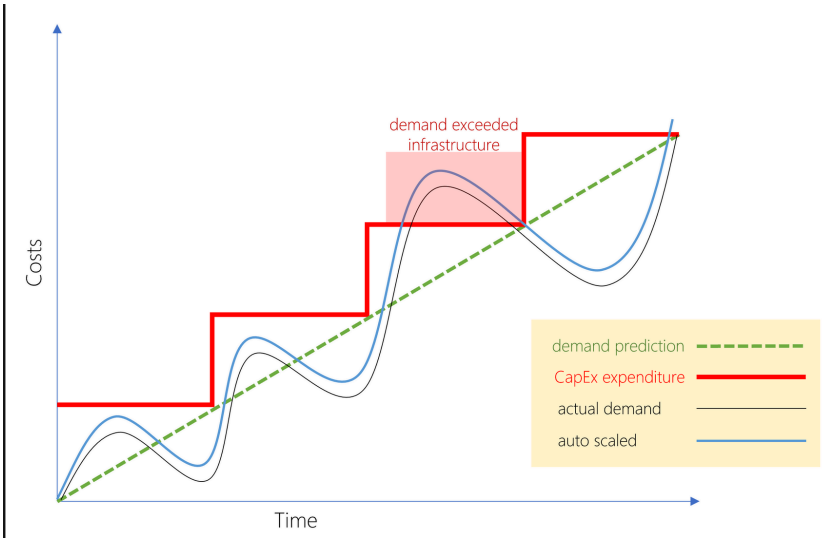
Few of the compliance offerings :

- Criminal Justice Information Services **(CJIS)**, Cloud Security Alliance **(CSA)** STAR Certification, General Data Protection Regulation **(GDPR),** Health Insurance Portability and Accountability Act **(HIPAA)**, International Organisation for Standardisation **(ISO)** and the International Electrotechnical Commission **(IEC)** 27018, Health Information Technology for Economic and Clinical Health **(HITECH),** Multi-Tier Cloud Security **(MTCS)** Singapore - 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 , UK Government G-Cloud, National Institute of Standards and Technology **(NIST)** Cybersecurity Framework **(CSF)**

- *Economies of scale* is the ability to do things more efficiently or at a lower-cost per unit when operating at a larger scale. This cost advantage is an important benefit in cloud computing.

- Cloud providers such as Microsoft, Google, and Amazon are large businesses leveraging the benefits of economies of scale. These providers can then pass the savings on to their customers.

- These savings are apparent to end users in a number of ways, one of which is the ability to acquire hardware at a lower cost. Cloud providers can also make deals with local governments and utilities to get tax savings, lowering the price of power, cooling, and high-speed network connectivity between sites. Cloud providers are then able to pass on these benefits to end users in the form of lower prices than what you could achieve on your own.

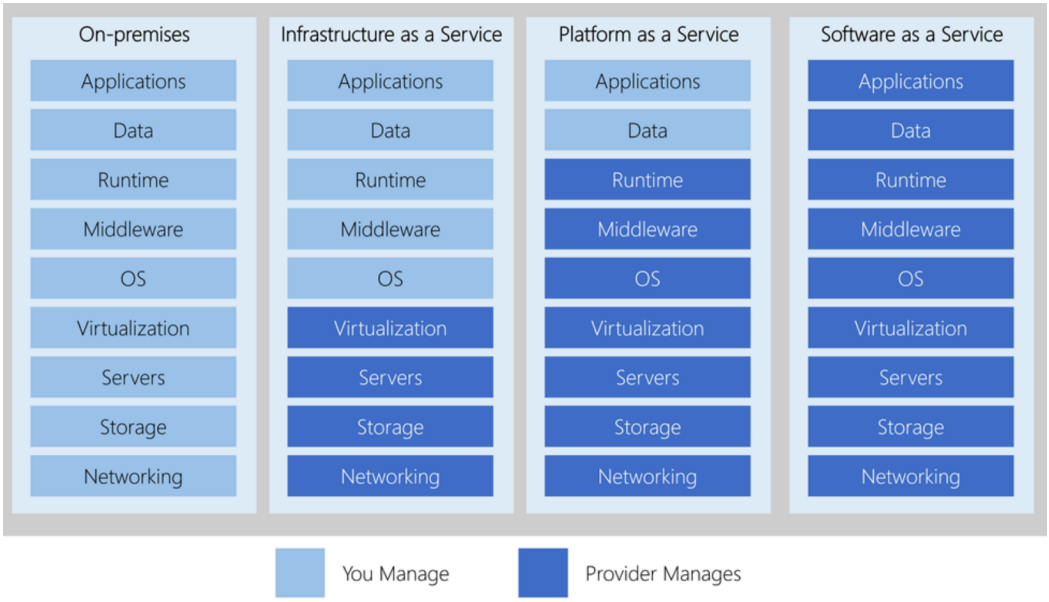


- Study Iaas, Saas, Paas from website.

- 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. (Automatic provision of resources and don't need to worry about allocating hardware)

- SaaS requires the least amount of management. The cloud provider is responsible for managing everything, and the end user just uses the software.



Azure purchasing options

With flexible purchasing options, you can choose the option that works best for you. Use one of the following three ways to buy Azure:

- **Azure.com**: Buying directly through Azure.com is the fastest and easiest way for organisations of all sizes to get started with Azure. You can manage your Azure deployments and usage yourself and get a monthly bill from Microsoft for the services used.
- **Microsoft representative**: Buying Azure through a Microsoft representative is intended for large organisations or customers who already work with one. You'll also manage your Azure deployments and usage yourself and get a monthly bill from Microsoft for the services used.
- **Microsoft partner**: If you buy Azure as a managed service through your partner, your partner will provide you with access to Azure, manage your billing, and provide support.

- When you sign up, an Azure subscription is created by default. An Azure subscription is a logical **container** used to provision resources in Azure. It holds the details of all your resources like virtual machines (VMs), databases, and more. When you create an Azure resource like a VM, you identify the subscription it belongs to. As you use the VM, the usage of the VM is aggregated and billed monthly.

Create additional Azure subscriptions

You might want to create additional subscriptions for resource or billing management purposes. For example, you might choose to create additional subscriptions to separate:

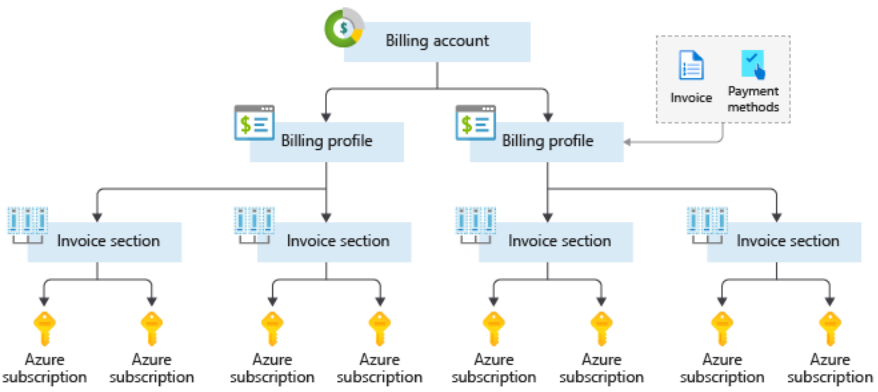
- **Environments**: When managing your resources, you can choose to create subscriptions to set up separate environments for development and testing, security, or to isolate data for compliance reasons. This is particularly useful because resource access control occurs at the subscription level.
- **Organisational structures**: You can create subscriptions to reflect different organisational structures. For example, you could limit a team to lower-cost resources, while allowing the IT department a full range. This design allows you to manage and control access to the resources that users provision within each subscription.
- **Billing**: You might want to also create additional subscriptions for billing purposes. Because costs are first aggregated at the subscription level, you might want to create subscriptions to manage and track costs based on your needs. For instance, you might want to create a subscription for your production workloads and another subscription for your development and testing workloads.

You might also need additional subscriptions due to:

- **Subscription limits**: Subscriptions are bound to some hard limitations. For example, the maximum number of Express Route circuits per subscription is 10. Those limits should be considered as you create subscriptions on your account. If there is a need to go over those limits in particular scenarios, then you might need additional subscriptions.

Customize billing to meet your needs

- If you have multiple subscriptions, you can organise them into invoice sections. Each invoice section is a line item on the invoice that shows the charges incurred that month. For example, you might need a single invoice for your organisation but want to organise charges by department, team, or project.
- Depending on your needs, you can set up multiple invoices within the same billing account. To do this, create additional billing profiles. Each billing profile has its own monthly invoice and payment method.
- The following diagram shows an overview of how billing is structured. If you've previously signed up for Azure or if your organisation has an Enterprise Agreement, your billing might be set up differently.



Azure Support Options

Azure free support resources

You have 24/7 access to the online documentation, community support, and new Azure capabilities demo videos on YouTube. Created by Azure engineers, these demo videos are available on Azure Friday, Microsoft Mechanics, and Azure portal how-to videos playlists. As an Azure customer, the following free support resources are available to you as well.

- Billing and subscription management support
- Azure Quickstart Center, a guided experience in the Azure portal available to anyone who wants to improve their knowledge of Azure
- Azure Service Health gives you insights on issues related to your Azure services
- Azure Advisor gives you personalized recommendations on how to optimize your cost and performance

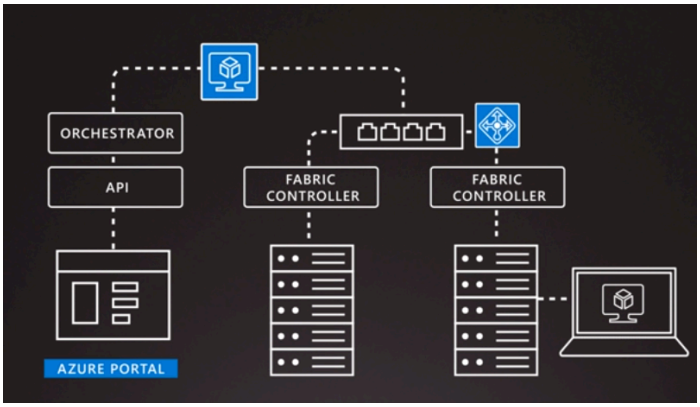
Azure support plans

Azure offers customers reactive and proactive technical support. Choose the support plan that best meet your needs. You can purchase the support plan on the Azure website or Azure portal. If you are working with a Microsoft representative or partner, you can purchase a support plan from them. Microsoft also provides support plans that cover Azure, Office 365 and Dynamics 365. Talk to your Microsoft representative or partner for more details.

	Developer	Standard	Professional Direct
Best for	Non-critical workloads	Production workloads	Business-critical workloads
Reactive technical support	1 business day response	1-hour response for critical cases	1-hour response + priority tracking of critical cases
Proactive technical support	Not applicable	Not applicable	Access to a pool of technical experts

	Basic Request support	DEVELOPER Purchase support	STANDARD Purchase support	PROFESSIONAL DIRECT Purchase support
Price	Included for all Azure customers	£21.614 per month	£74.531 per month	£745.309 per month
Scope	Included for all Azure customers	Trial and non-production environments	Production workload environments	Business-critical dependence
Billing and subscription management support	✓	✓	✓	✓
24/7 self-help resources, including Microsoft Learn, Azure portal how-to videos, documentation and community support	✓	✓	✓	✓
Ability to submit as many support tickets as you need	✓	✓	✓	✓
Azure Advisor – your free, personalised guide to Azure best practices	✓	✓	✓	✓
Azure health status and notifications	✓	✓	✓	✓
24/7 access to technical support by email and phone		Available during business hours by email only.	✓	✓

Case severity and response time		Minimal business impact (Sev C): Within eight business hours[1]	Minimal business impact (Sev C): Within eight business hours[1] Moderate business impact (Sev B): Within four hours Critical business impact (Sev A): Within one hour	Minimal business impact (Sev C): Within four business hours[1] Moderate business impact (Sev B): Within two hours Critical business impact (Sev A): Within one hour
Third-party software support with interoperability and configuration guidance and troubleshooting		✔	✔	✔
Architecture Support		General guidance	General guidance	Guidance from a pool of ProDirect delivery managers
Operations Support				Service reviews and advisory consultation from a pool of ProDirect delivery managers
Training				Webinars led by Azure engineers
Proactive Guidance				From a pool of ProDirect delivery managers



- Every server includes a hypervisor to run multiple VMs. A network switch provides connectivity to all servers. One server in each rack runs a special piece of software called a Fabric Controller which in turn is connected to another piece of software called the orchestrator.

The **Spec Picker** allows us to select a new pricing tier for our application.

- There are some global Azure services that do not require you to select a particular region, such as Microsoft Azure Active Directory, Microsoft Azure Traffic Manager, and Azure DNS. Azure has more global regions than any other cloud provider.

Special Azure regions

Azure has specialized regions that you might want to use when building out your applications for compliance or legal purposes. These include:

- *US DoD Central*, *US Gov Virginia*, *US Gov Iowa* and more: These are physical and logical network-isolated instances of Azure for US government agencies and partners. These datacenters are operated by screened US persons and include additional compliance certifications.

- *China East*, *China North* and more: These regions are available through a unique partnership between Microsoft and 21Vianet, whereby Microsoft does not directly maintain the datacenters.

Azure divides the world into *geographies* that are defined by geopolitical boundaries or country borders. An Azure geography is a discrete market typically containing two or more regions that preserve data residency and compliance boundaries. This division has several benefits.

- Geographies allow customers with specific data residency and compliance needs to keep their data and applications close.
- Geographies ensure that data residency, sovereignty, compliance, and resiliency requirements are honored within geographical boundaries.
- Geographies are fault-tolerant to withstand complete region failure through their connection to dedicated high-capacity networking infrastructure.

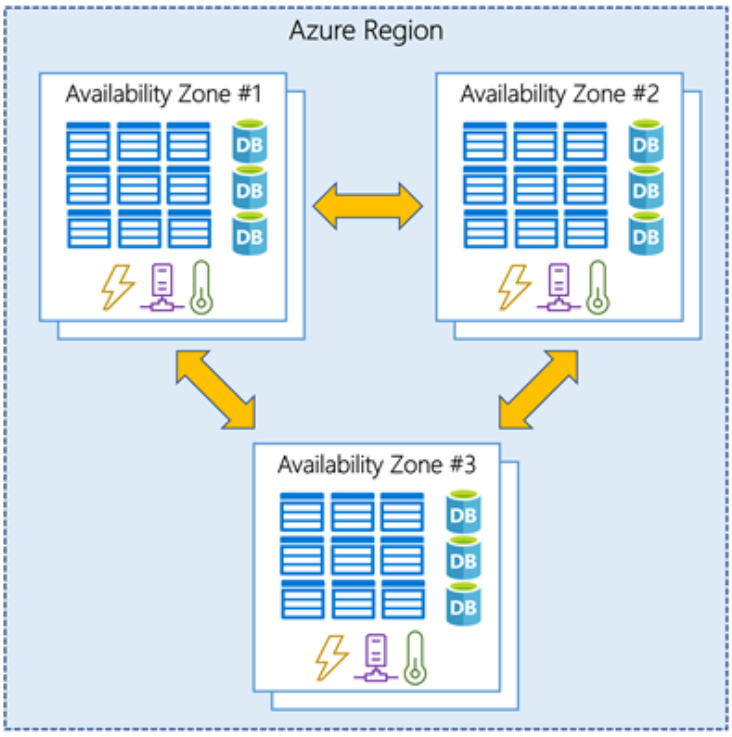
Geographies are broken up into the following areas:

- Americas
- Europe
- Asia Pacific
- Middle East and Africa

Each region belongs to a single geography and has specific service availability, compliance, and data residency/sovereignty rules applied to it.

- 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. Not every region has support for Availability Zones.



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).

- 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*.
- 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.

Azure does not provide SLAs for most services under the *Free* or *Shared* tiers. Also, free products such as Azure Advisor do not typically have an SLA.

SLAs for Azure products and services

There are three key characteristics of SLAs for Azure products and services:

1. Performance Targets
2. Uptime and Connectivity Guarantees
3. Service credits

Performance Targets

An SLA defines performance targets for an Azure product or service. The performance targets that an SLA defines are specific to each Azure product and service. For example, performance targets for some Azure services are expressed as uptime guarantees or connectivity rates.

Uptime and Connectivity Guarantees

A typical SLA specifies performance-target commitments that range from 99.9 percent ("three nines") to 99.999 percent ("five nines"), for each corresponding Azure product or service. These targets can apply to such performance criteria as uptime or response times for services.

Service Credits

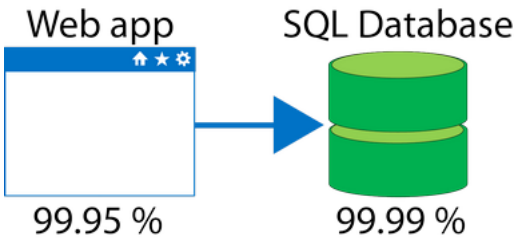
SLAs also describe how Microsoft will respond if an Azure product or service fails to perform to its governing SLA's specification.

For example, customers may have a discount applied to their Azure bill, as compensation for an under-performing Azure product or service. The table below explains this example in more detail.

- When combining SLAs across different service offerings, the resultant SLA is called a *Composite SLA*. The resulting composite SLA can provide higher or lower uptime values, depending on your application architecture.

Calculating downtime

Consider an App Service web app that writes to Azure SQL Database. These Azure services currently have the following SLAs:



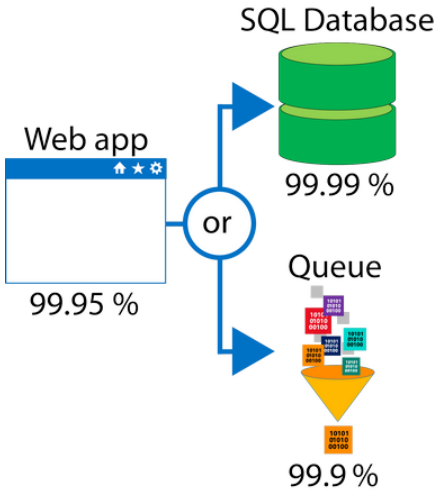
In this example, if either service fails the whole application will fail. In general, the individual probability values for each service are independent. However, the composite SLA value for this application is:

```
99.95 percent × 99.99 percent = 99.94 percent
```

For SLA and downtime calculations, remember that any time you see a number with the label or symbol for "percent" (`%`), that number is divided by 100. If you were entering the SLA calculation above, the actual values would be `0.9995 * 0.9999 = 0.9994`.

This means the **combined probability of failure** is higher than the individual SLA values. This isn't surprising, because an application that relies on multiple services has more potential failure points.

Conversely, you can improve the composite SLA by creating independent fallback paths. For example, if the SQL Database is unavailable, you can put transactions into a queue for processing at a later time.



With this design, the application is still available even if it can't connect to the database. However, it fails if both the database *and* the queue fail simultaneously.

If the expected percentage of time for a simultaneous failure is **0.0001 × 0.001**, the composite SLA for this combined path of a database *or* queue would be:

```
1.0 - (0.0001 × 0.001) = 99.99999 percent
```

Therefore, if we add the queue to our web app, the total composite SLA is:

```
99.95 percent × 99.99999 percent = ~99.95 percent
```

Notice we've improved our SLA behavior. However, there are trade-offs to using this approach: the application logic is more complicated, you are paying more to add the queue support, and there may be data-consistency issues you'll have to deal with due to retry behavior.

- You can use SLAs to evaluate how your Azure solutions meet business requirements and the needs of your clients and users. By creating your own SLAs, you can set performance targets to suit your specific Azure application. This approach is known as an *Application SLA*.
- *Resiliency* is the ability of a system to recover from failures and continue to function. It's not about avoiding failures, but responding to failures in a way that avoids downtime or data loss. The goal of resiliency is to return the application to a fully functioning state following a failure. High availability and disaster recovery are two crucial components of resiliency.
- When designing your architecture you need to design for resiliency, and you should perform a *Failure Mode Analysis* (FMA). The goal of an FMA is to identify possible points of failure and to define how the application will respond to those failures.

Availability refers to the time that a system is functional and working. Maximizing availability requires implementing measures to prevent possible service failures. However, devising preventative measures can be difficult and expensive, and often results in complex solutions.

As your solution grows in complexity, you will have more services depending on each other. Therefore, you might overlook possible failure points in your solution if you have several interdependent services.

Tip -

For example: A workload that requires 99.99 percent uptime shouldn't depend upon a service with a 99.9 percent SLA.

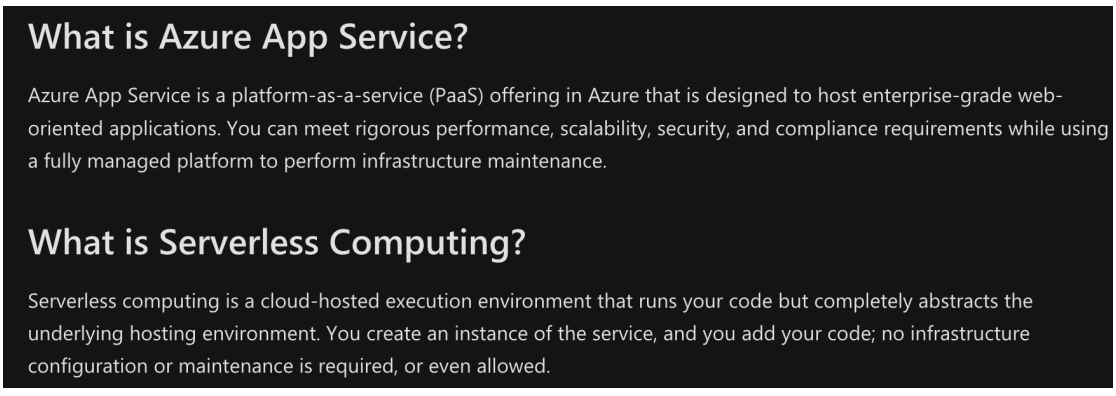
Summary - Microsoft provides more global presence than any other cloud provider with over 54 regions distributed worldwide. This infrastructure gives you the scale needed to bring your applications closer to users around the world. Azure also has dedicated regions to support government use and applications that need to be deployed in China so you can ensure data security and residency and meet compliance and resilience requirements for your customers no matter what type of business requirements you have.

- **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

- **Containers** are a virtualisation environment for running applications. Just like virtual machines, containers are 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.



- 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.

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 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

There's no cost for an availability set. You only pay for the VMs within the availability set.

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.

- 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.

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.

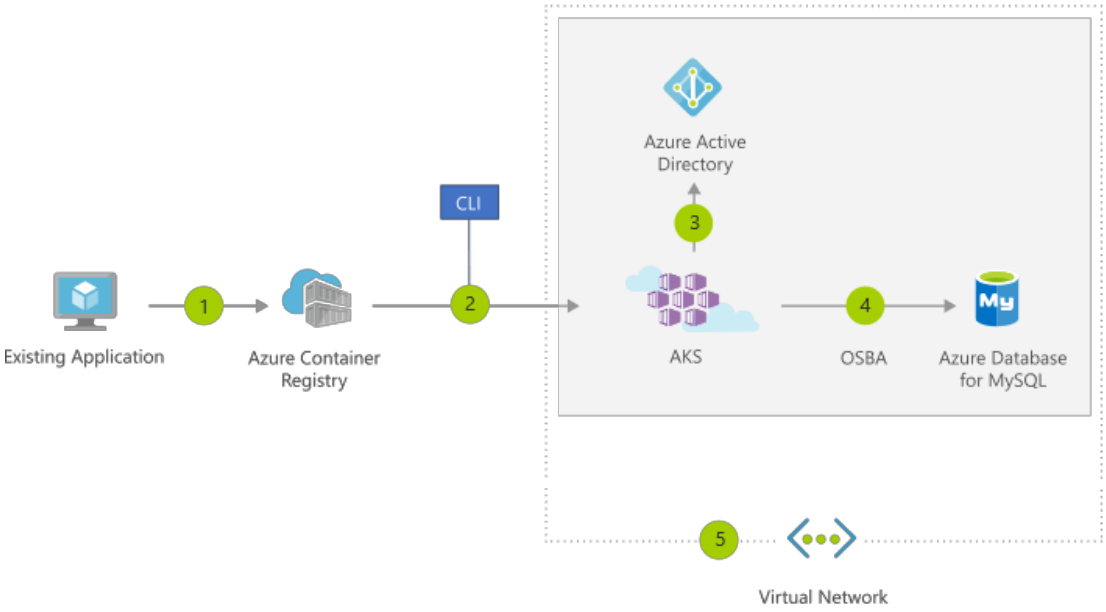
Using containers in your solutions

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.

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).



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 web apps

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

- Web Apps
- API Apps
- WebJobs
- Mobile Apps

Azure App Service handles most of the infrastructure decisions you deal with in hosting web 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.

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.

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**	**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.

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.

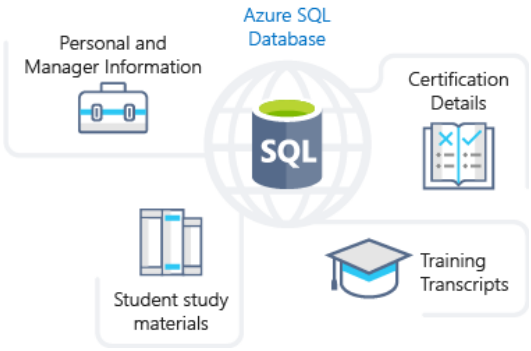
Azure SQL Database

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

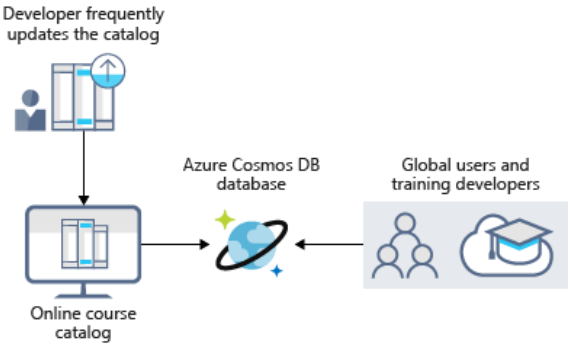
to performing a migration. Once you assess and perform any remediation required, you're ready to begin the migration process. The Azure Database Migration Service performs all of the required steps. You just change the connection string in your apps.

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.



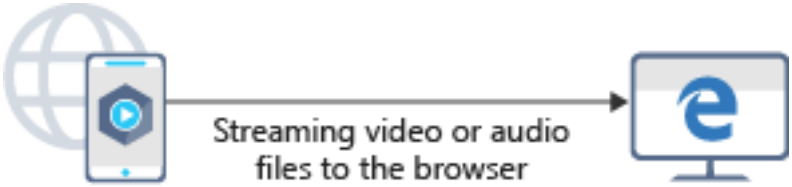
Azure Blob storage

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.

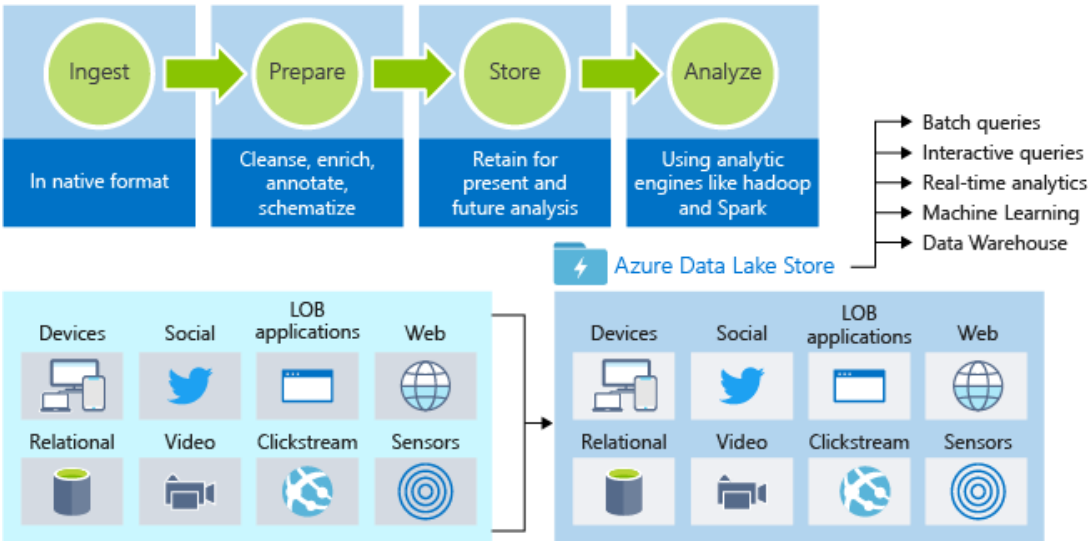
Azure Blob storage lets you stream large video or audio files directly to the user's browser from anywhere in the world. Blob storage is also used to store data for backup, disaster recovery, and archiving. It has the ability to store up to 8 TB of data for virtual machines. The following illustration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



Azure Data Lake Storage

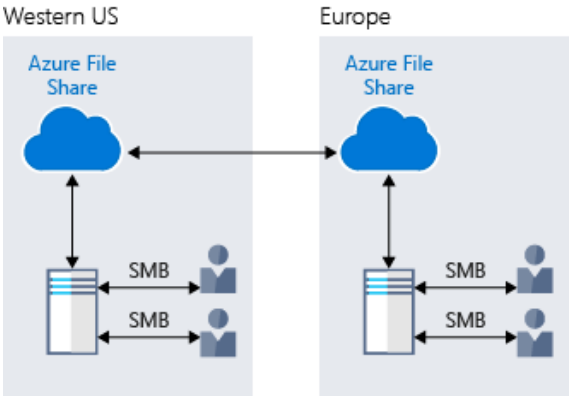
The Data Lake feature allows you to perform analytics on your data usage and prepare reports. Data Lake is a large repository that stores both structured and unstructured data.

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.



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.



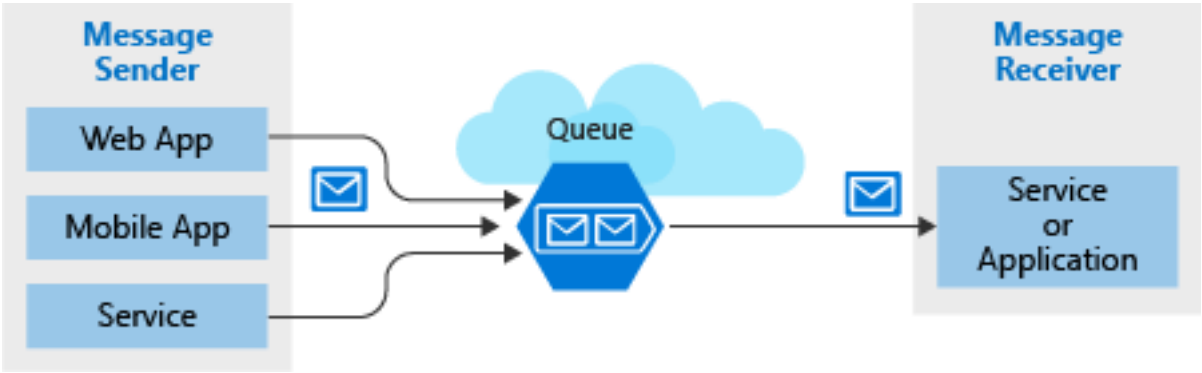
The following illustration shows Azure Files being used to share data between two geographical locations. Azure Files uses the Server Message Block (SMB) protocol that ensures the data is encrypted at rest and in transit.

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.



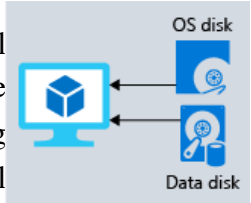
You can use queue storage to:

- 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 retrieving 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.

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.

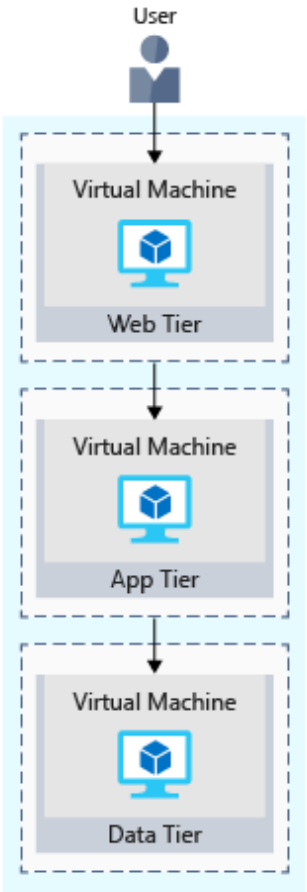
Tiers help separate concerns and are ideally designed to be reusable. Using a tiered architecture also simplifies maintenance. Tiers can be updated or replaced independently, and new tiers can be inserted if needed.

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.



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.

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, or 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.

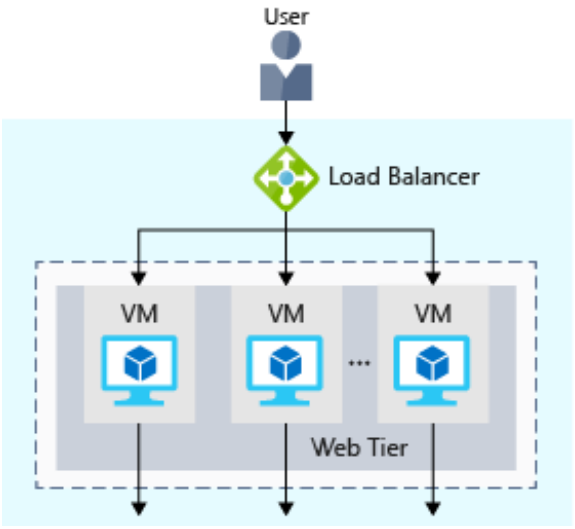
Say you start by adding additional VMs, each configured identically, to each tier. The idea is to have additional systems ready, in case one goes down, or is serving too many users at the same time.

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.



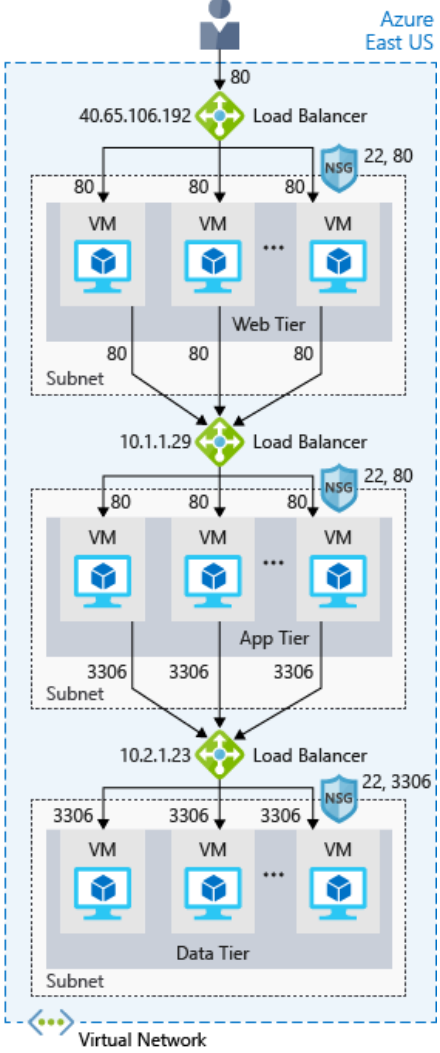
Load balancing enables you to run maintenance tasks without interrupting service. For example, you can stagger the maintenance window for each VM. During the maintenance window, the load balancer detects that the VM is unresponsive, and directs traffic to other VMs in the pool.

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.

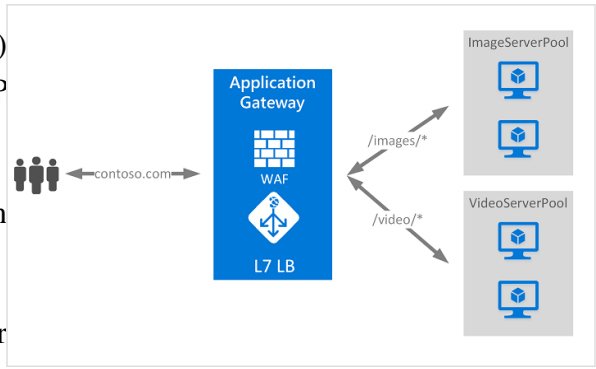
The following illustration shows the role of Azure load balancers in a multi-tier architecture.



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:

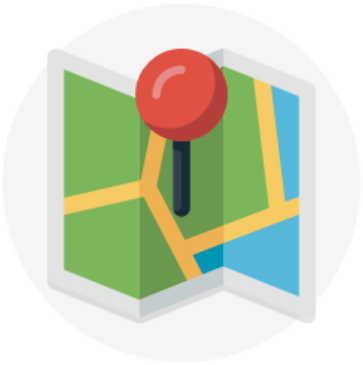
- **Cookie affinity**. Useful when you want to keep a user session on the same backend server.
- **SSL termination**. Application Gateway can manage your SSL certificates and pass unencrypted traffic to the backend servers to avoid encryption/decryption overhead. It also supports full end-to-end encryption 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?

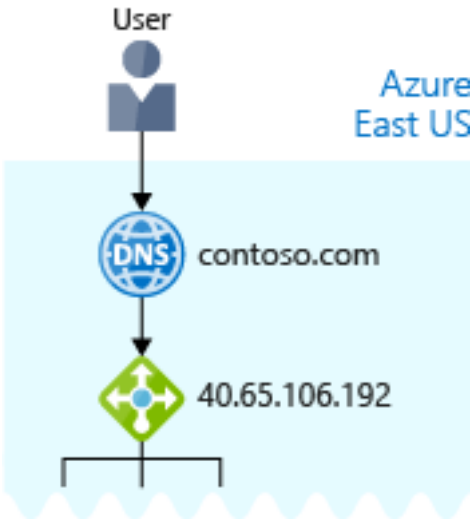


DNS, or Domain Name System, is a way to map user-friendly names to their IP addresses. You can think of DNS as the phonebook of the internet.

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.



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.

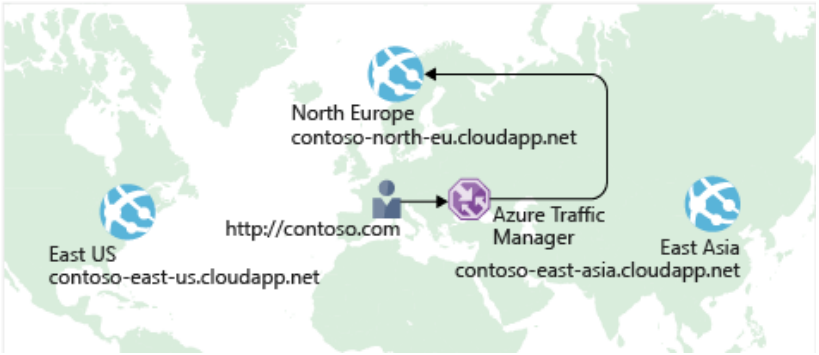
One way to reduce latency is to provide exact copies of your service in more than one region. The following illustration shows an example of global deployment.

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

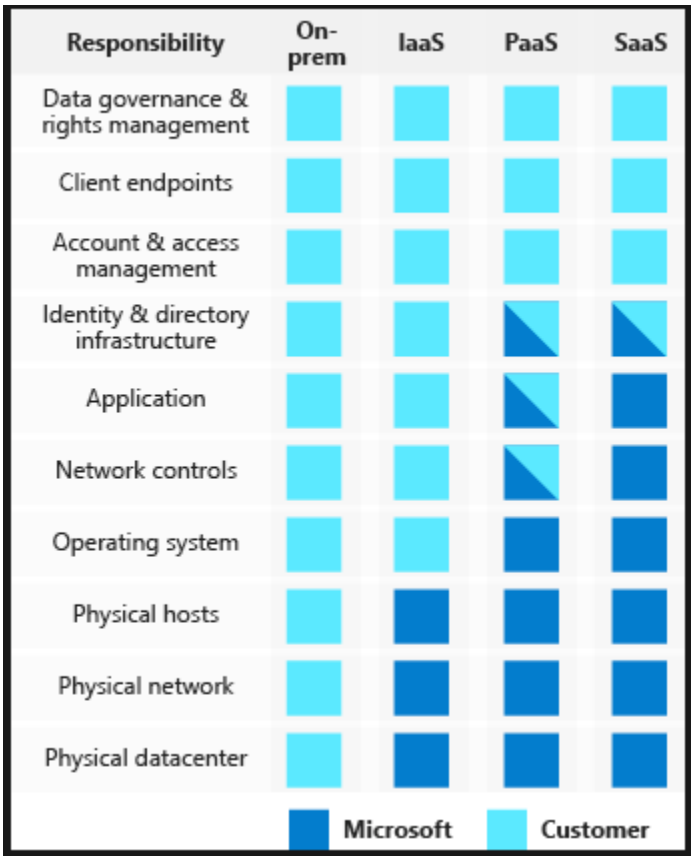
Azure Load Balancer distributes traffic within the same region to make your services more highly available and resilient. Traffic Manager works at the DNS level, and directs the client to a preferred endpoint. This endpoint can be to the region that's closest to your user.

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.

Note

Azure Event Hubs allow you to receive and process millions of events of real-time data each second via dynamic data pipelines. Event Hubs also integrates seamlessly with other Azure services.

SHARED RESPONSIBILITY MODEL

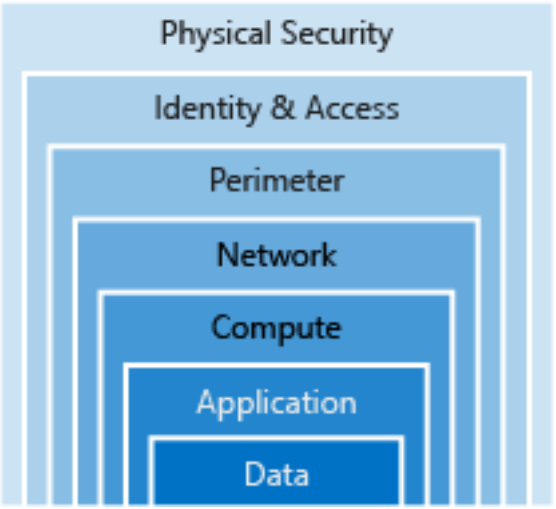


For all cloud deployment types, you own your data and identities. You are responsible for helping secure your data and identities, your on-premises resources, and the cloud components you control (which vary by service type).

Regardless of the deployment type, you always retain responsibility for the following items:

- Data
- Endpoints
- Accounts
- Access management

A layered approach to security



- *Defense in depth* is a strategy that employs a series of mechanisms to slow the advance of an attack aimed at acquiring unauthorized access to information. Each layer provides protection so that if one layer is breached, a subsequent layer is already in place to prevent further exposure. Microsoft applies a layered approach to security, both in physical data centers and across Azure services. The objective of defense in depth is to protect and prevent information from being stolen by individuals who are not authorized to access it.

- Defense in depth can be visualized as a set of concentric rings, with the data to be secured at the center. Each ring adds an additional layer of security around the data. This approach removes reliance on any single layer of protection and acts to slow down an attack and provide alert telemetry that can be acted upon, either automatically or manually. Let's take a look at each of the layers.

Data

In almost all cases, attackers are after data:

- Stored in a database
- Stored on disk inside virtual machines
- Stored on a SaaS application such as Office 365
- Stored in cloud storage

It's the responsibility of those storing and controlling access to data to ensure that it's properly secured. Often, there are regulatory requirements that dictate the controls and processes that must be in place to ensure the confidentiality, integrity, and availability of the data.

Application

- Ensure applications are secure and free of vulnerabilities.
- Store sensitive application secrets in a secure storage medium.
- Make security a design requirement for all application development.

Integrating security into the application development life cycle will help reduce the number of vulnerabilities introduced in code. We encourage all development teams to ensure their applications are

secure by default, and that they're making security requirements non-negotiable.

Compute

- Secure access to virtual machines.
- Implement endpoint protection and keep systems patched and current.

Malware, unpatched systems, and improperly secured systems open your environment to attacks. The focus in this layer is on making sure your compute resources are secure, and that you have the proper controls in place to minimize security issues.

Networking

- Limit communication between resources.
- Deny by default.
- Restrict inbound internet access and limit outbound, where appropriate.
- Implement secure connectivity to on-premises networks.

At this layer, the focus is on limiting the network connectivity across all your resources to allow only what is required. By limiting this communication, you reduce the risk of lateral movement throughout your network.

Perimeter

- Use distributed denial of service (DDoS) protection to filter large-scale attacks before they can cause a denial of service for end users.
- Use perimeter firewalls to identify and alert on malicious attacks against your network.

At the network perimeter, it's about protecting from network-based attacks against your resources. Identifying these attacks, eliminating their impact, and alerting you when they happen are important ways to keep your network secure.

Identity and access

- Control access to infrastructure and change control.

- Use single sign-on and multi-factor authentication.
- Audit events and changes.

The identity and access layer is all about ensuring identities are secure, access granted is only what is needed, and changes are logged.

Physical security

- Physical building security and controlling access to computing hardware within the data center is the first line of defense.

With physical security, the intent is to provide physical safeguards against access to assets. These safeguards ensure that other layers can't be bypassed, and loss or theft is handled appropriately.

Azure helps alleviate your security concerns. But security is still a **shared responsibility**. How much of that responsibility falls on us depends on which model we use with Azure. We use the *defense in depth* rings as a guideline for considering what protections are adequate for our data and environments.

Get tips from Azure Security Center

A great place to start when examining the security of your Azure-based solutions is **Azure Security Center**. Security Center is a monitoring service that provides threat protection across all of your services both in Azure, and on-premises. Security Center can:

- Provide security recommendations based on your configurations, resources, and networks.
- Monitor security settings across on-premises and cloud workloads, and automatically apply required security to new services as they come online.
- Continuously monitor all your services, and perform automatic security assessments to identify potential vulnerabilities before they can be exploited.
- Use machine learning to detect and block malware from being installed on your virtual machines and services. You can also define a list of allowed applications to ensure that only the apps you validate are allowed to execute.
- Analyze and identify potential inbound attacks, and help to investigate threats and any post-breach activity that might have occurred.
- Provide just-in-time access control for ports, reducing your attack surface by ensuring the network only allows traffic that you require.

Azure Security Center is part of the Center for Internet Security (CIS) recommendations.

Available tiers

Azure Security Center is available in two tiers:

1. *Free*. Available as part of your Azure subscription, this tier is limited to assessments and recommendations of Azure resources only.
2. *Standard*. This tier provides a full suite of security-related services including continuous monitoring, threat detection, just-in-time access control for ports, and more.

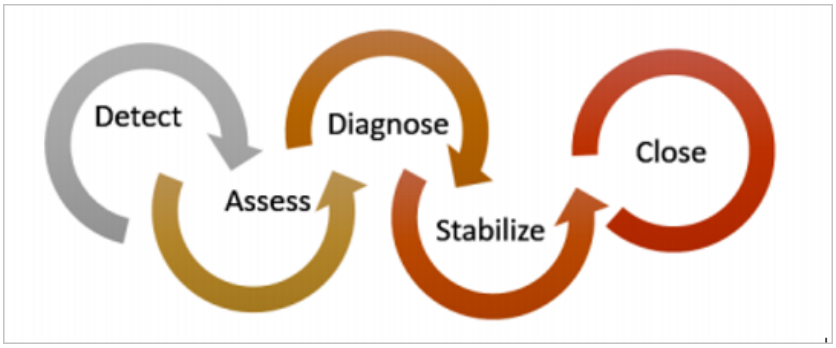
To access the full suite of Azure Security Center services, you will need to upgrade to a Standard tier subscription. You can access the 30-day free trial from within the Azure Security Center dashboard in the Azure portal. After the 30-day trial period is over, Azure Security Center is $15 per node per month.

Usage scenarios

You can integrate Security Center into your workflows and use it in many ways. Here are two examples.

1. Use Security Center for incident response.

   Many organizations learn how to respond to security incidents only after suffering an attack. To reduce costs and damage, it's important to have an incident response plan in place before an attack occurs. You can use Azure Security Center in different stages of an incident response.



   You can use Security Center during the detect, assess, and diagnose stages. Here are examples of how Security Center can be useful during the three initial incident response stages:

   - *Detect*. Review the first indication of an event investigation. For example, you can use the Security Center dashboard to review the initial verification that a high-priority security alert was raised.
   - *Assess*. Perform the initial assessment to obtain more information about the suspicious activity. For example, obtain more information about the security alert.
   - *Diagnose*. Conduct a technical investigation and identify containment, mitigation, and workaround strategies. For example, follow the remediation steps described by Security Center in that particular security alert.

2. Use Security Center recommendations to enhance security.

   You can reduce the chances of a significant security event by configuring a security policy, and then implementing the recommendations provided by Azure Security Center.

- A *security policy* defines the set of controls that are recommended for resources within that specified subscription or resource group. In Security Center, you define policies according to your company's security requirements.
- Security Center analyzes the security state of your Azure resources. When Security Center identifies potential security vulnerabilities, it creates recommendations based on the controls set in the security policy. The recommendations guide you through the process of configuring the needed security controls. For example, if you have workloads that do not require the *Azure SQL Database Transparent Data Encryption* (TDE) policy, turn off the policy at the subscription level and enable it only in the resources groups where SQL TDE is required.

Important

To upgrade a subscription to the Standard tier, you must be assigned the role of *Subscription Owner*, *Subscription Contributor*, or *Security Admin*.

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 Office 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 Office 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.

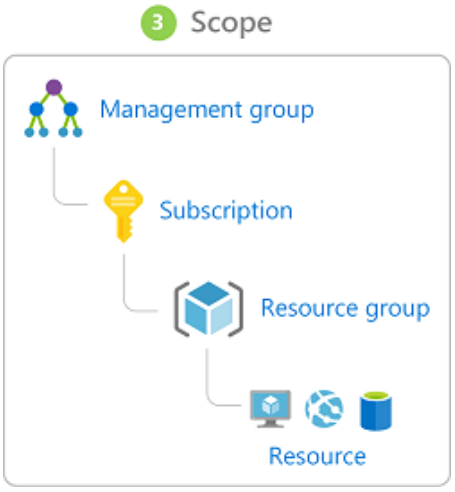
Role-based access control

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.

What is encryption?

Encryption is the process of making data unreadable and unusable to unauthorized viewers. To use or read the encrypted data, it must be *decrypted*, which requires the use of a secret key. There are two top-level types of encryption: **symmetric** and **asymmetric**.

Symmetric encryption uses the same key to encrypt and decrypt the data. Consider a desktop password manager application. You enter your passwords and they are encrypted with your own personal key (your key is often derived from your master password). When the data needs to be retrieved, the same key is used, and the data is decrypted.

Asymmetric encryption uses a public key and private key pair. Either key can encrypt but a single key can't decrypt its own encrypted data. To decrypt, you need the paired key. Asymmetric encryption is used for things like Transport Layer Security (TLS) (used in HTTPS) and data signing.

Both symmetric and asymmetric encryption play a role in properly securing your data. Encryption is typically approached in two ways:

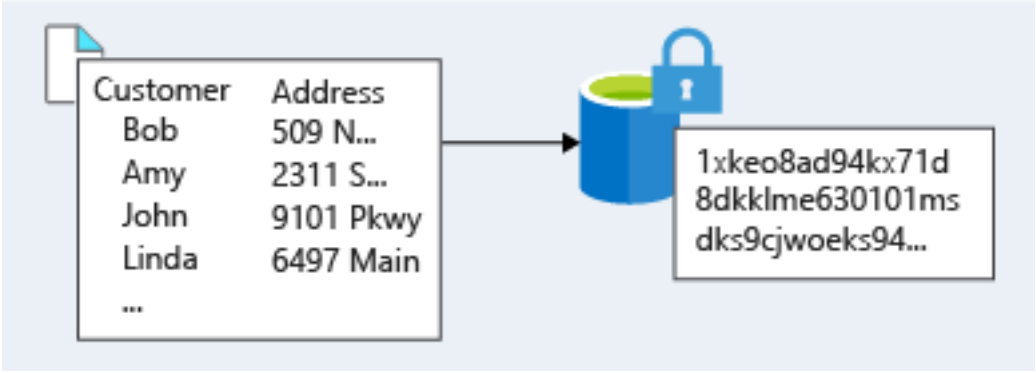
1. Encryption at rest
2. Encryption in transit

Encryption at rest

Data at rest is the data that has been stored on a physical medium. This data could be stored on the disk of a server, data stored in a database, or data stored in a storage account. Regardless of the storage mechanism, encryption of data at rest ensures that the stored data is unreadable without the keys and secrets needed to decrypt it. If an attacker was to obtain a hard drive with encrypted data and did not have access to the encryption keys, the attacker would not compromise the data without great difficulty.

The actual data that is encrypted could vary in its content, usage, and importance to the organization. This financial information could be critical to the business, intellectual property that has been developed by the business, personal data about customers or employees that the business stores, and even the keys and secrets used for the encryption of the data itself.

Here's a diagram that shows what encrypted customer data might look like as it sits in a database.



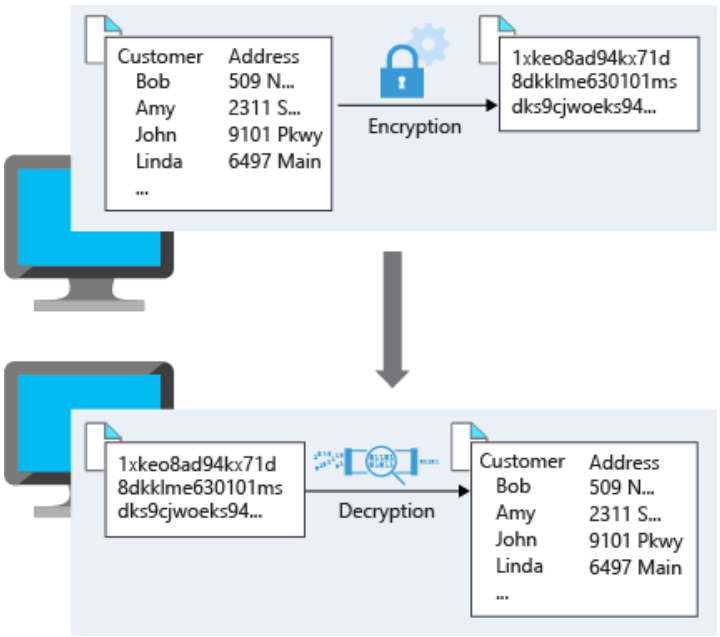
Encryption in transit

Data in transit is the data actively moving from one location to another, such as across the internet or through a private network. Secure transfer can be handled by several different layers. It could be done by encrypting the data at the application layer prior to sending it over a network. HTTPS is an example of application layer in transit encryption.

You can also set up a secure channel, like a virtual private network (VPN), at a network layer, to transmit data between two systems.

Encrypting data in transit protects the data from outside observers and provides a mechanism to transmit data while limiting risk of exposure.

This diagram shows the process. Here, customer data is encrypted as it's sent over the network. Only the receiver has the secret key that can decrypt the data to a usable form.



Encryption on Azure

Let's take a look at some ways that Azure enables you to encrypt data across services.

Encrypt raw storage

Azure Storage Service Encryption for data at rest helps you protect your data to meet your organizational security and compliance commitments. With this feature, the Azure storage platform automatically encrypts your data before persisting it to Azure Managed Disks, Azure Blob storage, Azure Files, or Azure Queue storage, and decrypts the data before retrieval. The handling of encryption, encryption at rest, decryption, and key management in Storage Service Encryption is transparent to applications using the services.

Encrypt virtual machine disks

Storage Service Encryption provides low-level encryption protection for data written to physical disk, but how do you protect the virtual hard disks (VHDs) of virtual machines? If malicious attackers gained access to your Azure subscription and got the VHDs of your virtual machines, how would you ensure they would be unable to access the stored data?

Azure Disk Encryption is a capability that helps you encrypt your Windows and Linux IaaS virtual machine disks. Azure Disk Encryption leverages the industry-standard BitLocker feature of Windows and the dm-crypt feature of Linux to provide volume encryption for the OS and data disks. The solution is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets (and you can use managed service identities for accessing Key Vault).

For Contoso Shipping, using VMs was one of the first moves toward the cloud. Having all the VHDs encrypted is an easy, low-impact way to ensure that you are doing all you can to secure your company's data.

Encrypt databases

Transparent data encryption (TDE) helps protect Azure SQL Database and Azure Data Warehouse against the threat of malicious activity. It performs real-time encryption and decryption of the database, associated backups, and

transaction log files at rest without requiring changes to the application. By default, TDE is enabled for all newly deployed Azure SQL Database instances.

TDE encrypts the storage of an entire database by using a symmetric key called the database encryption key. By default, Azure provides a unique encryption key per logical SQL Server instance and handles all the details. Bring your own key (BYOK) is also supported with keys stored in Azure Key Vault (see below).

Because TDE is enabled by default, you are confident that Contoso Shipping has the proper protections in place for data stored in the company's databases.

Encrypt secrets

We've seen that the encryption services all use keys to encrypt and decrypt data, so how do we ensure that the keys themselves are secure? Corporations may also have passwords, connection strings, or other sensitive pieces of information that they need to securely store. In Azure, we can use **Azure Key Vault** to protect our secrets.

Azure Key Vault is a centralized cloud service for storing your application secrets. Key Vault helps you control your applications' secrets by keeping them in a single, central location and by providing secure access, permissions control, and access logging capabilities. It is useful for a variety of scenarios:

- *Secrets management*. You can use Key Vault to securely store and tightly control access to tokens, passwords, certificates, *Application Programming Interface* (API) keys, and other secrets.
- *Key management*. You also can use Key Vault as a key management solution. Key Vault makes it easier to create and control the encryption keys used to encrypt your data.
- *Certificate management*. Key Vault lets you provision, manage, and deploy your public and private *Secure Sockets Layer/ Transport Layer Security* (SSL/ TLS) certificates for your Azure, and internally connected, resources more easily.
- *Store secrets backed by hardware security modules* (HSMs). The secrets and keys can be protected either by software, or by FIPS 140-2 Level 2 validated HSMs.

The benefits of using Key Vault include:

- *Centralized application secrets*. Centralizing storage for application secrets allows you to control their distribution, and reduces the chances that secrets may be accidentally leaked.
- *Securely stored secrets and keys*. Azure uses industry-standard algorithms, key lengths, and HSMs, and access requires proper authentication and authorization.
- *Monitor access and use*. Using Key Vault, you can monitor and control access to company secrets.

- *Simplified administration of application secrets*. Key Vault makes it easier to enroll and renew certificates from public Certificate Authorities (CAs). You can also scale up and replicate content within regions, and use standard certificate management tools.
- *Integrate with other Azure services*. You can integrate Key Vault with storage accounts, container registries, event hubs, and many more Azure services.

Because Azure AD identities can be granted access to use Azure Key Vault secrets, applications with managed service identities enabled can automatically and seamlessly acquire the secrets they need.

Overview of Azure certificates

As mentioned previously, Transport Layer Security (TLS) is the basis for encryption of website data in transit. TLS uses *certificates* to encrypt and decrypt data. However, these certificates have a lifecycle that requires administrator management. A common security problem with websites is having expired TLS certificates that open security vulnerabilities.

Certificates used in Azure are **x.509 v3** and can be signed by a trusted certificate authority, or they can be self-signed. A self-signed certificate is signed by its own creator; therefore, it is not trusted by default. Most browsers can ignore this problem. However, you should only use self-signed certificates when developing and testing your cloud services. These certificates can contain a private or a public key and have a thumbprint that provides a means to identify a certificate in an unambiguous way. This thumbprint is used in the Azure configuration file to identify which certificate a cloud service should use.

Types of certificates

Certificates are used in Azure for two primary purposes and are given a specific designation based on their intended use.

1. **Service certificates** are used for cloud services
2. **Management certificates** are used for authenticating with the management API

Service certificates

Service certificates are attached to cloud services and enable secure communication to and from the service. For example, if you deploy a web site, you would want to supply a certificate that can authenticate an exposed HTTPS endpoint. Service certificates, which are defined in your service definition, are automatically deployed to the VM that is running an instance of your role.

You can upload service certificates to Azure either using the Azure portal or by using the classic deployment model. Service certificates are associated with a specific cloud service. They are assigned to a deployment in the service definition file.

You can manage service certificates separately from your services, and you can have different people managing them. For example, a developer could upload a service package that refers to a certificate that an IT manager has previously uploaded to Azure. An IT manager can manage and renew that certificate (changing the configuration of the service) without needing to upload a new service package. Updating without a new service package is possible because the logical name, store name, and location of the certificate is in the service definition file, while the certificate thumbprint is specified in the service configuration file. To update the certificate, it's only necessary to upload a new certificate and change the thumbprint value in the service configuration file.

Management certificates

Management certificates allow you to authenticate with the classic deployment model. Many programs and tools (such as Visual Studio or the Azure SDK) use these certificates to automate configuration and deployment of various Azure services. However, these types of certificates are not related to cloud services.

Using Azure Key Vault with certificates

You can store your certificates in Azure Key Vault - much like any other secret. However, Key Vault provides additional features above and beyond the typical certificate management.

- You can create certificates in Key Vault, or import existing certificates
- You can securely store and manage certificates without interaction with private key material.
- You can create a policy that directs Key Vault to manage the life cycle of a certificate.
- You can provide contact information for notification about life-cycle events of expiration and renewal of certificate.
- You can automatically renew certificates with selected issuers - Key Vault partner x509 certificate providers / certificate authorities.

Automating certificate management helps to reduce or eliminate the error prone task of manual certificate management.

What is a Firewall?

A firewall is a service that grants server access based on the originating IP address of each request. You create firewall rules that specify ranges of IP addresses. Only clients from these granted IP addresses will be allowed to access the server. Firewall rules, generally speaking, also include specific network protocol and port information.

To provide inbound protection at the perimeter, you have several choices.

- **Azure Firewall** is a managed, cloud-based, network security service that protects your Azure Virtual Network resources. It is a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability. Azure Firewall provides inbound protection for non-HTTP/S protocols. Examples of non-HTTP/S protocols include: Remote Desktop Protocol (RDP), Secure Shell (SSH), and File Transfer Protocol (FTP). It also provides

outbound, network-level protection for all ports and protocols, and application-level protection for outbound HTTP/S.

- **Azure Application Gateway** is a load balancer that includes a Web Application Firewall (WAF) that provides protection from common, known vulnerabilities in websites. It is designed to protect HTTP traffic.

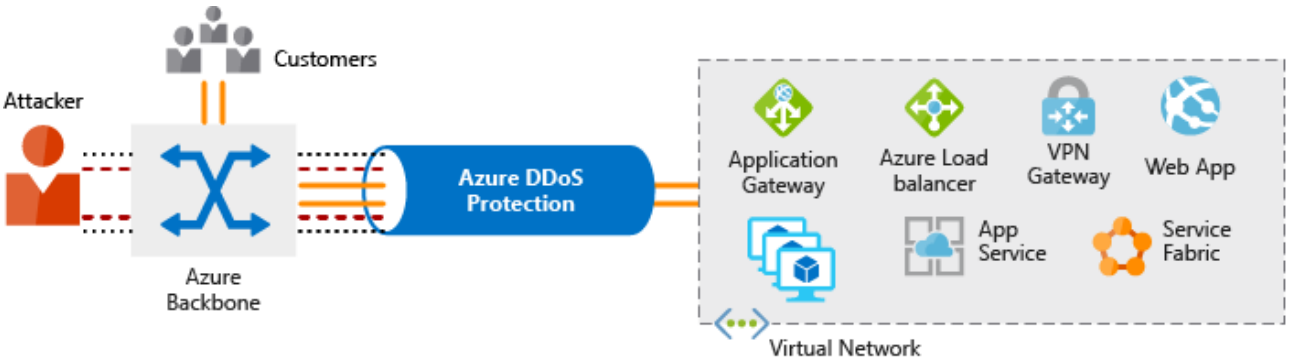
- **Network virtual appliances (NVAs)** are ideal options for non-HTTP services or advanced configurations, and are similar to hardware firewall appliances.

Stopping Distributed Denial of Service (DDoS) attacks

Any resource exposed on the internet is at risk of being attacked by a denial of service attack. These types of attacks attempt to overwhelm a network resource by sending so many requests that the resource becomes slow or unresponsive.

When you combine **Azure DDoS Protection** with application design best practices, you help provide defense against DDoS attacks. DDoS Protection leverages the scale and elasticity of Microsoft's global network to bring DDoS mitigation capacity to every Azure region. The Azure DDoS Protection service protects your Azure applications by monitoring traffic at the Azure network edge before it can impact your service's availability. Within a few minutes of attack detection, you are notified using Azure Monitor metrics.

This diagram shows network traffic flowing into Azure from both customers and an attacker. Azure DDoS protection identifies the attacker's attempt to overwhelm the network and blocks further traffic from reaching Azure services. Legitimate traffic from customers still flows into Azure without any interruption of service.



Azure DDoS Protection provides the following service tiers:

- **Basic** - The Basic service tier is automatically enabled as part of the Azure platform. Always-on traffic monitoring and real-time mitigation of common network-level attacks

provide the same defenses that Microsoft's online services use. Azure's global network is used to distribute and mitigate attack traffic across regions.

- **Standard** - The Standard service tier provides additional mitigation capabilities that are tuned specifically to Microsoft Azure Virtual Network resources. DDoS Protection Standard is simple to enable and requires no application changes. Protection policies are tuned through dedicated traffic monitoring and machine learning algorithms. Policies are applied to public IP addresses associated with resources deployed in virtual networks, such as Azure Load Balancer and Application Gateway. DDoS standard protection can mitigate the following types of attacks:
    - Volumetric attacks. The attackers goal 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 (application) layer attacks. These attacks target web application packets to disrupt the transmission of data between hosts.

Controlling the traffic inside your virtual network

Virtual network security

Once inside a virtual network (VNet), it's crucial that you limit communication between resources to only what is required.

For communication between virtual machines, *Network Security Groups* (NSGs) are a critical piece to restrict unnecessary communication.

Network Security Groups allow you to filter network traffic to and from Azure resources in an Azure virtual network. An NSG can contain multiple inbound and outbound security rules that enable you to filter traffic to and from resources by source and destination IP address, port, and protocol. They provide a list of allowed and denied communication to and from network interfaces and subnets, and are fully customizable.

You can completely remove public internet access to your services by restricting access to service endpoints. With service endpoints, Azure service access can be limited to your virtual network.

Network integration

It's common to have existing network infrastructure that needs to be integrated to provide communication from on-premises networks or to provide improved communication between services in Azure. There are a few key ways to handle this integration and improve the security of your network.

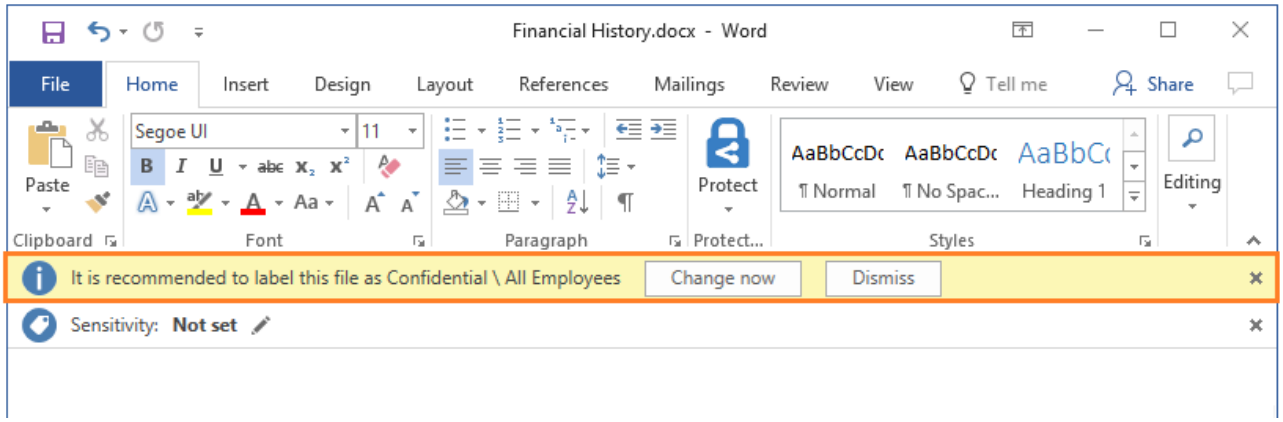
Virtual private network (VPN) connections are a common way of establishing secure communication channels between networks. Connections between Azure Virtual Network and an on-premises VPN device are a great way to provide secure communication between your network and your VNet on Azure.

To provide a dedicated, private connection between your network and Azure, you can use Azure ExpressRoute. ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Office 365, and Dynamics 365. ExpressRoute connections improve the security of your on-premises communication by sending this traffic over the private circuit instead of over the public internet. You don't need to allow access to these services for your end users over the public internet, and you can send this traffic through appliances for further traffic inspection.

Microsoft Azure Information Protection (sometimes referred to as AIP) is a cloud-based solution that helps organizations classify and optionally protect documents and emails by applying labels.

Labels can be applied automatically based on rules and conditions. Labels can also be applied manually. You can also guide users to choose recommended labels with a combination of automatic and manual steps.

The following screen capture is an example of AIP in action on a user's computer. In this example, the administrator has configured a label with rules that detect sensitive data. When a user saves a Microsoft Word document containing a credit card number, a custom tooltip is displayed. The tooltip recommends labeling the file as *Confidential - All Employees*. This label is configured by the administrator. Using this label classifies the document and protects it.



Note

You can purchase AIP either as a standalone solution, or through one of the following Microsoft licensing suites: Enterprise Mobility + Security, or Microsoft 365 Enterprise.

Azure Advanced Threat Protection

Azure Advanced Threat Protection (Azure ATP) is a cloud-based security solution that identifies, detects, and helps you investigate advanced threats, compromised identities, and malicious insider actions directed at your organization.

Azure ATP is capable of detecting known malicious attacks and techniques, security issues, and risks against your network.

Azure ATP components

Azure ATP consists of several components.

Azure ATP portal

Azure ATP has its own portal, through which you can monitor and respond to suspicious activity. The Azure ATP portal allows you to create your Azure ATP instance, and view the data received from Azure ATP sensors. You can also use the portal to monitor, manage, and investigate threats in your network environment. You can sign in to the Azure ATP portal at https://portal.atp.azure.com . Your user accounts must be assigned to an Azure AD security group that has access to the Azure ATP portal to be able to sign in.

Azure ATP sensor

Azure ATP sensors are installed directly on your domain controllers. The sensor monitors domain controller traffic without requiring a dedicated server or configuring port mirroring.

Azure ATP cloud service

Azure ATP cloud service runs on Azure infrastructure and is currently deployed in the United States, Europe, and Asia. Azure ATP cloud service is connected to Microsoft's intelligent security graph.

Purchasing Azure Advanced Threat Protection

Azure ATP is available as part of the Enterprise Mobility + Security E5 suite (EMS E5) and as a standalone license. You can acquire a license directly from the Enterprise Mobility + Security Pricing Options page or through the Cloud Solution Provider (CSP) licensing model. It is not available to purchase via the Azure portal.