**Describe Cloud computing:**

Cloud computing is the delivery of computing services over the internet. Computing services include common IT infrastructure such as virtual machines, storage, databases, and networking. Cloud services also expand the traditional IT offerings to include things like Internet of Things (IoT), machine learning (ML), and artificial intelligence (AI).

Because cloud computing uses the internet to deliver these services, it doesn’t have to be constrained by physical infrastructure the same way that a traditional datacenter is. That means if you need to increase your IT infrastructure rapidly, you don’t have to wait to build a new datacenter—you can use the cloud to rapidly expand your IT footprint.

**Shared responsibility model:**

Start with a traditional corporate datacenter. The company is responsible for maintaining the physical space, ensuring security, and maintaining or replacing the servers if anything happens. The IT department is responsible for maintaining all the infrastructure and software needed to keep the datacenter up and running. They’re also likely to be responsible for keeping all systems patched and on the correct version

With the shared responsibility model, these responsibilities get shared between the cloud provider and the consumer. Physical security, power, cooling, and network connectivity are the responsibility of the cloud provider. The consumer isn’t collocated with the datacenter, so it wouldn’t make sense for the consumer to have any of those responsibilities.

At the same time, the consumer is responsible for the data and information stored in the cloud. (You wouldn’t want the cloud provider to be able to read your information.) The consumer is also responsible for access security, meaning you only give access to those who need it.

Then, for some things, the responsibility depends on the situation. If you’re using a cloud SQL database, the cloud provider would be responsible for maintaining the actual database. However, you’re still responsible for the data that gets ingested into the database. If you deployed a virtual machine and installed an SQL database on it, you’d be responsible for database patches and updates, as well as maintaining the data and information stored in the database.

With an on-premises datacenter, you’re responsible for everything. With cloud computing, those responsibilities shift. The shared responsibility model is heavily tied into the cloud service types (covered later in this learning path): infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). IaaS places the most responsibility on the consumer, with the cloud provider being responsible for the basics of physical security, power, and connectivity. On the other end of the spectrum, SaaS places most of the responsibility with the cloud provider. PaaS, being a middle ground between IaaS and SaaS, rests somewhere in the middle and evenly distributes responsibility between the cloud provider and the consumer.

You’ll always be responsible for:

* The information and data stored in the cloud
* Devices that are allowed to connect to your cloud (cell phones, computers, and so on)
* The accounts and identities of the people, services, and devices within your organization

The cloud provider is always responsible for:

* The physical datacenter
* The physical network
* The physical hosts

Your service model will determine responsibility for things like:

* Operating systems
* Network controls
* Applications
* Identity and infrastructure

**Define Cloud models:**

The cloud models define the deployment type of cloud resources. The three main cloud models are: private, public, and hybrid.

Private Cloud:

Let’s start with a private cloud. A private cloud is, in some ways, the natural evolution from a corporate datacenter. It’s a cloud (delivering IT services over the internet) that’s used by a single entity. Private cloud provides much greater control for the company and its IT department. However, it also comes with greater cost and fewer of the benefits of a public cloud deployment. Finally, a private cloud may be hosted from your on-site datacenter. It may also be hosted in a dedicated datacenter offsite, potentially even by a third party that has dedicated that datacenter to your company.

Public Cloud:

A public cloud is built, controlled, and maintained by a third-party cloud provider. With a public cloud, anyone that wants to purchase cloud services can access and use resources. The general public availability is a key difference between public and private clouds.

Hybrid Cloud:

A hybrid cloud is a computing environment that uses both public and private clouds in an inter-connected environment. A hybrid cloud environment can be used to allow a private cloud to surge for increased, temporary demand by deploying public cloud resources. Hybrid cloud can be used to provide an extra layer of security. For example, users can flexibly choose which services to keep in public cloud and which to deploy to their private cloud infrastructure.

|  |  |  |
| --- | --- | --- |
| Public cloud | Private cloud | Hybrid Cloud |
| No capital expenditures to scale up | Organizations have complete control over resources and security | Provides the most flexibility |
| Applications can be quickly provisioned and deprovisioned | Data is not collocated with other organizations’ data | Organizations determine where to run their applications |
| Organizations pay only for what they use | Hardware must be purchased for startup and maintenance | Organizations control security, compliance, or legal requirements |
| Organizations don’t have complete control over resources and security | Organizations are responsible for hardware maintenance and updates |  |

Multi-cloud:

A fourth, and increasingly likely scenario is a multi-cloud scenario. In a multi-cloud scenario, you use multiple public cloud providers. Maybe you use different features from different cloud providers. Or maybe you started your cloud journey with one provider and are in the process of migrating to a different provider. Regardless, in a multi-cloud environment you deal with two (or more) public cloud providers and manage resources and security in both environments.

Azure Arc:

Azure Arc is a set of technologies that helps manage your cloud environment. Azure Arc can help manage your cloud environment, whether it's a public cloud solely on Azure, a private cloud in your datacenter, a hybrid configuration, or even a multi-cloud environment running on multiple cloud providers at once.

Azure VMware Solution:

What if you’re already established with VMware in a private cloud environment but want to migrate to a public or hybrid cloud? Azure VMware Solution lets you run your VMware workloads in Azure with seamless integration and scalability.

**Describe the Consumption-based model:**

When comparing IT infrastructure models, there are two types of expenses to consider. Capital expenditure (CapEx) and operational expenditure (OpEx).

CapEx is typically a one-time, up-front expenditure to purchase or secure tangible resources. A new building, repaving the parking lot, building a datacenter, or buying a company vehicle are examples of CapEx.

In contrast, OpEx is spending money on services or products over time. Renting a convention center, leasing a company vehicle, or signing up for cloud services are all examples of OpEx.

Cloud computing falls under OpEx because cloud computing operates on a consumption-based model. With cloud computing, you don’t pay for the physical infrastructure, the electricity, the security, or anything else associated with maintaining a datacenter. Instead, you pay for the IT resources you use. If you don’t use any IT resources this month, you don’t pay for any IT resources.

This consumption-based model has many benefits, including:

* No upfront costs.
* No need to purchase and manage costly infrastructure that users might not use to its fullest potential.
* The ability to pay for more resources when they're needed.
* The ability to stop paying for resources that are no longer needed.

With a traditional datacenter, you try to estimate the future resource needs. If you overestimate, you spend more on your datacenter than you need to and potentially waste money. If you underestimate, your datacenter will quickly reach capacity and your applications and services may suffer from decreased performance. Fixing an under-provisioned datacenter can take a long time. You may need to order, receive, and install more hardware. You'll also need to add power, cooling, and networking for the extra hardware.

In a cloud-based model, you don’t have to worry about getting the resource needs just right. If you find that you need more virtual machines, you add more. If the demand drops and you don’t need as many virtual machines, you remove machines as needed. Either way, you’re only paying for the virtual machines that you use, not the “extra capacity” that the cloud provider has on hand.

**Compare cloud pricing models:**

Cloud computing is the delivery of computing services over the internet by using a pay-as-you-go pricing model. You typically pay only for the cloud services you use, which helps you:

* Plan and manage your operating costs.
* Run your infrastructure more efficiently.
* Scale as your business needs change.

To put it another way, cloud computing is a way to rent compute power and storage from someone else’s datacenter. You can treat cloud resources like you would resources in your own datacenter. However, unlike in your own datacenter, when you're done using cloud resources, you give them back. You’re billed only for what you use.

Instead of maintaining CPUs and storage in your datacenter, you rent them for the time that you need them. The cloud provider takes care of maintaining the underlying infrastructure for you. The cloud enables you to quickly solve your toughest business challenges and bring cutting-edge solutions to your users.

**Describe the benefits of using cloud services:**

When building or deploying a cloud application, two of the biggest considerations are uptime (or availability) and the ability to handle demand (or scale).

**High availability:**

When you’re deploying an application, a service, or any IT resources, it’s important the resources are available when needed. High availability focuses on ensuring maximum availability, regardless of disruptions or events that may occur.

When you’re architecting your solution, you’ll need to account for service availability guarantees. Azure is a highly available cloud environment with uptime guarantees depending on the service. These guarantees are part of the service-level agreements (SLAs).

Service Level Agreement (SLA) guarantees a certain level of availability as a percentage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **High Availability** | | | | |
| **Availability %** | **Downtime per year** | **Downtime per month** | **Downtime per week** | **Downtime per day** |
| 99 % “two nines” | 3.65 days | 7.31 hours | 1.68 hours | 14.40 minutes |
| 99.9% “three nines” | 8.77 hours | 43.83 hours | 10.08 hours | 1.44 mins |
| 99.99 % “four nines” | 52.60 mins | 4.38 mins | 1.01 mins | 8.64 secs |
| 99.999% “five nines” | 5.26 mins | 26.30 secs | 6.05 secs | 864.00 milli secs |
| 99.9999% “six nines” | 31.56 secs | 2.63 secs | 604.80 milli secs | 86.40 milli secs |
| 99.99999% “seven nines” | 3.16 secs | 262.97 milli secs | 60.48 milli secs | 8.64 milli secs |
| 99.999999% “eight nines” | 315.58 milli secs | 26.30 milli secs | 6.05 milli secs | 864.00 micro secs |
| 99.9999999% “nine nines” | 31.56 milli secs | 2.63 milli secs | 604.80 micro secs | 86.40 micro secs |

**Scalability:**

Another major benefit of cloud computing is the scalability of cloud resources. Scalability refers to the ability to adjust resources to meet demand. If you suddenly experience peak traffic and your systems are overwhelmed, the ability to scale means you can add more resources to better handle the increased demand.

The other benefit of scalability is that you aren't overpaying for services. Because the cloud is a consumption-based model, you only pay for what you use. If demand drops off, you can reduce your resources and thereby reduce your costs.

Scaling generally comes in two varieties: vertical and horizontal.

Vertical scaling is focused on increasing or decreasing the capabilities of resources. Horizontal scaling is adding or subtracting the number of resources.

Vertical scaling:

With vertical scaling, if you were developing an app and you needed more processing power, you could vertically scale up to add more CPUs or RAM to the virtual machine. Conversely, if you realized you had over-specified the needs, you could vertically scale down by lowering the CPU or RAM specifications.

Horizontal scaling:

With horizontal scaling, if you suddenly experienced a steep jump in demand, your deployed resources could be scaled out (either automatically or manually). For example, you could add additional virtual machines or containers, scaling out. In the same manner, if there was a significant drop in demand, deployed resources could be scaled in (either automatically or manually), scaling in.

**Benefits of Reliability & Predictability in the cloud:**

Reliability and predictability are two crucial cloud benefits that help you develop solutions with confidence.

Reliability:

Reliability is the ability of a system to recover from failures and continue to function. It's also one of the pillars of the Microsoft Azure Well-Architected Framework.

The cloud, by virtue of its decentralized design, naturally supports a reliable and resilient infrastructure. With a decentralized design, the cloud enables you to have resources deployed in regions around the world. With this global scale, even if one region has a catastrophic event other regions are still up and running. You can design your applications to automatically take advantage of this increased reliability. In some cases, your cloud environment itself will automatically shift to a different region for you, with no action needed on your part. You’ll learn more about how Azure leverages global scale to provide reliability later in this series.

Predictability

Predictability in the cloud lets you move forward with confidence. Predictability can be focused on performance predictability or cost predictability. Both performance and cost predictability are heavily influenced by the Microsoft Azure Well-Architected Framework. Deploy a solution that’s built around this framework and you have a solution whose cost and performance are predictable.

Performance:

Performance predictability focuses on predicting the resources needed to deliver a positive experience for your customers. Autoscaling, load balancing, and high availability are just some of the cloud concepts that support performance predictability. If you suddenly need more resources, autoscaling can deploy additional resources to meet the demand, and then scale back when the demand drops. Or if the traffic is heavily focused on one area, load balancing will help redirect some of the overload to less stressed areas.

Cost:

Cost predictability is focused on predicting or forecasting the cost of the cloud spend. With the cloud, you can track your resource use in real time, monitor resources to ensure that you’re using them in the most efficient way, and apply data analytics to find patterns and trends that help better plan resource deployments. By operating in the cloud and using cloud analytics and information, you can predict future costs and adjust your resources as needed. You can even use tools like the Total Cost of Ownership (TCO) or Pricing Calculator to get an estimate of potential cloud spend.

**Benefits of security and governance in the cloud:**

Whether you’re deploying infrastructure as a service or software as a service, cloud features support governance and compliance. Things like set templates help ensure that all your deployed resources meet corporate standards and government regulatory requirements. Plus, you can update all your deployed resources to new standards as standards change. Cloud-based auditing helps flag any resource that’s out of compliance with your corporate standards and provides mitigation strategies. Depending on your operating model, software patches and updates may also automatically be applied, which helps with both governance and security.

On the security side, you can find a cloud solution that matches your security needs. If you want maximum control of security, infrastructure as a service provides you with physical resources but lets you manage the operating systems and installed software, including patches and maintenance. If you want patches and maintenance taken care of automatically, platform as a service or software as a service deployments may be the best cloud strategies for you.

And because the cloud is intended as an over-the-internet delivery of IT resources, cloud providers are typically well suited to handle things like distributed denial of service (DDoS) attacks, making your network more robust and secure.

By establishing a good governance footprint early, you can keep your cloud footprint updated, secure, and well managed.

**Benefits of manageability in the cloud:**

A major benefit of cloud computing is the manageability options. There are two types of manageability for cloud computing that you’ll learn about in this series, and both are excellent benefits.

Management of the cloud:

Management of the cloud speaks to managing your cloud resources. In the cloud, you can:

Automatically scale resource deployment based on need.

* Deploy resources based on a preconfigured template, removing the need for manual configuration.
* Monitor the health of resources and automatically replace failing resources.
* Receive automatic alerts based on configured metrics, so you’re aware of performance in real time.

Management in the cloud:

Management in the cloud speaks to how you’re able to manage your cloud environment and resources. You can manage these:

* Through a web portal.
* Using a command line interface.
* Using APIs.
* Using PowerShell.

**Cloud Service types:**

Infrastructure as a Service (Iaas):

Infrastructure as a service (IaaS) is the most flexible category of cloud services, as it provides you the maximum amount of control for your cloud resources. In an IaaS model, the cloud provider is responsible for maintaining the hardware, network connectivity (to the internet), and physical security. You’re responsible for everything else: operating system installation, configuration, and maintenance; network configuration; database and storage configuration; and so on. With IaaS, you’re essentially renting the hardware in a cloud datacenter, but what you do with that hardware is up to you.

Shared responsibility model:

The shared responsibility model applies to all the cloud service types. IaaS places the largest share of responsibility with you. The cloud provider is responsible for maintaining the physical infrastructure and its access to the internet. You’re responsible for installation and configuration, patching and updates, and security.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Responsibility | SaaS | PaaS | IaaS | On-Prem |
| Responsibility always retained by the customer | Information and data | Customer | Customer | Customer | Customer |
| Devices (Mobile & PCs) | Customer | Customer | Customer | Customer |
| Accounts and identities | Customer | Customer | Customer | Customer |
| Responsibility varies by type | Identity and directory infrastructure | Both | Both | Customer | Customer |
| Applications | Microsoft | Both | Customer | Customer |
| Network Controls | Microsoft | Both | Customer | Customer |
| Operating system | Microsoft | Microsoft | Customer | Customer |
| Responsibility transfers to cloud provider | Physical hosts | Microsoft | Microsoft | Microsoft | Customer |
| Physical network | Microsoft | Microsoft | Microsoft | Customer |
| Physical datacenter | Microsoft | Microsoft | Microsoft | Customer |

Scenarios:

Some common scenarios where IaaS might make sense include:

* Lift-and-shift migration: You’re standing up cloud resources similar to your on-prem datacenter, and then simply moving the things running on-prem to running on the IaaS infrastructure.
* Testing and development: You have established configurations for development and test environments that you need to rapidly replicate. You can stand up or shut down the different environments rapidly with an IaaS structure, while maintaining complete control.

**Platform as a Service (PaaS):**

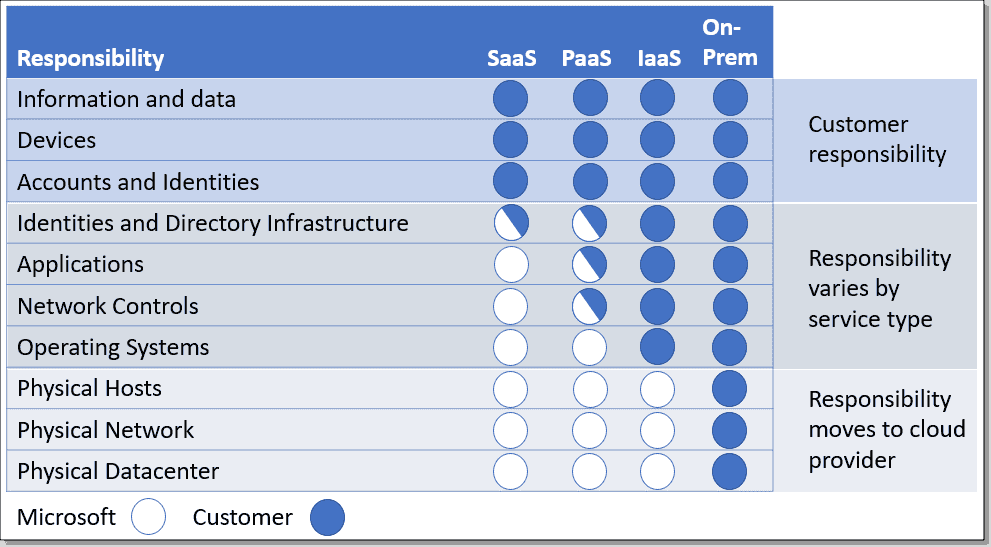
Platform as a service (PaaS) is a middle ground between renting space in a datacenter (infrastructure as a service) and paying for a complete and deployed solution (software as a service). In a PaaS environment, the cloud provider maintains the physical infrastructure, physical security, and connection to the internet. They also maintain the operating systems, middleware, development tools, and business intelligence services that make up a cloud solution. In a PaaS scenario, you don't have to worry about the licensing or patching for operating systems and databases.

PaaS is well suited to provide a complete development environment without the headache of maintaining all the development infrastructure.

Shared responsibility model for PaaS:

The shared responsibility model applies to all the cloud service types. PaaS splits the responsibility between you and the cloud provider. The cloud provider is responsible for maintaining the physical infrastructure and its access to the internet, just like in IaaS. In the PaaS model, the cloud provider will also maintain the operating systems, databases, and development tools. Think of PaaS like using a domain joined machine: IT maintains the device with regular updates, patches, and refreshes.

Depending on the configuration, you or the cloud provider may be responsible for networking settings and connectivity within your cloud environment, network and application security, and the directory infrastructure.



Scenarios:

Some common scenarios where PaaS might make sense include:

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

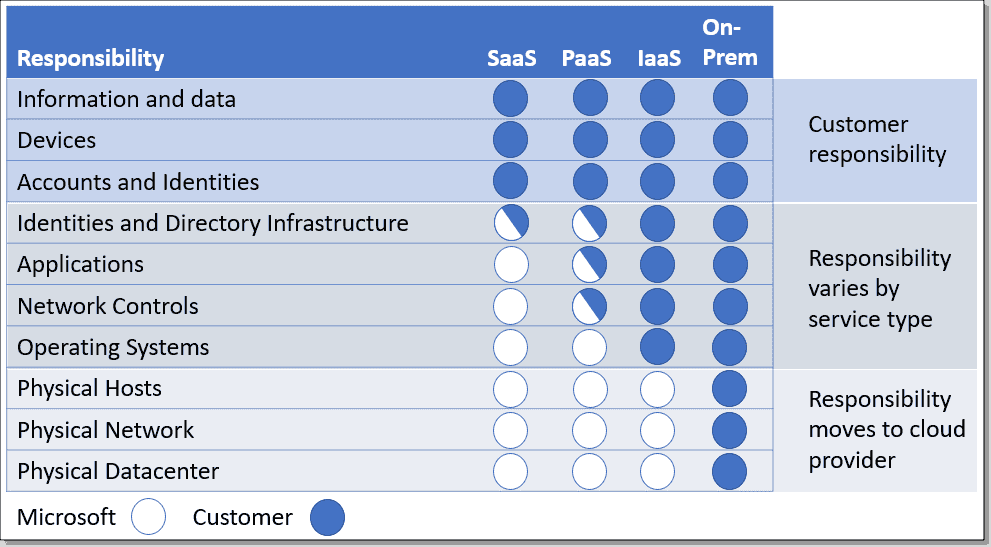
**Software as a Service (SaaS):**

Software as a service (SaaS) is the most complete cloud service model from a product perspective. With SaaS, you’re essentially renting or using a fully developed application. Email, financial software, messaging applications, and connectivity software are all common examples of a SaaS implementation.

While the SaaS model may be the least flexible, it’s also the easiest to get up and running. It requires the least amount of technical knowledge or expertise to fully employ.

Shared responsibility model

The shared responsibility model applies to all the cloud service types. SaaS is the model that places the most responsibility with the cloud provider and the least responsibility with the user. In a SaaS environment you’re responsible for the data that you put into the system, the devices that you allow to connect to the system, and the users that have access. Nearly everything else falls to the cloud provider. The cloud provider is responsible for physical security of the datacenters, power, network connectivity, and application development and patching.



Scenarios:

Some common scenarios for SaaS are:

* Email and messaging.
* Business productivity applications.
* Finance and expense tracking.

**Core Architectural Components of Azure:**

Azure is a continually expanding set of cloud services that help you meet current and future business challenges. Azure gives you the freedom to build, manage, and deploy applications on a massive global network using your favorite tools and frameworks.

What does Azure offer?

With help from Azure, you have everything you need to build your next great solution. The following lists several of the benefits that Azure provides, so you can easily invent with purpose:

* **Be ready for the future**: Continuous innovation from Microsoft supports your development today and your product visions for tomorrow.
* **Build on your terms**: You have choices. With a commitment to open source, and support for all languages and frameworks, you can build how you want and deploy where you want.
* **Operate hybrid seamlessly**: On-premises, in the cloud, and at the edge, we'll meet you where you are. Integrate and manage your environments with tools and services designed for a hybrid cloud solution.
* **Trust your cloud**: Get security from the ground up, backed by a team of experts, and proactive compliance trusted by enterprises, governments, and startups.

What can I do with Azure?

Azure provides more than 100 services that enable you to do everything from running your existing applications on virtual machines to exploring new software paradigms, such as intelligent bots and mixed reality.

Many teams start exploring the cloud by moving their existing applications to virtual machines (VMs) that run in Azure. Migrating your existing apps to VMs is a good start, but the cloud is much more than a different place to run your VMs.

For example, Azure provides artificial intelligence (AI) and machine-learning (ML) services that can naturally communicate with your users through vision, hearing, and speech. It also provides storage solutions that dynamically grow to accommodate massive amounts of data. Azure services enable solutions that aren't feasible without the power of the cloud.

**Azure Accounts:**

To create and use Azure services, you need an Azure subscription. When you're completing Learn modules, most of the time a temporary subscription is created for you, which runs in an environment called the Learn sandbox. When you're working with your own applications and business needs, you need to create an Azure account, and a subscription will be created for you.

After you've created an Azure account, you're free to create additional subscriptions. For example, your company might use a single Azure account for your business and separate subscriptions for development, marketing, and sales departments. After you've created an Azure subscription, you can start creating Azure resources within each subscription.

What is the Azure free account?

The Azure free account includes:

* Free access to popular Azure products for 12 months.
* A credit to use for the first 30 days.
* Access to more than 25 products that are always free.

The Azure free account is an excellent way for new users to get started and explore. To sign up, you need a phone number, a credit card, and a Microsoft or GitHub account. The credit card information is used for identity verification only. You won't be charged for any services until you upgrade to a paid subscription.

What is the Azure free student account?

The Azure free student account offer includes:

* Free access to certain Azure services for 12 months.
* A credit to use in the first 12 months.
* Free access to certain software developer tools.

The [Azure free student account](https://azure.microsoft.com/free/students/) is an offer for students that gives $100 credit and free developer tools. Also, you can sign up without a credit card.

What is the Microsoft Learn sandbox?

Many of the Learn exercises use a technology called the sandbox, which creates a temporary subscription that's added to your Azure account. This temporary subscription allows you to create Azure resources during a Learn module. Learn automatically cleans up the temporary resources for

you after you've completed the module.

When you're completing a Learn module, you're welcome to use your personal subscription to complete the exercises in a module. However, the sandbox is the preferred method to use because it allows you to create and test Azure resources at no cost to you.

Azure Physical Infrastructure:

Throughout your journey with Microsoft Azure, you’ll hear and use terms like Regions, Availability Zones, Resources, Subscriptions, and more. This module focuses on the core architectural components of Azure. The core architectural components of Azure may be broken down into two main groupings: the physical infrastructure, and the management infrastructure.

Physical infrastructure

The physical infrastructure for Azure starts with datacenters. Conceptually, the datacenters are the same as large corporate datacenters. They’re facilities with resources arranged in racks, with dedicated power, cooling, and networking infrastructure.

As a global cloud provider, Azure has datacenters around the world. However, these individual datacenters aren’t directly accessible. Datacenters are grouped into Azure Regions or Azure Availability Zones that are designed to help you achieve resiliency and reliability for your business-critical workloads.

The [Global infrastructure](https://infrastructuremap.microsoft.com/) site gives you a chance to interactively explore the underlying Azure infrastructure.

**Regions:**

A region is a geographical area on the planet that contains at least one, but potentially multiple datacenters that are nearby and networked together with a low-latency network. Azure intelligently assigns and controls the resources within each region to ensure workloads are appropriately balanced.

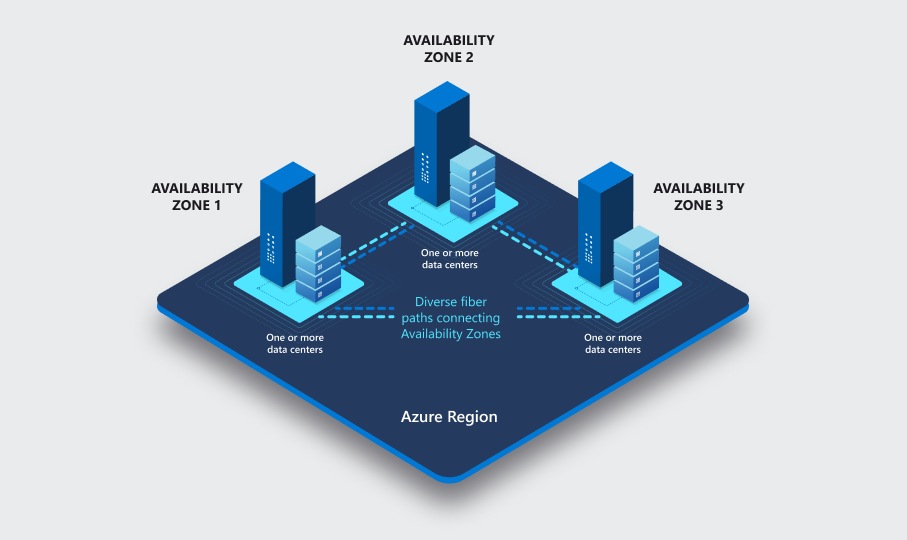
When you deploy a resource in Azure, you'll often need to choose the region where you want your resource deployed.

Some services or virtual machine (VM) features are only available in certain regions, such as specific VM sizes or storage types. There are also some global Azure services that don't require you to select a particular region, such as Azure Active Directory, Azure Traffic Manager, and Azure DNS.

**Availability Zones:**

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. An availability zone 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.

To ensure resiliency, a minimum of three separate availability zones are present in all availability zone-enabled regions. However, not all Azure Regions currently support availability zones.



**Use availability zones in your apps:**

You want to ensure your services and data are redundant so you can protect your information in case of failure. When you host your infrastructure, setting up your own redundancy requires that you create duplicate hardware environments. Azure can help make your app highly available through availability zones.

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

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

* Zonal services: You pin the resource to a specific zone (for example, VMs, managed disks, IP addresses).
* Zone-redundant services: The platform replicates automatically across zones (for example, zone-redundant storage, SQL Database).
* Non-regional services: Services are always available from Azure geographies and are resilient to zone-wide outages as well as region-wide outages.
* Even with the additional resiliency that availability zones provide, it’s possible that an event could be so large that it impacts multiple availability zones in a single region. To provide even further resilience, Azure has Region Pairs.

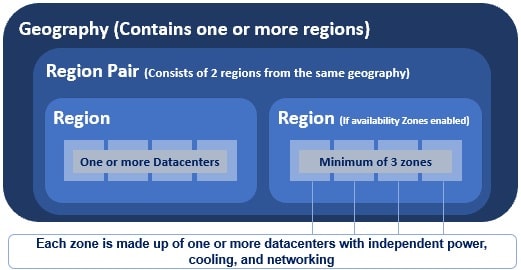
**Region pairs:**

Most Azure regions are 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 across a geography that helps reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages, or physical network outages that affect an entire region. For example, if a region in a pair was affected by a natural disaster, services would automatically fail over to the other region in its region pair.

 Important:

Not all Azure services automatically replicate data or automatically fall back from a failed region to cross-replicate to another enabled region. In these scenarios, recovery and replication must be configured by the customer.

Examples of region pairs in Azure are West US paired with East US and South-East Asia paired with East Asia. Because the pair of regions are directly connected and far enough apart to be isolated from regional disasters, you can use them to provide reliable services and data redundancy.



**Additional advantages of region pairs:**

* If an extensive Azure outage occurs, one region out of every pair is prioritized to make sure at least one is restored as quickly 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.

 Important:

Most directions are paired in two directions, meaning they are the backup for the region that provides a backup for them (West US and East US back each other up). However, some regions, such as West India and Brazil South, are paired in only one direction. In a one-direction pairing, the Primary region does not provide backup for its secondary region. So, even though West India’s secondary region is South India, South India does not rely on West India. West India's secondary region is South India, but South India's secondary region is Central India. Brazil South is unique because it's paired with a region outside of its geography. Brazil South's secondary region is South Central US. The secondary region of South Central US isn't Brazil South.



**Sovereign Regions:**

In addition to regular regions, Azure also has sovereign regions. Sovereign regions are instances of Azure that are isolated from the main instance of Azure. You may need to use a sovereign region for compliance or legal purposes.

**Azure sovereign regions include:**

* US DoD Central, US Gov Virginia, US Gov Iowa and more: These regions are physical and logical network-isolated instances of Azure for U.S. government agencies and partners. These datacenters are operated by screened U.S. personnel 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 doesn't directly maintain the datacenters.

**Azure Management Infrastructure:**

The management infrastructure includes Azure resources and resource groups, subscriptions, and accounts. Understanding the hierarchical organization will help you plan your projects and products within Azure.

**Azure resources and resource groups:**

A resource is the basic building block of Azure. Anything you create, provision, deploy, etc. is a resource. Virtual Machines (VMs), virtual networks, databases, cognitive services, etc. are all considered resources within Azure.

Resource groups are simply groupings of resources. When you create a resource, you’re required to place it into a resource group. While a resource group can contain many resources, a single resource can only be in one resource group at a time. Some resources may be moved between resource groups, but when you move a resource to a new group, it will no longer be associated with the former group. Additionally, resource groups can't be nested, meaning you can’t put resource group B inside of resource group A.

Resource groups provide a convenient way to group resources together. When you apply an action to a resource group, that action will apply to all the resources within the resource group. If you delete a resource group, all the resources will be deleted. If you grant or deny access to a resource group, you’ve granted or denied access to all the resources within the resource group.

When you’re provisioning resources, it’s good to think about the resource group structure that best suits your needs.

For example, if you’re setting up a temporary dev environment, grouping all the resources together means you can deprovision all of the associated resources at once by deleting the resource group. If you’re provisioning compute resources that will need three different access schemas, it may be best to group resources based on the access schema, and then assign access at the resource group level.

There aren’t hard rules about how you use resource groups, so consider how to set up your resource groups to maximize their usefulness for you.

Azure subscriptions

In Azure, subscriptions are a unit of management, billing, and scale. Similar to how resource groups are a way to logically organize resources, subscriptions allow you to logically organize your resource groups and facilitate billing.

Using Azure requires an Azure subscription. A subscription provides you with authenticated and authorized access to Azure products and services. It also allows you to provision resources. An Azure subscription links to an Azure account, which is an identity in Azure Active Directory (Azure AD) or in a directory that Azure AD trusts.

An account can have multiple subscriptions, but it’s only required to have one. In a multi-subscription account, you can use the subscriptions to configure different billing models and apply different access-management policies. You can use Azure subscriptions to define boundaries around Azure products, services, and resources. There are two types of subscription boundaries that you can use:

* Billing boundary: This subscription type determines how an Azure account is billed for using Azure. You can create multiple subscriptions for different types of billing requirements. Azure generates separate billing reports and invoices for each subscription so that you can organize and manage costs.
* Access control boundary: Azure applies access-management policies at the subscription level, and you can create separate subscriptions to reflect different organizational structures. An example is that within a business, you have different departments to which you apply distinct Azure subscription policies. This billing model allows you to manage and control access to the resources that users provision with specific subscriptions.

**Create additional Azure subscriptions:**

Similar to using resource groups to separate resources by function or access, 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**: You can choose to create subscriptions to set up separate environments for development and testing, security, or to isolate data for compliance reasons. This design is particularly useful because resource access control occurs at the subscription level.
* **Organizational structures**: You can create subscriptions to reflect different organizational structures. For example, you could limit one 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 can 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 one subscription for your production workloads and another subscription for your development and testing workloads.

**Azure management groups:**

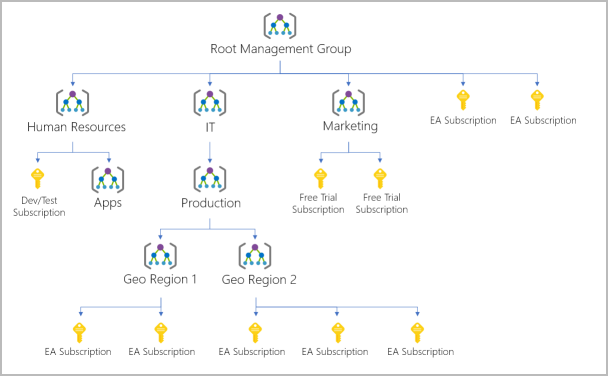
The final piece is the management group. Resources are gathered into resource groups, and resource groups are gathered into subscriptions. If you’re just starting in Azure that might seem like enough hierarchy to keep things organized. But imagine if you’re dealing with multiple applications, multiple development teams, in multiple geographies.

If you have many subscriptions, you might need a way to efficiently manage access, policies, and compliance for those subscriptions. Azure management groups provide a level of scope above subscriptions. You organize subscriptions into containers called management groups and apply governance conditions to the management groups. All subscriptions within a management group automatically inherit the conditions applied to the management group, the same way that resource groups inherit settings from subscriptions and resources inherit from resource groups.

Management groups give you enterprise-grade management at a large scale, no matter what type of subscriptions you might have. Management groups can be nested.

**Management group, subscriptions, and resource group hierarchy:**

You can build a flexible structure of management groups and subscriptions to organize your resources into a hierarchy for unified policy and access management. The following diagram shows an example of creating a hierarchy for governance by using management groups.



**Some examples of how you could use management groups might be:**

* **Create a hierarchy that applies a policy**. You could limit VM locations to the US West Region in a group called Production. This policy will inherit onto all the subscriptions that are descendants of that management group and will apply to all VMs under those subscriptions. This security policy can't be altered by the resource or subscription owner, which allows for improved governance.
* **Provide user access to multiple subscriptions**. By moving multiple subscriptions under a management group, you can create one Azure role-based access control (Azure RBAC) assignment on the management group. Assigning Azure RBAC at the management group level means that all sub-management groups, subscriptions, resource groups, and resources underneath that management group would also inherit those permissions. One assignment on the management group can enable users to have access to everything they need instead of scripting Azure RBAC over different subscriptions.

**Important facts about management groups:**

* 10,000 management groups can be supported in a single directory.
* A management group tree can support up to six levels of depth. This limit doesn't include the root level or the subscription level.
* Each management group and subscription can support only one parent.

**Azure Compute & Networking Services:**

**Azure Virtual Machine:**

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

* Total control over the operating system (OS).
* The ability to run custom software.
* To use custom hosting configurations.

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

You can even create or use an already created image to rapidly provision VMs. You can create and provision a VM in minutes when you select a preconfigured VM image. An image is a template used to create a VM and may already include an OS and other software, like development tools or web hosting environments.

**Scale VMs in Azure:**

You can run single VMs for testing, development, or minor tasks. Or you can group VMs together to provide high availability, scalability, and redundancy. Azure can also manage the grouping of VMs for you with features such as scale sets and availability sets.

**Virtual machine scale sets:**

Virtual machine scale sets let you create and manage a group of identical, load-balanced VMs. If you simply created multiple VMs with the same purpose, you’d need to ensure they were all configured identically and then set up network routing parameters to ensure efficiency. You’d also have to monitor the utilization to determine if you need to increase or decrease the number of VMs.

Instead, with virtual machine scale sets, Azure automates most of that work. Scale sets allow you to centrally manage, configure, and update a large number of VMs in minutes. The number of VM instances can automatically increase or decrease in response to demand, or you can set it to scale based on a defined schedule. Virtual machine scale sets also automatically deploy a load balancer to make sure that your resources are being used efficiently. With virtual machine scale sets, you can build large-scale services for areas such as compute, big data, and container workloads.

**Virtual machine availability sets:**

Virtual machine availability sets are another tool to help you build a more resilient, highly available environment. Availability sets are designed to ensure that VMs stagger updates and have varied power and network connectivity, preventing you from losing all your VMs with a single network or power failure.

Availability sets do this by grouping VMs in two ways: update domain and fault domain.

* **Update domain**: The update domain groups VMs that can be rebooted at the same time. This allows you to apply updates while knowing that only one update domain grouping will be offline at a time. All of the machines in one update domain will be updated. An update group going through the update process is given a 30-minute time to recover before maintenance on the next update domain starts.
* **Fault domain**: The fault domain groups your VMs by common power source and network switch. By default, an availability set will split your VMs across up to three fault domains. This helps protect against a physical power or networking failure by having VMs in different fault domains (thus being connected to different power and networking resources).

Best of all, there’s no additional cost for configuring an availability set. You only pay for the VM instances you create.

Examples of when to use VMs:

Some common examples or use cases for virtual machines include:

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

Move to the cloud with VMs:

VMs are also an excellent choice when you move from a physical server to the cloud (also known as lift and shift). You can create an image of the physical server and host it within a VM with little or no changes. Just like a physical on-premises server, you must maintain the VM: you’re responsible for maintaining the installed OS and software.

VM Resources:

When you provision a VM, you’ll also have the chance to pick the resources that are associated with that VM, including:

* Size (purpose, number of processor cores, and amount of RAM)
* Storage disks (hard disk drives, solid state drives, etc.)
* Networking (virtual network, public IP address, and port configuration)

Azure Virtual Desktop:

Another type of virtual machine is the Azure Virtual Desktop. Azure Virtual Desktop is a desktop and application virtualization service that runs on the cloud. It enables you to use a cloud-hosted version of Windows from any location. Azure Virtual Desktop works across devices and operating systems and works with apps that you can use to access remote desktops or most modern browsers.

Enhance security:

Azure Virtual Desktop provides centralized security management for users' desktops with Azure Active Directory (Azure AD). You can enable multifactor authentication to secure user sign-ins. You can also secure access to data by assigning granular role-based access controls (RBACs) to users.

With Azure Virtual Desktop, the data and apps are separated from the local hardware. The actual desktop and apps are running in the cloud, meaning the risk of confidential data being left on a personal device is reduced. Additionally, user sessions are isolated in both single and multi-session environments.

Multi-session Windows 10 or Windows 11 deployment:

Azure Virtual Desktop lets you use Windows 10 or Windows 11 Enterprise multi-session, the only Windows client-based operating system that enables multiple concurrent users on a single VM. Azure Virtual Desktop also provides a more consistent experience with broader application support compared to Windows Server-based operating systems.

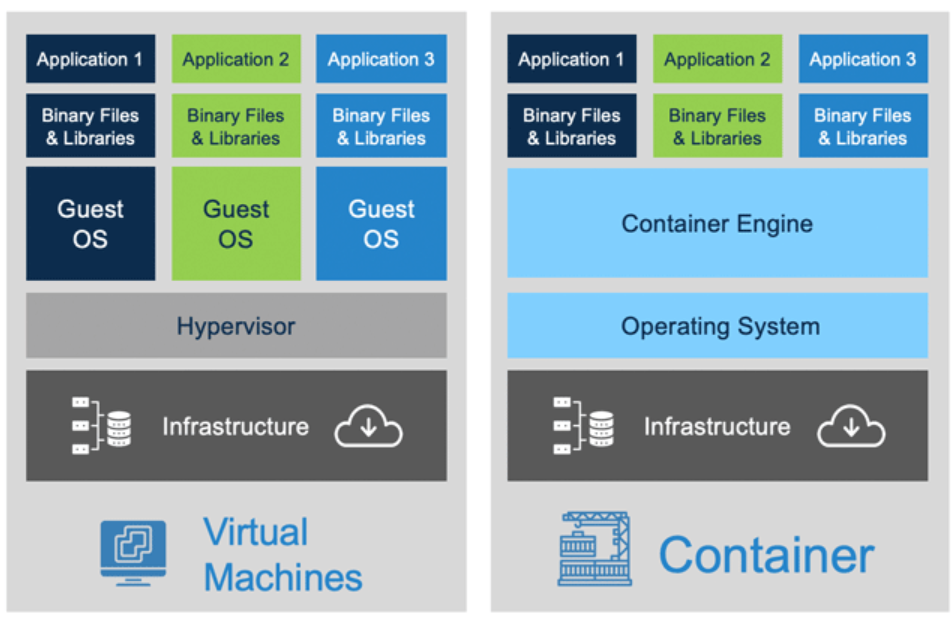
**Azure Containers:**

While virtual machines are an excellent way to reduce costs versus the investments that are necessary for physical hardware, they're still limited to a single operating system per virtual machine. If you want to run multiple instances of an application on a single host machine, containers are an excellent choice.

**What are containers?**

Containers are a virtualization environment. Much like running multiple virtual machines on a single physical host, you can run multiple containers on a single physical or virtual host. Unlike virtual machines, you don't manage the operating system for a container. Virtual machines appear to be an instance of an operating system that you can connect to and manage. Containers are lightweight and designed to be created, scaled out, and stopped dynamically. It's possible to create and deploy virtual machines as application demand increases, but containers are a lighter weight, more agile method. Containers are designed to allow you to respond to changes on demand. With containers, you can quickly restart if there's a crash or hardware interruption. One of the most popular container engines is Docker, which is supported by Azure.

**Compare virtual machines to containers:**



**Azure Container Instances:**

Azure Container Instances offer the fastest and simplest way to run a container in Azure; without having to manage any virtual machines or adopt any additional services. Azure Container Instances are a platform as a service (PaaS) offering. Azure Container Instances allow you to upload your containers and then the service will run the containers for you.

Use containers in your solutions

Containers are often used to create solutions by using a microservice architecture. This architecture is where you break solutions into smaller, independent pieces. For example, you might 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.

Imagine your website back-end has reached capacity but the front end and storage aren't being stressed. With containers, you could scale the back end separately to improve performance. If something necessitated such a change, you could also choose to change the storage service or modify the front end without impacting any of the other components.

**Azure Functions:**

Azure Functions is an event-driven, serverless compute option that doesn’t require maintaining virtual machines or containers. If you build an app using VMs or containers, those resources have to be “running” in order for your app to function. With Azure Functions, an event wakes the function, alleviating the need to keep resources provisioned when there are no events.

**Benefits of Azure Functions:**

Using Azure Functions is ideal when you're only concerned about the code running your service and not about the underlying platform or infrastructure. Functions are 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.

Functions scale automatically based on demand, so they may be a good choice when demand is variable.

Azure Functions 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.

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

Functions are a key component of serverless computing. 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. This flexibility allows you to manage scaling, run on virtual networks, and even completely isolate the functions.

**Application Hosting Options in Azure:**

If you need to host your application on Azure, you might initially turn to a virtual machine (VM) or containers. Both VMs and containers provide excellent hosting solutions. VMs give you maximum control of the hosting environment and allow you to configure it exactly how you want. VMs also may be the most familiar hosting method if you’re new to the cloud. Containers, with the ability to isolate and individually manage different aspects of the hosting solution, can also be a robust and compelling option.

There are other hosting options that you can use with Azure, including Azure App Service.

**Azure App Service:**

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

Azure App Service is a robust hosting option that you can use to host your apps in Azure. Azure App Service lets you focus on building and maintaining your app, and Azure focuses on keeping the environment up and running.

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. It supports multiple languages, including .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. It also supports both Windows and Linux environments.

**Types of app services:**

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

* Web apps
* API apps
* WebJobs
* Mobile apps

App Service handles most of the infrastructure decisions you deal with in hosting web-accessible apps:

* Deployment and management are integrated into the platform.
* Endpoints can be secured.
* Sites can be scaled quickly to handle high traffic loads.
* 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 by 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 by using your choice of language and framework. You get full Swagger support and the ability to package and publish your API in Azure Marketplace. The produced apps can be consumed from any HTTP- or HTTPS-based client.

**WebJobs:**

You can use the WebJobs feature 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 apps:**

Use the Mobile Apps feature of App Service to quickly build a back end for iOS and Android apps. With just a few actions 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's SDK support for native iOS and Android, Xamarin, and React native apps.

**Azure Virtual Networking:**

Azure virtual networks and virtual subnets enable Azure resources, such as VMs, web apps, and databases, to communicate with each other, with users on the internet, and with your on-premises client computers. You can think of an Azure network as an extension of your on-premises network with resources that link other Azure resources.

Azure virtual networks provide the following key networking capabilities:

* Isolation and segmentation
* Internet communications
* Communicate between Azure resources
* Communicate with on-premises resources
* Route network traffic
* Filter network traffic
* Connect virtual networks

Azure virtual networking supports both public and private endpoints to enable communication between external or internal resources with other internal resources.

* Public endpoints have a public IP address and can be accessed from anywhere in the world.
* Private endpoints exist within a virtual network and have a private IP address from within the address space of that virtual network.

**Isolation and segmentation:**

Azure virtual network allows you to create multiple isolated virtual networks. When you set up a virtual network, you define a private IP address space by using either public or private IP address ranges. The IP range only exists within the virtual network and isn't internet routable. You can divide that IP address space into subnets and allocate part of the defined address space to each named subnet.

For name resolution, you can use the name resolution service that's built into Azure. You also can configure the virtual network to use either an internal or an external DNS server.

**Internet communications:**

You can enable incoming connections from the internet by assigning a public IP address to an Azure resource, or putting the resource behind a public load balancer.

**Communicate between Azure resources:**

You'll want to enable Azure resources to communicate securely with each other. You can do that in one of two ways:

* Virtual networks can connect not only VMs but other Azure resources, such as the App Service Environment for Power Apps, Azure Kubernetes Service, and Azure virtual machine scale sets.
* Service endpoints can connect to other Azure resource types, such as Azure SQL databases and storage accounts. This approach enables you to link multiple Azure resources to virtual networks to improve security and provide optimal routing between resources.

**Communicate with on-premises resources:**

Azure virtual networks enable you to link resources together in your on-premises environment and within your Azure subscription. In effect, you can create a network that spans both your local and cloud environments. There are three mechanisms for you to achieve this connectivity:

* Point-to-site virtual private network connections are from a computer outside your organization back into your corporate network. In this case, the client computer initiates an encrypted VPN connection to connect to the Azure virtual network.
* Site-to-site virtual private networks link your on-premises VPN device or gateway to the Azure VPN gateway in a virtual network. In effect, the devices in Azure can appear as being on the local network. The connection is encrypted and works over the internet.
* Azure ExpressRoute provides a dedicated private connectivity to Azure that doesn't travel over the internet. ExpressRoute is useful for environments where you need greater bandwidth and even higher levels of security.

**Route network traffic:**

By default, Azure routes traffic between subnets on any connected virtual networks, on-premises networks, and the internet. You also can control routing and override those settings, as follows:

* Route tables allow you to define rules about how traffic should be directed. You can create custom route tables that control how packets are routed between subnets.
* Border Gateway Protocol (BGP) works with Azure VPN gateways, Azure Route Server, or Azure ExpressRoute to propagate on-premises BGP routes to Azure virtual networks.

**Filter network traffic:**

Azure virtual networks enable you to filter traffic between subnets by using the following approaches:

* Network security groups are Azure resources that can contain multiple inbound and outbound security rules. You can define these rules to allow or block traffic, based on factors such as source and destination IP address, port, and protocol.
* Network virtual appliances are specialized VMs that can be compared to a hardened network appliance. A network virtual appliance carries out a particular network function, such as running a firewall or performing wide area network (WAN) optimization.

**Connect virtual networks:**

You can link virtual networks together by using virtual network peering. Peering allows two virtual networks to connect directly to each other. Network traffic between peered networks is private, and travels on the Microsoft backbone network, never entering the public internet. Peering enables resources in each virtual network to communicate with each other. These virtual networks can be in separate regions, which allows you to create a global interconnected network through Azure.

User-defined routes (UDR) allow you to control the routing tables between subnets within a virtual network or between virtual networks. This allows for greater control over network traffic flow.

**Azure Virtual Private Network:**

A virtual private network (VPN) uses an encrypted tunnel within another network. VPNs are typically deployed to connect two or more trusted private networks to one another over an untrusted network (typically the public internet). Traffic is encrypted while traveling over the untrusted network to prevent eavesdropping or other attacks. VPNs can enable networks to safely and securely share sensitive information.

**VPN gateways:**

A VPN gateway is a type of virtual network gateway. Azure VPN Gateway instances are deployed in a dedicated subnet of the virtual network and enable the following connectivity:

* Connect on-premises datacenters to virtual networks through a site-to-site connection.
* Connect individual devices to virtual networks through a point-to-site connection.
* Connect virtual networks to other virtual networks through a network-to-network connection.

All data transfer is encrypted inside a private tunnel as it crosses the internet. You can deploy only one VPN gateway in each virtual network. However, you can use one gateway to connect to multiple locations, which includes other virtual networks or on-premises datacenters.

When you deploy a VPN gateway, you specify the VPN type: either policy-based or route-based. The main difference between these two types of VPNs is how traffic to be encrypted is specified. In Azure, both types of VPN gateways use a pre-shared key as the only method of authentication.

* Policy-based VPN gateways specify statically the IP address of packets that should be encrypted through each tunnel. This type of device evaluates every data packet against those sets of IP addresses to choose the tunnel where that packet is going to be sent through.
* In Route-based gateways, IPSec tunnels are modeled as a network interface or virtual tunnel interface. IP routing (either static routes or dynamic routing protocols) decides which one of these tunnel interfaces to use when sending each packet. Route-based VPNs are the preferred connection method for on-premises devices. They're more resilient to topology changes such as the creation of new subnets.

Use a route-based VPN gateway if you need any of the following types of connectivity:

* Connections between virtual networks
* Point-to-site connections
* Multisite connections
* Coexistence with an Azure ExpressRoute gateway

High-availability scenarios:

If you’re configuring a VPN to keep your information safe, you also want to be sure that it’s a highly available and fault tolerant VPN configuration. There are a few ways to maximize the resiliency of your VPN gateway.

Active/standby:

By default, VPN gateways are deployed as two instances in an active/standby configuration, even if you only see one VPN gateway resource in Azure. When planned maintenance or unplanned disruption affects the active instance, the standby instance automatically assumes responsibility for connections without any user intervention. Connections are interrupted during this failover, but they're typically restored within a few seconds for planned maintenance and within 90 seconds for unplanned disruptions.

Active/active:

With the introduction of support for the BGP routing protocol, you can also deploy VPN gateways in an active/active configuration. In this configuration, you assign a unique public IP address to each instance. You then create separate tunnels from the on-premises device to each IP address. You can extend the high availability by deploying an additional VPN device on-premises.

ExpressRoute failover:

Another high-availability option is to configure a VPN gateway as a secure failover path for ExpressRoute connections. ExpressRoute circuits have resiliency built in. However, they aren't immune to physical problems that affect the cables delivering connectivity or outages that affect the complete ExpressRoute location. In high-availability scenarios, where there's risk associated with an outage of an ExpressRoute circuit, you can also provision a VPN gateway that uses the internet as an alternative method of connectivity. In this way, you can ensure there's always a connection to the virtual networks.

Zone-redundant gateways:

In regions that support availability zones, VPN gateways and ExpressRoute gateways can be deployed in a zone-redundant configuration. This configuration brings resiliency, scalability, and higher availability to virtual network gateways. Deploying gateways in Azure availability zones physically and logically separates gateways within a region while protecting your on-premises network connectivity to Azure from zone-level failures. These gateways require different gateway stock keeping units (SKUs) and use Standard public IP addresses instead of Basic public IP addresses.

**Azure ExpressRoute:**

Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection, with the help of a connectivity provider. This connection is called an ExpressRoute Circuit. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure and Microsoft 365. This allows you to connect offices, datacenters, or other facilities to the Microsoft cloud. Each location would have its own ExpressRoute circuit.

Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a colocation facility. ExpressRoute connections don't go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, consistent latencies, and higher security than typical connections over the Internet.

Features and benefits of ExpressRoute:

There are several benefits to using ExpressRoute as the connection service between Azure and on-premises networks.

* Connectivity to Microsoft cloud services across all regions in the geopolitical region.
* Global connectivity to Microsoft services across all regions with the ExpressRoute Global Reach.
* Dynamic routing between your network and Microsoft via Border Gateway Protocol (BGP).
* Built-in redundancy in every peering location for higher reliability.

Connectivity to Microsoft cloud services:

ExpressRoute enables direct access to the following services in all regions:

* Microsoft Office 365
* Microsoft Dynamics 365
* Azure compute services, such as Azure Virtual Machines
* Azure cloud services, such as Azure Cosmos DB and Azure Storage

Global connectivity:

You can enable ExpressRoute Global Reach to exchange data across your on-premises sites by connecting your ExpressRoute circuits. For example, say you had an office in Asia and a datacenter in Europe, both with ExpressRoute circuits connecting them to the Microsoft network. You could use ExpressRoute Global Reach to connect those two facilities, allowing them to communicate without transferring data over the public internet.

Dynamic routing:

ExpressRoute uses the BGP. BGP is used to exchange routes between on-premises networks and resources running in Azure. This protocol enables dynamic routing between your on-premises network and services running in the Microsoft cloud.

Built-in redundancy:

Each connectivity provider uses redundant devices to ensure that connections established with Microsoft are highly available. You can configure multiple circuits to complement this feature.

ExpressRoute connectivity models:

* ExpressRoute supports four models that you can use to connect your on-premises network to the Microsoft cloud:
* CloudExchange colocation
* Point-to-point Ethernet connection
* Any-to-any connection
* Directly from ExpressRoute sites

Co-location at a cloud exchange:

Co-location refers to your datacenter, office, or other facility being physically co-located at a cloud exchange, such as an ISP. If your facility is co-located at a cloud exchange, you can request a virtual cross-connect to the Microsoft cloud.

Point-to-point Ethernet connection:

Point-to-point ethernet connection refers to using a point-to-point connection to connect your facility to the Microsoft cloud.

Any-to-any networks:

With any-to-any connectivity, you can integrate your wide area network (WAN) with Azure by providing connections to your offices and datacenters. Azure integrates with your WAN connection to provide a connection like you would have between your datacenter and any branch offices.

Directly from ExpressRoute sites:

You can connect directly into the Microsoft's global network at a peering location strategically distributed across the world. ExpressRoute Direct provides dual 100 Gbps or 10-Gbps connectivity, which supports Active/Active connectivity at scale.

Security considerations:

With ExpressRoute, your data doesn't travel over the public internet, so it's not exposed to the potential risks associated with internet communications. ExpressRoute is a private connection from your on-premises infrastructure to your Azure infrastructure. Even if you have an ExpressRoute connection, DNS queries, certificate revocation list checking, and Azure Content Delivery Network requests are still sent over the public internet.

**Azure DNS:**

Azure DNS is a hosting service for DNS domains that provides name resolution by using Microsoft Azure infrastructure. By hosting your domains in Azure, you can manage your DNS records using the same credentials, APIs, tools, and billing as your other Azure services.

Benefits of Azure DNS:

Azure DNS leverages the scope and scale of Microsoft Azure to provide numerous benefits, including:

* Reliability and performance
* Security
* Ease of Use
* Customizable virtual networks
* Alias records

Reliability and performance:

DNS domains in Azure DNS are hosted on Azure's global network of DNS name servers, providing resiliency and high availability. Azure DNS uses anycast networking, so each DNS query is answered by the closest available DNS server to provide fast performance and high availability for your domain.

Security:

Azure DNS is based on Azure Resource Manager, which provides features such as:

* Azure role-based access control (Azure RBAC) to control who has access to specific actions for your organization.
* Activity logs to monitor how a user in your organization modified a resource or to find an error when troubleshooting.
* Resource locking to lock a subscription, resource group, or resource. Locking prevents other users in your organization from accidentally deleting or modifying critical resources.

Ease of use:

Azure DNS can manage DNS records for your Azure services and provide DNS for your external resources as well. Azure DNS is integrated in the Azure portal and uses the same credentials, support contract, and billing as your other Azure services.

Because Azure DNS is running on Azure, it means you can manage your domains and records with the Azure portal, Azure PowerShell cmdlets, and the cross-platform Azure CLI. Applications that require automated DNS management can integrate with the service by using the REST API and SDKs.

Customizable virtual networks with private domains:

Azure DNS also supports private DNS domains. This feature allows you to use your own custom domain names in your private virtual networks, rather than being stuck with the Azure-provided names.

Alias records:

Azure DNS also supports alias record sets. You can use an alias record set to refer to an Azure resource, such as an Azure public IP address, an Azure Traffic Manager profile, or an Azure Content Delivery Network (CDN) endpoint. If the IP address of the underlying resource changes, the alias record set seamlessly updates itself during DNS resolution. The alias record set points to the service instance, and the service instance is associated with an IP address.

Important:

You can't use Azure DNS to buy a domain name. For an annual fee, you can buy a domain name by using App Service domains or a third-party domain name registrar. Once purchased, your domains can be hosted in Azure DNS for record management.

**Azure Storage services:**

**Azure storage accounts:**

A storage account provides a unique namespace for your Azure Storage data that's accessible from anywhere in the world over HTTP or HTTPS. Data in this account is secure, highly available, durable, and massively scalable.

When you create your storage account, you’ll start by picking the storage account type. The type of account determines the storage services and redundancy options and has an impact on the use cases. Below is a list of redundancy options that will be covered later in this module:

* Locally redundant storage (LRS)
* Geo-redundant storage (GRS)
* Read-access geo-redundant storage (RA-GRS)
* Zone-redundant storage (ZRS)
* Geo-zone-redundant storage (GZRS)
* Read-access geo-zone-redundant storage (RA-GZRS)

| Type | Supported services | Redundancy Options | Usage |
| --- | --- | --- | --- |
| Standard general-purpose v2 | Blob Storage (including Data Lake Storage), Queue Storage, Table Storage, and Azure Files | LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS | Standard storage account type for blobs, file shares, queues, and tables. Recommended for most scenarios using Azure Storage. If you want support for network file system (NFS) in Azure Files, use the premium file shares account type. |
| Premium block blobs | Blob Storage (including Data Lake Storage) | LRS, ZRS | Premium storage account type for block blobs and append blobs. Recommended for scenarios with high transaction rates or that use smaller objects or require consistently low storage latency. |
| Premium file shares | Azure Files | LRS, ZRS | Premium storage account type for file shares only. Recommended for enterprise or high-performance scale applications. Use this account type if you want a storage account that supports both Server Message Block (SMB) and NFS file shares. |
| Premium page blobs | Page blobs only | LRS | Premium storage account type for page blobs only. |

Storage account endpoints:

One of the benefits of using an Azure Storage Account is having a unique namespace in Azure for your data. In order to do this, every storage account in Azure must have a unique-in-Azure account name. The combination of the account name and the Azure Storage service endpoint forms the endpoints for your storage account.

When naming your storage account, keep these rules in mind:

* Storage account names must be between 3 and 24 characters in length and may contain numbers and lowercase letters only.
* Your storage account name must be unique within Azure. No two storage accounts can have the same name. This supports the ability to have a unique, accessible namespace in Azure.

The following table shows the endpoint format for Azure Storage services.

| **Storage service** | **Endpoint** |
| --- | --- |
| Blob Storage | https://<storage-account-name>.blob.core.windows.net |
| Data Lake Storage Gen2 | https://<storage-account-name>.dfs.core.windows.net |
| Azure Files | https://<storage-account-name>.file.core.windows.net |
| Queue Storage | https://<storage-account-name>.queue.core.windows.net |
| Table Storage | https://<storage-account-name>.table.core.windows.net |

**Azure storage redundancy:**

Azure Storage always stores multiple copies of your data so that it's protected from planned and unplanned events such as transient hardware failures, network or power outages, and natural disasters. Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures.

When deciding which redundancy option is best for your scenario, consider the tradeoffs between lower costs and higher availability. The factors that help determine which redundancy option you should choose include:

* How your data is replicated in the primary region.
* Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters.
* Whether your application requires read access to the replicated data in the secondary region if the primary region becomes unavailable.

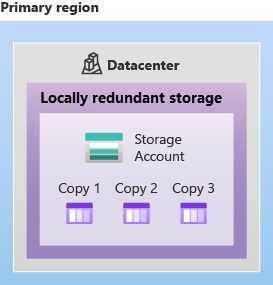
Redundancy in the primary region:

Data in an Azure Storage account is always replicated three times in the primary region. Azure Storage offers two options for how your data is replicated in the primary region, locally redundant

storage (LRS) and zone-redundant storage (ZRS).

Locally redundant storage:

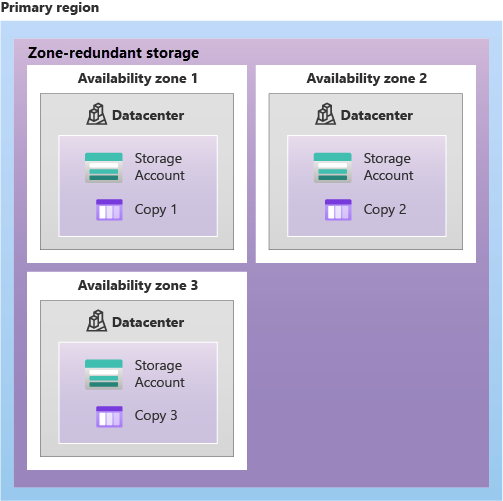
Locally redundant storage (LRS) replicates your data three times within a single data center in the primary region. LRS provides at least 11 nines of durability (99.999999999%) of objects over a given year.



LRS is the lowest-cost redundancy option and offers the least durability compared to other options. LRS protects your data against server rack and drive failures. However, if a disaster such as fire or flooding occurs within the data center, all replicas of a storage account using LRS may be lost or unrecoverable. To mitigate this risk, Microsoft recommends using zone-redundant storage (ZRS), geo-redundant storage (GRS), or geo-zone-redundant storage (GZRS).

Zone-redundant storage:

For Availability Zone-enabled Regions, zone-redundant storage (ZRS) replicates your Azure Storage data synchronously across three Azure availability zones in the primary region. ZRS offers durability for Azure Storage data objects of at least 12 nines (99.9999999999%) over a given year.



With ZRS, your data is still accessible for both read and write operations even if a zone becomes unavailable. No remounting of Azure file shares from the connected clients is required. If a zone becomes unavailable, Azure undertakes networking updates, such as DNS repointing. These updates may affect your application if you access data before the updates have completed.

Microsoft recommends using ZRS in the primary region for scenarios that require high availability. ZRS is also recommended for restricting replication of data within a country or region to meet data governance requirements.

Redundancy in a secondary region:

For applications requiring high durability, you can choose to additionally copy the data in your storage account to a secondary region that is hundreds of miles away from the primary region. If the data in your storage account is copied to a secondary region, then your data is durable even in the event of a catastrophic failure that prevents the data in the primary region from being recovered.

When you create a storage account, you select the primary region for the account. The paired secondary region is based on Azure Region Pairs, and can't be changed.

Azure Storage offers two options for copying your data to a secondary region: geo-redundant storage (GRS) and geo-zone-redundant storage (GZRS). GRS is similar to running LRS in two regions, and GZRS is similar to running ZRS in the primary region and LRS in the secondary region.

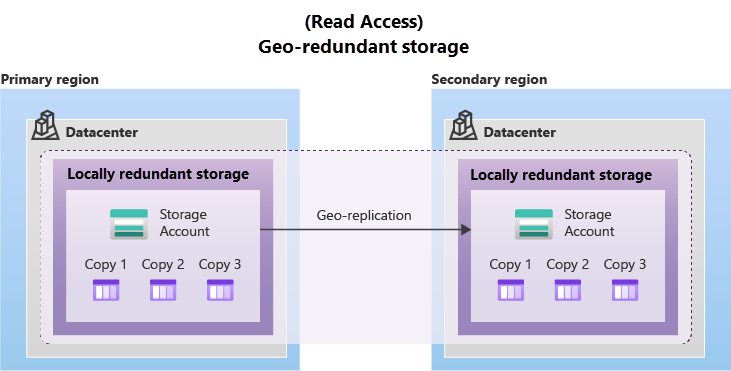
By default, data in the secondary region isn't available for read or write access unless there's a failover to the secondary region. If the primary region becomes unavailable, you can choose to fail over to the secondary region. After the failover has completed, the secondary region becomes the primary region, and you can again read and write data.

 Important:

Because data is replicated to the secondary region asynchronously, a failure that affects the primary region may result in data loss if the primary region can't be recovered. The interval between the most recent writes to the primary region and the last write to the secondary region is known as the recovery point objective (RPO). The RPO indicates the point in time to which data can be recovered. Azure Storage typically has an RPO of less than 15 minutes, although there's currently no SLA on how long it takes to replicate data to the secondary region.

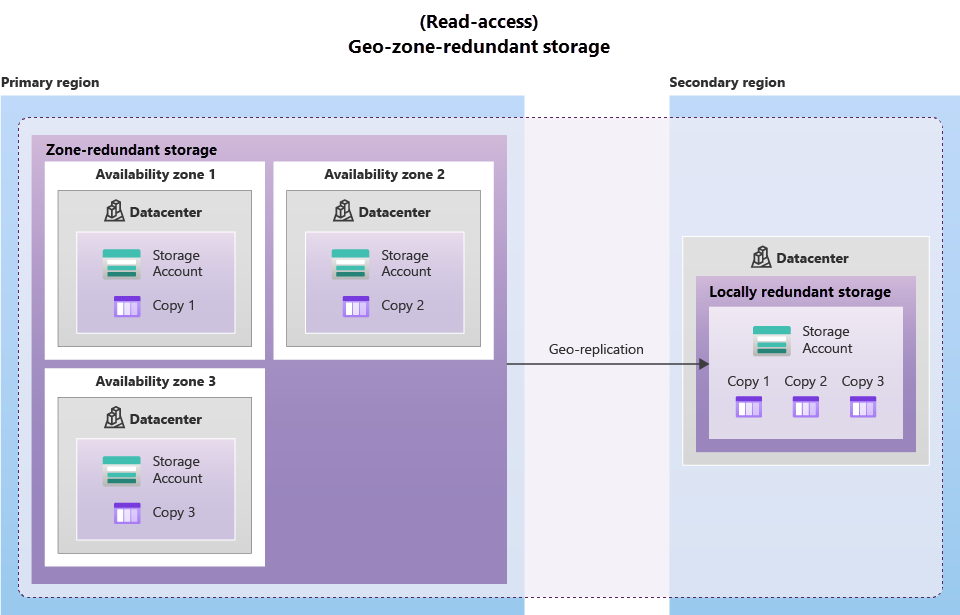
Geo-redundant storage:

GRS copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region (the region pair) using LRS. GRS offers durability for Azure Storage data objects of at least 16 nines (99.99999999999999%) over a given year.



Geo-zone-redundant storage:

GZRS combines the high availability provided by redundancy across availability zones with protection from regional outages provided by geo-replication. Data in a GZRS storage account is copied across three Azure availability zones in the primary region (similar to ZRS) and is also replicated to a secondary geographic region, using LRS, for protection from regional disasters. Microsoft recommends using GZRS for applications requiring maximum consistency, durability, and availability, excellent performance, and resilience for disaster recovery.



GZRS is designed to provide at least 16 nines (99.99999999999999%) of durability of objects over a given year.

Read access to data in the secondary region:

Geo-redundant storage (with GRS or GZRS) replicates your data to another physical location in the secondary region to protect against regional outages. However, that data is available to be read only if the customer or Microsoft initiates a failover from the primary to secondary region. However, if you enable read access to the secondary region, your data is always available, even when the primary region is running optimally. For read access to the secondary region, enable read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS).

Important:

Remember that the data in your secondary region may not be up-to-date due to RPO.

Azure storage services:

The Azure Storage platform includes the following data services:

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

Benefits of Azure Storage:

Azure Storage services offer the following benefits for application developers and IT professionals:

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

Blob storage:

Azure Blob Storage is an object storage solution for the cloud. It can store massive amounts of data, such as text or binary data. Azure Blob Storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold. 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. One advantage of blob storage over disk storage is that it doesn't require developers to think about or manage disks. Data is uploaded as blobs, and Azure takes care of the physical storage needs.

Blob storage is ideal for:

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

Accessing blob storage:

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

Blob storage tiers:

Data stored in the cloud can grow at an exponential pace. To manage costs for your expanding storage needs, it's helpful to organize your data based on attributes like frequency of access and planned retention period. Data stored in the cloud can be handled differently based on how it's generated, processed, and accessed over its lifetime. Some data is actively accessed and modified throughout its lifetime. Some data is accessed frequently early in its lifetime, with access dropping drastically as the data ages. Some data remains idle in the cloud and is rarely, if ever, accessed after it's stored. To accommodate these different access needs, Azure provides several access tiers, which you can use to balance your storage costs with your access needs.

Azure Storage offers different access tiers for your blob storage, helping you store object data in the most cost-effective manner. The available access tiers include:

* **Hot access tier**: Optimized for storing data that is accessed frequently (for example, images for your website).
* **Cool access tier**: Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers).
* **Archive access tier**: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups).

The following considerations apply to the different access tiers:

* Only the hot and cool access tiers can be set at the account level. The archive access tier isn't available at the account level.
* Hot, cool, and archive tiers can be set at the blob level, during or after upload.
* Data in the cool access tier can tolerate slightly lower availability, but still requires high durability, retrieval latency, and throughput characteristics similar to hot data. For cool data, a slightly lower availability service-level agreement (SLA) and higher access costs compared to hot data are acceptable trade-offs for lower storage costs.
* Archive storage stores data offline and offers the lowest storage costs, but also the highest costs to rehydrate and access data.

Azure Files:

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) or Network File System (NFS) protocols. Azure Files file shares can be mounted concurrently by cloud or on-premises deployments. SMB Azure file shares are accessible from Windows, Linux, and macOS clients. NFS Azure Files shares are accessible from Linux or macOS clients. Additionally, SMB Azure file shares can be cached on Windows Servers with Azure File Sync for fast access near where the data is being used.

Azure Files key benefits:

* **Shared access**: Azure file shares support the industry standard SMB and NFS protocols, meaning you can seamlessly replace your on-premises file shares with Azure file shares without worrying about application compatibility.
* **Fully managed**: Azure file shares can be created without the need to manage hardware or an OS. This means you don't have to deal with patching the server OS with critical security upgrades or replacing faulty hard disks.
* **Scripting and tooling**: PowerShell cmdlets and Azure CLI can be used to create, mount, and manage Azure file shares as part of the administration of Azure applications. You can create and manage Azure file shares using Azure portal and Azure Storage Explorer.
* **Resiliency**: Azure Files has been built from the ground up to always be available. Replacing on-premises file shares with Azure Files means you don't have to wake up in the middle of the night to deal with local power outages or network issues.
* **Familiar programmability**: Applications running in Azure can access data in the share via file system I/O APIs. Developers can therefore leverage their existing code and skills to migrate existing applications. In addition to System IO APIs, you can use Azure Storage Client Libraries or the Azure Storage REST API.

Queue storage:

Azure Queue Storage is a service for storing large numbers of messages. Once stored, you can access the messages from anywhere in the world via authenticated calls using HTTP or HTTPS. A queue can contain as many messages as your storage account has room for (potentially millions). Each individual message can be up to 64 KB in size. Queues are commonly used to create a backlog of work to process asynchronously.

Queue storage can be combined with compute functions like Azure Functions to take an action when a message is received. For example, you want to perform an action after a customer uploads a form to your website. You could have the submit button on the website trigger a message to the Queue storage. Then, you could use Azure Functions to trigger an action once the message was received.

Disk storage:

Disk storage, or Azure managed disks, are block-level storage volumes managed by Azure for use with Azure VMs. Conceptually, they’re the same as a physical disk, but they’re virtualized – offering greater resiliency and availability than a physical disk. With managed disks, all you have to do is provision the disk, and Azure will take care of the rest.

**Identify Azure data migration options:**

Now that you understand the different storage options within Azure, it’s important to also understand how to get your data and information into Azure. Azure supports both real-time migration of infrastructure, applications, and data using Azure Migrate as well as asynchronous migration of data using Azure Data Box.

**Azure Migrate:**

Azure Migrate is a service that helps you migrate from an on-premises environment to the cloud. Azure Migrate functions as a hub to help you manage the assessment and migration of your on-premises datacenter to Azure. It provides the following:

* **Unified migration platform**: A single portal to start, run, and track your migration to Azure.
* **Range of tools**: A range of tools for assessment and migration. Azure Migrate tools include Azure Migrate: Discovery and assessment and Azure Migrate: Server Migration. Azure Migrate also integrates with other Azure services and tools, and with independent software vendor (ISV) offerings.
* **Assessment and migration**: In the Azure Migrate hub, you can assess and migrate your on-premises infrastructure to Azure.

Integrated tools:

In addition to working with tools from ISVs, the Azure Migrate hub also includes the following tools to help with migration:

* **Azure Migrate: Discovery and assessment**. Discover and assess on-premises servers running on VMware, Hyper-V, and physical servers in preparation for migration to Azure.
* **Azure Migrate: Server Migration**. Migrate VMware VMs, Hyper-V VMs, physical servers, other virtualized servers, and public cloud VMs to Azure.
* **Data Migration Assistant**. Data Migration Assistant is a stand-alone tool to assess SQL Servers. It helps pinpoint potential problems blocking migration. It identifies unsupported features, new features that can benefit you after migration, and the right path for database migration.
* **Azure Database Migration Service**. Migrate on-premises databases to Azure VMs running SQL Server, Azure SQL Database, or SQL Managed Instances.
* **Web app migration assistant**. Azure App Service Migration Assistant is a standalone tool to assess on-premises websites for migration to Azure App Service. Use Migration Assistant to migrate .NET and PHP web apps to Azure.
* **Azure Data Box**. Use Azure Data Box products to move large amounts of offline data to Azure.

**Azure Data Box:**

Azure Data Box is a physical migration service that helps transfer large amounts of data in a quick, inexpensive, and reliable way. The secure data transfer is accelerated by shipping you a proprietary Data Box storage device that has a maximum usable storage capacity of 80 terabytes. The Data Box is transported to and from your datacenter via a regional carrier. A rugged case protects and secures the Data Box from damage during transit.

You can order the Data Box device via the Azure portal to import or export data from Azure. Once the device is received, you can quickly set it up using the local web UI and connect it to your network. Once you’re finished transferring the data (either into or out of Azure), simply return the Data Box. If you’re transferring data into Azure, the data is automatically uploaded once Microsoft receives the Data Box back. The entire process is tracked end-to-end by the Data Box service in the Azure portal.

Use cases:

Data Box is ideally suited to transfer data sizes larger than 40 TBs in scenarios with no to limited network connectivity. The data movement can be one-time, periodic, or an initial bulk data transfer followed by periodic transfers.

Here are the various scenarios where Data Box can be used to import data to Azure.

* Onetime migration - when a large amount of on-premises data is moved to Azure.
* Moving a media library from offline tapes into Azure to create an online media library.
* Migrating your VM farm, SQL server, and applications to Azure.
* Moving historical data to Azure for in-depth analysis and reporting using HDInsight.
* Initial bulk transfer - when an initial bulk transfer is done using Data Box (seed) followed by incremental transfers over the network.
* Periodic uploads - when large amount of data is generated periodically and needs to be moved to Azure.

Here are the various scenarios where Data Box can be used to export data from Azure.

* Disaster recovery - when a copy of the data from Azure is restored to an on-premises network. In a typical disaster recovery scenario, a large amount of Azure data is exported to a Data Box. Microsoft then ships this Data Box, and the data is restored on your premises in a short time.
* Security requirements - when you need to be able to export data out of Azure due to government or security requirements.
* Migrate back to on-premises or to another cloud service provider - when you want to move all the data back to on-premises, or to another cloud service provider, export data via Data Box to migrate the workloads.

Once the data from your import order is uploaded to Azure, the disks on the device are wiped clean in accordance with NIST 800-88r1 standards. For an export order, the disks are erased once the device reaches the Azure datacenter.

**Identify Azure file movement options:**

In addition to large scale migration using services like Azure Migrate and Azure Data Box, Azure also has tools designed to help you move or interact with individual files or small file groups. Among those tools are AzCopy, Azure Storage Explorer, and Azure File Sync.

**AzCopy:**

AzCopy is a command-line utility that you can use to copy blobs or files to or from your storage account. With AzCopy, you can upload files, download files, copy files between storage accounts, and even synchronize files. AzCopy can even be configured to work with other cloud providers to help move files back and forth between clouds.

Important:

Synchronizing blobs or files with AzCopy is one-direction synchronization. When you synchronize, you designated the source and destination, and AzCopy will copy files or blobs in that direction. It doesn't synchronize bi-directionally based on timestamps or other metadata.

**Azure Storage Explorer:**

Azure Storage Explorer is a standalone app that provides a graphical interface to manage files and blobs in your Azure Storage Account. It works on Windows, macOS, and Linux operating systems and uses AzCopy on the backend to perform all of the file and blob management tasks. With Storage Explorer, you can upload to Azure, download from Azure, or move between storage accounts.

**Azure File Sync:**

Azure File Sync is a tool that lets you centralize your file shares in Azure Files and keep the flexibility, performance, and compatibility of a Windows file server. It’s almost like turning your Windows file server into a miniature content delivery network. Once you install Azure File Sync on your local Windows server, it will automatically stay bi-directionally synced with your files in Azure.

With Azure File Sync, you can:

* Use any protocol that's available on Windows Server to access your data locally, including SMB, NFS, and FTPS.
* Have as many caches as you need across the world.
* Replace a failed local server by installing Azure File Sync on a new server in the same datacenter.
* Configure cloud tiering so the most frequently accessed files are replicated locally, while infrequently accessed files are kept in the cloud until requested.