**AZ900-PART2:**

**Describe Azure architecture and services (35–40%)**

**Learning objectives**

After completing this part, you’ll be able to:

1. Describe Azure regions, region pairs, and sovereign regions.
2. Describe Availability Zones.
3. Describe Azure datacenters.
4. Describe Azure resources and Resource Groups.
5. Describe subscriptions.
6. Describe management groups.
7. Describe the hierarchy of resource groups, subscriptions, and management groups.

To create and use **Azure services**, you need an **Azure subscription**.

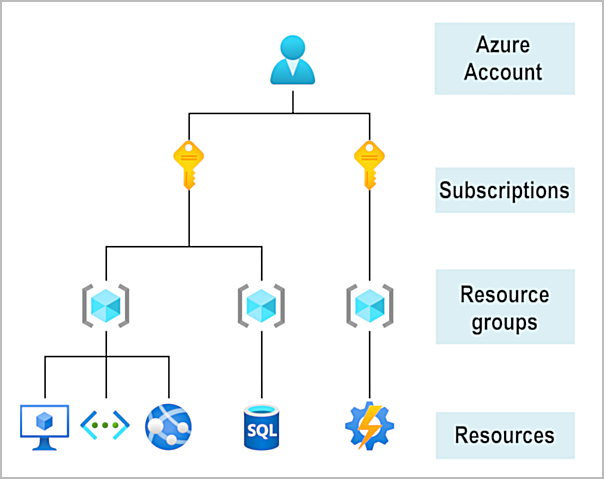
**Learn Sandbox** is temporary subscription.

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.

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 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.

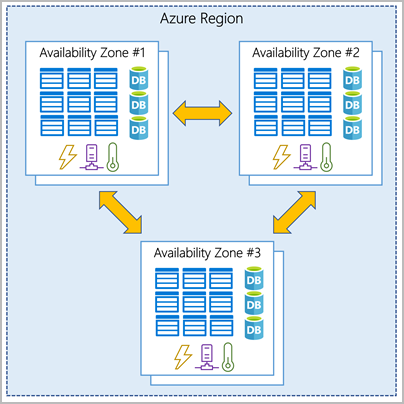
--------------------------------------------------------------------------------------------------------------------------------------

The 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.  Azure has datacenters around the world!

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.

**Note:** 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 **Microsoft Entra ID, Azure Traffic Manager, and Azure DNS.**

**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.

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**. 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:

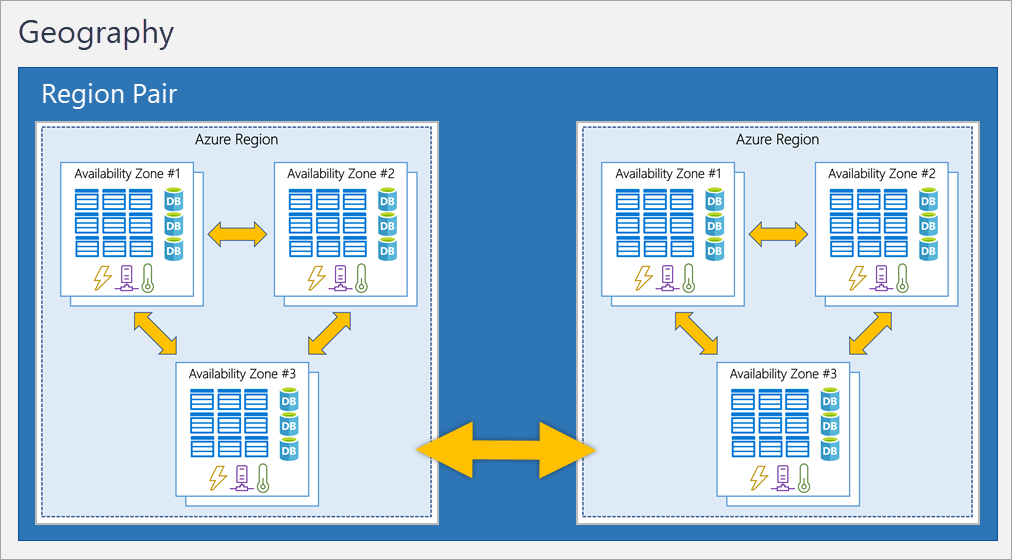
* **Zonal services**: You pin the resource to a specific zone (for example, VMs, managed disks, IP addresses).
* **Zone-redundant services**: The platform replicates automatically across zones (for example, zone-redundant storage, SQL Database).
* **Non-regional services**: Services are always available 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**.

Most Azure regions are paired with another region within the same geography (such as US, Europe, or Asia) at least **300 miles** away. (replication of resources )

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.



**Important**

Most regions 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** are instances of Azure that are isolated from the main instance of Azure. These regions are physical and logical network-isolated instances of Azure for U.S. government agencies and partners.

* 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.

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. 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.

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. 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 **Microsoft Entra ID** or in a directory that Microsoft Entra ID trusts.

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.

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.

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.

**Resource group permissions are inherited by all resources in the resource group, current and future.**

**A region pair is when an Azure Region is paired with another region at least 300 miles away.**

**Describe Azure compute and networking services**

**Learning objectives**

Upon completion of this module, you'll be able to:

1. Compare compute types, including container instances, virtual machines, and functions
2. Describe virtual machine (VM) options, including VMs, Virtual Machine Scale Sets, availability sets, Azure Virtual Desktop
3. Describe resources required for virtual machines
4. Describe application hosting options, including Azure Web Apps, containers, and virtual machines
5. Describe virtual networking, including the purpose of Azure Virtual Networks, Azure virtual subnets, peering, Azure DNS, VPN Gateway, and ExpressRoute
6. Define public and private endpoints

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.

**Virtual machine scale sets** let you create and manage a group of identical, load-balanced VMs. 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** are another tool to help you build a more resilient, highly available environment. Availability sets do this by grouping VMs in two ways: **update domain:** The update domain groups VMs that can be rebooted at the same time. An update group going through the update process is given a 30-minute time to recover before maintenance on the next update domain starts. and **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. – Things for you:

* 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** 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 provides **centralized security management for users' desktops with Microsoft Entra ID.** 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.

While virtual machines are an excellent way to reduce costs, 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.

**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**, and Azure supports Docker. Azure Container Instances are a **platform as a service (PaaS) offering.**

**Azure Kubernetes Service (AKS)** is a container orchestration service. An **orchestration service manages the lifecycle of containers.** When you're deploying a fleet of containers, AKS can make fleet management simpler and more efficient.

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,* 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** 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. 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 scale automatically based on demand, so they may be a good choice when demand is variable.  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 you need to host your application on Azure, you might initially turn to a virtual machine (VM) or containers. 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.

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.

**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.

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 endpoin**ts 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.

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.

* **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.

 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:

1. **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.
2. **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.
3. **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

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:

1. **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.
2. **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.

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.

**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.

**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 gateway connectivities:**

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

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 setting up a VPN gateway, you must specify the type of VPN - either **policy-based or route-based**. The primary distinction between these two types is how they determine which traffic needs encryption. In Azure, regardless of the VPN type, the method of authentication employed is a pre-shared key.

* **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. typically restored within a few seconds for planned maintenance and within 90 seconds for unplanned disruptions.

### **Active/active**: you assign a unique public IP address to each instance.

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

**ExpressRoute uses the BGP(Border Gateway Protocol ).** BGP is used to exchange routes between on-premises networks and resources running in Azure.

With ExpressRoute, your data doesn't travel over the public internet, so it's not exposed to the potential risks associated with internet communications. 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** 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:

1. Reliability and performance: Azure DNS uses anycast networking, so each DNS query is closest available DNS answered by the server to provide fast performance and high availability for your domain.
2. Security: i) Azure role-based access control (Azure RBAC) to control who has access to specific actions for your organization.

ii)Activity logs to monitor how a user in your organization modified a resource or to find an error when troubleshooting.

iii)Resource locking to lock a subscription, resource group, or resource. Locking prevents other users in your organization from accidentally deleting or modifying critical resources.

3.Ease of Use: Azure DNS can manage DNS records for your Azure services and provide DNS for your external resources as well.

4.Customizable virtual networks: Azure DNS also supports private DNS domains.

5.Alias records: 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

**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.

----------------------------------------------------------------------------------------------

**Describe Azure storage accounts**

**Learning objectives**

After completing this module, you’ll be able to:

1. Compare Azure storage services.
2. Describe storage tiers.
3. Describe redundancy options.
4. Describe storage account options and storage types.
5. Identify options for moving files, including AzCopy, Azure Storage Explorer, and Azure File Sync.
6. Describe migration options, including Azure Migrate and Azure Data Box.

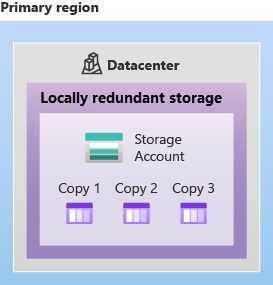
When you create your storage account, you’ll start by picking the storage account type. The type of account determines the storage services and redundancy options and has an impact on the use cases. Every storage account in Azure must have a unique-in-Azure account name. Has Redundancy : Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures.

## **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**

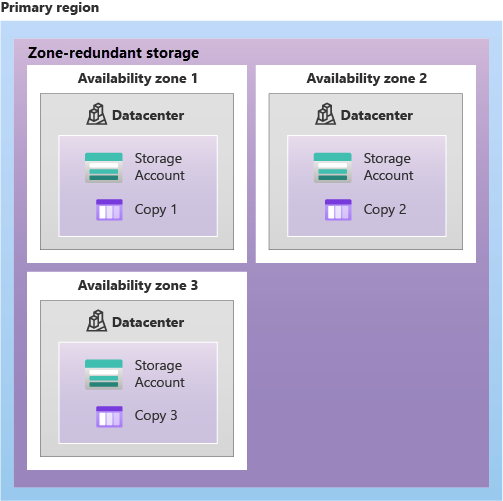
Locally redundant storage (LRS) replicates your data three times within a single data center in the primary region. LRS provides at least 11 nines of durability (99.999999999%) of objects over a given year.

****

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.



## **Redundancy in a secondary region**

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.

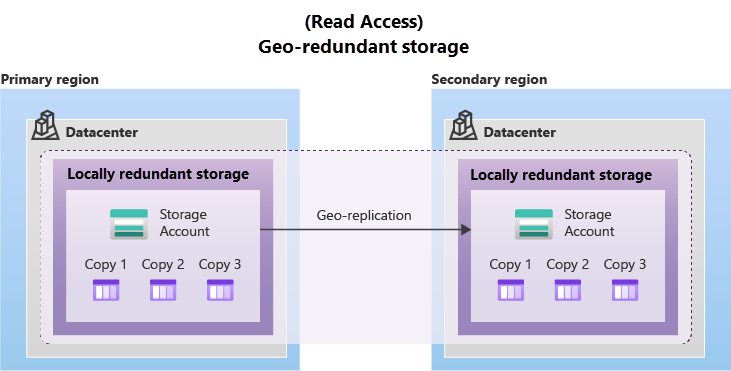
By default, data in the secondary region isn't available for read or write access unless there's a failover to the secondary region.  After the failover has completed, the secondary region becomes the primary region, and you can again read and write data.

**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.

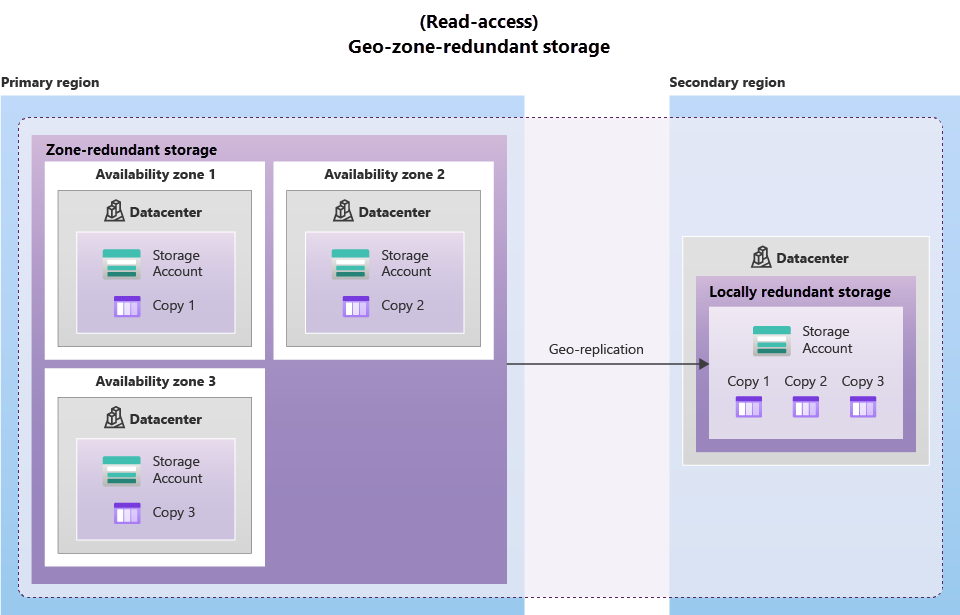


**Geo-zone-redundant storage**

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.

For read access to the secondary region, enable read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS).

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



* **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.
* **Azure Tables:** NoSQL table option for structured, non-relational data.

Blob storage is ideal for:

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

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).
* **Cold access tier**: Optimized for storing data that is infrequently accessed and stored for at least 90 days.
* **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:

* Hot and cool access tiers can be set at the account level. The cold and archive access tiers aren't available at the account level.
* Hot, cool, cold, and archive tiers can be set at the blob level, during or after upload.
* Data in the cool and cold access tiers can tolerate slightly lower availability, but still requires high durability, retrieval latency, and throughput characteristics similar to hot data. For cool and cold data, a 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 File storage** 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 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.

**Azure 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.

**Azure Table storage** stores large amounts of structured data. Azure tables are a NoSQL datastore that accepts authenticated calls from inside and outside the Azure cloud. This enables you to use Azure tables to build your hybrid or multi-cloud solution and have your data always available. Azure tables are ideal for storing structured, non-relational data.

**Azure Migrate** is a service that helps you migrate from an on-premises environment to the cloud. [independent software vendor (ISV) o]

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** is a command-line utility that you can use to copy blobs or files to or from your storage account. Synchronizing blobs or files with AzCopy is one-direction synchronization.

**Azure Storage Explorer** is a standalone app that provides a graphical interface to manage files and blobs in your Azure Storage Account. With Storage Explorer, you can upload to Azure, download from Azure, or move between storage accounts.

**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. 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.

**Azure File Sync maintains a bidirectional synchronization of files between your on-premises and cloud Windows servers.**

**Geo-redundant storage (GRS) and geo-zone-redundant storage (GZRS) both provide 16 nines of durability.**

**Azure Blobs is a massively scalable object store for text and binary data. Azure Blobs also includes support for big data analytics through Data Lake Storage Gen2.**

# **Describe Azure identity, access, and security:**

## **Learning objectives:**

By the end of this module, you will be able to:

1. Describe directory services in Azure, including Microsoft Entra ID and Microsoft Entra Domain Services
2. Describe authentication methods in Azure, including single sign-on (SSO), multifactor authentication (MFA), and passwordless
3. Describe external identities and guest access in Azure
4. Describe Microsoft Entra Conditional Access
5. Describe Azure Role Based Access Control (RBAC)
6. Describe the concept of Zero Trust
7. Describe the purpose of the defense in depth model
8. Describe the purpose of Microsoft Defender for Cloud

**Microsoft Entra ID** is a directory service that enables you to sign in and access both Microsoft cloud applications and cloud applications that you develop. Microsoft Entra ID can also help you maintain your on-premises Active Directory deployment. Microsoft Entra ID is Microsoft's cloud-based identity and access management service.

When you connect Active Directory with Microsoft Entra ID, Microsoft can help protect you by detecting suspicious sign-in attempts at no extra cost. For example, Microsoft Entra ID can detect sign-in attempts from unexpected locations or unknown devices.

Microsoft Entra ID is for:

* **IT administrators**. Administrators can use Microsoft Entra ID to control access to applications and resources based on their business requirements.
* **App developers**. Developers can use Microsoft Entra ID to provide a standards-based approach for adding functionality to applications that they build, such as adding SSO functionality to an app or enabling an app to work with a user's existing credentials.
* **Users**. Users can manage their identities and take maintenance actions like self-service password reset.
* **Online service subscribers**. Microsoft 365, Microsoft Office 365, Azure, and Microsoft Dynamics CRM Online subscribers are already using Microsoft Entra ID to authenticate into their account.

Microsoft Entra ID provides services such as:

* **Authentication**: This includes verifying identity to access applications and resources. It also includes providing functionality such as self-service password reset, multifactor authentication, a custom list of banned passwords, and smart lockout services.
* **Single sign-on**: Single sign-on (SSO) enables you to remember only one username and one password to access multiple applications. A single identity is tied to a user, which simplifies the security model. As users change roles or leave an organization, access modifications are tied to that identity, which greatly reduces the effort needed to change or disable accounts.
* **Application management**: You can manage your cloud and on-premises apps by using Microsoft Entra ID. Features like Application Proxy, SaaS apps, the My Apps portal, and single sign-on provide a better user experience.
* **Device management**: Along with accounts for individual people, Microsoft Entra ID supports the registration of devices. Registration enables devices to be managed through tools like **Microsoft Intune**. It also allows for device-based Conditional Access policies to restrict access attempts to only those coming from known devices, regardless of the requesting user account.

You can connect Active Directory with Microsoft Entra ID, enabling a consistent identity experience between cloud and on-premises. **Microsoft Entra Connect** synchronizes user identities between on-premises Active Directory and Microsoft Entra ID. Microsoft Entra Connect synchronizes changes between both identity systems, so you can use features like SSO, multifactor authentication, and self-service password reset under both systems.

**Microsoft Entra Domain Services** is a service that provides managed domain services such as domain join, group policy, lightweight directory access protocol (LDAP), and Kerberos/NTLM authentication.

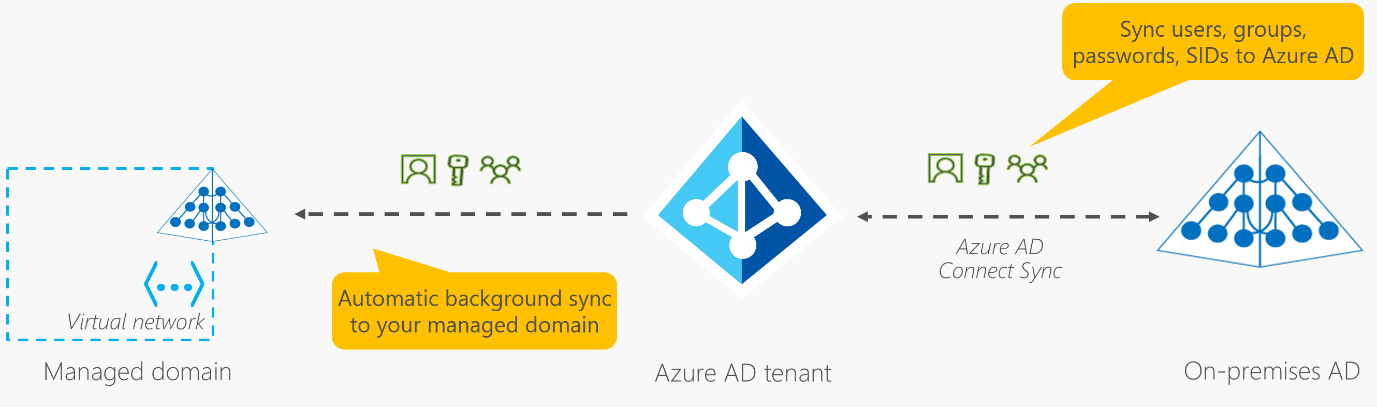
When you create a Microsoft Entra Domain Services managed domain, you define a unique namespace. This namespace is the domain name. Two Windows Server domain controllers are then deployed into your selected Azure region. This deployment of DCs (Domain Controllers)is known as a replica set.

You don't need to manage, configure, or update these DCs. The Azure platform handles the DCs as part of the managed domain, including backups and encryption at rest using Azure Disk Encryption.

### **Is information synchronized?**

A managed domain is configured to perform a one-way synchronization from Microsoft Entra ID to Microsoft Entra Domain Services. You can create resources directly in the managed domain, but they aren't synchronized back to Microsoft Entra ID. In a hybrid environment with an on-premises AD DS environment, Microsoft Entra Connect synchronizes identity information with Microsoft Entra ID, which is then synchronized to the managed domain.

Applications, services, and VMs in Azure that connect to the managed domain can then use common Microsoft Entra Domain Services features such as domain join, group policy, LDAP, and Kerberos/NTLM authentication.



**Authentication** is the process of establishing the identity of a person, service, or device. It requires the person, service, or device to provide some type of credential to prove who they are. Authentication is like presenting ID when you’re traveling. It doesn’t confirm that you’re ticketed, it just proves that you're who you say you are. Azure supports multiple authentication methods, including standard passwords, single sign-on (SSO), multifactor authentication (MFA), and passwordless.

Notice **Passwordless authentication** is high security and high convenience while passwords on their own are low security but high convenience.

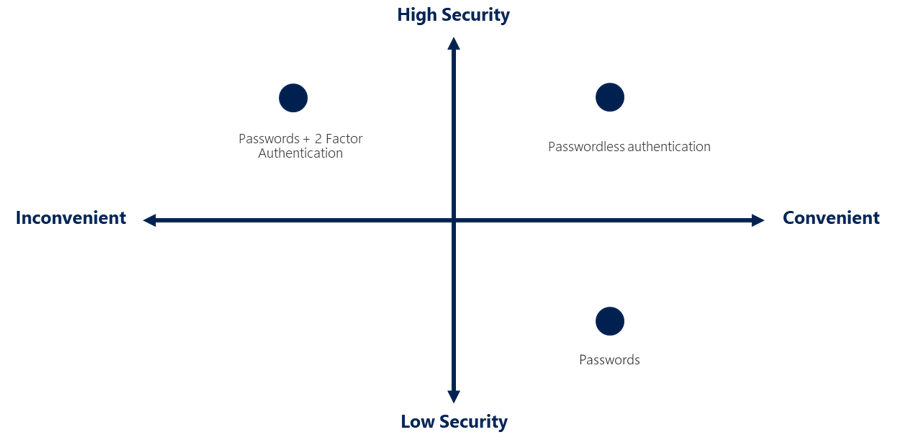
**Single sign-on (SSO)** enables a user to sign in one time and use that credential to access multiple resources and applications from different providers.With SSO, you need to remember only one ID and one password. Access across applications is granted to a single identity that's tied to the user, which simplifies the security model. As users change roles or leave an organization, access is tied to a single identity. This change greatly reduces the effort needed to change or disable accounts. Using SSO for accounts makes it easier for users to manage their identities and for IT to manage users.

**Important:**Single sign-on is only as secure as the initial authenticator because the subsequent connections are all based on the security of the initial authenticator.

**Multifactor authentication** is the process of prompting a user for an extra form (or factor) of identification during the sign-in process. MFA helps protect against a password compromise in situations where the password was compromised but the second factor wasn't. Multifactor authentication provides additional security for your identities by requiring two or more elements to fully authenticate. These elements fall into three categories:

* Something the user knows – this might be a challenge question.
* Something the user has – this might be a code that's sent to the user's mobile phone.
* Something the user is – this is typically some sort of biometric property, such as a fingerprint or face scan.

**Microsoft Entra multifactor authentication** is a Microsoft service that provides multifactor authentication capabilities. Microsoft Entra multifactor authentication enables users to choose an additional form of authentication during sign-in, such as a phone call or mobile app notification.

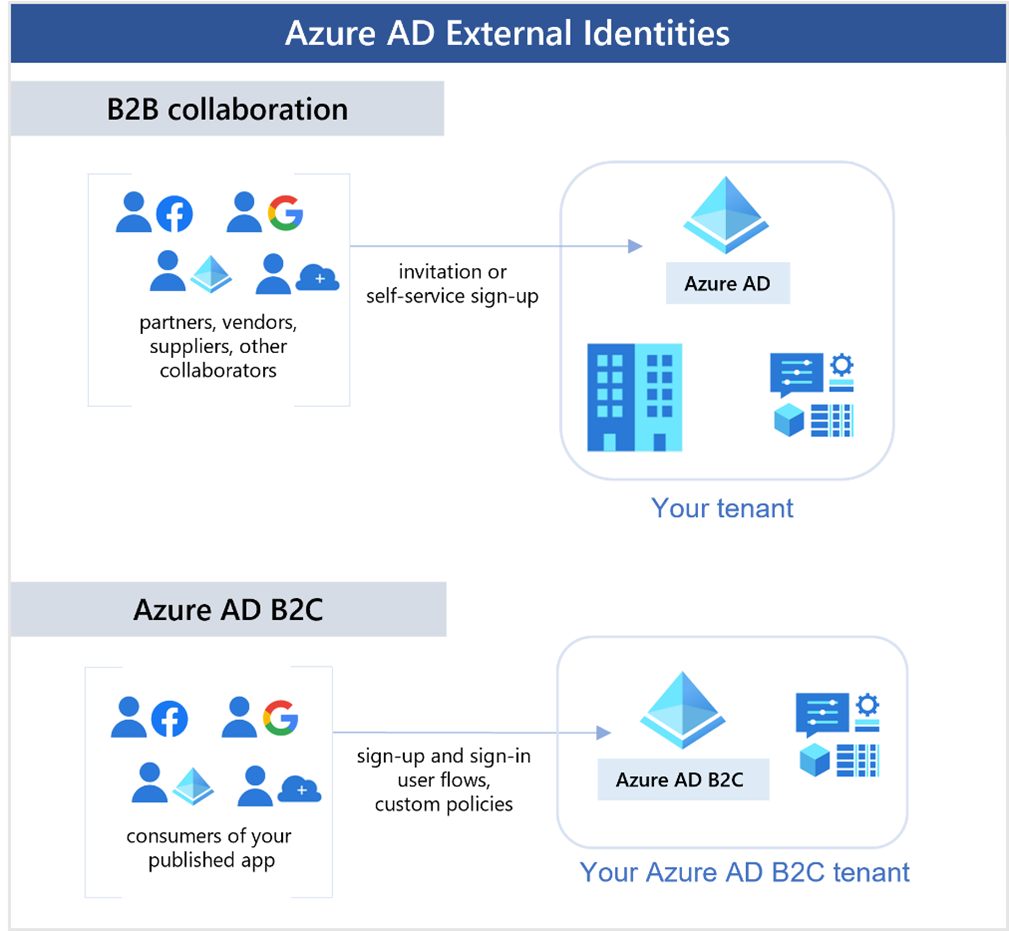


People are more likely to comply when it's easy and convenient to do so. **Passwordless** authentication methods are more convenient because the password is removed and replaced with something you have, plus something you are, or something you know. Passwordless authentication needs to be set up on a device before it can work. For example, your computer is something you have. Once it’s been registered or enrolled, Azure now knows that it’s associated with you. Now that the computer is known, once you provide something you know or are (such as a PIN or fingerprint), you can be authenticated without using a password.

Microsoft global Azure and Azure Government offer the following three passwordless authentication options that integrate with Microsoft Entra ID:

1. **Windows Hello for Business:** Windows Hello for Business is ideal for information workers that have their own designated Windows PC. The biometric and PIN credentials are directly tied to the user's PC, which prevents access from anyone other than the owner.
2. **Microsoft Authenticator app:** The Authenticator App turns any iOS or Android phone into a strong, passwordless credential. Users can sign-in to any platform or browser by getting a notification to their phone, matching a number displayed on the screen to the one on their phone, and then using their biometric (touch or face) or PIN to confirm.
3. **FIDO2 security keys:** The FIDO (**Fast IDentity Online**) Alliance helps to promote open authentication standards and reduce the use of passwords as a form of authentication. FIDO2 is the latest standard that incorporates the web authentication (WebAuthn) standard.

An **external identity** is a person, device, service, etc. that is outside your organization. **Microsoft Entra External ID** refers to all the ways you can securely interact with users outside of your organization. If you want to collaborate with partners, distributors, suppliers, or vendors, you can share your resources and define how your internal users can access external organizations.



The following capabilities make up External Identities:

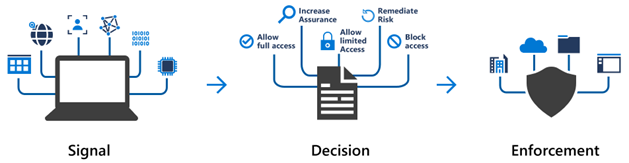
* **Business to business (B2B) collaboration** - Collaborate with external users by letting them use their preferred identity to sign-in to your Microsoft applications or other enterprise applications (SaaS apps, custom-developed apps, etc.). B2B collaboration users are represented in your directory, typically as guest users.
* **B2B direct connect** - Establish a mutual, two-way trust with another Microsoft Entra organization for seamless collaboration. B2B direct connect currently supports Teams shared channels, enabling external users to access your resources from within their home instances of Teams. B2B direct connect users aren't represented in your directory, but they're visible from within the Teams shared channel and can be monitored in Teams admin center reports.
* **Microsoft Azure Active Directory business to customer (B2C)** - Publish modern SaaS apps or custom-developed apps (excluding Microsoft apps) to consumers and customers, while using Azure AD B2C for identity and access management.

**Conditional Access** is a tool that Microsoft Entra ID uses to allow (or deny) access to resources based on **identity signals**. These signals include who the user is, where the user is, and what device the user is requesting access from.

Conditional Access also provides a more granular multifactor authentication experience for users. For example, a user might not be challenged for second authentication factor if they're at a known location. However, they might be challenged for a second authentication factor if their sign-in signals are unusual or they're at an unexpected location.

During sign-in, Conditional Access collects signals from the user, makes decisions based on those signals, and then enforces that decision by allowing or denying the access request or challenging for a multifactor authentication response.

The following diagram illustrates this flow:



Here, the signal might be the user's location, the user's device, or the application that the user is trying to access.

Based on these signals, the decision might be to allow full access if the user is signing in from their usual location. If the user is signing in from an unusual location or a location that's marked as high risk, then access might be blocked entirely or possibly granted after the user provides a second form of authentication.

Enforcement is the action that carries out the decision. For example, the action is to allow access or require the user to provide a second form of authentication.

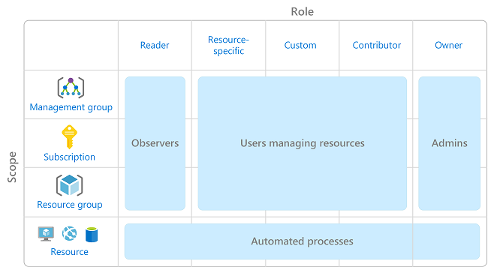
Conditional Access is useful when you need to:

* **Require multifactor authentication (MFA) to access an application depending on the requester’s role, location, or network.** For example, you could require MFA for administrators but not regular users or for people connecting from outside your corporate network.
* **Require access to services only through approved client applications**. For example, you could limit which email applications are able to connect to your email service.
* **Require users to access your application only from managed devices**. A managed device is a device that meets your standards for security and compliance.
* **Block access from untrusted sources, such as access from unknown or unexpected locations.**

Azure enables you to control access through **Azure role-based access control (Azure RBAC).**Azure provides built-in roles that describe common access rules for cloud resources. You can also define your own roles. Each role has an associated set of access permissions that relate to that role. When you assign individuals or groups to one or more roles, they receive all the associated access permissions.

So, if you hire a new engineer and add them to the Azure RBAC group for engineers, they automatically get the same access as the other engineers in the same Azure RBAC group. Similarly, if you add additional resources and point Azure RBAC at them, everyone in that Azure RBAC group will now have those permissions on the new resources as well as the existing resources.

**Role-based access control** is applied to **a scope**, which is a resource or set of resources that this access applies to.

****

Azure RBAC is hierarchical, in that when you grant access at a parent scope, those permissions are inherited by all child scopes. For example:

* When you assign the Owner role to a user at the management group scope, that user can manage everything in all subscriptions within the management group.
* When you assign the Reader role to a group at the subscription scope, the members of that group can view every resource group and resource within the subscription.

Azure RBAC is enforced on any action that's initiated against an Azure resource that passes through **Azure Resource Manager**. Resource Manager is a management service that provides a way to organize and secure your cloud resources.

You typically access Resource Manager from the Azure portal, Azure Cloud Shell, Azure PowerShell, and the Azure CLI. Azure RBAC doesn't enforce access permissions at the application or data level. Application security must be handled by your application.

Azure RBAC uses an allow model. When you're assigned a role, Azure RBAC allows you to perform actions within the scope of that role. If one role assignment grants you read permissions to a resource group and a different role assignment grants you write permissions to the same resource group, you have both read and write permissions on that resource group.

**Zero Trust** is a security model that assumes the worst case scenario and protects resources with that expectation.

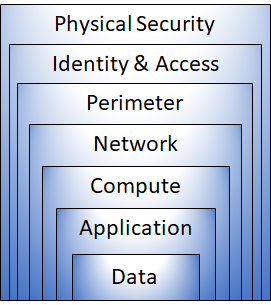
To address this new world of computing, Microsoft highly recommends the Zero Trust security model, which is based on these guiding principles:

* **Verify explicitly** - Always authenticate and authorize based on all available data points.
* **Use least privilege access** - Limit user access with Just-In-Time and Just-Enough-Access (JIT/JEA), risk-based adaptive policies, and data protection.
* **Assume breach** - Minimize blast radius and segment access. Verify end-to-end encryption. Use analytics to get visibility, drive threat detection, and improve defenses.

Traditionally, corporate networks were restricted, protected, and generally assumed safe. Only managed computers could join the network, VPN access was tightly controlled, and personal devices were frequently restricted or blocked.

The Zero Trust model flips that scenario. Instead of assuming that a device is safe because it’s within the corporate network, it requires everyone to authenticate. Then grants access based on authentication rather than location.

The objective of **defense-in-depth** is to protect information and prevent it from being stolen by those who aren't authorized to access it. A defense-in-depth strategy uses a series of mechanisms to slow the advance of an attack that aims at acquiring unauthorized access to data.

****

Each layer provides protection so that if one layer is breached, a subsequent layer is already in place to prevent further exposure. This approach removes reliance on any single layer of protection. It slows down an attack and provides alert information that security teams can act upon, either automatically or manually.

Here's a brief overview of the role of each layer:

1. The **physical security laye**r is the first line of defense to protect computing hardware in the datacenter.
2. The **identity and access layer** controls access to infrastructure and change control.
3. The **perimeter layer** uses distributed denial of service (DDoS) protection to filter large-scale attacks before they can cause a denial of service for users.
4. The **network layer** limits communication between resources through segmentation and access controls.
5. The **compute layer** secures access to virtual machines.
6. The **application layer** helps ensure that applications are secure and free of security vulnerabilities.
7. The **data layer** controls access to business and customer data that you need to protect.

**Defender for Cloud** is a monitoring tool for security posture management and threat protection. It monitors your cloud, on-premises, hybrid, and multi-cloud environments to provide guidance and notifications aimed at strengthening your security posture. Defender for Cloud provides the tools needed to harden your resources, track your security posture, protect against cyber attacks, and streamline security management. Deployment of Defender for Cloud is easy, it’s already natively integrated to Azure.

When necessary, Defender for Cloud can automatically deploy a **Log Analytics** agent to gather security-related data. For Azure machines, deployment is handled directly. For hybrid and multi-cloud environments, Microsoft Defender plans are extended to non Azure machines with the help of Azure Arc. **Cloud security posture management (CSPM**) features are extended to multi-cloud machines without the need for any agents. Defender for Cloud fills three vital needs as you manage the security of your resources and workloads in the cloud and on-premises:



This section is not that important in AZ-900 exam POV. For more info, Check : <https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/9-describe-microsoft-defender-for-cloud>

\*\*\*:

**Conditional Access is a tool that Microsoft Entra ID uses to allow (or deny) access to resources based on identity signals. Conditional Access might challenge you for a second authentication factor if your sign-in signals are unusual or from an unexpected location.**

**Passwordless is an authentication method that relies on something you have; plus something you are or something you know. For example, Windows Hello is a passwordless authentication method.**

**Zero Trust is a security model that assumes the worst case scenario and protects resources with that expectation.**

**Role-based access control, using an allow model, grants all of the permissions assigned in all of the assigned roles. *SEE YOU IN PART-3~~***