MODULE-2 : AZURE SERVICES

TABLE OF CONTENTS

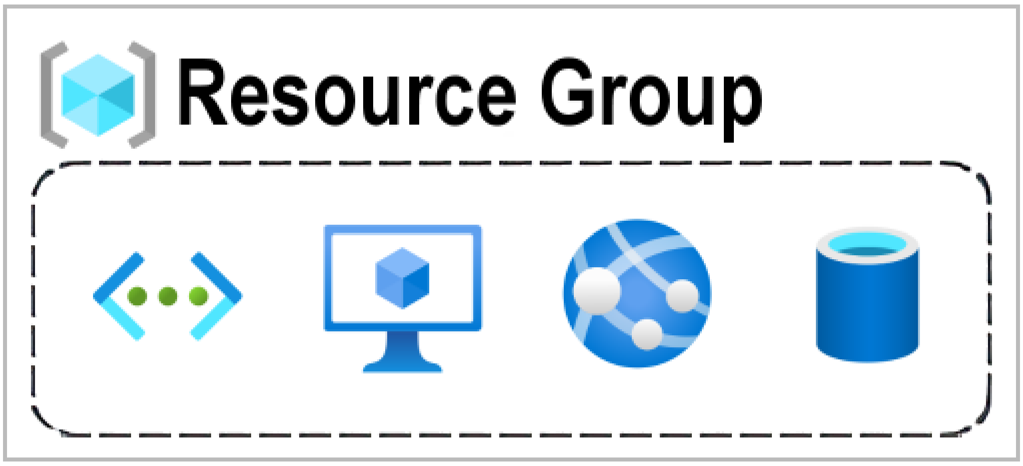
1.CORE ARCHITECTURAL COMPONENTS

Describe 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.

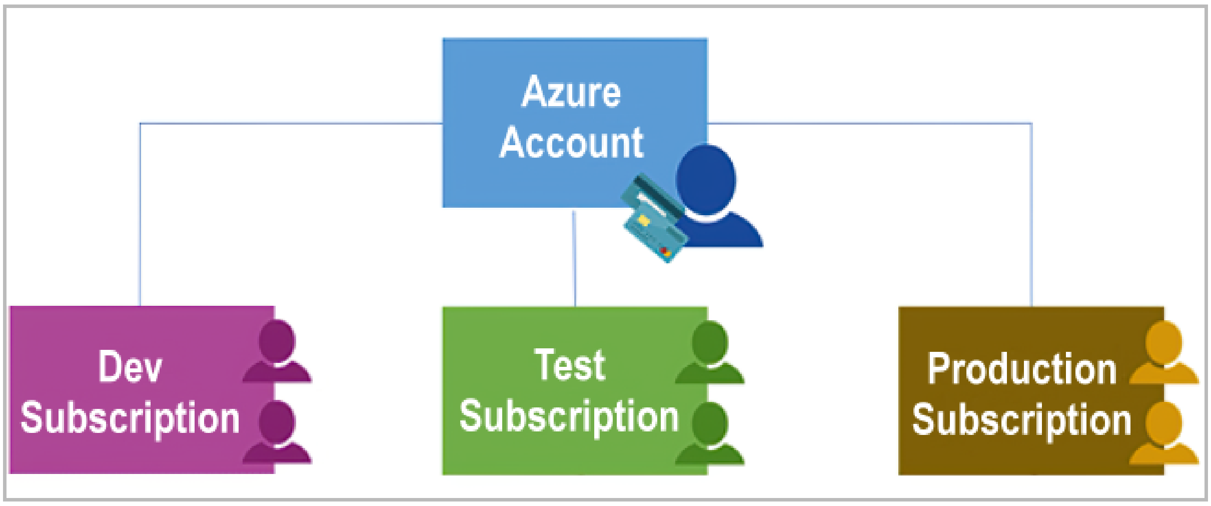
When you're provisioning resources, it's good to think about the resource group structure that best suits your needs.

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.

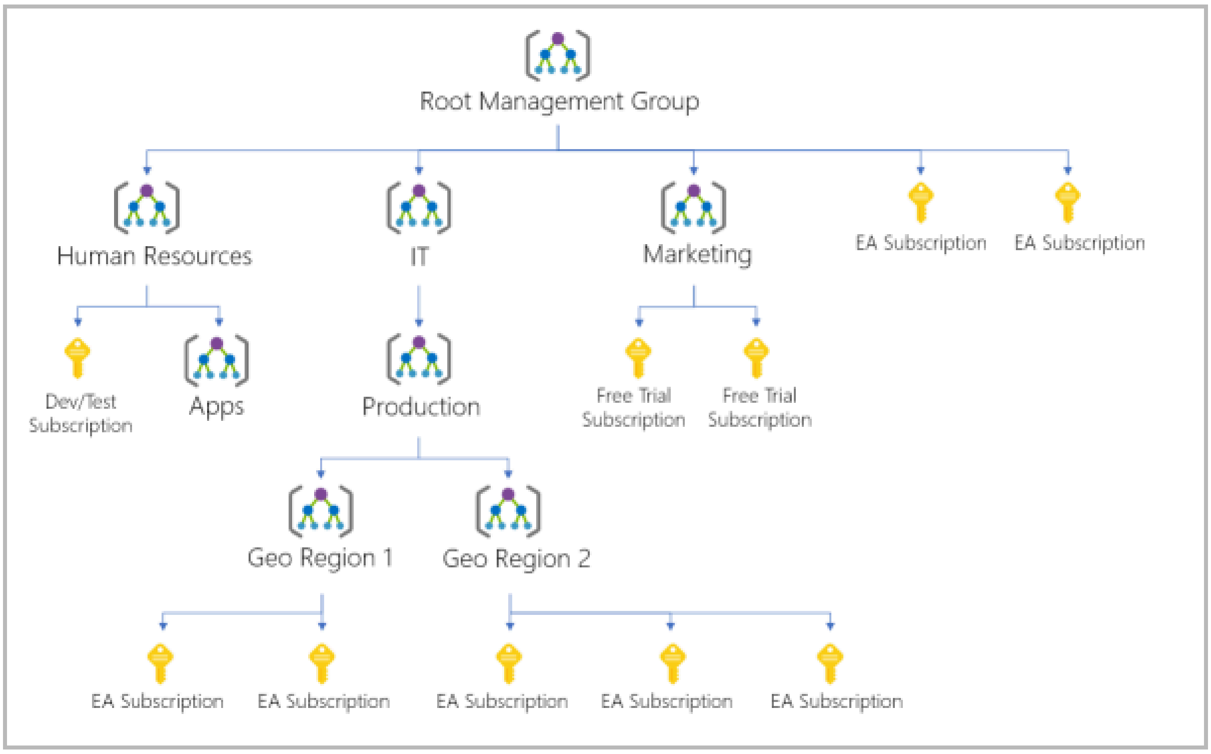
Azure management groups

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.

2.Describe Azure compute and networking services

2.1Virtual machines

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:

1. update domain
2. 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 datacentre fails, you can create VMs running on Azure to run your critical applications and then shut them down when the primary datacentre 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)

Describe 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.

2.2Azure 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 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.

Describe 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 Express Route

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, datacentres, or other facilities to the Microsoft cloud. Each location would have its own ExpressRoute circuit.

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.

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

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.

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.

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.

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.

3.Describe Azure storage services

This module introduces you to storage in Azure, including things such as different types of storage and how a distributed infrastructure can make your data more resilient.

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)

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.

Describe 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:

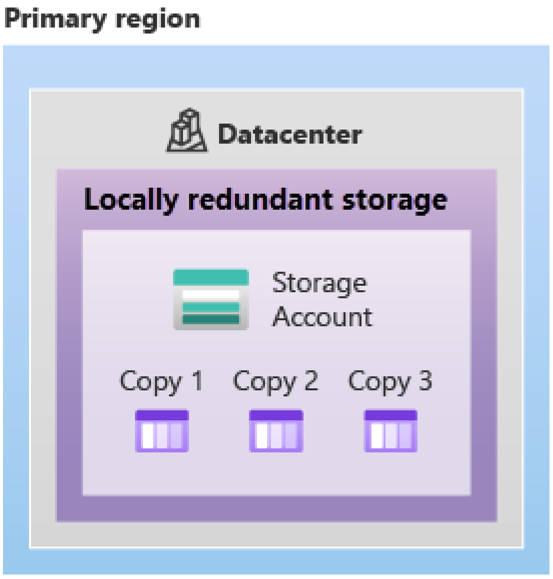
- How your data is replicated in the primary region.
- Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters.
- Whether your application requires read access to the replicated data in the secondary region if the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

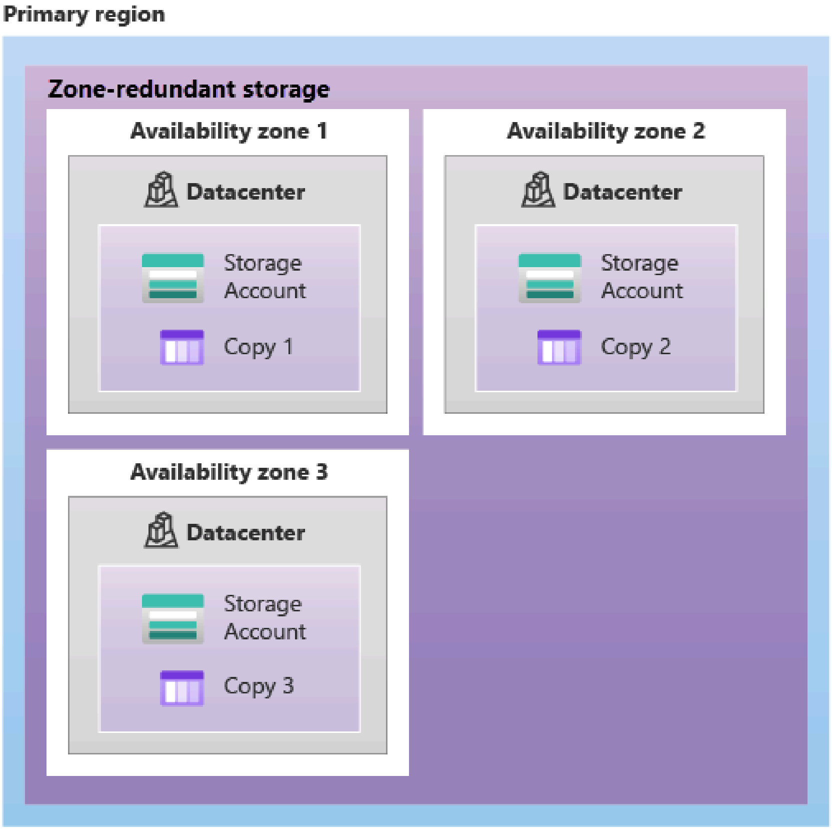
Locally redundant storage (LRS) replicates your data three times within a single data center in the primary region. LRS provides at least 11 nines of durability (99.999999999%) of objects over a given year.



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.

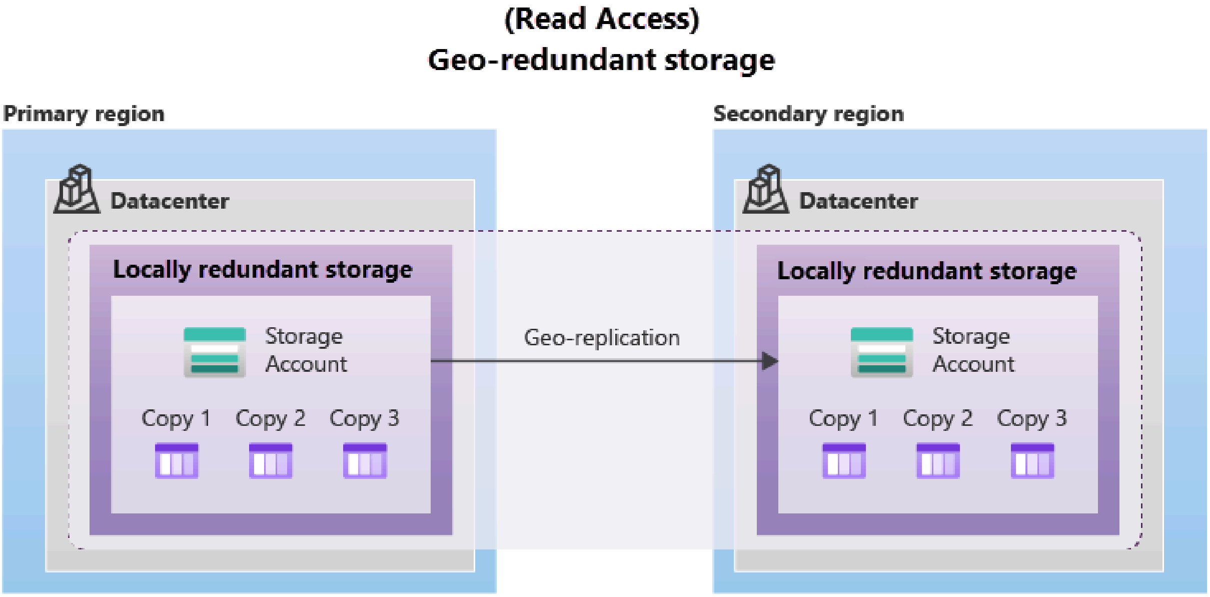
By default, data in the secondary region isn't available for read or write access unless there's a failover to the secondary region. If the primary region becomes unavailable, you can choose to fail over to the secondary region. After the failover has completed, the secondary region becomes the primary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

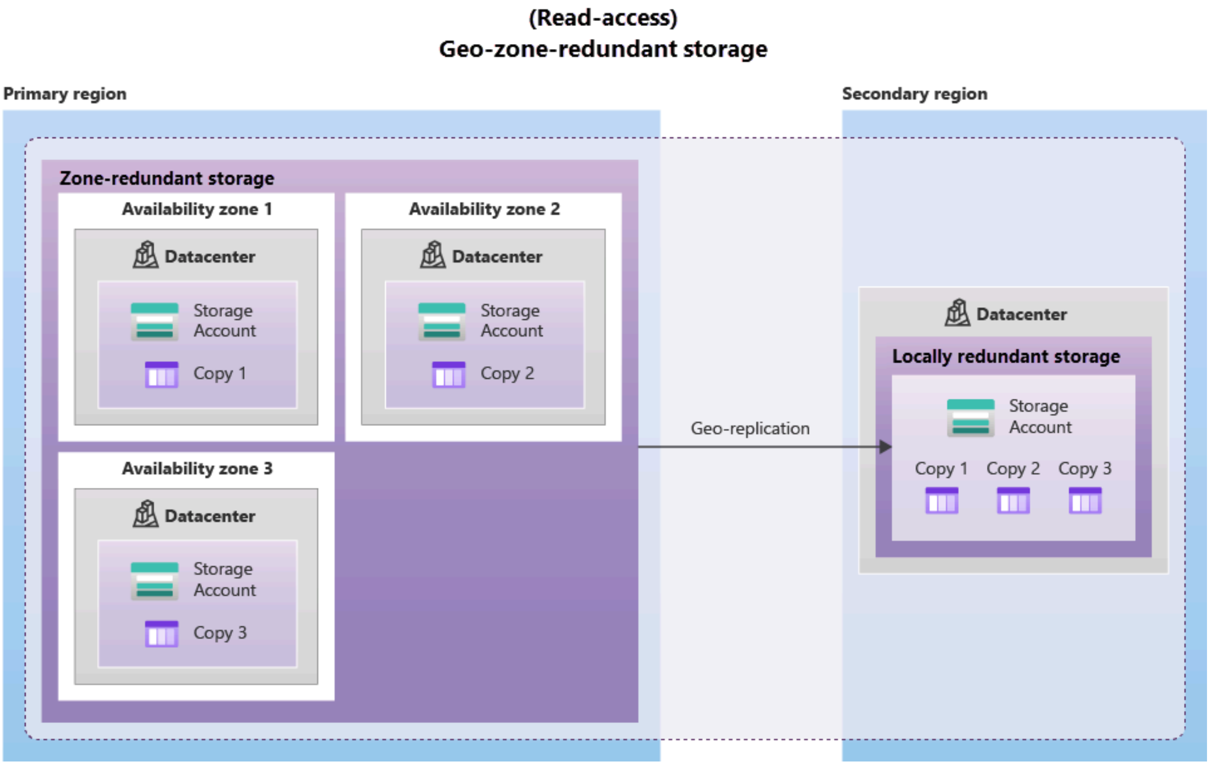
Geo-redundant storage

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.



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.

3.Describe Azure storage services

he 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.

4.Azure Functions overview

Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs. Instead of worrying about deploying and maintaining servers, the cloud infrastructure provides all the up-to-date resources needed to keep your applications running.

You focus on the code that matters most to you, in the most productive language for you, and Azure Functions handles the rest. In addition, azure functions can be written in multiple languages such as C#, Java, JavaScript, TypeScript, and Python.

Azure functions are scalable. When demand for execution increases, more resources are allocated automatically to the service, and when requests fall, all extra resources and application instances drop off automatically.

Top of Form

Let's say you have to send a birthday email to your customers. You're an ASP.NET web developer. Instead of building a website in ASP.NET and employing and hosting it on IIS just for one feature, you can write an azure function, put your email login in the function, and deploy it on the Azure cloud. The azure functions will directly connect to your data source, get your customers' emails, and send them an email on a scheduled date and time.

Azure functions are best suited for smaller apps with events that can work independently of other websites. For example, some azure functions are sending emails, starting back up, order processing, task scheduling such as database cleanup, sending notifications, messages, and IoT data processing.

Why use Azure Functions

Here are some of the reasons why you should use azure functions.

1. Azure functions are lightweight and serverless.
2. Azure functions are easier to write and deploy.
3. Azure functions are fast to execute because there is no large application, startup time, initialization, and other events fired before the code is executed.
4. Azure functions' execution is triggered when an event is fired.
5. Azure functions are compute-on-demand, and that is scalable. When the demand for the execution increases, more resources are allocated automatically to the service, and when requests fall, all extra resources and application instances drop off automatically.
6. Azure functions support programming languages, including ding C#, F#, Java, JavaScript, TypeScript, and Python. You choose your choice of language.
7. Azure functions do not need any infrastructure and have 0 maintenance.
8. The azure function can be built, tested, and deployed in the Azure portal using a browser.
9. Azure functions are easy to upgrade and don't affect other parts of the website.
10. Azure functions use industry standards and can communicate with other APIs, databases, and libraries.
11.

When should you use Azure Functions?

Azure Functions is a lightweight serverless service. It has its own specific uses, and you can't just replace an entire website simply by using it.

The most common use cases of Azure Functions include:

- Reminders and notifications
- Scheduled tasks and messages
- File processing
- Data or data streams processing
- Running background backup tasks
- Computing backend calculations
- Lightweight Web APIs, proofs of concept, MVPs

Importantly, not all applications can or should use Azure Functions as it's a service that uses triggers. Once an event has been triggered, the task is executed in the background.

It's worth keeping in mind that Azure Functions is not a replacement for Web APIs. Web applications tend to use Web APIs as the middleman to bunch together data and business logic tasks, while Azure Functions receives input, runs its logic and provides output. Nevertheless, it could be a great extension for Web APIs for specific use cases.

Importantly, Azure Functions is not designed to carry out multiple tasks. The service was created to perform one or very few tasks as fast as possible.

Azure Functions is not recommended for infrequent, time-sensitive or long and computationally intensive tasks. Since Azure Functions is a compute-on-demand service, attempting to replace any APIs with multiple Azure functions could result in severely increased costs in terms of development, maintenance and computations.

5.Introduction to Azure SQL

Azure SQL is a family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud.

Many organizations have an aging or under-engineered data-platform strategy. To modernize their IT resources, companies are moving systems to the cloud, building new applications quickly with the cloud, and offloading some on-premises costs.

You need a plan for how to move some workloads to the cloud, and you need to understand how to set up your organization for success. You also need to understand how the role of a database administrator (DBA) or data professional stays the same, and what changes you have to make.

This module starts with a brief history of why and how Microsoft built Azure SQL. You'll then learn about the various deployment options and service tiers, including what to use for your organization and when. These options include Azure SQL Database, Azure SQL Managed Instance, and SQL Server in an Azure virtual machine. Understanding what platform as a service (PaaS) encompasses and how it compares to a traditional SQL server environment can help you understand what you do and don't get when you move to the cloud.

Benefits of Azure SQL Database

Azure SQL Database offers many benefits over traditional on-premises databases, including cost savings, increased flexibility, and scalability. Perhaps the most significant advantage of Azure SQL Database is its pay-as-you-go pricing model, which can help organizations save money on their database costs. Additionally, Azure SQL Database can be scaled up or down as needed, making it a more flexible solution than an on-premises database. Finally, Azure SQL Database is highly available and provides built-in disaster recovery capabilities, ensuring that your data is always safe and accessible.

Azure SQL history

Software and services for relational databases have been a large part of the Microsoft product offering over the years. Before you learn about Azure SQL and where it's going, let's briefly consider where it started.

Launch of Windows Azure

At the Microsoft Professional Developers Conference in 2008, Microsoft's Chief Software Architect at the time, Ray Ozzie, announced the new cloud computing operating system: Windows Azure. One of the five key components of the Azure Services Platform launch was Microsoft SQL Services. From the beginning, SQL has been a large part of Azure. SQL Azure was created to provide a cloud-hosted version of SQL Server. Windows Azure was later renamed to Microsoft Azure, SQL Azure was renamed to Azure SQL, and both have since dramatically expanded services.

Evolution of database services on Azure

Azure SQL is a cloud database offering that Microsoft provides as part of the Azure cloud computing platform. Unlike other editions of SQL Server, you do not need to provision hardware for, install or patch Azure SQL; Microsoft maintains the platform for you. You also do not need to architect a database installation for scalability, high availability, or disaster recovery as these features are provided automatically by the service. Any application that uses Azure SQL must have Internet access in order to connect to the database.

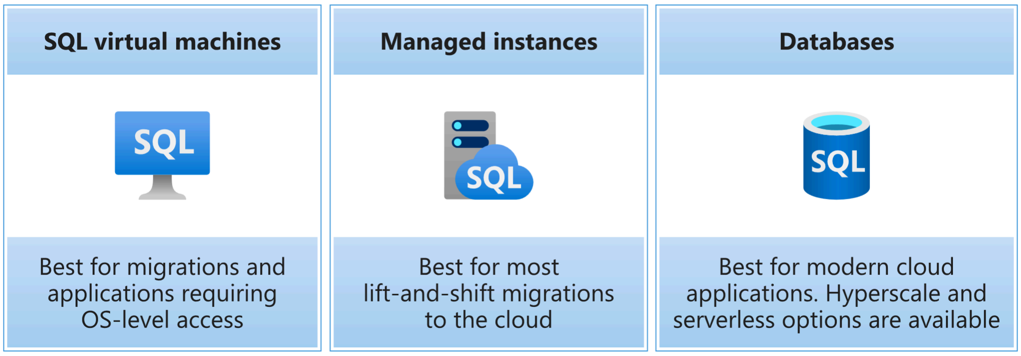
This explanation remains valid, but the capabilities around security, performance, availability, and scale have been enhanced greatly. Azure SQL has evolved over the years to include Azure virtual machines, managed instances, and several options for databases. There are now multiple deployment options that have the flexibility to scale to your needs. There have been more than seven million deployments of some form of Azure SQL. The architecture for Azure SQL has also evolved to meet the growing demands of applications. For example, the architecture introduced in 2014 set the stage for new possibilities like elastic database pools, vCore choices, business-critical deployments, hyperscale, and serverless architectures.

Since 2008, SQL Server and Azure SQL have both evolved to become more available, scalable, and performant to meet the demands of any application. The database services offered have expanded from SQL Server to include open-source databases like Azure Database for PostgreSQL and Azure Database for MariaDB.

Azure SQL deployment options

Within the umbrella of the Azure SQL platform, there are many deployment options and choices that you can make. These options give you the flexibility to get and pay for exactly what you need.

This unit covers some of the considerations you need to make when you choose various Azure SQL deployment options. You'll also learn about technical specifications for each of these deployment options. The deployment options discussed here include SQL Server on virtual machines, Azure SQL Managed Instance, Azure SQL Database, Azure SQL Managed Instance pools, and Azure SQL Database elastic database pools.



SQL virtual machines	Managed instances	Databases
Best for migrations and applications requiring OS-level access	Best for most lift-and-shift migrations to the cloud	Best for modern cloud applications. Hyperscale and serverless options are available

SQL Server on Azure Virtual Machines

SQL Server on a virtual machine (VM) is a version of SQL Server that runs in an Azure VM. It's just SQL Server, so all your SQL Server skills should directly transfer, though Azure can help automate backups and security patches. SQL Server on an Azure VM is referred to as *infrastructure as a service (IaaS)*. You're responsible for updating and patching the OS and SQL Server, apart from critical SQL Server security patches, but you have access to the full capabilities of SQL Server.

Here are some considerations for optimally deploying and managing SQL Server on VMs:

- Deploy specific SQL Server and operating-system versions from preinstalled Azure gallery images. If you self-install SQL Server on an Azure VM, you can take advantage of the SQL Server IaaS Agent Extension for licensing flexibility and to enable automatic backups and updates.

IaaS vs. PaaS

SQL Server on a VM is considered IaaS. The other deployment options in the Azure SQL platform, Azure SQL Managed Instance and Azure SQL Database, are platform as a service (PaaS) deployments. These PaaS Azure SQL deployment options contain a fully managed database engine that automates most of the database management functions, like upgrading, patching, backups, and monitoring. Here are some key features of SQL Managed Instance and SQL Database:

- Business continuity allows your business to continue operating in the face of disruption.

- High availability guarantees your databases are up and running 99.99% of the time. There's no need to worry about maintenance or downtimes.

- Automated backups are created and use Azure read-access geo-redundant storage (RA-GRS) to provide geo-redundancy.

- Long-term backup retention lets you store specific full databases for up to 10 years.

- Geo-replication creates readable replicas of your database in the same datacenter (region) or a different one.

- Scalability lets you easily add more resources (CPU, memory, storage) without long provisioning.

- Network security features protect your data over the network. These features include firewalls to restrict connectivity, Azure Private Link to ensure your data isn't exposed to the internet, and integration with virtual networks for connectivity to on-premises environments.

- Advanced security detects threats and vulnerabilities in your databases and allows you to secure your data.

- Automatic tuning analyzes your workload. It provides recommendations that can optimize performance of your applications by adding indexes, removing unused indexes, and automatically fixing query plan problems.

- Built-in monitoring capabilities provide insights into the performance of your databases and workload, and help you troubleshoot performance problems.

- Built-in intelligence automatically identifies potential problems in your workload, and provides recommendations that can help you to fix those problems.

Versionless database services

Another significant difference between IaaS and PaaS is versionless SQL. Unlike IaaS, which is tied to a specific SQL Server version, SQL Database and SQL Managed Instance are versionless. The main "branch" of the SQL Server engine codebase powers SQL Server 2019, SQL Database, and SQL Managed Instance.

Although SQL Server versions come out every few years, PaaS services allow Microsoft to continually update SQL databases and instances. Microsoft rolls out fixes and features as appropriate. As a consumer of the service, you don't have control over these updates, and the result of @@VERSION doesn't line up to a specific SQL Server version. But versionless SQL allows for worry-free patching for both the underlying OS and SQL Server and for Microsoft to give you the latest bits.

As new features are developed, some customers are granted access to specific features before they're publicly available. These new features then become available in public previews. Public previews allow everyone to access new features, but there's typically limited support and often discount pricing.

SQL Managed Instance

SQL Managed Instance is a PaaS deployment option of Azure SQL. It gives you an instance of SQL Server, but removes much of the overhead of managing a VM. Most of the features available in SQL Server are available in SQL Managed Instance. This option is ideal for customers who want to use instance-scoped features and want to move to Azure without rearchitecting their applications. Instance-scoped features are tied to an instance of SQL Server, as opposed to features that are tied to a database in an instance of SQL Server.

Instance-scoped features of SQL Managed Instance include SQL Server Agent, Service Broker, common language runtime (CLR), Database Mail, linked servers, distributed transactions (preview), and Machine Learning Services. SQL Managed Instance allows you to access instance-scoped features, but you don't have to worry about, nor do you have access to, the OS or the infrastructure underneath.

- Consider memory-optimized or storage-optimized VM sizes for maximum performance.

- Use the right storage configuration and take advantage of Azure Blob storage read caching.

- Integrate your VMs into on-premises networks by using Azure virtual networks.

- Take advantage of automated backups, backups to Azure Blob storage, and integration with Azure Backup.

- Always On Failover Cluster Instances is supported with Azure premium file share.

- Always On availability groups are supported, including Cloud Witness.

Companies around the world use SQL Server on VMs. One example is Allscripts. Allscripts is a leading manufacturer of healthcare software, serving physician practices, hospitals, health plans, and the pharmaceutical industry. To transform its applications frequently and to host them securely and reliably, Allscripts wanted to move to Azure quickly. In just three weeks, the company used Azure Site Recovery to migrate dozens of acquired applications running on approximately 1,000 VMs to Azure.

SQL Database

SQL Database is a PaaS deployment option of Azure SQL that abstracts both the OS and the SQL Server instance away from users. This deployment option allows you to just get a database and start developing applications. SQL Database is also the only deployment option that supports scenarios that require unlimited database storage (hyperscale) and autoscaling for unpredictable workloads (serverless). SQL Database has the industry's highest availability SLA. It provides other intelligent capabilities related to monitoring and performance, partly because Microsoft manages instances

AccuWeather provides a great example of using SQL Database. AccuWeather has been analyzing and predicting the weather for more than 55 years. The company wanted to access Azure for its big data, machine learning, and AI capabilities. AccuWeather wants to focus on building new models and applications, not on managing databases. The company chose SQL Database to use with other services, like Azure Data Factory and Azure Machine Learning, to quickly and easily deploy new internal applications to make sales and customer predictions.

Elastic database pool

You've now learned about the three main deployment options within Azure SQL: virtual machines, managed instances, and SQL Database. For SQL Database and SQL Managed Instance, there are other options if you have multiple instances or databases. These options are referred to as *elastic database pools*. Elastic database pools allow you to share resources among multiple instances and databases and optimize your costs.

SQL Database elastic pools allow you to host many databases within a single set of provisioned SQL Database resources. This option is ideal for software as a service (SaaS) application or provider because you can manage and monitor performance in a simplified way for many databases.

SQL Managed Instance pools allow you to host multiple managed instances and share resources. You can pre-provision compute resources. Doing so can reduce overall deployment time to make migrations easier. You can also host smaller managed instances in an instance pool than in a single managed instance. This offer is currently in public preview.

Paychex is a good example of a company that uses SQL Database elastic database pools. Paychex is a Human Capital Management firm that serves more than 650,000 businesses across the US and Europe. Paychex needed a way to separately manage the time and pay management for each of its customers and cut costs. The company chose SQL Database elastic database pools, which allowed it to simplify management and enable resource sharing between separate databases to lower costs.