

Domain 2: Introduction to Cloud

Infrastructure: Describe Azure architecture and services

Describe the core architectural components of Azure

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/1-introduction>

Introduction

Completed

In this module, you'll be introduced to the core architectural components of Azure. You'll learn about Azure's physical layout: datacenters, availability zones, and regions; and you'll learn about Azure's management structure: resources and resource groups, subscriptions, and management groups.

Learning objectives

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

- Describe Azure regions, region pairs, and sovereign regions.
- Describe Availability Zones.
- Describe Azure datacenters.
- Describe Azure resources and Resource Groups.
- Describe subscriptions.
- Describe management groups.
- Describe the hierarchy of resource groups, subscriptions, and management groups.

2. What is Microsoft Azure

What is Microsoft Azure

Completed



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

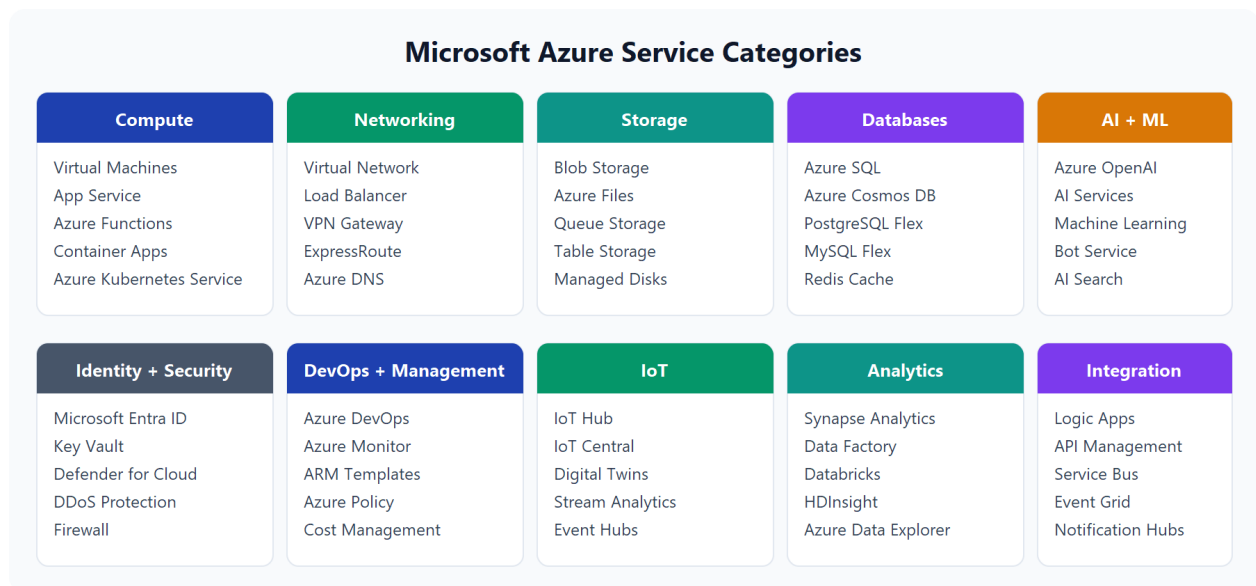
What does Azure offer?

Limitless innovation. Build intelligent apps and solutions with advanced technology, tools, and services to take your operations to the next level. Seamlessly unify your technology to simplify platform management and deliver innovations efficiently and securely on a trusted cloud.

- **Bring ideas to life:** Build on a trusted platform to advance your team's capabilities with industry-leading AI and cloud services.
- **Seamlessly unify:** Efficiently manage all your infrastructure, data, analytics, and AI solutions across an integrated platform.
- **Innovate on trust:** Rely on trusted technology from a partner who's dedicated to security and responsibility.

What can I do with Azure?

Azure provides hundreds of services that enable you to do everything from running your existing applications on virtual machines to exploring new software paradigms, such as intelligent bots and generative AI.



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

As your skills grow, you can modernize one workload at a time, such as moving from manually managed servers to managed databases, autoscaling web apps, or event-driven services.

Practical example

Suppose your organization runs an internal app with seasonal demand spikes. In Azure, you can host the app on virtual machines or managed app services, store data in managed databases, and monitor health from a centralized dashboard. As demand increases, you can scale resources up or out and then scale back when demand drops so you're not paying for unused capacity year-round.

For example, Azure provides Azure AI services and Azure OpenAI Service so you can add language, vision, speech, and generative AI capabilities to your applications. It also provides Azure Machine Learning, Internet of Things (IoT) services, and storage solutions that dynamically grow to accommodate massive amounts of data. Azure services enable solutions that aren't feasible without the power of the cloud.

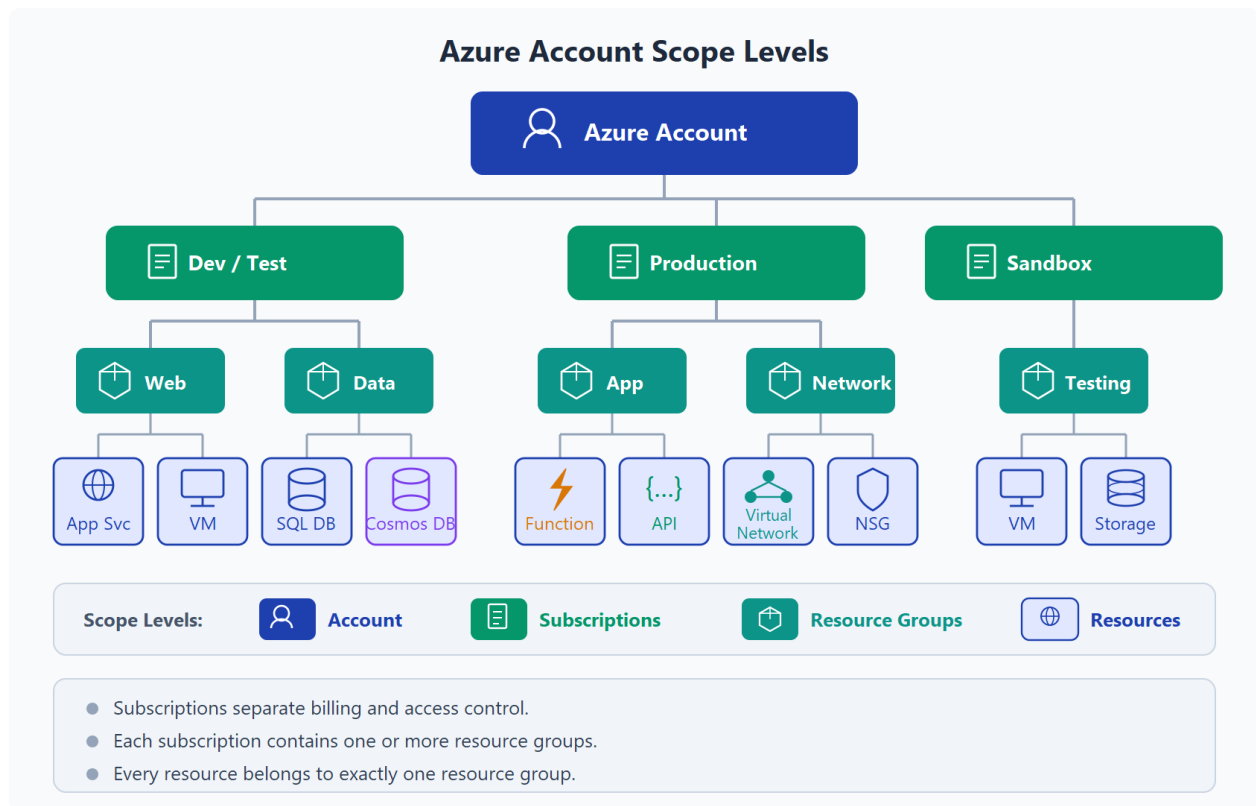
3. Get started with Azure accounts

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/3-get-started-azure-accounts>

Get started with Azure accounts

Completed

To create and use Azure services, you need an Azure subscription. When you're working with your own applications and workloads, you create an Azure account, and a subscription is created for you. After you've created an Azure account, you're free to create additional subscriptions. For example, your team might use a single Azure account and separate subscriptions for development, testing, and production workloads. After you've created an Azure subscription, you can start creating Azure resources within each subscription.



If you're new to Azure, you can sign up for a free account on the Azure website to start exploring at no cost to you. When you're ready, you can choose to upgrade your free account. You can also create a new subscription that enables you to start paying for Azure services you need beyond the limits of a free account.

Create an Azure account

You can purchase Azure access directly from Microsoft by signing up on the Azure website or through a Microsoft representative. You can also purchase Azure access through a Microsoft partner. Cloud Solution Provider partners offer a range of complete managed-cloud solutions for Azure.



What is the Azure free account?

The Azure free account includes:

- Free access to popular Azure products for 12 months.
- A credit to use for the first 30 days.
- Access to more than 65 services that are always free.

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

What is the Azure free student account?

The Azure free student account offer includes:

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

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

If you're practicing by creating resources in Azure, monitor usage and remove resources you no longer need to avoid unexpected costs.

4. Describe Azure physical infrastructure

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/5-describe-azure-physical-infrastructure>

Describe Azure physical infrastructure

Completed

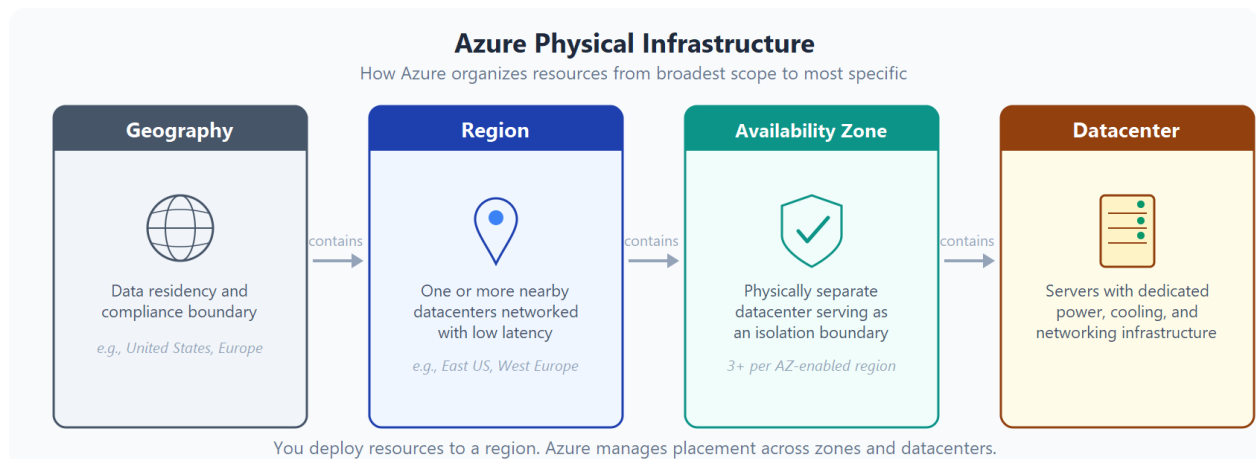
Azure's core architectural components can be broken down into two main groupings: the physical infrastructure and the management infrastructure. This unit covers the physical side — how Azure organizes its datacenters, regions, and availability zones to deliver reliable services worldwide.

Physical infrastructure

The physical infrastructure for Azure starts with datacenters. These datacenters are facilities with servers arranged in racks, with dedicated power, cooling, and networking infrastructure — similar to an on-premises datacenter, but at a much larger scale.

As a global cloud provider, Azure has datacenters around the world. However, you don't interact with individual datacenters directly. Instead, datacenters are grouped into Azure Regions and Azure Availability Zones that provide resiliency and reliability for your workloads.

The [Global infrastructure](#) site gives you a chance to interactively explore the underlying Azure infrastructure.



Regions

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

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

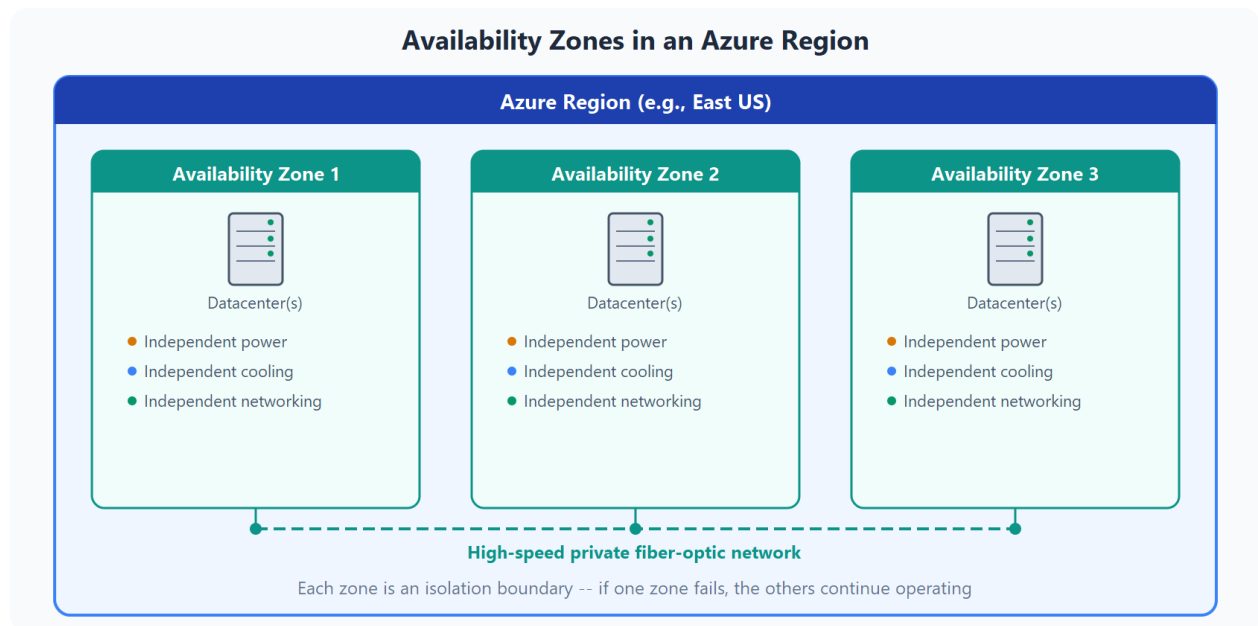
Note

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

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.



Important

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

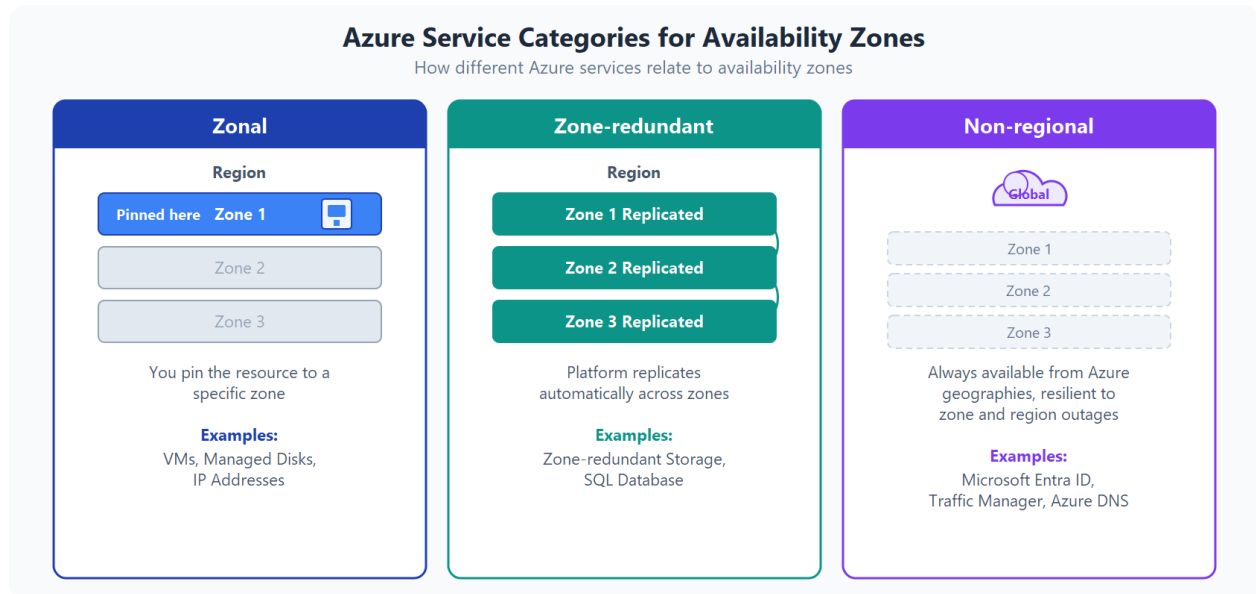
Use availability zones for your workloads

When you run your own on-premises infrastructure, setting up redundancy means buying and maintaining duplicate hardware. With Azure, you can protect your workloads by spreading them across availability zones within a region.

You place your VMs, storage, databases, and other resources in one availability zone and replicate them to other zones within the same region. Keep in mind that there could be a cost to duplicating your services and transferring data between zones.

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 resiliency, Azure has Region Pairs.

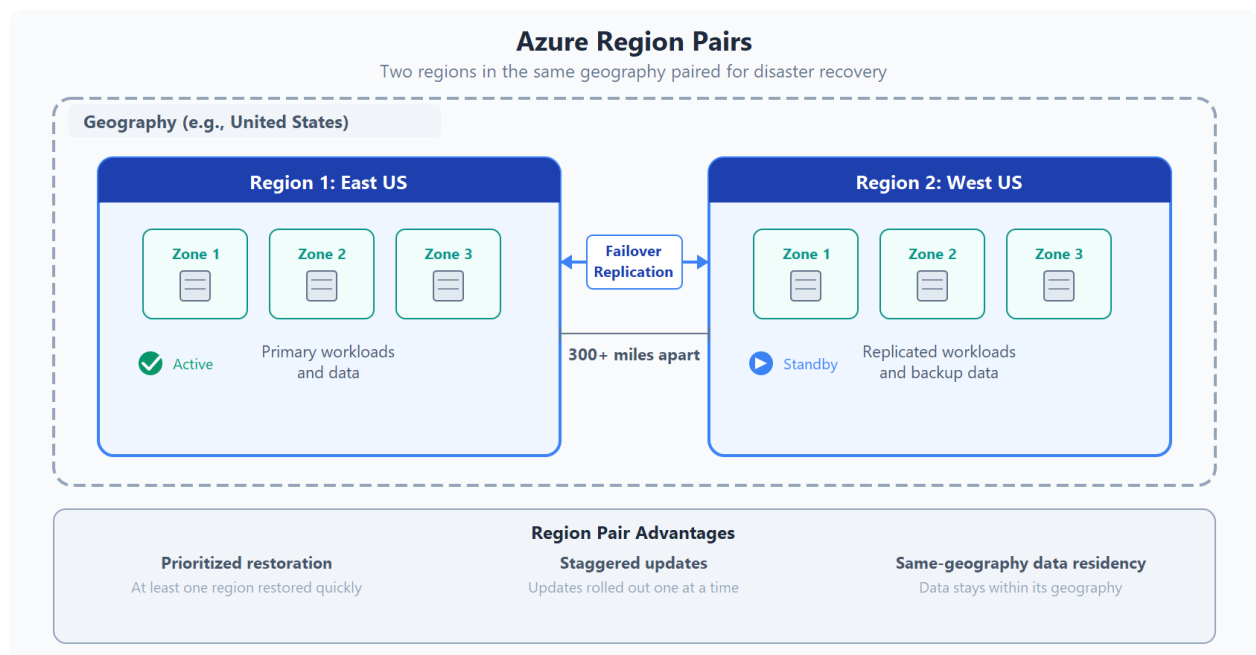
Region pairs

Most Azure regions are paired with another region within the same geography (such as US, Europe, or Asia) at least 300 miles away. This approach allows for the replication of resources across a geography that helps reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages, or physical network outages that affect an entire region. For example, if a region in a pair was affected by a natural disaster, services would automatically fail over to the other region in its region pair.

Important

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

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



Additional advantages of region pairs:

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

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 Brazil South, are paired in only one direction. In a one-direction pairing, the Primary region does not provide backup for its secondary region. 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. Additionally, some regions (such as Italy North, Poland Central, and Israel Central) don't have a traditional region pair and instead rely on availability zones and geo-redundant storage for resiliency.

Sovereign Regions

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

Azure sovereign regions include:

- US DoD Central, US Gov Virginia, US Gov Arizona, and more: These regions are physical and logical network-isolated instances of Azure for U.S. government agencies and partners. These datacenters are operated by screened U.S. personnel and include additional compliance certifications.
- China East, China North, and more: These regions are available through a unique partnership between Microsoft and 21Vianet, whereby Microsoft doesn't directly maintain the datacenters.

5. Describe Azure management infrastructure

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/6-describe-azure-management-infrastructure>

Describe Azure management infrastructure

Completed

The management infrastructure includes Azure resources and resource groups, subscriptions, and accounts. Understanding this hierarchy helps you organize resources, control who can access what, and manage costs as your Azure usage grows.

Azure resources and resource groups

A resource is the basic building block of Azure. Anything you create, provision, or deploy is a resource. VMs, virtual networks, databases, and Azure AI services are all examples of resources.



Resource groups are groupings of resources. Every resource must belong to exactly one resource group. You can move some resources between groups, but a resource is only associated with one

group at a time. Resource groups can't be nested, and they can't be renamed after creation, so choose a clear naming convention from the start.

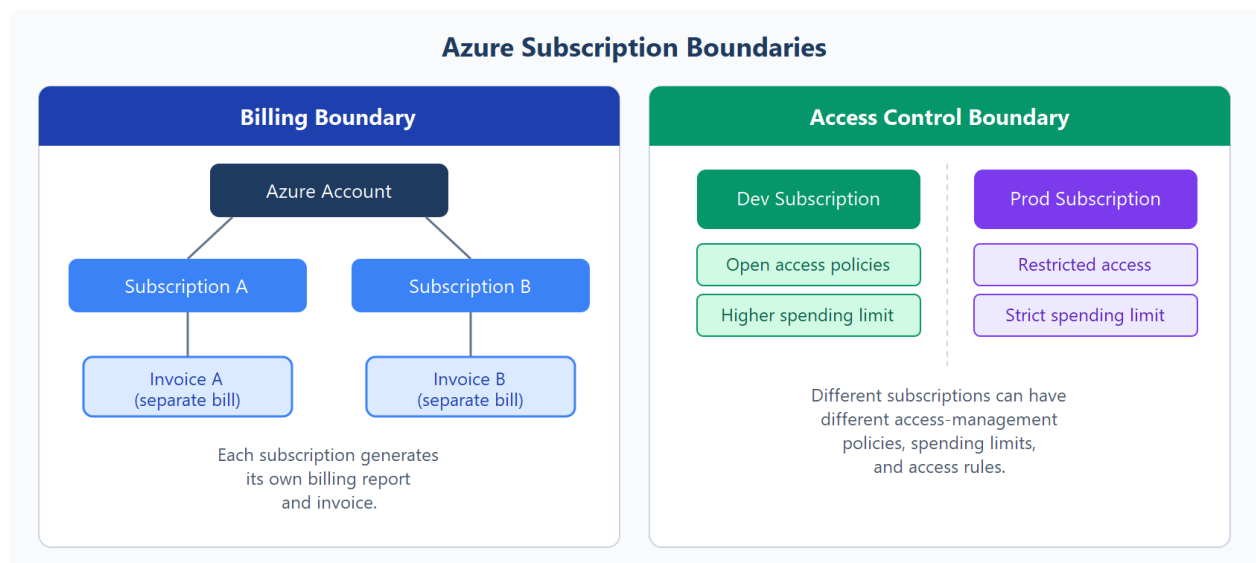
Actions you apply to a resource group affect all resources inside it. Deleting a resource group deletes everything in it. Granting or denying access applies to all its resources.

For example, if you're setting up a temporary dev environment, grouping all the resources together lets you delete the entire group when you're done. If you're running multiple projects, create a separate resource group for each so each team only sees and manages its own resources.

There are no hard rules for structuring resource groups — choose the approach that works best for your situation.

Azure subscriptions

In Azure, subscriptions are a unit of management, billing, and scale. Subscriptions let you organize resource groups and control billing separately from access.



Using Azure requires an Azure subscription. A subscription provides access to Azure products and services and serves as a billing unit. 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.

An account can have multiple subscriptions, but only one is required. In a multi-subscription account, you can configure different billing models and access policies. There are two types of subscription boundaries:

- **Billing boundary:** Determines how an Azure account is billed. You can create multiple subscriptions for different billing requirements. Azure generates separate billing reports and invoices for each subscription.
- **Access control boundary:** Azure applies access-management policies at the subscription level. For example, you might create one subscription for your development work and another for production, each with different spending limits and access rules.

Create additional Azure subscriptions

You might create additional subscriptions to separate:

- **Environments:** Subscriptions for lifecycle stages such as sandbox, development, test, and production. Access control occurs at the subscription level, making this a natural boundary.
- **Team and workload boundaries:** Give each project its own subscription so costs are easy to track, or separate sandbox environments from production.
- **Billing:** Create subscriptions to track costs separately — for instance, one for production workloads and another for development and testing.

Azure management groups

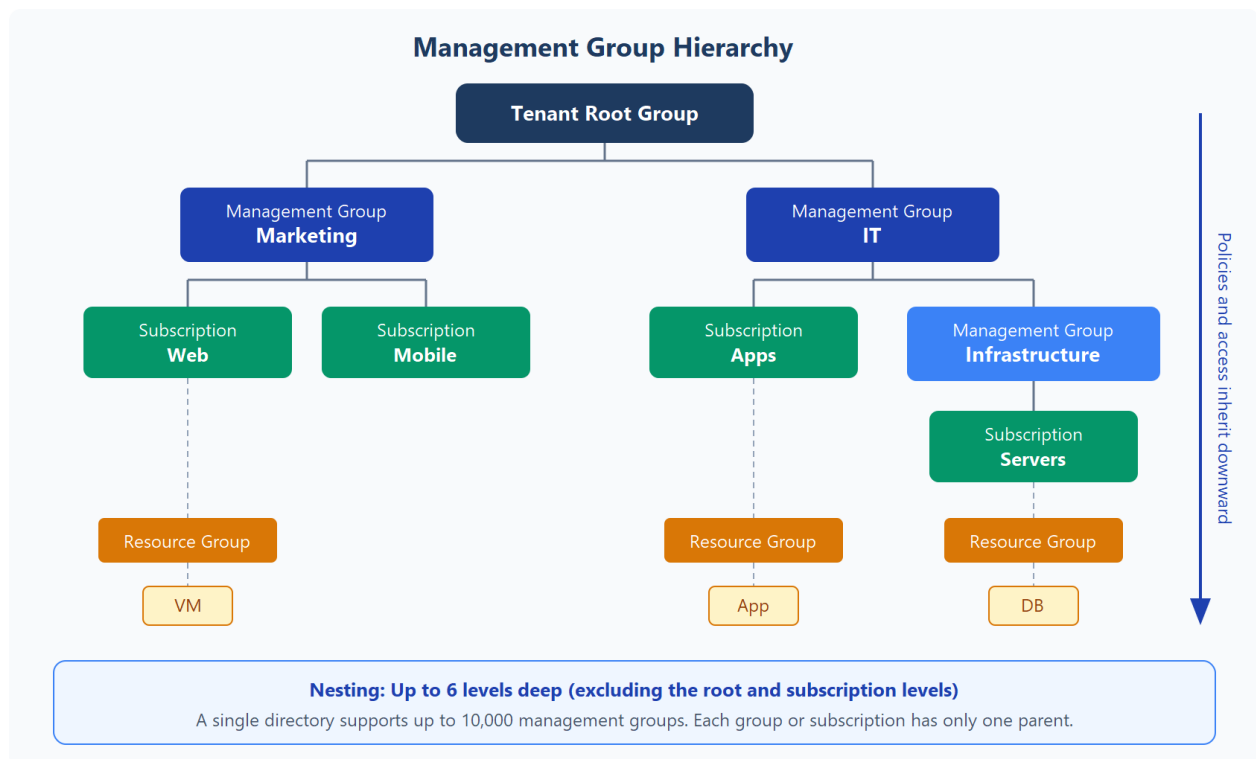
Resources go into resource groups, and resource groups go into subscriptions. For a small environment, that's enough. But when you have many subscriptions across multiple teams or geographies, you need a way to manage access and policies at a higher level.

Azure management groups sit above subscriptions. You organize subscriptions into management groups and apply governance conditions — like access policies or compliance rules — to the group. All subscriptions in a management group automatically inherit those conditions, just as resources inherit settings from their resource group. Management groups can be nested up to six levels deep (not counting the root level or the subscription level), letting you build a hierarchy that mirrors your organization.

Every Microsoft Entra tenant has a single top-level Tenant Root Group. All other management groups and subscriptions fold up to this root group, which lets you apply governance policies globally.

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.



Examples of how you could use management groups:

- **Apply a policy across subscriptions.** You could limit VM locations to the US West Region in a group called Production. This policy inherits to all subscriptions under that management group and applies to all VMs in those subscriptions. The resource or subscription owner can't override it, which strengthens governance.
- **Grant access to multiple subscriptions at once.** By placing subscriptions under a management group, you can create one Azure RBAC assignment on the group. All sub-management groups, subscriptions, resource groups, and resources underneath inherit those permissions — no need to script Azure RBAC across individual subscriptions.

Important facts about management groups:

- A single directory supports up to 10,000 management groups.
- Each management group and subscription can have only one parent.

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/8-knowledge-check>

Module assessment

Completed

7. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/9-summary>

Summary

Completed

In this module, you learned about the physical and management structure of Microsoft Azure. You were introduced to the relationship between datacenters, availability zones, and regions. You explored how the infrastructure supports the benefits of the cloud, such as high availability and reliability. You also learned about the management infrastructure of Azure. You explored how resources and resource groups are related, and how subscriptions and management groups can help manage resources.

Learning objectives

You should now be able to:

- Describe Azure regions, region pairs, and sovereign regions.
- Describe Availability Zones.
- Describe Azure datacenters.
- Describe Azure resources and Resource Groups.
- Describe subscriptions.
- Describe management groups.
- Describe the hierarchy of resource groups, subscriptions, and management groups.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Build a concept map that connects regions, Availability Zones, datacenters, resources, resource groups, subscriptions, and management groups."
- "Design a subscription and management-group structure for an organization with multiple departments and compliance boundaries."

- "Explain region pairs and sovereign regions, then recommend a resiliency approach for a workload with strict data residency requirements."

Describe Azure compute services

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/1-introduction>

Introduction

Completed

This module introduces you to Azure compute services and related innovation options. You learn about core compute choices such as virtual machines, containers, and Azure Functions. You also review application hosting options with Azure App Service and fundamentals of Azure AI, machine learning, and IoT/Edge services.

Learning objectives

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

- Compare compute types, including container instances, virtual machines, and functions.
- Describe virtual machine options, including virtual machines (VMs), virtual machine scale sets, and virtual machine availability sets.
- Describe resources required for virtual machines.
- Describe application hosting options, including Azure Web Apps, containers, and virtual machines.
- Describe AI, machine learning, and IoT/Edge service categories in Azure.

2. Describe Azure virtual machines

<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/2-virtual-machines>

Describe Azure virtual machines

Completed

With Azure Virtual Machines (VMs), you can run virtualized servers in Azure as infrastructure as a service (IaaS). Like a physical server, you control the operating system and installed software. VMs are a good fit when you need:

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

Azure VMs remove the need to buy and maintain physical server hardware. As an IaaS service, you still manage patching, updates, and configuration inside the VM.

You can deploy VMs quickly from prebuilt images. An image is a template that already includes an operating system and tools such as web hosting components.

Examples of when to use VMs

Common VM use cases include:

- **Testing and development.** Create different OS and app configurations quickly, then remove the VM when testing is complete.
- **Cloud application hosting.** Run applications in Azure and scale capacity up or down as demand changes.
- **Datacenter extension.** Extend an on-premises network into Azure and host workloads in a connected virtual network.
- **Disaster recovery.** Keep failover capacity in Azure and run critical workloads there if your primary site is unavailable.
- **Lift and shift migration.** Move existing server workloads with minimal application redesign.

VM resources and sizing

When you provision a VM, you choose resources such as:

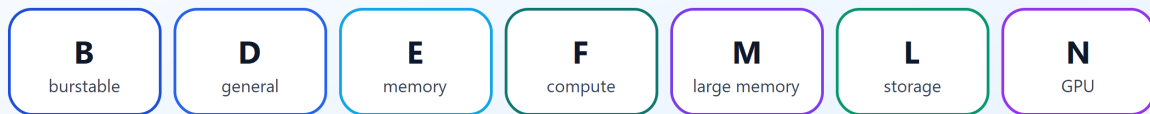
- Size (purpose, number of processor cores, and amount of RAM)
- Storage disks (hard disk drives, solid state drives, etc.)
- Networking (virtual network, public IP address, and port configuration)

Understand VM size families and names

Azure VM sizes are grouped into families so you can quickly choose a size based on your workload needs.

VM size families

Choose the family that best matches workload behavior.



The following table highlights common use cases for the Azure VM families.

Family	Typical focus	Example use
B-series	Burstable, cost-efficient	Dev/test workloads with occasional CPU spikes
D-series	General purpose	Web servers, small-to-medium app servers
E-series	Memory optimized	In-memory databases, analytics workloads
F-series	Compute optimized	CPU-intensive application tiers
M-series	Large memory footprint	Large enterprise databases
L-series	Storage optimized	High-throughput storage and data processing
N-series	GPU enabled	AI training/inference and graphics workloads

Each VM also has options that you can customize based on your needs. You can adjust the number of virtual CPUs (vCPUs), the amount of RAM, and the storage disk configuration.

Virtual machine options

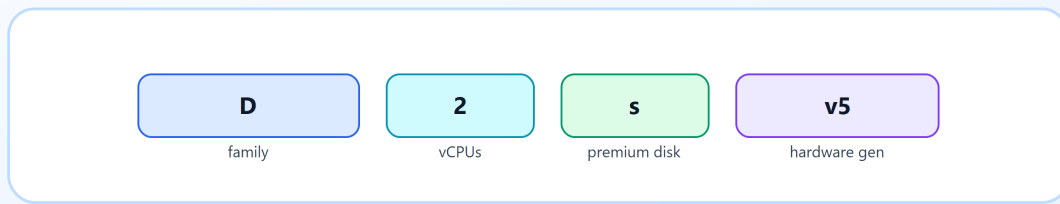


Some of the adjustments you can make are:

- **vCPU count:** affects compute capacity for concurrent and CPU-bound workloads.
- **RAM:** affects how much working data the VM can keep in memory.
- **Disk configuration:** affects storage capacity, IOPS, and throughput.
- **Network throughput:** affects data transfer performance in and out of the VM.
- **Premium SSD support:** indicates whether the size supports premium managed disks.
- **Hardware generation:** indicates platform generation and can affect baseline performance.

VM names also carry useful sizing information. The base name indicates the family (or purpose), the number of vCPUs, and the hardware generation. It may also indicate features such as premium storage support.

How to read Standard_D2s_v5



- **D** : the VM family (general purpose in this case)
- **2** : the number of vCPUs in this size
- **s** : supports Premium SSD storage
- **v5** : hardware generation for that family

At a fundamentals level, start by selecting a family that matches workload behavior, then choose the size that meets performance needs and scale as demand grows.

Scale and resiliency options for VMs

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 manage these groupings with features such as scale sets and availability sets.

Virtual machine scale sets

Virtual machine scale sets let you create and manage groups of identical, load-balanced VMs. Without scale sets, you must manually keep VM configuration consistent, monitor utilization, and adjust instance counts. Scale sets centralize configuration and can automatically scale out or in based on demand or schedules. They also integrate with load balancing so traffic is distributed efficiently.

Virtual machine availability sets

Virtual machine availability sets improve VM resiliency inside a region. They reduce the chance that all VMs are affected by one maintenance event or hardware failure.

Availability sets group VMs by:

- **Update domain:** VMs that can be rebooted together during planned maintenance.
- **Fault domain:** VMs that share a potential power or network failure point.

Use availability sets for VM-level redundancy. In regions that support Availability Zones, zone-based designs are often preferred because they provide broader failure isolation. Availability sets themselves don't add cost; you pay for the VM instances.

3. Describe Azure virtual desktop

Describe Azure virtual desktop

Completed

Azure Virtual Desktop is a desktop and application virtualization service in Azure. It lets users securely access Windows desktops and apps from many device types and locations.

At a fundamentals level, Azure Virtual Desktop is a managed option for remote desktop access where desktops and apps stay in the cloud instead of on local devices.

The following video gives you an overview of Azure Virtual Desktop:



When to use Azure Virtual Desktop

Use Azure Virtual Desktop when a team needs centralized desktop and app access across distributed users, contractors, or hybrid workers. For example, a support team can use standardized cloud-hosted desktops so each shift has the same tools, access policies, and security controls.

If you need a full desktop experience for many users with centralized management, Azure Virtual Desktop is often easier to operate than building separate VM-based desktop environments for each user group.



Key points

- It centralizes desktop and app delivery in Azure and integrates with Microsoft Entra ID for identity and access controls.
- It helps reduce data exposure on local devices because apps and data can remain in Azure-hosted sessions.
- It supports single-session and multi-session Windows experiences, depending on user and workload needs.

4. Describe Azure containers

<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/5-containers>

Describe Azure containers

Completed

Virtual machines reduce costs compared to physical hardware, but they're still limited to a single operating system per VM. If you want to run multiple instances of an application on a single host machine, containers are an excellent choice.

What are containers?

Containers are a virtualization environment. Much like running multiple virtual machines on a single physical host, you can run multiple containers on a single physical or virtual host. Unlike virtual machines, you don't manage the operating system for a container. Each virtual machine runs its own operating system that you can connect to and manage. Containers are lightweight and designed to be created, scaled out, and stopped dynamically. You can create and deploy virtual machines as application demand increases, but containers are a lighter-weight, more agile method. Containers help you respond to changes on demand and restart quickly after a crash or hardware interruption. One of the most popular container engines is Docker, and Azure supports Docker.

Compare virtual machines to containers

The following video highlights several of the important differences between virtual machines and containers:

Azure Container Instances

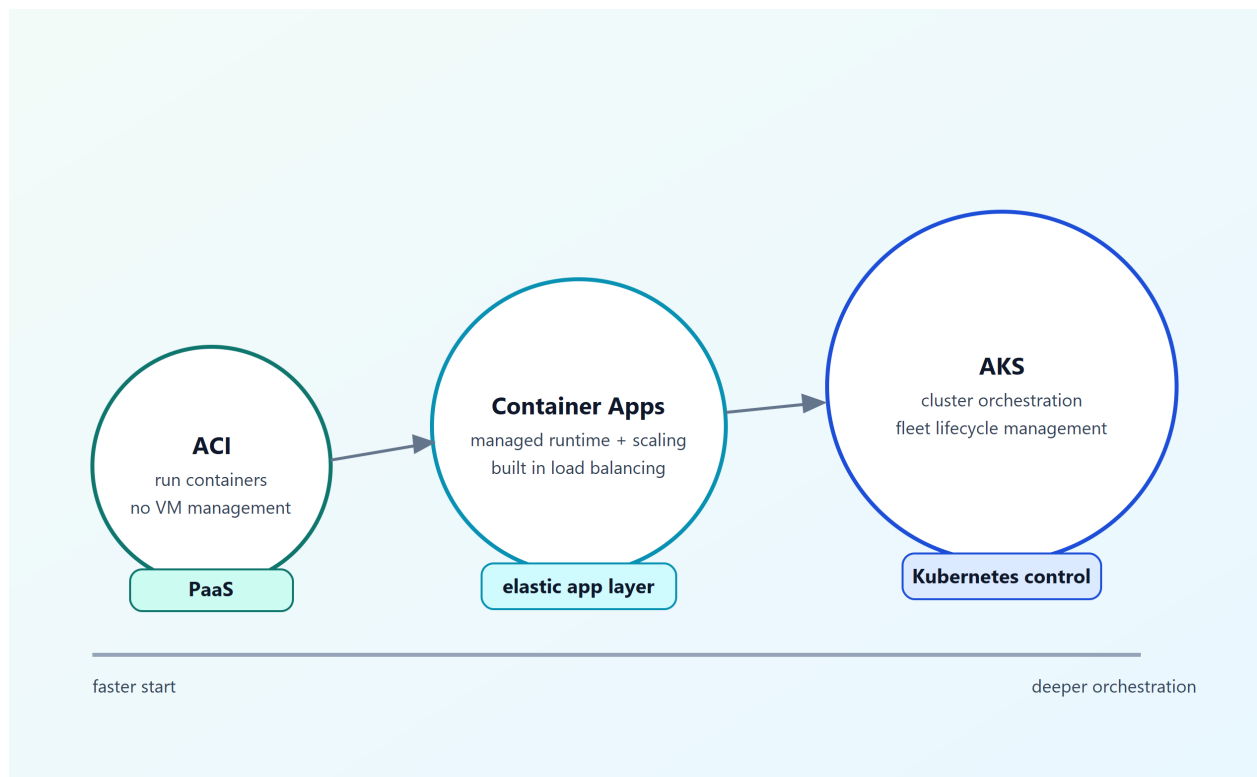
Azure Container Instances offer the fastest and simplest way to run a container in Azure, without managing any virtual machines or adopting extra services. Azure Container Instances are a platform as a service (PaaS) offering. You upload your containers and the service runs them for you.

Azure Container Apps

Azure Container Apps are similar in many ways to a container instance. They let you get up and running right away, they remove the container management overhead, and they're a PaaS offering. Container Apps also include built-in load balancing and scaling, so your design can adapt to changing demand.

Azure Kubernetes Service

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.



Use containers in your solutions

Containers are often used to create solutions that use a microservice architecture. In this architecture, you break solutions into smaller, independent pieces. For example, you might split a website into a container hosting your front end, another hosting your back end, and a third for storage. This split lets you maintain, scale, or update each part of your app independently.

Imagine your website back end reaches capacity, but the front end and storage aren't stressed. With containers, you can scale the back end separately to improve performance. You can also change the storage service or modify the front end without affecting the other components.

5. Describe Azure functions

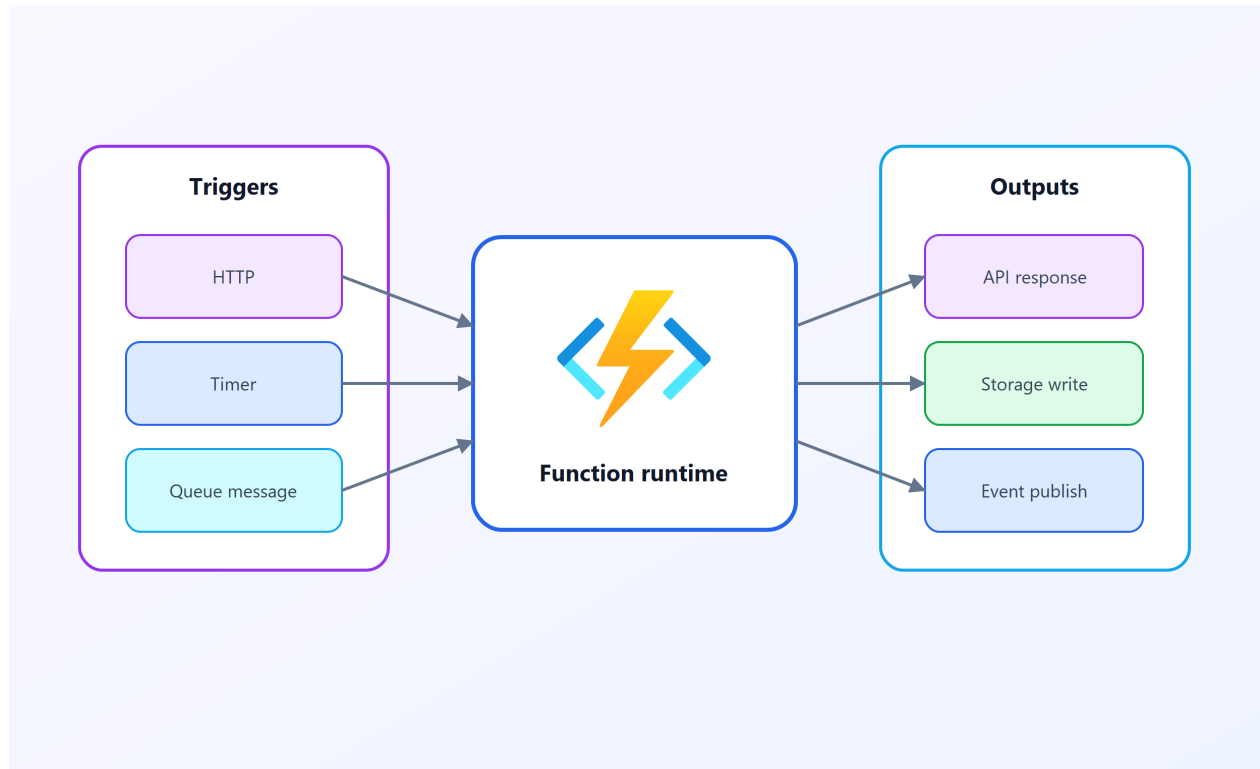
<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/6-functions>

Describe Azure functions

Completed

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.



Serverless computing in Azure



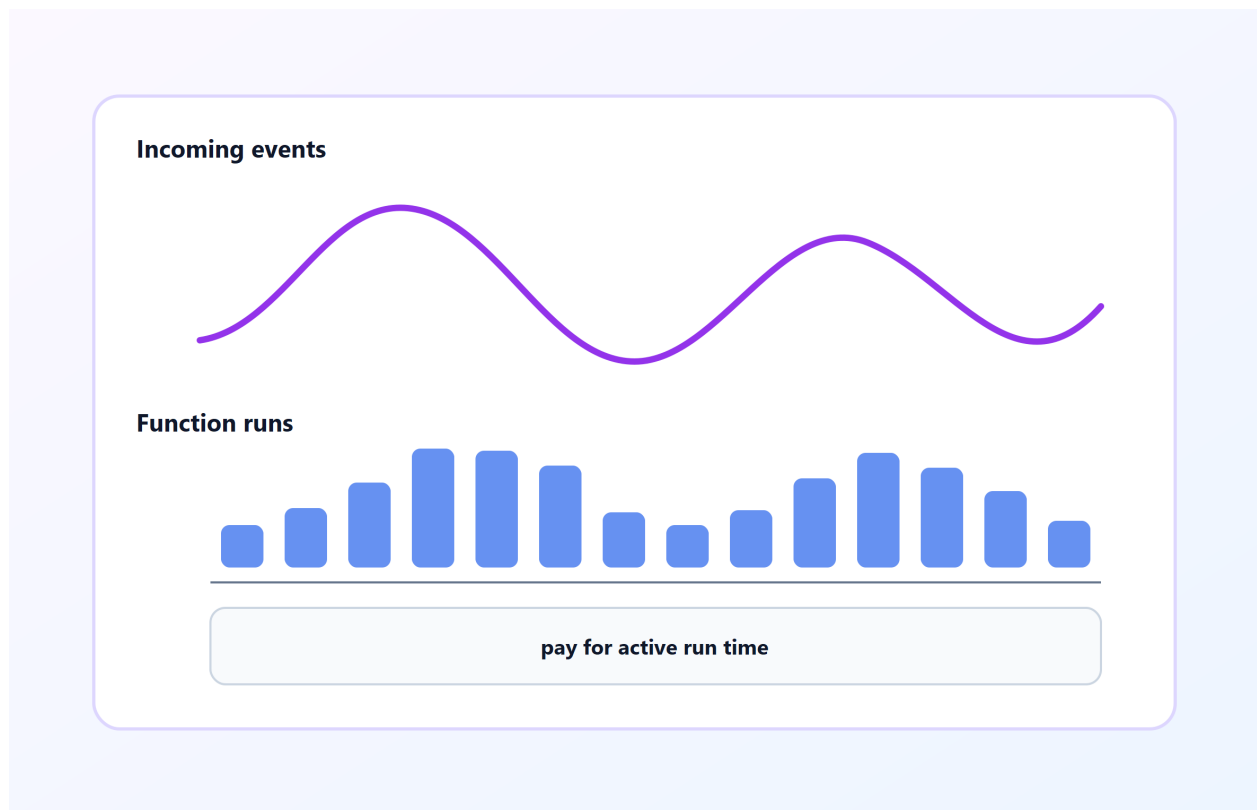
Benefits of Azure Functions

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 finishes. In this model, Azure charges you only 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 restart every time they respond to an event. When they're stateful (called Durable Functions), the runtime passes a context 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 your app change, you can deploy the project in an environment that isn't serverless. This flexibility lets you manage scaling, run on virtual networks, and even completely isolate the functions.

6. Describe application hosting options

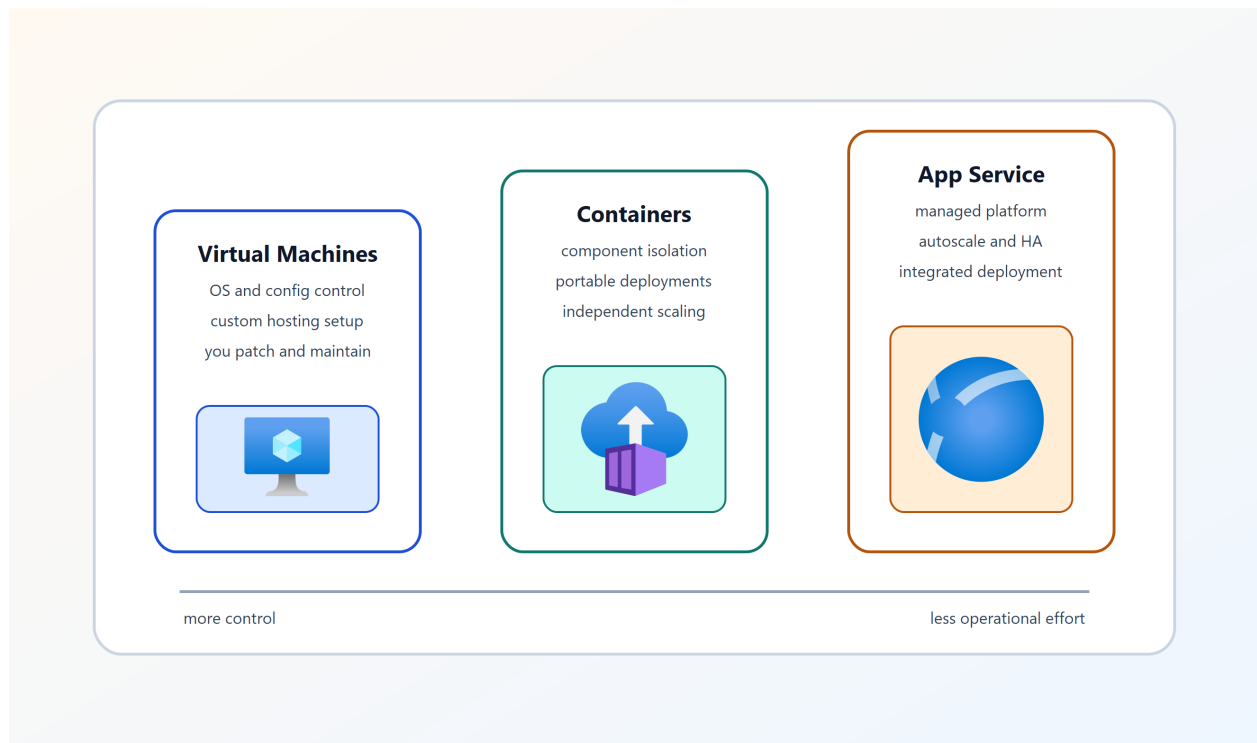
<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/7-describe-application-hosting-options>

Describe application hosting options

Completed

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

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



Azure App Service

App Service lets you 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 and supports automated deployments from GitHub, Azure DevOps, or any Git repo for continuous deployment.

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. It lets you focus on building and maintaining your app while Azure keeps the environment up and running. App Service supports multiple languages and frameworks, including .NET, .NET Core, Java, PHP, Python, and Node.js.

Types of app services

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

- Web apps
- API apps

- WebJobs
- Mobile apps

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

- Deployment and management are integrated into the platform.
- Endpoints can be secured.
- Sites can be scaled quickly to handle high traffic loads.
- The built-in load balancing and traffic manager provide high availability.

All of these app styles are hosted in the same infrastructure and share these benefits. This flexibility makes App Service the ideal choice to host web-oriented applications.

Web apps

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

API apps

Much like hosting a website, you can build REST-based web APIs by using your choice of language and framework. App Service provides full Swagger support and the ability to package and publish your API in Azure Marketplace. The published APIs 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, X, 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.

7. Describe AI, machine learning, and IoT/Edge services in Azure

<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/12a-describe-ai-ml-iot-edge-services>

Describe AI, machine learning, and IoT/Edge services in Azure

Completed

Azure includes service categories that help you build intelligent and connected solutions without starting from scratch.

Azure AI services

Azure AI services provides prebuilt capabilities for common AI scenarios, such as language, speech, vision, and document processing. These services are useful when you want to add intelligent features through APIs instead of training your own model first.

Azure OpenAI Service is another AI option in Azure that supports generative AI scenarios, such as chat and content generation, with built-in security and governance controls.

Agentic AI patterns

Agentic applications combine an AI model with instructions, context, and tool use to complete multistep goals. In Azure, you usually build these patterns by combining Azure AI services and Azure OpenAI Service with your own application logic.

At a fundamentals level, treat agentic AI as an application pattern built from Azure AI capabilities, not as a separate compute service category.

Azure Machine Learning

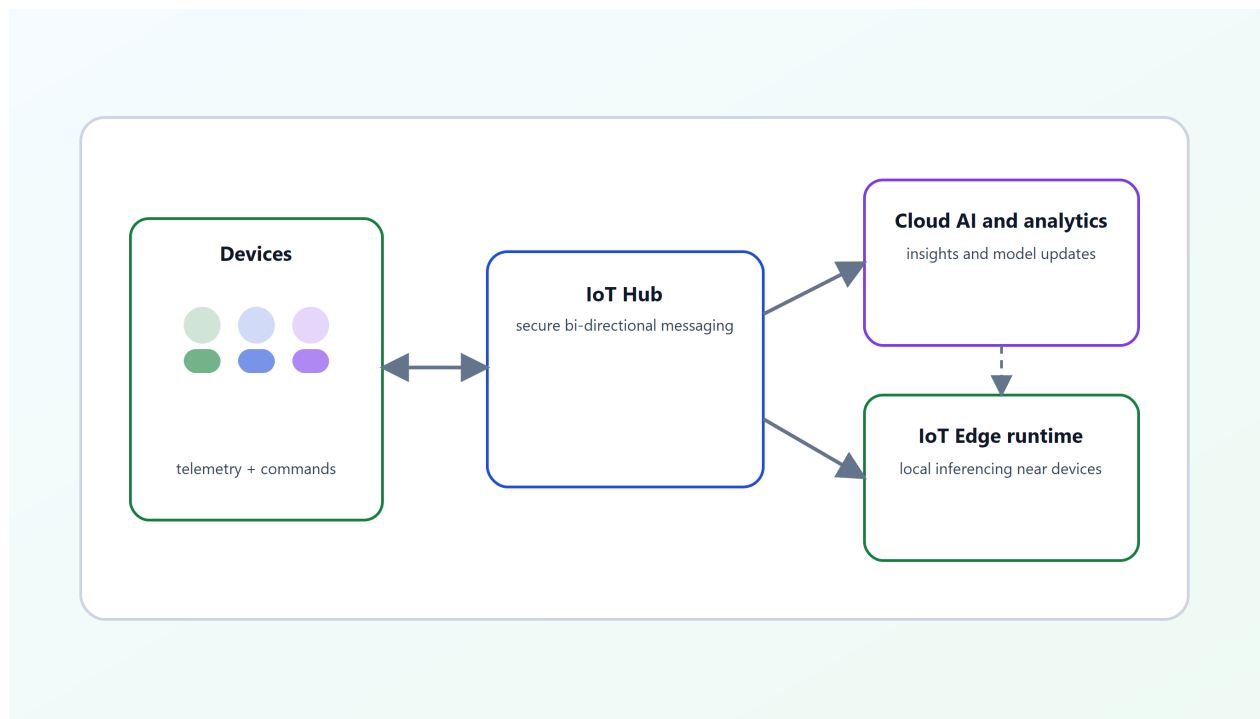
Use Azure Machine Learning when you need to build, train, and manage custom machine learning models. This option is a better fit when your scenario requires model development, experimentation, and lifecycle management.

IoT and Edge services

Azure IoT services help you connect, monitor, and manage devices.

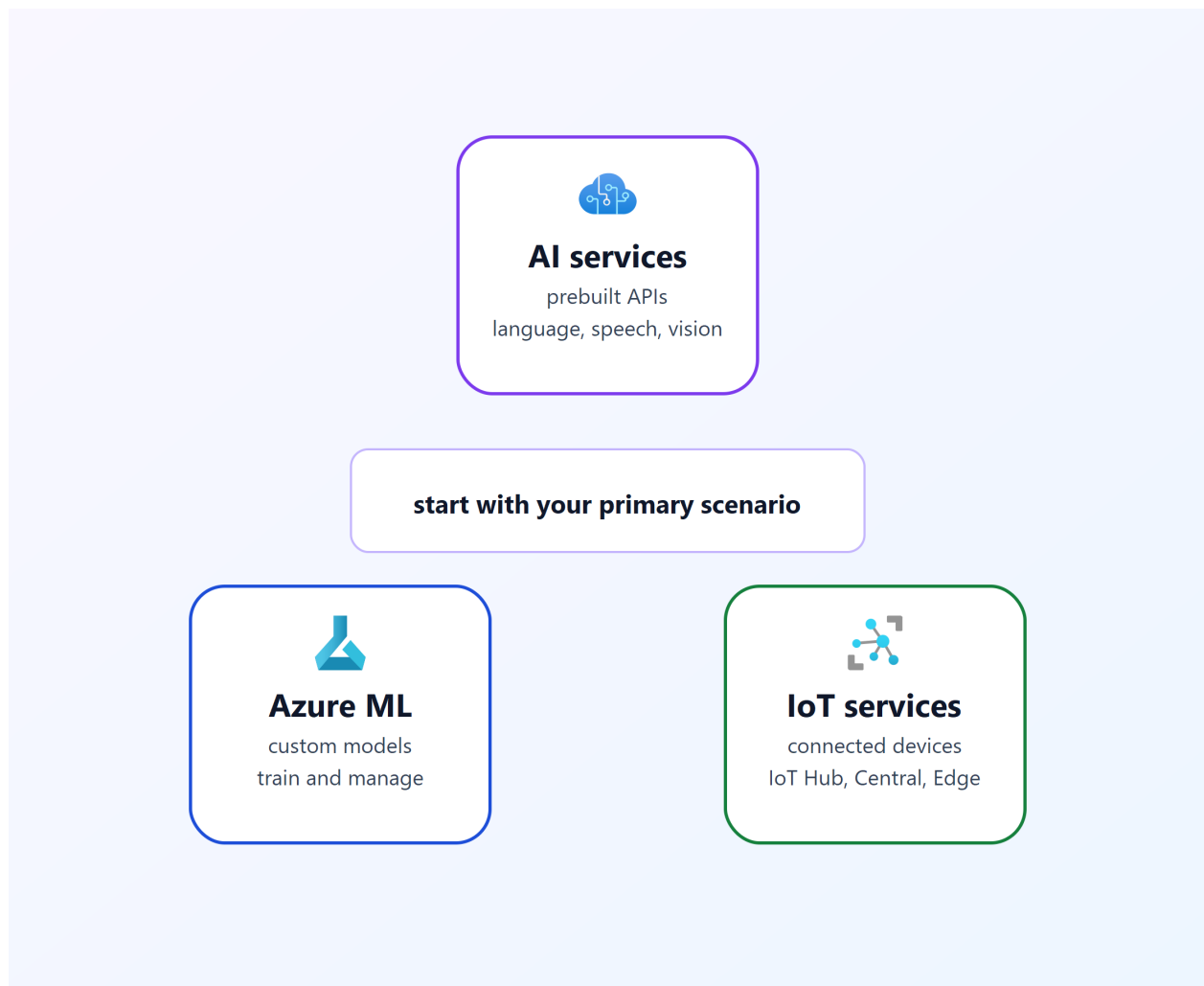
- Azure IoT Hub enables secure, bi-directional communication between cloud services and IoT devices.
- Azure IoT Central provides a simplified software as a service (SaaS) IoT platform for solution builders.
- Azure IoT Edge extends cloud capabilities to edge devices so some workloads can run closer to where data is generated.

This flow commonly starts with devices sending telemetry through IoT Hub, then uses cloud analytics to generate insights and model updates that IoT Edge runtime applies near the devices. IoT Central can simplify solution management for these connected-device scenarios.



Choosing the right option

At a fundamentals level, use this decision pattern:



- Use Azure AI services when you need prebuilt AI features exposed through APIs.
- Use Azure Machine Learning when you need custom model development and management.
- Use Azure IoT services when your solution centers on connected devices and telemetry.

8. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/13-knowledge-check>

Module assessment

Completed

9. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-azure-compute-networking-services/14-summary>

Summary

Completed

In this module, you learned about core Azure compute services and when to use them. You reviewed virtual machines and options such as virtual machine scale sets and virtual machine availability sets. You explored containers, Azure Functions, application hosting options, and Azure AI, machine learning, and IoT/Edge service categories.

You learned that agentic AI is an application pattern that can be built by combining Azure AI capabilities with application logic when a solution needs multistep reasoning and tool use.

Learning objectives

You should now be able to:

- Compare compute types, including container instances, virtual machines, and functions.
- Describe virtual machine options, including virtual machines (VMs), virtual machine scale sets, and virtual machine availability sets.
- Describe resources required for virtual machines.
- Describe application hosting options, including Azure Web Apps, containers, and virtual machines.
- Describe AI, machine learning, and IoT/Edge service categories in Azure.
- Recognize agentic AI as an application pattern built from Azure AI capabilities.

Additional resources

The following additional resources are intended to provide more information on topics in this module or on additional topics related to this module.

- [Host a web application with Azure App Service](#) is a Microsoft Learn module that explores the process of hosting a web application in Azure.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Build a decision matrix for when to use virtual machines, containers, Azure Functions, and Azure App Service across three realistic workloads."
- "Explain scale sets, availability sets, and Availability Zones with one failure scenario that shows how each option improves resiliency."
- "Create a short architecture walkthrough that combines compute hosting with one AI or IoT Edge service and explains why each component was chosen."

Describe Azure networking services

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-azure-networking-services/1-introduction>

Introduction

Completed

This module introduces Azure networking services that help you connect, secure, and route traffic between resources. You explore foundational capabilities for communication within Azure, between Azure and on-premises environments, and across internet-connected clients.

Learning objectives

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

- Describe Azure virtual networking, including subnets and endpoints.
- Describe connectivity options with Azure VPN Gateway.
- Describe when to use Azure ExpressRoute.
- Describe Azure DNS capabilities.
- Configure basic network access for an Azure resource.

2. Describe Azure virtual networking

<https://learn.microsoft.com/en-us/training/modules/describe-azure-networking-services/2-virtual-network>

Describe Azure virtual networking

Completed

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 lets you 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 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

Azure resources can communicate securely with each other 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 lets you link multiple Azure resources to virtual networks to improve security and provide optimal routing between resources.

Communicate with on-premises resources

Azure virtual networks let you 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 ways to achieve this connectivity:

- Point-to-site virtual private network connections are from a computer outside your environment back into your private 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

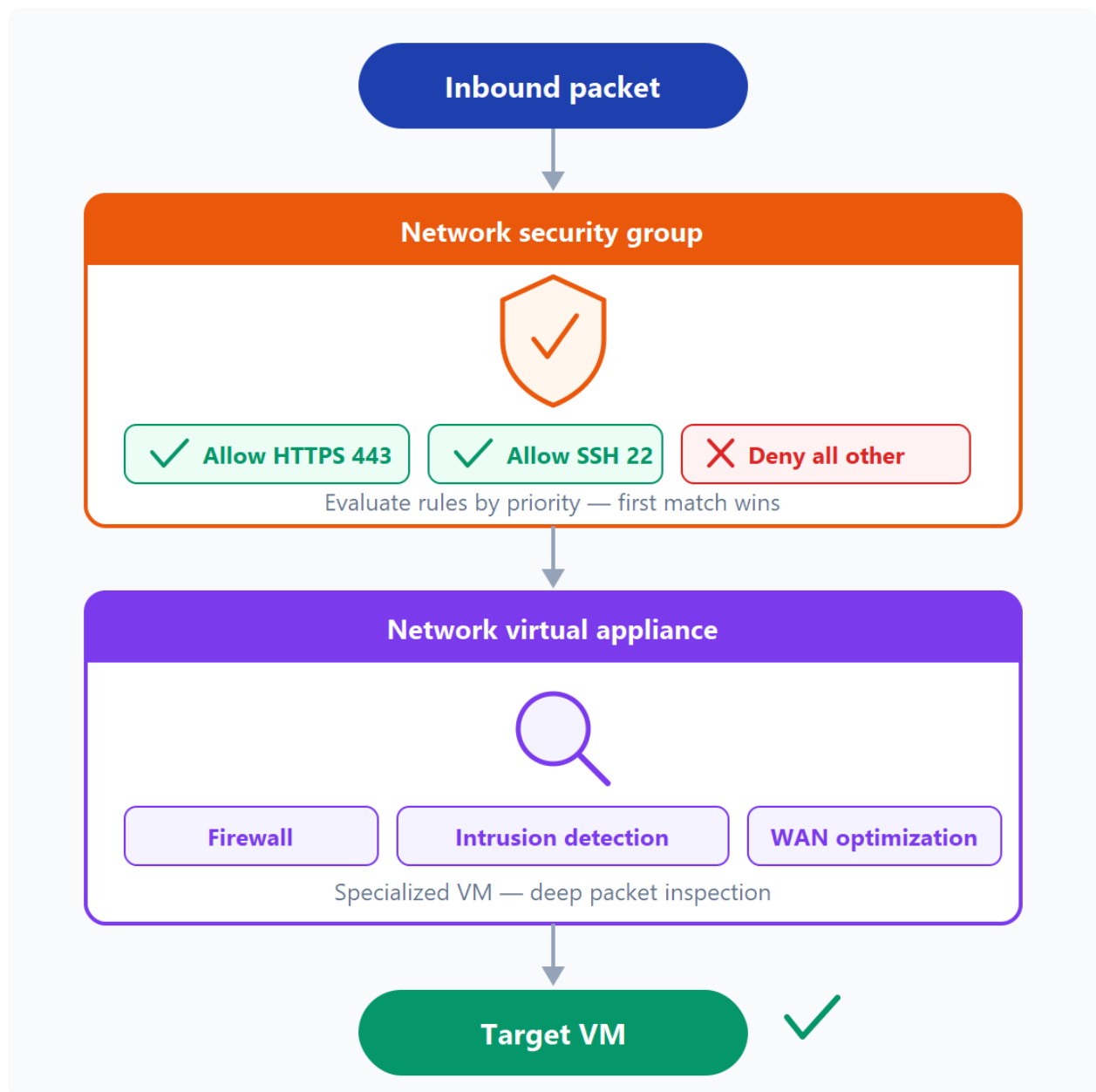
By default, Azure routes traffic between subnets on any connected virtual networks, on-premises networks, and the internet. You also can control routing and override those settings, as follows:

- Route tables let you 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.
- User-defined routes (UDR) let you control the routing tables between subnets within a virtual network or between virtual networks, giving you greater control over network traffic flow.

Filter network traffic

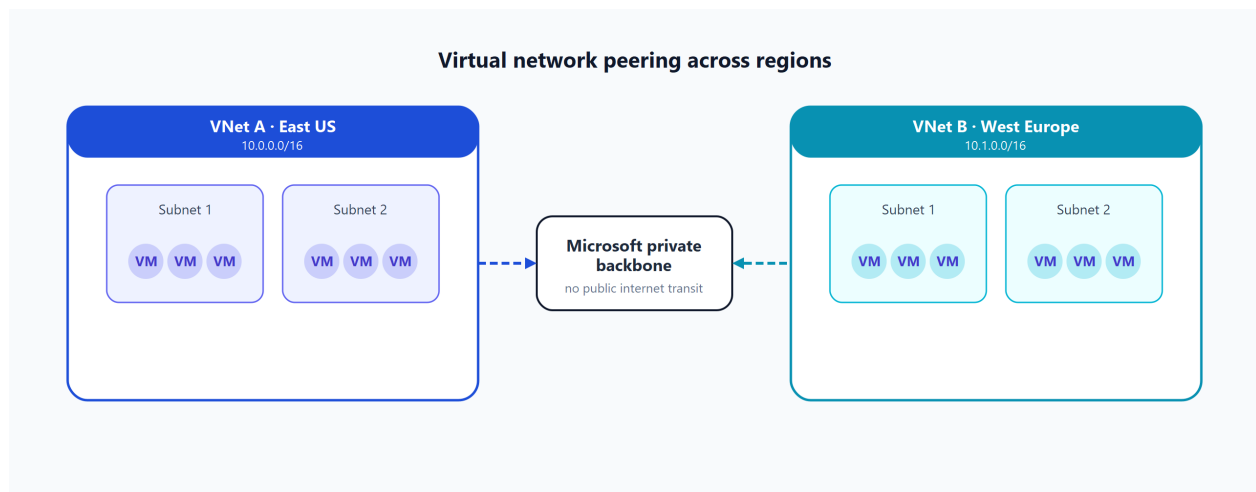
Azure virtual networks let you filter traffic between subnets by using the following approaches:

- 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 lets you create a global interconnected network through Azure.



3. Describe Azure virtual private networks

<https://learn.microsoft.com/en-us/training/modules/describe-azure-networking-services/3-virtual-private-networks>

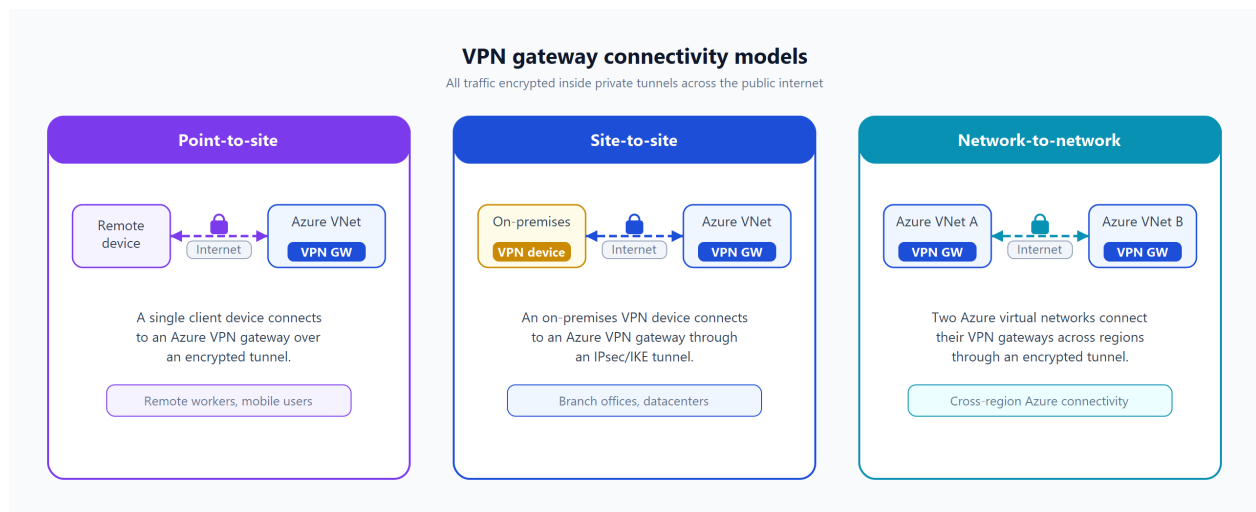
Describe Azure virtual private networks

Completed

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

VPN gateways

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



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

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

When 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 preshared 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 through which that packet is sent.
- 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:

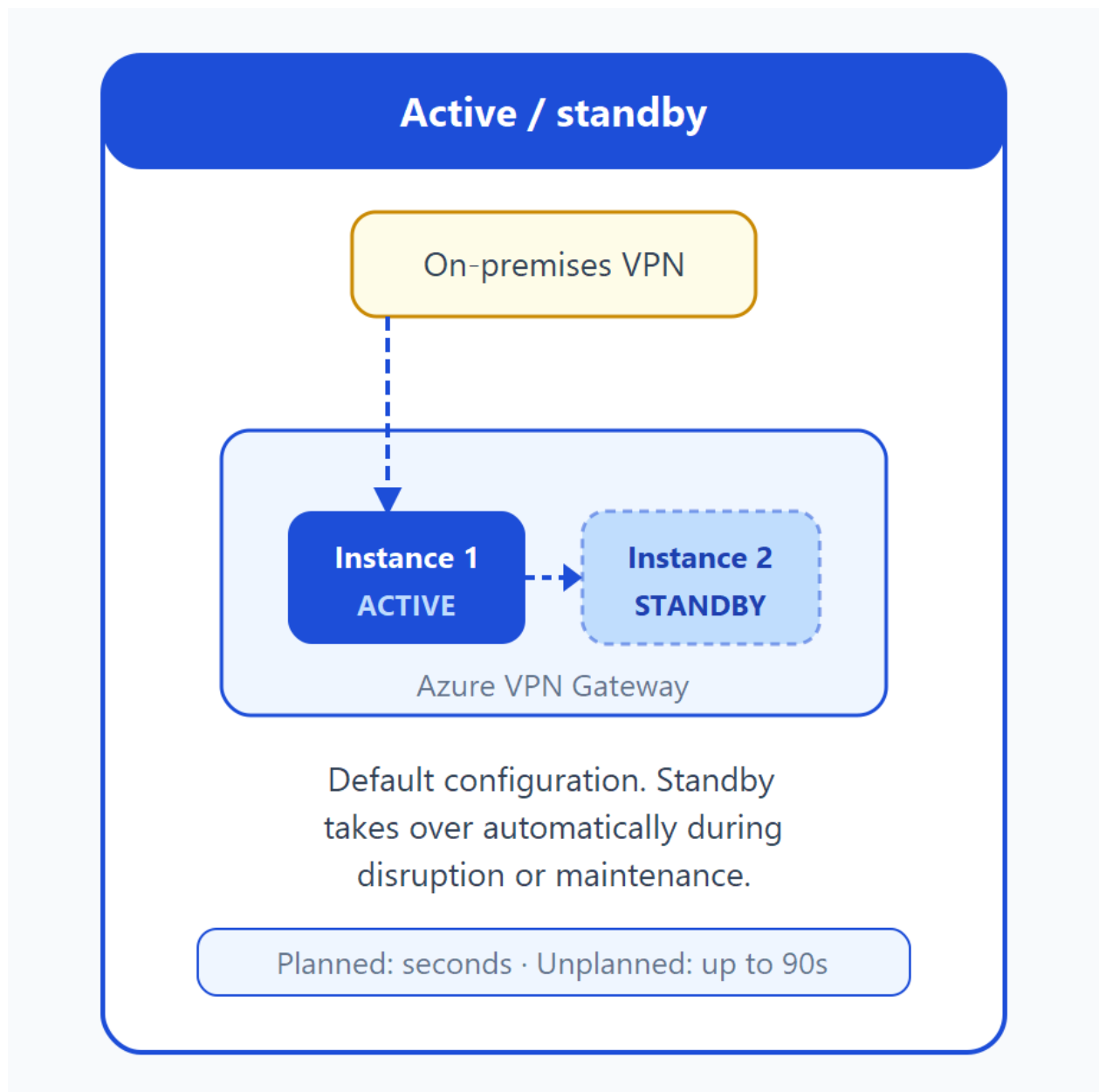
- Connections between virtual networks
- Point-to-site connections
- Multisite connections
- Coexistence with an Azure ExpressRoute gateway

High-availability scenarios

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

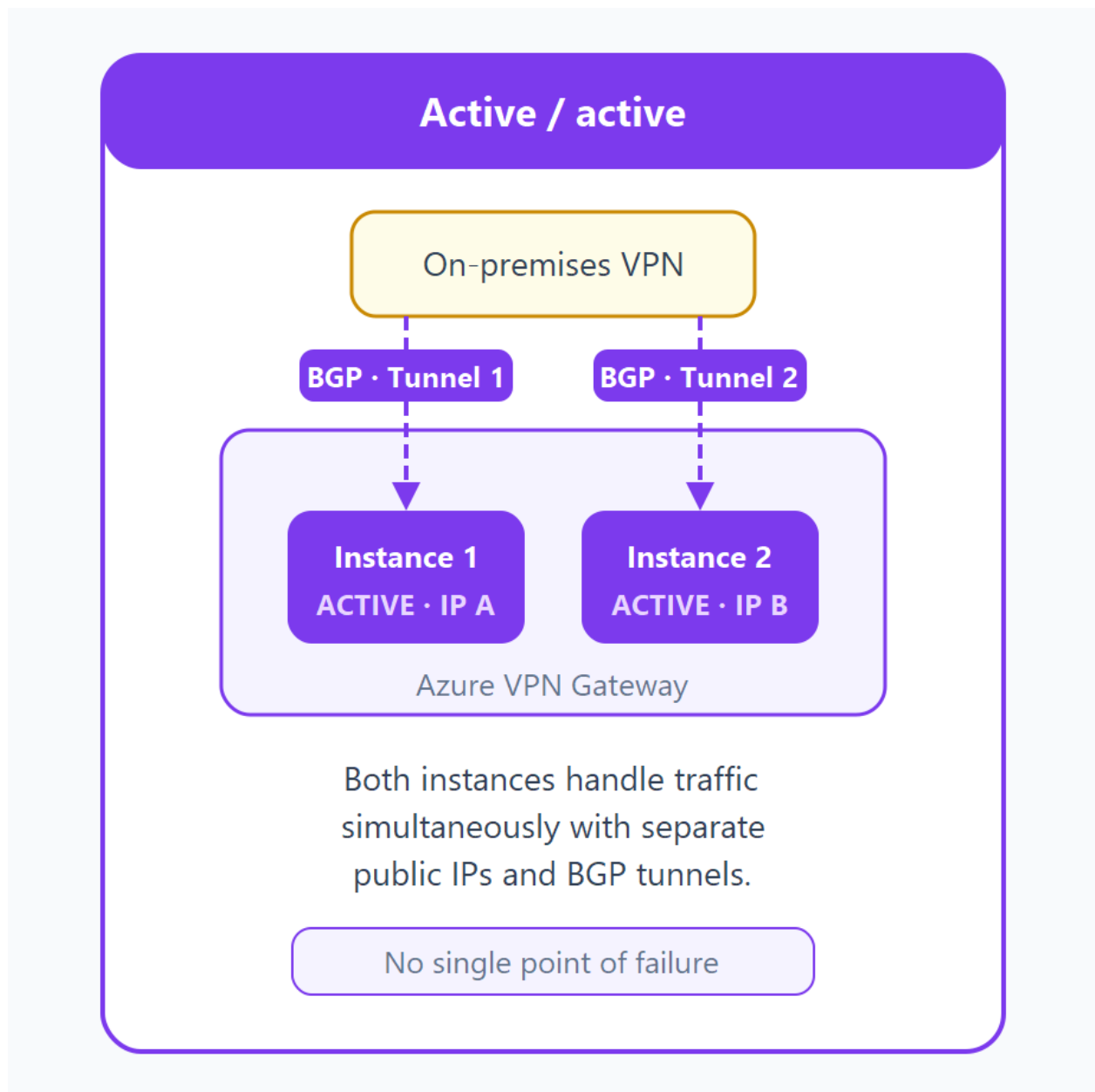
your VPN gateway.

Active/standby



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

Active/active

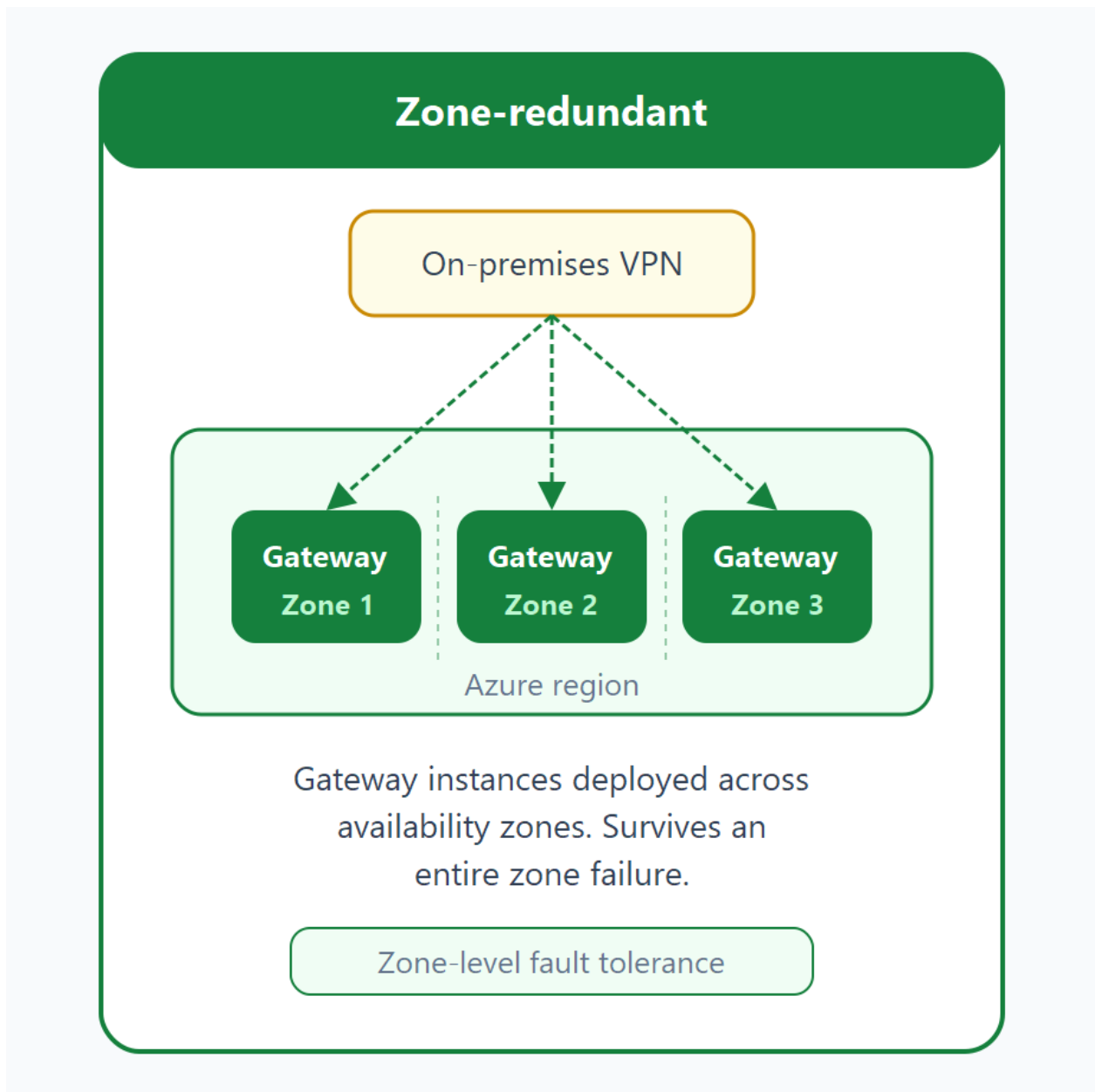


With the introduction of support for the BGP routing protocol, you can also deploy VPN gateways in an active/active configuration. In this configuration, you assign a unique public IP address to each instance. You then create separate tunnels from the on-premises device to each IP address. You can extend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

ExpressRoute failover

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

Zone-redundant gateways



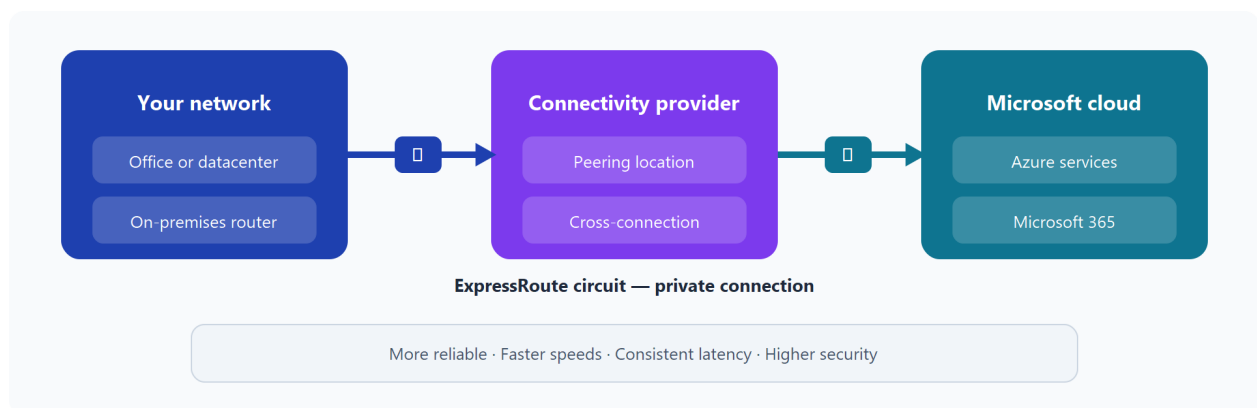
In regions that support availability zones, VPN gateways and ExpressRoute gateways can be deployed in a zone-redundant configuration. This configuration brings resiliency, scalability, and higher availability to virtual network gateways. Deploying gateways in Azure availability zones physically and logically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

4. Describe Azure ExpressRoute

Describe Azure ExpressRoute

Completed

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. ExpressRoute lets you connect offices, datacenters, or other facilities to the Microsoft cloud. Each location would have its own ExpressRoute circuit.



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

Features and benefits of ExpressRoute

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

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

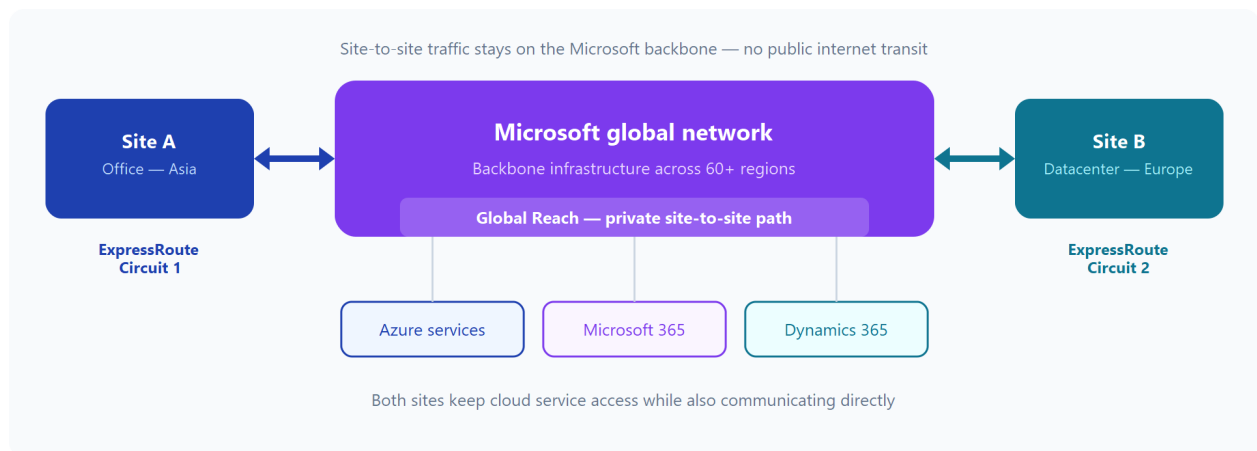
Connectivity to Microsoft cloud services

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

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

Global connectivity

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



Dynamic routing

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

Built-in redundancy

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

Connectivity model options

ExpressRoute supports several connectivity options depending on your provider and network design, including provider-based connectivity, point-to-point Ethernet, and direct connectivity at ExpressRoute locations. At this level, focus on when to use ExpressRoute rather than detailed implementation models.

At a high level, choose ExpressRoute when:

- You need private, consistent connectivity between on-premises networks and Azure.

- Your team has strict compliance or data-transfer requirements.
- You need predictable latency and high-throughput network performance.
- You want to avoid sending critical traffic over the public internet.

Security considerations

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

5. Describe Azure DNS

<https://learn.microsoft.com/en-us/training/modules/describe-azure-networking-services/5-domain-name-system>

Describe Azure DNS

Completed

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.

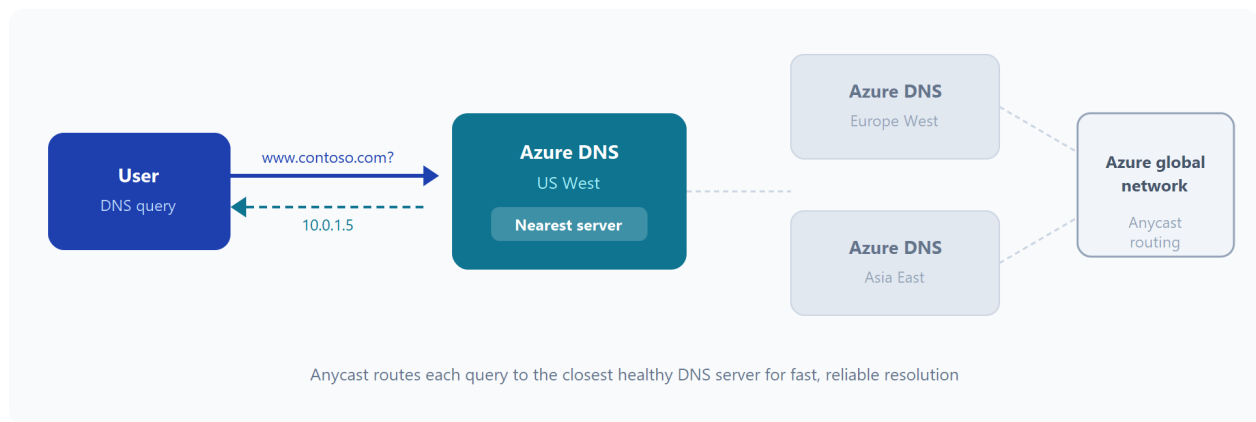
Benefits of Azure DNS

Azure DNS uses 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 the closest available DNS server answers each DNS query, providing fast performance and high availability for your domain.



Security

Azure DNS is based on Azure Resource Manager, which provides features such as Azure role-based access control (Azure RBAC), activity logs, and resource locks to help protect DNS resources and records.

Ease of use

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

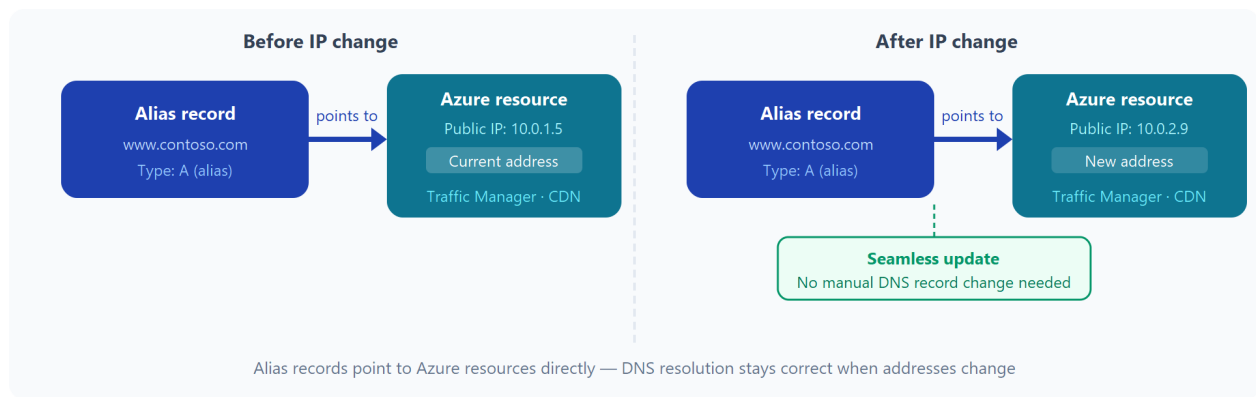
Because Azure DNS runs on Azure, you can manage your domains and records with the Azure portal, Azure PowerShell cmdlets, and the cross-platform Azure CLI. Applications that require automated DNS management can integrate with the service by using the REST API and SDKs.

Customizable virtual networks with private domains

Azure DNS also supports private DNS domains. This feature lets you use your own custom domain names in your private virtual networks instead of the Azure-provided names.

Alias records

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



Important

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

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-azure-networking-services/6-knowledge-check>

Module assessment

Completed

7. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-azure-networking-services/7-summary>

Summary

Completed

In this module, you learned how Azure networking services support secure and reliable connectivity. You reviewed virtual networking fundamentals, including segmentation with subnets and communication options through public and private endpoints.

You also explored VPN Gateway, ExpressRoute, and Azure DNS, and when each service is most appropriate. With these fundamentals, you can make better decisions about connectivity options for cloud-only and hybrid scenarios.

Learning objectives

You should now be able to:

- Describe Azure virtual networking, including subnets and endpoints.
- Describe connectivity options with Azure VPN Gateway.
- Describe when to use Azure ExpressRoute.
- Describe Azure DNS capabilities.
- Configure basic network access for an Azure resource.

Additional resources

- [Introduction to Azure network foundation services](#) is a Microsoft Learn course that provides deeper networking guidance in Azure.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Design a secure hybrid connectivity plan that chooses between VPN Gateway and ExpressRoute, and explain the tradeoffs in cost, latency, and reliability."
- "Walk me through a DNS strategy for an app that has both internet-facing endpoints and private internal services in Azure."
- "Create a troubleshooting flowchart for a connectivity issue from on-premises users to an Azure workload, including likely root causes and checks."

Describe Azure storage services

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/1-introduction>

Introduction

Completed

In this module, you'll be introduced to the Azure storage services. You'll learn about the Azure Storage Account and how that relates to the different storage services that are available. You'll also learn about blob storage tiers, data redundancy options, and ways to move data or even entire infrastructures to Azure.

Learning objectives

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

- Compare Azure storage services.
- Describe storage tiers.
- Describe redundancy options.
- Describe storage account options and storage types.
- Identify options for moving files, including AzCopy, Azure Storage Explorer, and Azure File Sync.
- Describe migration options, including Azure Migrate and Azure Data Box.

2. Describe Azure storage accounts

<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/2-accounts>

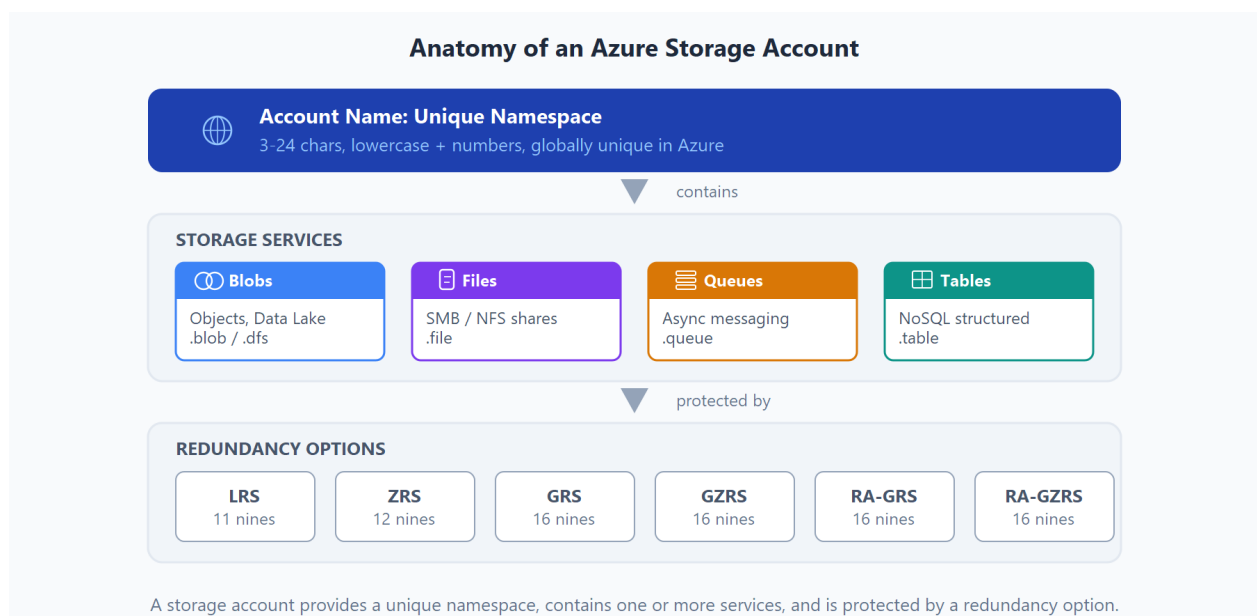
Describe Azure storage accounts

Completed

The following video introduces the services available with Azure Storage.



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. These redundancy options are covered later in this module:

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

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

Azure Storage Account Types

Standard GP v2	Premium Block Blobs	Premium File Shares	Premium Page Blobs
Services Blob (+ Data Lake) Queue Storage Table Storage Azure Files	Services Blob (+ Data Lake)	Services Azure Files	Services Page blobs only
Redundancy LRS, GRS, RA-GRS ZRS, GZRS, RA-GZRS	Redundancy LRS, ZRS	Redundancy LRS, ZRS	Redundancy LRS
Best For Most scenarios. Recommended default for blobs, files, queues, and tables.	Best For High transaction rates, small objects, or low latency requirements.	Best For High-scale or high-performance apps. SMB + NFS support.	Best For Page blob workloads (e.g., VM disks).

Standard GP v2 is recommended for most scenarios. Premium types offer lower latency for specialized workloads.

Storage account endpoints

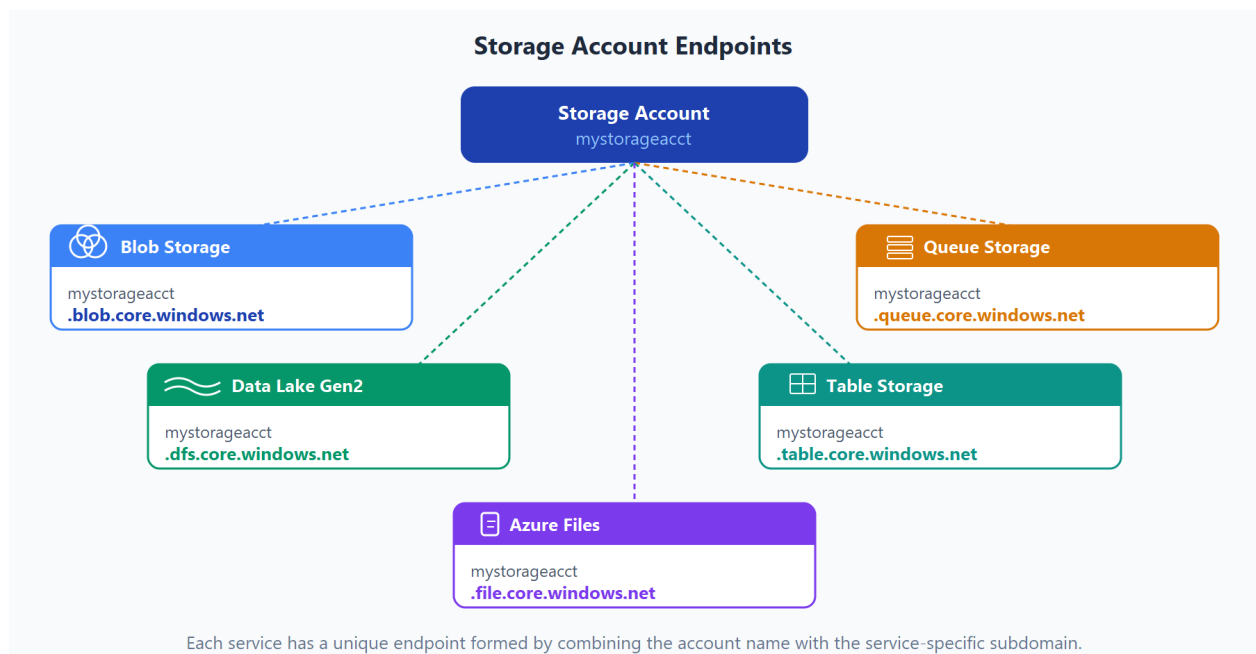
One of the benefits of using an Azure storage account is having a unique namespace in Azure for your data. Every storage account must have a unique account name within Azure. The combination of the account name and the Azure Storage service endpoint forms the endpoints for your storage account.

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

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

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

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



3. Describe Azure storage redundancy

<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/3-redundancy>

Describe Azure storage redundancy

Completed

Azure Storage keeps multiple copies of data to protect against hardware failures, outages, and regional disasters. Redundancy choices determine availability, durability, and cost.

When selecting a redundancy option, balance lower cost against higher availability. Key decision factors include:

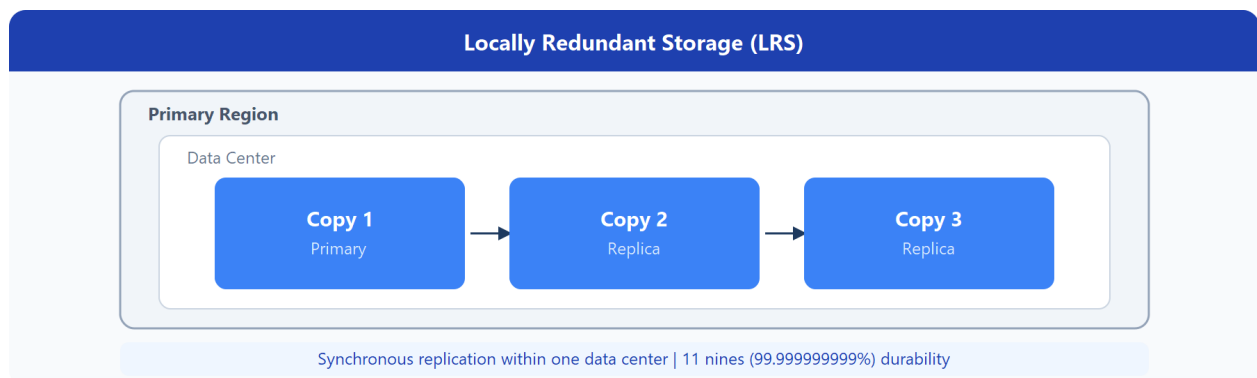
- How your data is replicated in the primary region.
- Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters.
- Whether your application requires read access to the replicated data in the secondary region if the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

Redundancy in the primary region

Data in Azure Storage is always replicated three times in the primary region. Primary-region options are locally redundant storage (LRS) and zone-redundant storage (ZRS).

Locally redundant storage

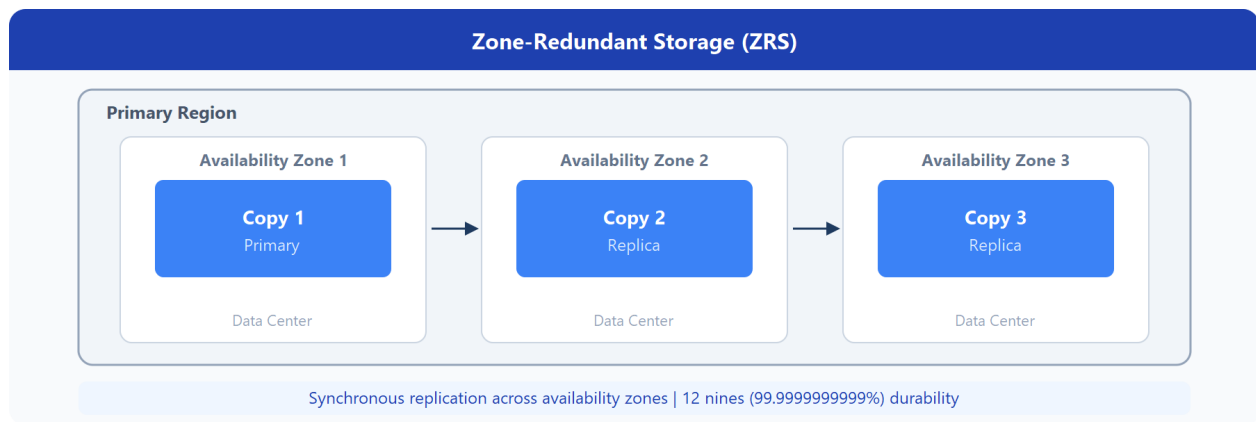
Locally redundant storage (LRS) replicates your data three times within a single data center in the primary region. LRS provides at least 11 nines of durability (99.999999999%) of objects over a given year.



LRS is the lowest-cost option and protects against server rack and drive failures in one datacenter. Because all replicas stay in that datacenter, a datacenter-level disaster can still cause data loss. For stronger resilience, use ZRS, GRS, or 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, data stays available for read and write operations even if one zone is unavailable. Azure performs networking updates, such as DNS repointing, during recovery.

Microsoft recommends ZRS for high availability in-region scenarios and for requirements that keep replication within a country or region.

Redundancy in a secondary region

For higher durability, you can replicate data to a secondary region that is geographically distant from the primary region.

When you create a storage account, you choose the primary region. Azure assigns the paired secondary region based on region pairs.

Secondary-region options are geo-redundant storage (GRS) and geo-zone-redundant storage (GZRS). GRS uses LRS in both regions, while GZRS uses ZRS in the primary region and LRS in the secondary region.

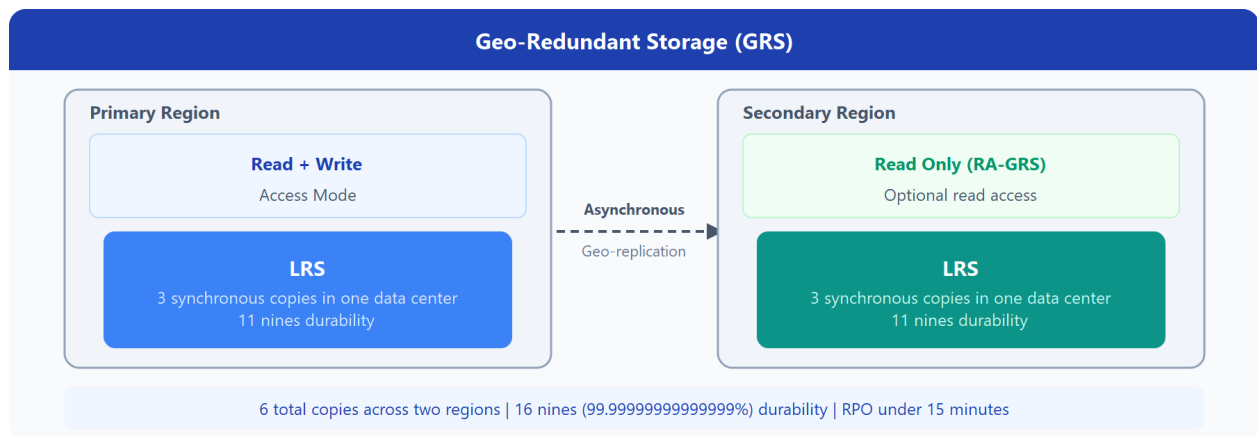
By default, secondary-region data isn't readable unless failover occurs. If the primary region is unavailable, you can fail over so the secondary region becomes primary.

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.

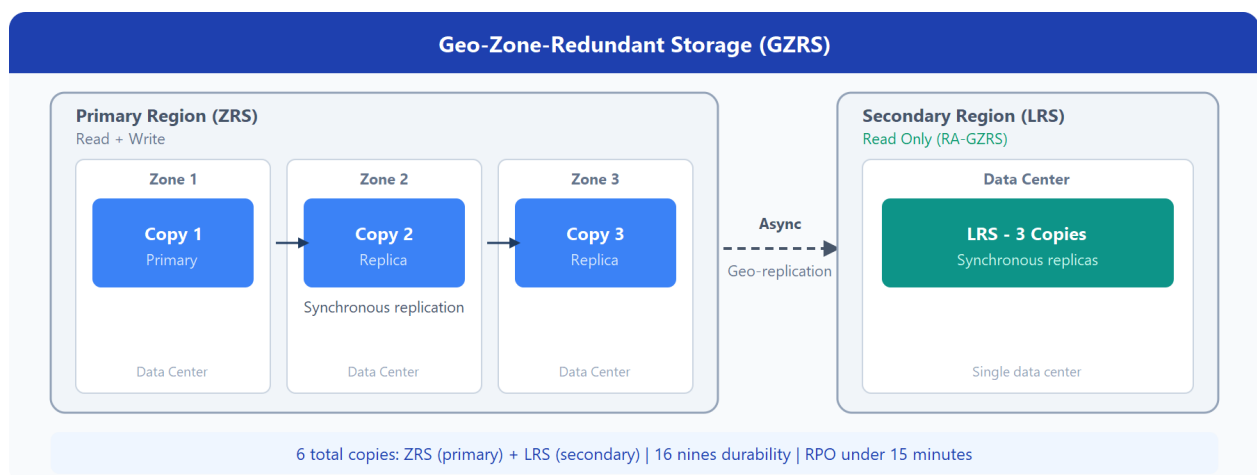
Geo-redundant storage

GRS copies data synchronously three times in the primary region (LRS), then asynchronously to the secondary region (also LRS). It provides at least 16 nines of durability over a year.



Geo-zone-redundant storage

GZRS combines zone-level resilience in the primary region with geo-replication to a secondary region. Data is copied across three availability zones in the primary region and replicated to the paired secondary region using LRS. Microsoft recommends GZRS for workloads that need maximum consistency, availability, and disaster recovery resilience.



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

Read access to data in the secondary region

GRS and GZRS protect against regional outages by replicating data to a secondary location. To read secondary-region data before failover, 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.

4. Describe Azure storage services

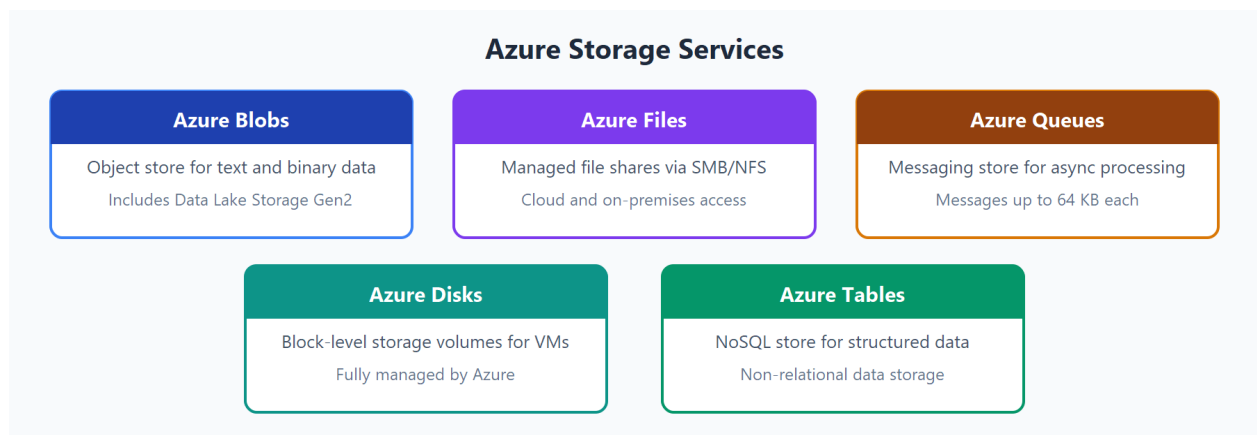
<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/4-describe-azure-storage-services>

Describe Azure storage services

Completed

Azure Storage includes these core 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.
- **Azure Tables:** NoSQL table option for structured, non-relational data.



Benefits of Azure Storage

Azure Storage provides these benefits:

- **Durable and highly available.** Redundancy options protect data from hardware failures, outages, and regional incidents.
- **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 can be accessed globally over HTTP or HTTPS by using REST APIs, SDKs, Azure CLI, Azure PowerShell, the Azure portal, or Azure Storage Explorer.

Azure Blobs

Azure Blob storage is an unstructured object storage service for large volumes of text or binary data. It's designed for scenarios such as large uploads, media content, log files, and analytics datasets. Developers store objects as blobs while Azure handles the underlying storage infrastructure.

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.

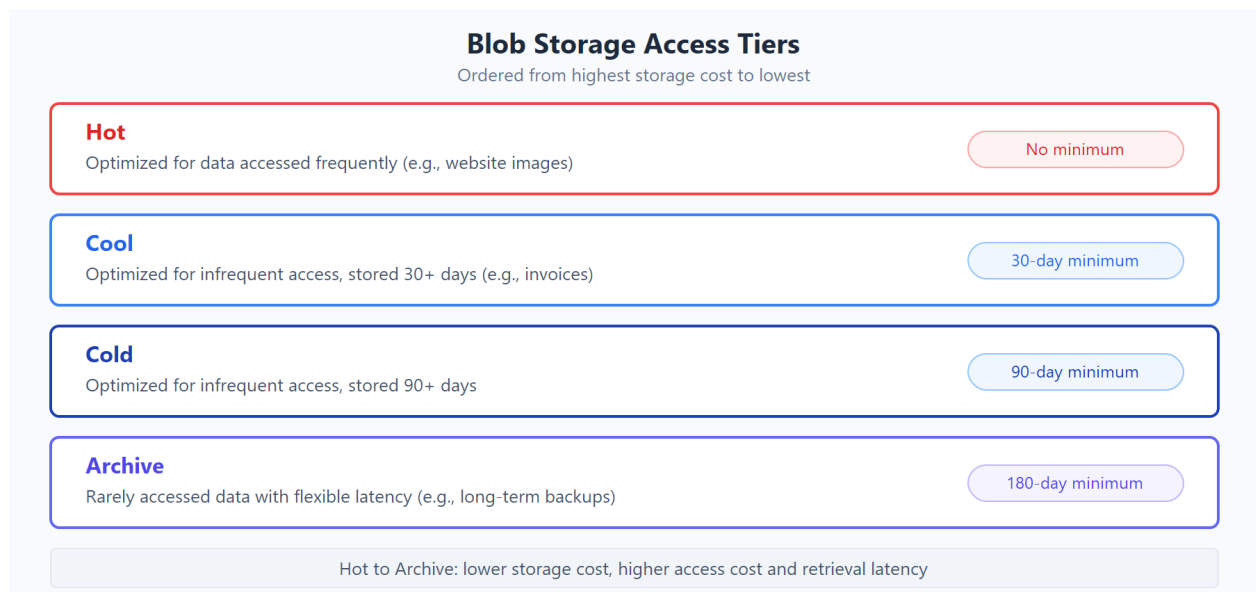
Blob storage tiers

Data access patterns change over time, so Azure Blob storage offers access tiers that balance storage cost against retrieval speed. 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).

Considerations for access tiers:

- Hot, cool, and cold access tiers can be set at the account level. The archive access tier isn't available at the account level.
- Hot, cool, cold, and archive tiers can be set at the blob level, during or after upload.
- Cool and cold tiers have lower storage costs but higher access costs and lower availability SLAs.
- Archive tier has the lowest storage cost and highest rehydration latency and access cost.



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 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 file 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:** SMB and NFS protocol support enables compatibility with existing applications.
- **Fully managed:** No server hardware or OS patching to manage.
- **Scripting and tooling:** Manage shares with Azure CLI, Azure PowerShell, the Azure portal, and Storage Explorer.
- **Resiliency:** Built for high availability.
- **Familiar programmability:** Applications can use standard file I/O APIs plus Azure SDKs and REST APIs.

Azure Queues

Azure Queue storage stores large numbers of messages for asynchronous processing. Queues are accessed by authenticated HTTP/HTTPS calls, can hold millions of messages, and support messages up to 64 KB.

Queue storage is commonly paired with Azure Functions so messages trigger background actions.

Azure Disks

Azure Disk storage (managed disks) provides block-level volumes for Azure VMs. They are virtualized and managed by Azure for improved resiliency and simpler operations.

Azure Tables

Azure Table storage is a NoSQL store for large amounts of structured, non-relational data, accessible through authenticated calls from cloud and hybrid environments.

5. Identify Azure data migration options

<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/6-identify-azure-data-migration-options>

Identify Azure data migration options

Completed

Now that you understand the different storage options within Azure, it's important to also understand how to get your data 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 Data Migration Options

Azure Migrate	Azure Data Box
Online Real-time migration hub Single portal to assess and migrate on-premises infrastructure to Azure Discovery and assessment Server migration Database migration Web app migration ISV tool integration	Physical Offline bulk data transfer Ship a secure storage device with up to 80 TB capacity One-time bulk migration Periodic large uploads Data export from Azure NIST 800-88r1 disk wipe End-to-end tracking in portal

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 includes integrated options for discovery and assessment, server migration, database migration, and web app migration.

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 for moving very large data sets (often tens of terabytes or more) when network bandwidth is limited.

Common Data Box scenarios include:

- One-time bulk migration of on-premises data to Azure.
- Periodic large data uploads where online transfer is too slow.
- Exporting large data sets from Azure for recovery or regulatory needs.

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.

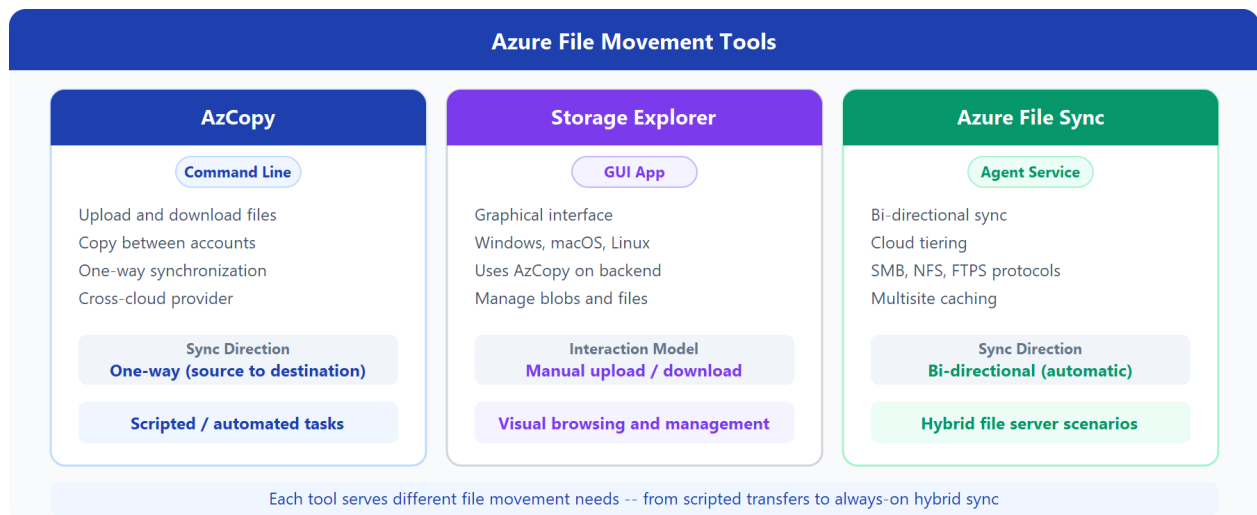
6. Identify Azure file movement options

<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/7-identify-azure-file-movement-options>

Identify Azure file movement options

Completed

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



AzCopy

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

Important

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

Azure Storage Explorer

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

Azure File Sync

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

With Azure File Sync, you can:

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

7. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-azure-storage-services/8-knowledge-check>

Module assessment

Completed

8. Summary

Summary

Completed

In this module, you learned about Azure storage services. You learned about Azure storage accounts and how they relate to different storage services. You explored storage blobs, redundancy options, and ways to migrate and move your data both into and within Azure.

Learning objectives

You should now be able to:

- Compare Azure storage services.
- Describe storage tiers.
- Describe redundancy options.
- Describe storage account options and storage types.
- Identify options for moving files, including AzCopy, Azure Storage Explorer, and Azure File Sync.
- Describe migration options, including Azure Migrate and Azure Data Box.

Additional resources

The following resources provide more information on topics in this module or related to this module.

- [Store data in Azure](#) is a Microsoft Learn course that covers more information about storing data in Azure.
- [Microsoft Certified: Azure Data Fundamentals](#) is an entire certification, with associated training that dives deeper into data fundamentals on Azure.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Map four workload patterns to the right Azure storage service and justify each choice for performance, durability, and access requirements."
- "Compare LRS, ZRS, GRS, and GZRS for a business-critical workload and recommend one option based on RPO, availability, and budget constraints."

- "Create a migration checklist that uses Azure Migrate, Data Box, AzCopy, Storage Explorer, or File Sync where each tool fits best."

Describe Azure identity, access, and security

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/1-introduction>

Introduction

Completed

In this module, you'll be introduced to the Azure identity, access, and security services and tools. You'll learn about directory services in Azure, authentication methods, and access control. You'll also cover Zero Trust, defense in depth, and how they keep your cloud safer. Finally, you'll review encryption concepts, key management with Azure Key Vault, and Microsoft Defender for Cloud.

Learning objectives

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

- Describe directory services in Azure, including Microsoft Entra ID and Microsoft Entra Domain Services.
- Describe authentication methods in Azure, including single sign-on (SSO), multifactor authentication (MFA), and passwordless.
- Describe external identities and guest access in Azure.
- Describe Microsoft Entra Conditional Access.
- Describe Azure Role Based Access Control (RBAC).
- Describe the concept of Zero Trust.
- Describe the purpose of the defense in depth model.
- Describe encryption concepts and key management options in Azure.
- Describe the purpose of Microsoft Defender for Cloud.

2. Describe Azure directory services

Describe Azure directory services

Completed

Microsoft Entra ID is Microsoft's cloud-based identity and access management service. It lets you sign in and access both Microsoft cloud applications and cloud applications that you develop.

If you've worked with on-premises Active Directory, Microsoft Entra ID will feel familiar. The key difference is that you control the identity accounts while Microsoft ensures the service is available globally.

Connecting the two unlocks extra protection. On its own, on-premises Active Directory doesn't monitor sign-in behavior. Once connected to Microsoft Entra ID, Microsoft can detect suspicious sign-in attempts at no extra cost — for example, sign-ins from unexpected locations or unknown devices.

Who uses Microsoft Entra ID?

Microsoft Entra ID is for:

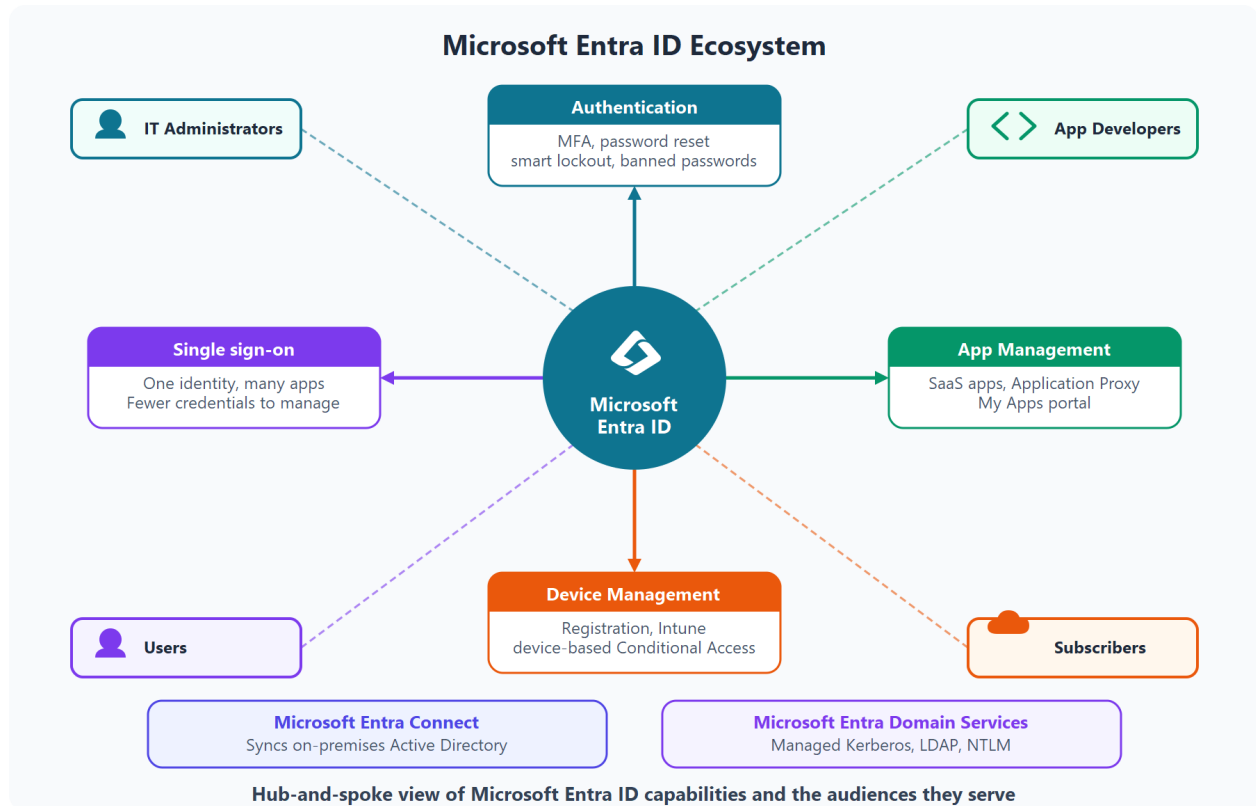
- **IT administrators.** Administrators can use Microsoft Entra ID to control access to applications and resources based on workload and security requirements.
- **App developers.** Developers can use Microsoft Entra ID to provide a standards-based approach for adding functionality to applications that 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.

What does Microsoft Entra ID do?

Microsoft Entra ID provides services such as:

- **Authentication** — Verifies identity before granting access. Includes self-service password reset, multifactor authentication, banned password lists, and smart logout.
- **Single sign-on (SSO)** — Lets one identity access multiple applications. SSO benefits and behavior are covered in the authentication methods unit.
- **Application management** — Manages cloud and on-premises apps through features like Application Proxy, SaaS app integration, and the My Apps portal.

- **Device management** — Supports device registration and management through tools like Microsoft Intune. Enables device-based Conditional Access policies that restrict access to known devices.



Can I connect my on-premises AD with Microsoft Entra ID?

Without a connection, an on-premises Active Directory deployment and a cloud Microsoft Entra ID deployment require you to maintain two separate identity sets. Microsoft Entra Connect bridges that gap.

Microsoft Entra Connect synchronizes user identities between on-premises Active Directory and Microsoft Entra ID. Because changes flow between both systems, users get a consistent experience — including SSO, multifactor authentication, and self-service password reset — whether they're accessing on-premises or cloud resources.

What is Microsoft Entra Domain Services?

Microsoft Entra Domain Services provides managed domain services — domain join, group policy, LDAP, and Kerberos/NTLM authentication — without requiring you to deploy or maintain domain controllers in the cloud.

This is especially useful for legacy applications that can't use modern authentication. You can lift and shift those applications from on-premises into a managed domain without managing an AD DS environment in the cloud.

Because Microsoft Entra Domain Services integrate with your existing Microsoft Entra tenant, users can sign in to the managed domain with their existing credentials. Existing groups and user accounts also carry over, providing a smoother migration path.

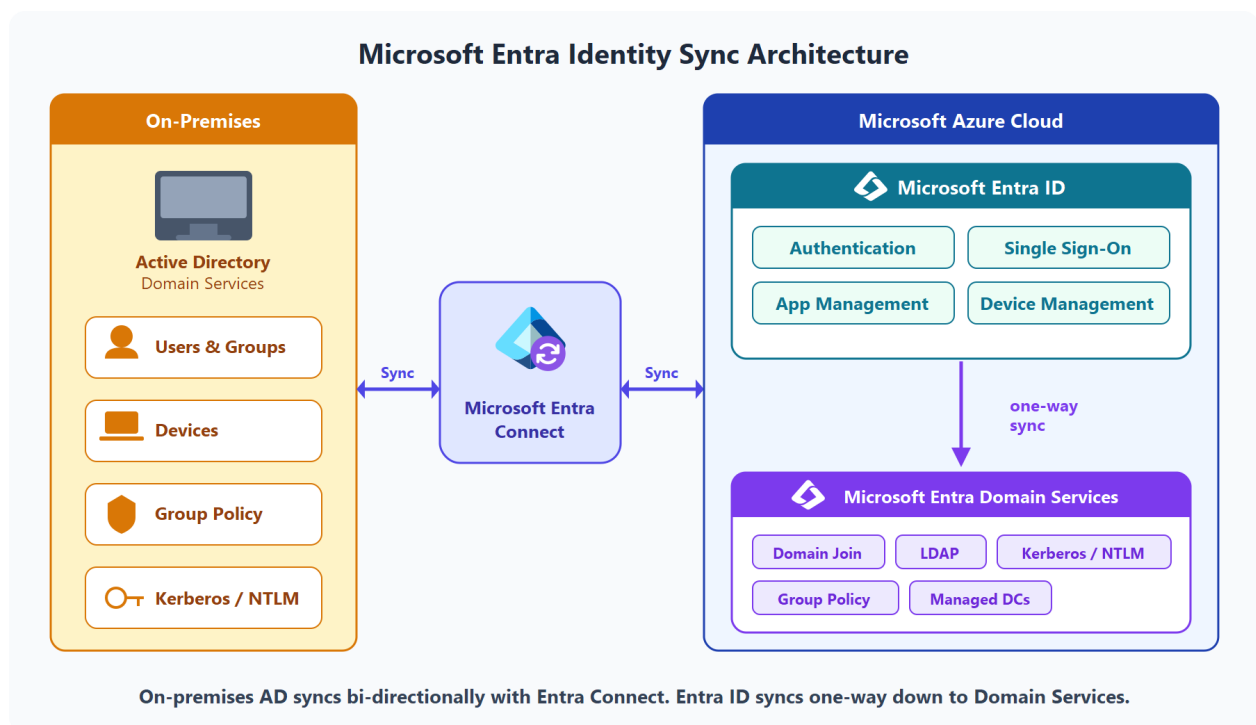
How does Microsoft Entra Domain Services work?

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 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.

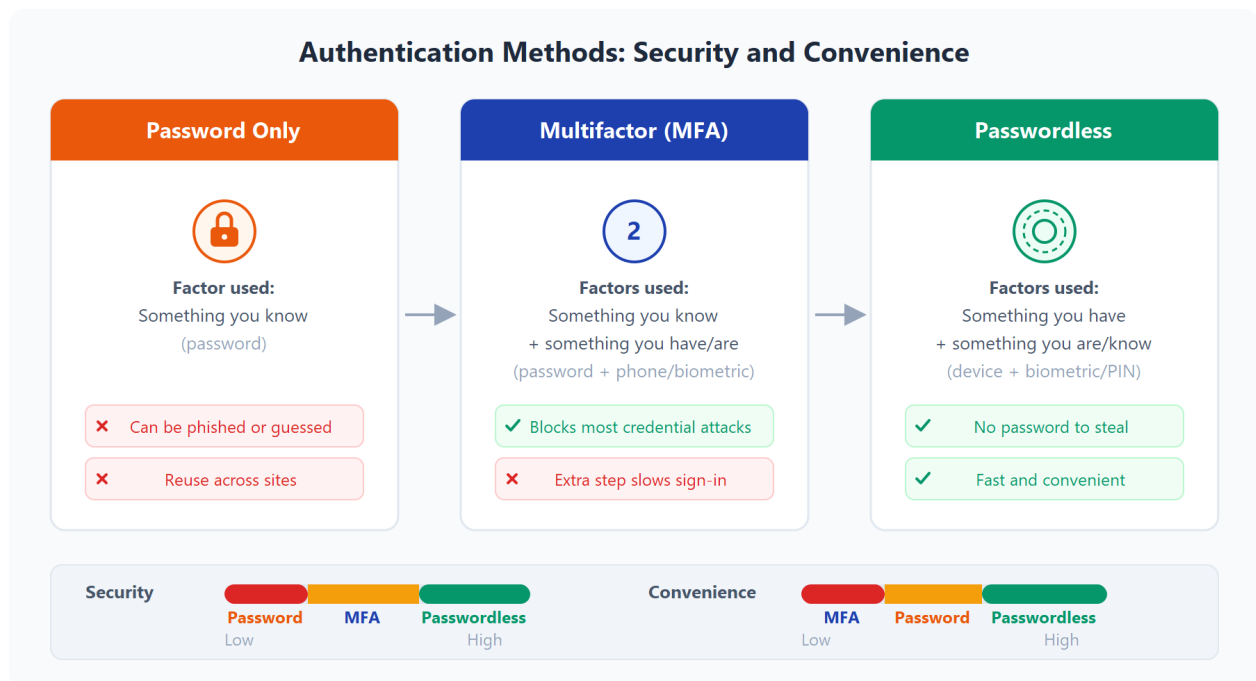
3. Describe Azure authentication methods

Describe Azure authentication methods

Completed

Authentication establishes the identity of a person, service, or device by requiring credentials. In Azure, common methods include passwords, single sign-on (SSO), multifactor authentication (MFA), and passwordless sign-in. Modern approaches are designed to improve both security and user convenience.

The following diagram shows the security level compared to the convenience. Notice Passwordless authentication is high security and high convenience while passwords on their own are low security but high convenience.



What's single sign-on?

Single sign-on (SSO) lets a user sign in once and access multiple trusted applications. SSO reduces password sprawl, which lowers the chance of credential-related incidents and decreases account lockout and reset overhead.

From an operations perspective, SSO also simplifies lifecycle management. Access is tied to one identity, making it easier to update or remove access when roles change.

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.

What's multifactor authentication?

Multifactor authentication (MFA) prompts a user for an extra factor during sign-in, so a compromised password alone isn't enough for access. These factors fall into three categories:

- **Something the user knows** — a password or challenge question.
- **Something the user has** — a code sent to a mobile phone.
- **Something the user is** — a biometric signal such as a fingerprint or face scan.

With MFA enabled, an attacker who obtains a password still needs the second factor — such as a phone prompt or biometric signal — to complete sign-in.

What's Microsoft Entra multifactor authentication?

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.

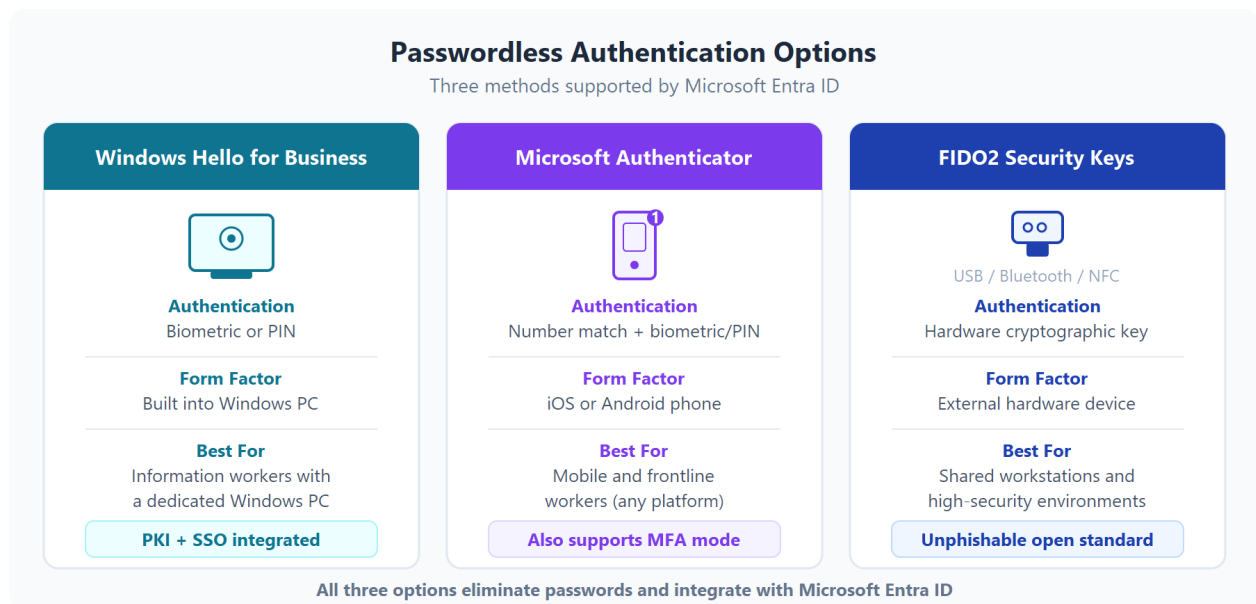
What's passwordless authentication?

While MFA adds security, passwords themselves remain a usability and risk challenge. Passwordless methods eliminate the password entirely, replacing it with a trusted device plus a biometric signal or PIN.

After initial registration, the user signs in with a factor they know or are — such as a PIN or fingerprint — instead of typing a password.

Microsoft Entra ID supports three passwordless options:

- Windows Hello for Business
- Microsoft Authenticator app
- FIDO2 security keys



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. With public key infrastructure (PKI) integration and built-in support for single sign-on (SSO), Windows Hello for Business provides a convenient method for seamlessly accessing work resources on-premises and in the cloud.

Microsoft Authenticator app

The Microsoft Authenticator app can also serve as a passwordless credential, turning any iOS or Android phone into a strong sign-in factor.

To sign in, the user receives a notification on their phone, matches a number displayed on screen, and confirms with a biometric signal (touch or face) or PIN. No password is needed.

FIDO2 security keys

FIDO2 is an open standard for passwordless authentication built on the web authentication (WebAuthn) specification. FIDO2 security keys are unphishable hardware devices — typically USB, but also available with Bluetooth or NFC — that handle authentication without a username or password.

Users register a FIDO2 key and then select it at the sign-in screen as their primary authentication method. Because the hardware device handles authentication, there's no password that could be exposed or guessed.

4. Describe Azure external identities

Describe Azure external identities

Completed

An external identity is a person, device, or service that exists outside your tenant. Microsoft Entra External ID includes the capabilities used to securely interact with users beyond your tenant boundary.

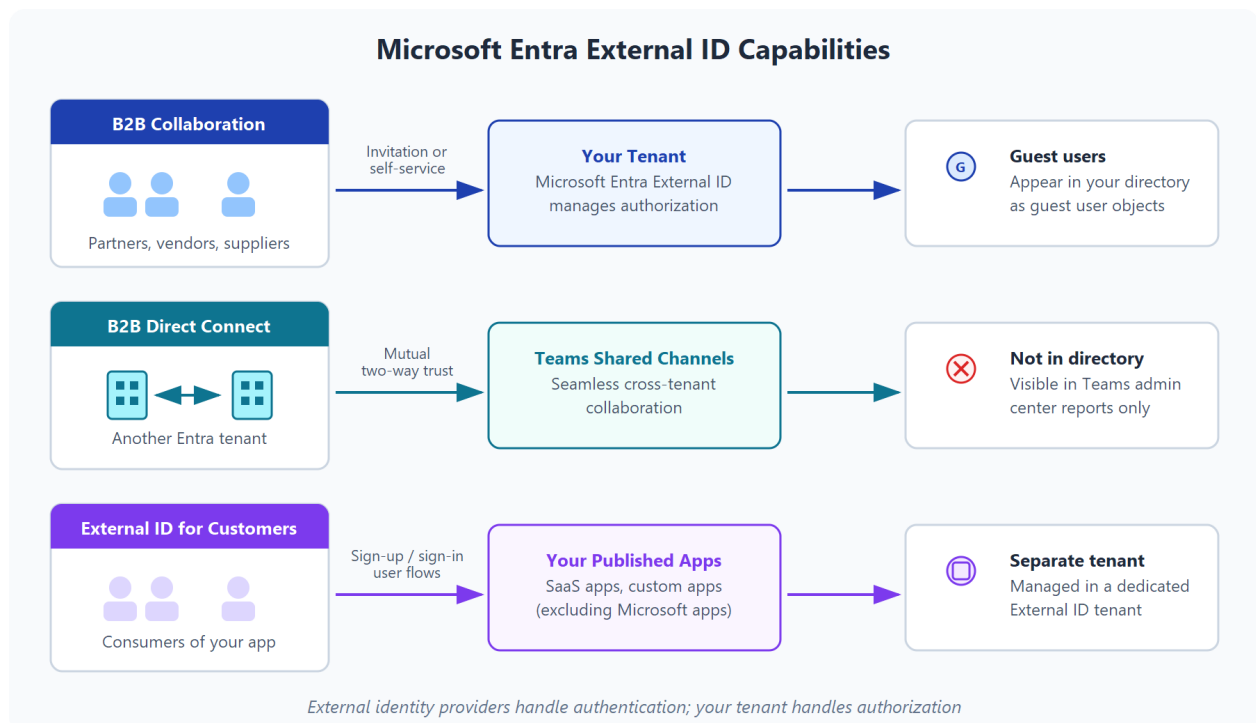
Why external identities matter

Organizations often need to collaborate with partners, suppliers, vendors, and contractors. External identities let those users access approved resources by using their existing credentials, while your team still enforces access policies.

For example, a project team can invite a supplier's users into specific collaboration spaces without creating and managing separate local credentials for each guest.

External identities can sound similar to single sign-on, but they're used for cross-tenant and consumer access scenarios. External users can bring their own identities, whether those are work accounts, government-issued digital identities, or social identities such as Google or Facebook.

The external identity provider manages authentication, and you manage authorization to your applications with Microsoft Entra ID or Azure AD B2C.



External ID capabilities

The following capabilities make up External Identities:

- **B2B collaboration** - Collaborate with external users by letting them use their preferred identity to sign in to your Microsoft applications or other internal 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 tenant 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 Entra External ID for customers (formerly Azure AD B2C)** - Publish modern SaaS apps or custom-developed apps (excluding Microsoft apps) to consumers and customers, while using Entra External ID for identity and access management.

External ID Capabilities at a Glance		
B2B Collaboration	B2B Direct Connect	External ID for Customers
WHO Partners, vendors, suppliers using work or social accounts	WHO Users from another Microsoft Entra tenant	WHO Consumers of your published SaaS or custom app
HOW Invitation or self-service sign-up into your tenant	HOW Mutual two-way trust between Entra tenants	HOW Sign-up and sign-in user flows with custom policies
DIRECTORY PRESENCE ✓ Represented as guest user objects	DIRECTORY PRESENCE ✗ Not in your directory; visible in Teams admin	DIRECTORY PRESENCE ✓ Managed in a separate External ID tenant
BEST FOR Sharing internal apps, SaaS apps, or custom apps	BEST FOR Teams shared channels across organizations	BEST FOR Consumer-facing apps (excluding Microsoft apps)

External providers authenticate; Microsoft Entra ID authorizes access to your resources

Depending on your collaboration goals and the resources being shared, you can combine these capabilities.

Manage guest access over time

With Microsoft Entra ID, you can enable collaboration across tenant boundaries by using B2B features. Guest users from other tenants can be invited by administrators or authorized users. This capability also applies to social identities such as Microsoft accounts.

You also can easily ensure that guest users have appropriate access. You can ask the guests themselves or a decision maker to participate in an access review and recertify (or attest) to the guests' access. The reviewers can give their input on each user's need for continued access, based on

suggestions from Microsoft Entra ID. When an access review is finished, you can then make changes and remove access for guests who no longer need it.

5. Describe Azure conditional access

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/5-conditional-access>

Describe Azure conditional access

Completed

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.

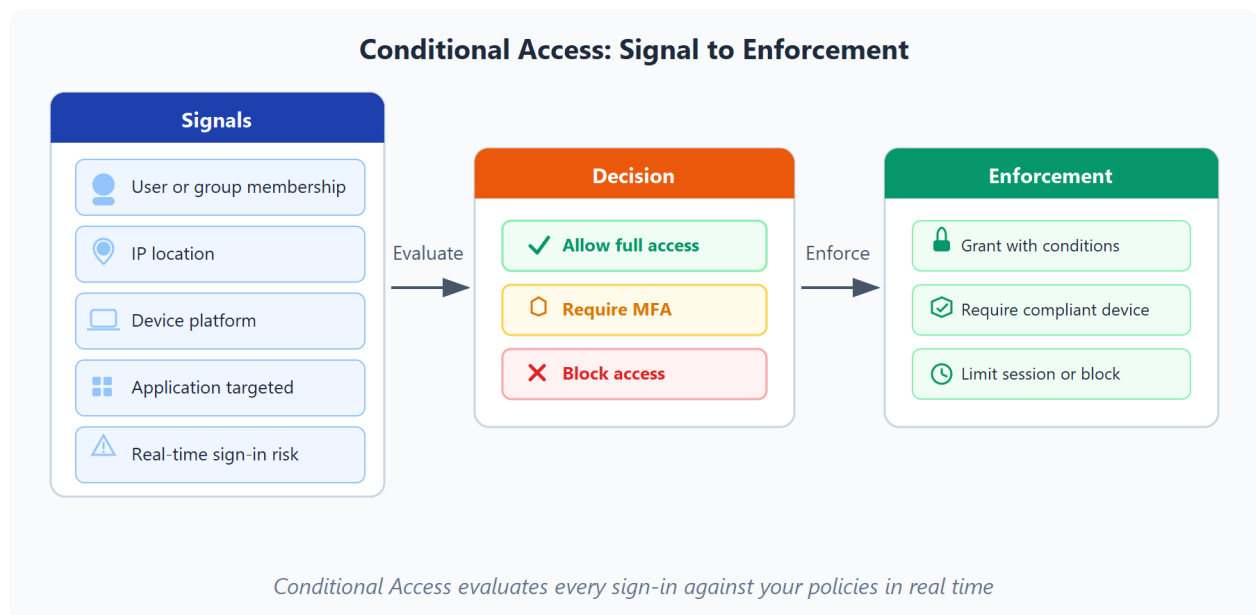
Conditional Access helps IT administrators:

- Empower users to be productive wherever and whenever.
- Protect critical assets.

Conditional Access also provides a more granular multifactor authentication experience for users. For example, a user might not be challenged for a 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.

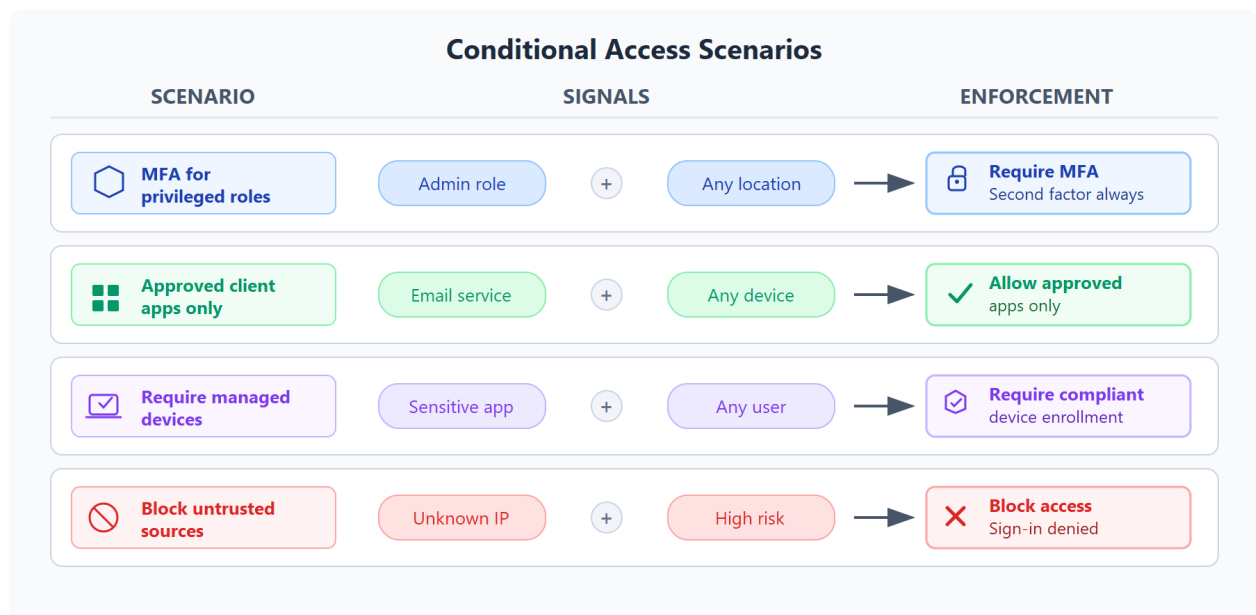
Enforcement is the action that carries out the decision. For example, the action is to allow access or require the user to provide a second form of authentication.

When can I use Conditional Access?

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, or for people connecting from outside trusted network locations.
- 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.

The following diagram shows how these common scenarios map signals to enforcement actions:



Common IT tasks include requiring MFA for privileged roles, restricting access to sensitive apps from unmanaged devices, and blocking risky sign-ins from unexpected locations.

6. Describe Azure role-based access control

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/6-role-based-access-control>

Describe Azure role-based access control

Completed

When you have multiple IT and engineering teams, how can you control what access they have to the resources in your cloud environment? The principle of least privilege says you should only grant access up to the level needed to complete a task. If you only need read access to a storage blob, then you should only be granted read access to that storage blob — not write access, and not access to other blobs. It's a good security practice to follow.

However, managing that level of permissions for an entire team would become tedious. Instead of defining the detailed access requirements for each individual, and then updating access requirements when new resources are created or new people join the team, 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.

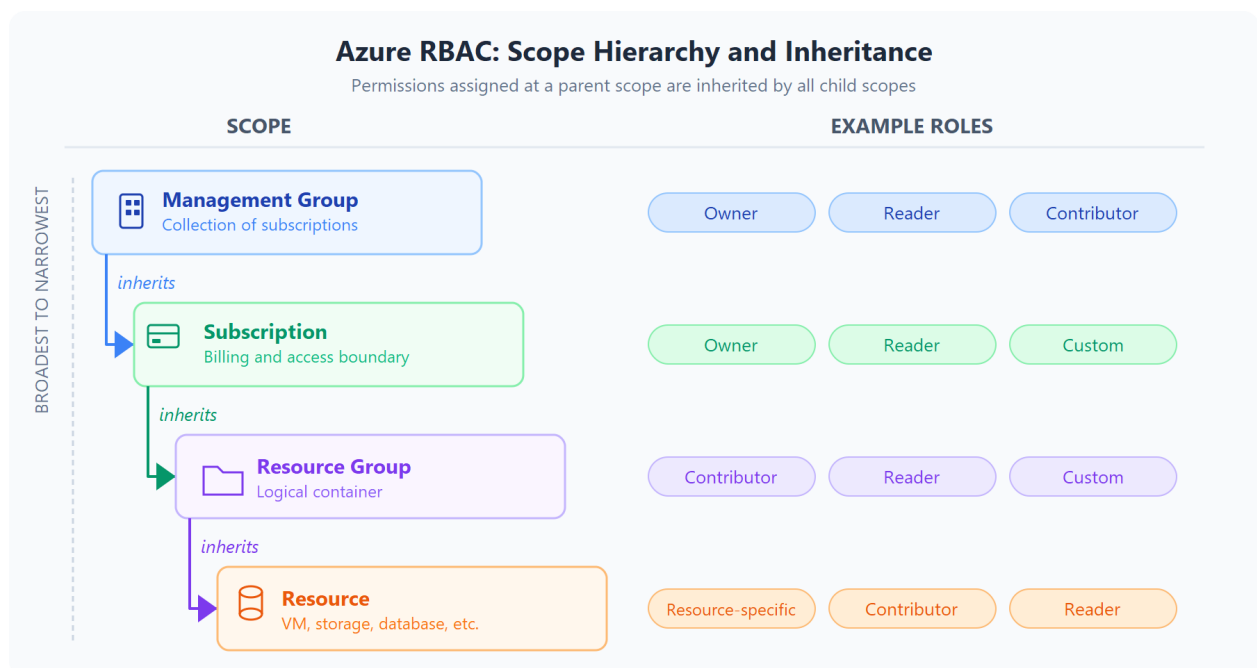
When you assign individuals or groups to one or more roles, they receive all the associated access permissions.

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.

How is role-based access control applied to resources?

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

The following diagram shows the relationship between roles and scopes. A management group, subscription, or resource group might be given the role of owner, so they have increased control and authority. An observer, who isn't expected to make any updates, might be given a role of Reader for the same scope, enabling them to review or observe the management group, subscription, or resource group.



Scopes include:

- A management group (a collection of multiple subscriptions).
- A single subscription.
- A resource group.
- A single resource.

Observers, users managing resources, admins, and automated processes illustrate the kinds of users or accounts that would typically be assigned each of the various roles.

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.

How is Azure RBAC enforced?

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.

7. Describe Zero Trust model

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/7-describe-zero-trust-model>

Describe Zero Trust model

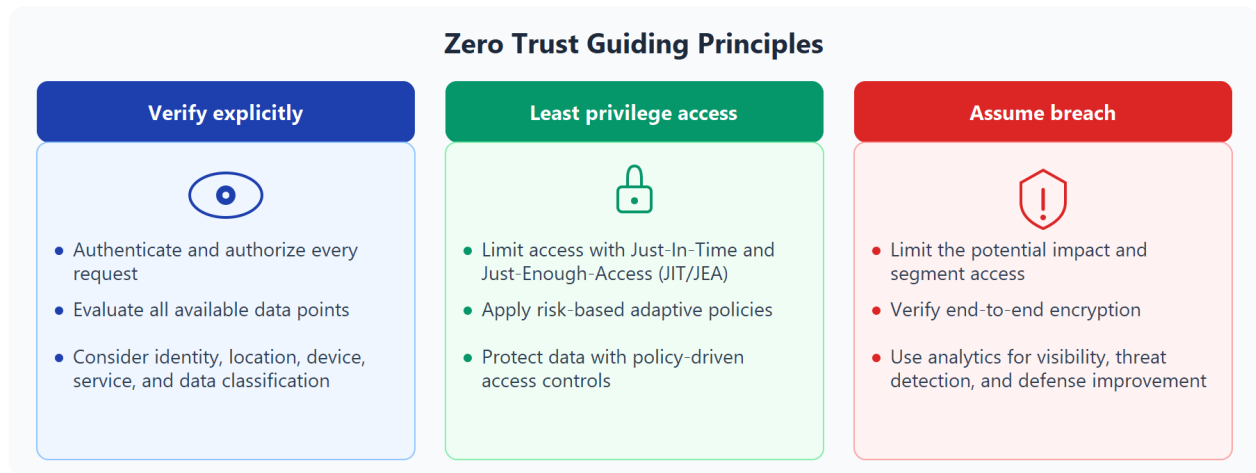
Completed

Zero Trust is a security model that assumes the worst case scenario and protects resources with that expectation. Zero Trust assumes breach at the outset, and then verifies each request as though it originated from an uncontrolled network.

Today, IT teams need a security model that effectively adapts to the complexity of the modern environment; embraces the mobile workforce; and protects people, devices, applications, and data wherever they're located.

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** - Limit the potential impact and segment access. Verify end-to-end encryption. Use analytics to get visibility, drive threat detection, and improve defenses.



Adjusting to Zero Trust

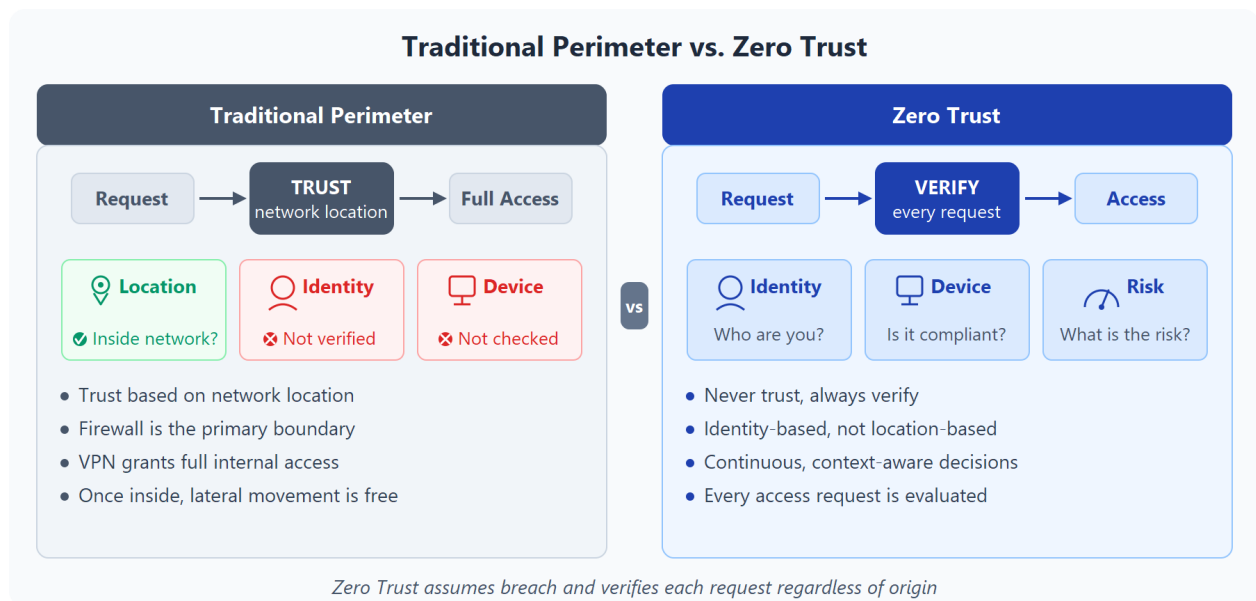
Traditionally, perimeter-based networks were restricted, protected, and generally assumed safe. Only managed computers could join the network, VPN access was tightly controlled, and personal devices were frequently restricted or blocked.

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

Example in practice

Suppose a user signs in from an unmanaged device on a public network. A Zero Trust approach still evaluates identity signals, device state, and risk conditions before granting access. You might allow access to low-risk apps, require MFA for sensitive systems, or block sign-in entirely when risk is too high.

At a fundamentals level, Zero Trust means access decisions are continuous and context-aware, not based only on where the request originates.



8. Describe defense-in-depth

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/8-describe-defense-depth>

Describe defense-in-depth

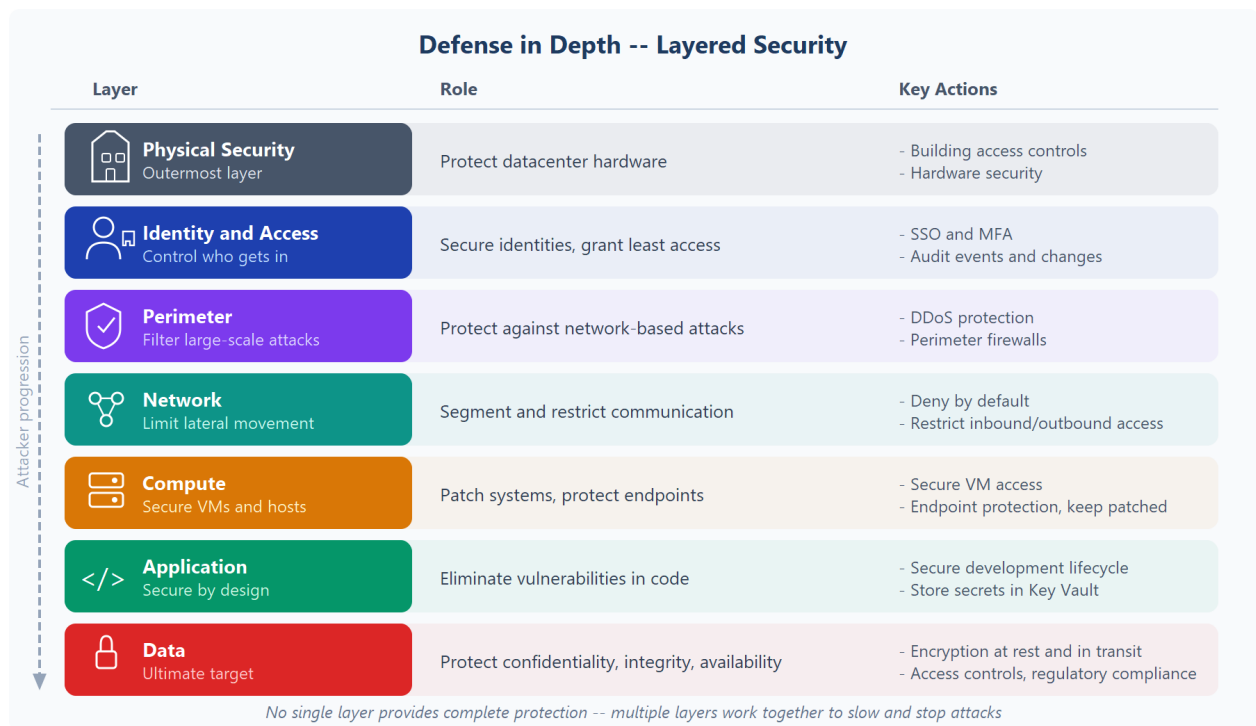
Completed

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.

Layers of defense-in-depth

You can visualize defense-in-depth as a set of layers, with the data to be secured at the center and all the other layers functioning to protect that central data layer.



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:

- The physical security layer is the first line of defense to protect computing hardware in the datacenter.
- The identity and access layer controls access to infrastructure and change control.
- 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.
- The network layer limits communication between resources through segmentation and access controls.
- The compute layer secures access to virtual machines.
- The application layer helps ensure that applications are secure and free of security vulnerabilities.
- The data layer controls access to operational and customer data that you need to protect.

These layers provide a guideline for you to help make security configuration decisions in all of the layers of your applications.

Azure provides security tools and features at every level of the defense-in-depth concept. Let's take a closer look at each layer:

Physical security

Physically securing access to buildings and controlling access to computing hardware within the datacenter are the first line of defense.

With physical security, the intent is to provide physical safeguards against access to assets. These safeguards ensure that other layers can't be bypassed, and loss or theft is handled appropriately. Microsoft uses various physical security mechanisms in its cloud datacenters.

Identity and access

The identity and access layer is all about ensuring that identities are secure, that access is granted only to what's needed, and that sign-in events and changes are logged.

At this layer, it's important to:

- Control access to infrastructure and change control.
- Use single sign-on (SSO) and multifactor authentication.
- Audit events and changes.

Perimeter

The network perimeter protects from network-based attacks against your resources. Identifying these attacks, eliminating their impact, and alerting you when they happen are important ways to keep your network secure.

At this layer, it's important to:

- Use DDoS protection to filter large-scale attacks before they can affect the availability of a system for users.
- Use perimeter firewalls to identify and alert on malicious attacks against your network.

Network

At this layer, the focus is on limiting the network connectivity across all your resources to allow only what's required. By limiting this communication, you reduce the risk of an attack spreading to other systems in your network.

At this layer, it's important to:

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

Compute

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

At this layer, it's important to:

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

Application

Integrating security into the application development lifecycle helps reduce the number of vulnerabilities introduced in code. Every development team should ensure that its applications are secure by default.

At this layer, it's important to:

- Ensure that applications are secure and free of vulnerabilities.
- Store sensitive application secrets in a secure storage medium, such as Azure Key Vault.
- Make security a design requirement for all application development.

Data

Those who store and control access to data are responsible for ensuring that it's properly secured. Often, regulatory requirements dictate the controls and processes that must be in place to ensure the confidentiality, integrity, and availability of the data.

At this layer, it's also important to protect data with encryption at rest and encryption in transit. Azure services provide encryption capabilities, and Azure Key Vault can help secure and manage the keys and secrets used by your applications.

In almost all cases, attackers are after data:

- Stored in a database.
- Stored on disk inside virtual machines.
- Stored in software as a service (SaaS) applications, such as Office 365.
- Managed through cloud storage.

9. Describe encryption and key management in Azure

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/9a-describe-encryption-key-management>

Describe encryption and key management in Azure

Completed

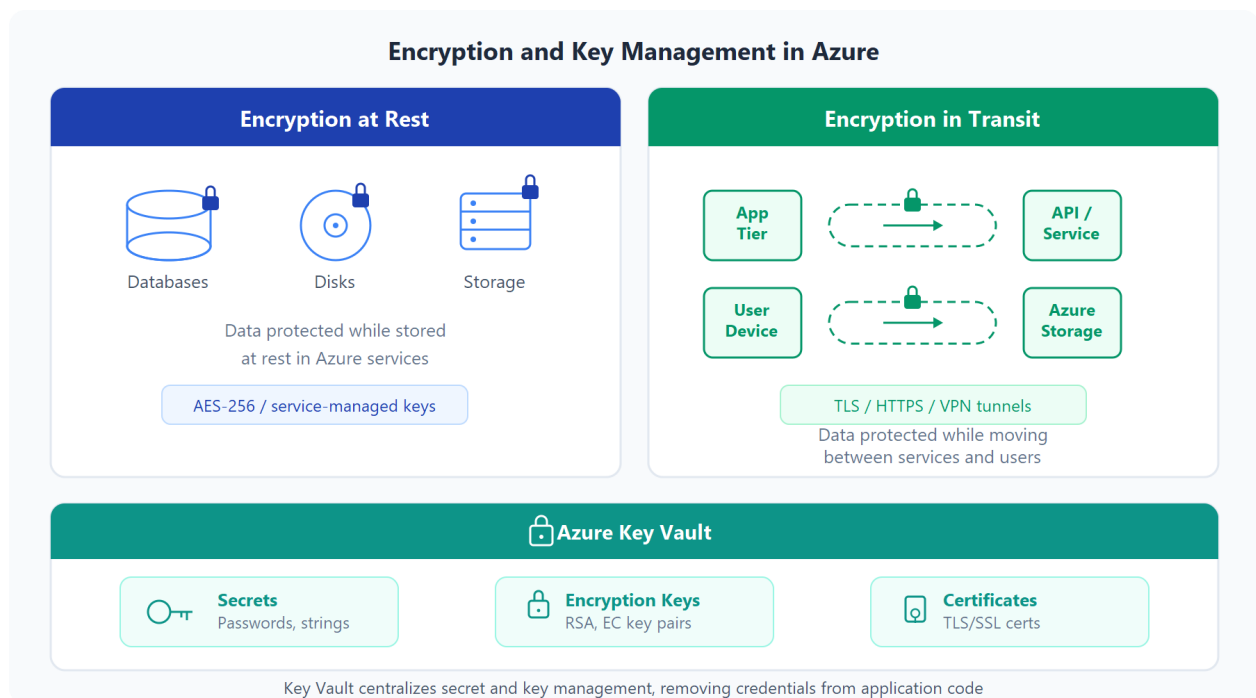
Encryption helps protect data confidentiality by making data unreadable to unauthorized users.

Encryption at rest and in transit

In Azure, encryption is commonly discussed in two forms:

- Encryption at rest protects data when it is stored, such as in databases, disks, and storage accounts.
- Encryption in transit protects data while it moves between services, applications, and users.

A strong security posture generally includes both.



Practical example

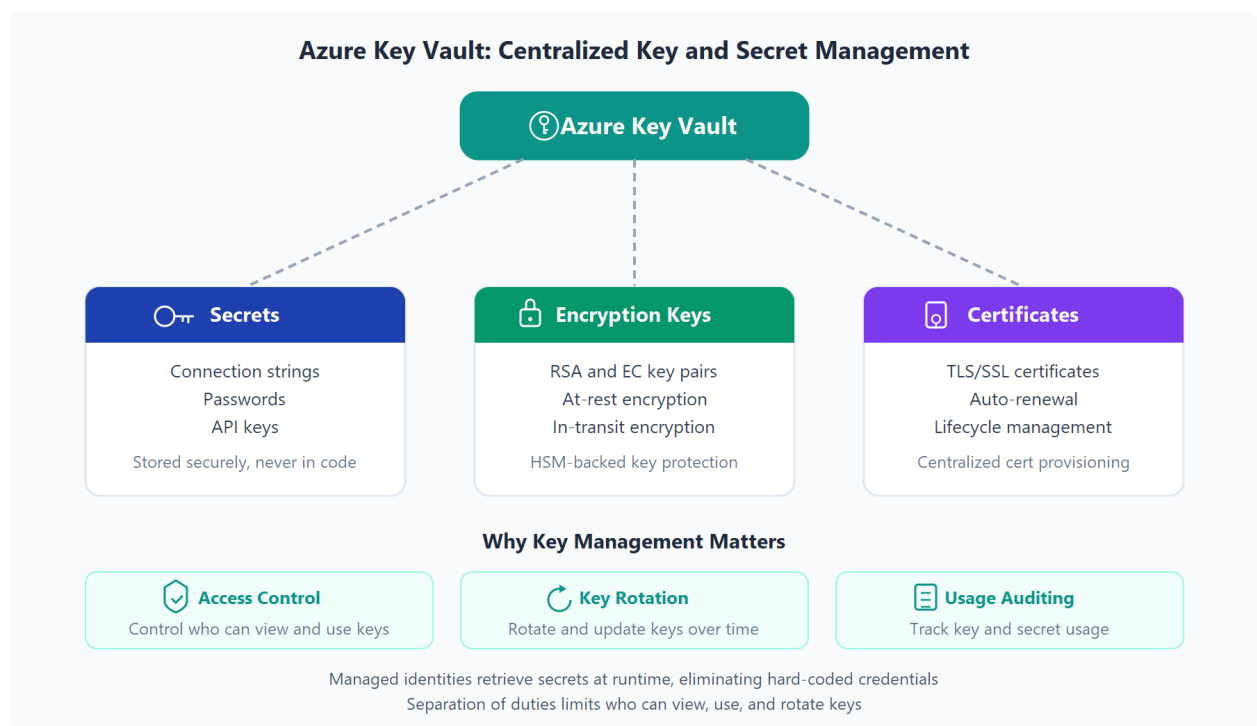
Consider a line-of-business application that stores customer records in Azure Storage and Azure SQL. Data should be encrypted while stored in those services and while moving between application tiers, APIs, and user devices.

Key management with Azure Key Vault

Azure Key Vault is a service for securely storing and controlling access to:

- Secrets (such as connection strings and passwords)
- Encryption keys
- Certificates

Using Key Vault helps centralize secret and key management instead of storing sensitive values directly in application code or configuration files.



Why key management matters

Key management supports security and compliance goals by helping you:

- Control who can access keys and secrets
- Rotate and update keys over time
- Audit key and secret usage

You can also separate duties by limiting who can view, use, and rotate keys. This approach helps reduce risk and supports compliance reporting.

Another common practice is to set key rotation policies and alerting so teams are notified before keys expire. This helps avoid service disruption while maintaining secure key hygiene.

When applications retrieve secrets from Key Vault at runtime by using managed identities, teams can reduce hard-coded credentials and improve overall secret governance.

At a fundamentals level, remember that Azure Key Vault is the primary Azure service for secure key and secret storage.

10. Describe Microsoft Defender for Cloud

Describe Microsoft Defender for Cloud

Completed

Defender for Cloud is a security posture management and threat protection service. It monitors cloud, on-premises, hybrid, and multicloud resources and provides recommendations and alerts to improve security posture.

Defender for Cloud helps you harden resources, track risk, and respond to threats. In Azure, it's natively integrated and can be enabled quickly.

Coverage across cloud and hybrid environments

Because Defender for Cloud is Azure-native, many Azure services can be monitored without extra deployment. For hybrid and multicloud estates, Defender for Cloud extends visibility so security teams can work from one control plane.

When needed, Defender for Cloud can deploy data collection components for security insights. Azure Arc can extend Microsoft Defender plans to non-Azure machines, and Cloud Security Posture Management (CSPM) capabilities can assess multicloud resources agentlessly.

Azure-native protections

Defender for Cloud helps you detect threats across:

- Azure PaaS services - Detects threats across services such as App Service, Azure SQL, and Azure Storage.
- Azure data services - Provides data security assessments and recommendations for services such as Azure SQL and Storage.
- Networks - Helps reduce brute-force exposure through controls such as just-in-time VM access and restrictive port policies.

Defend your hybrid resources

In addition to Azure coverage, Defender for Cloud can protect non-Azure servers in hybrid environments and prioritize alerts based on your environment.

To extend protection to on-premises machines, deploy Azure Arc and enable Defender for Cloud's enhanced security features.

Defend resources running on other clouds

Defender for Cloud can also protect resources in other clouds, including AWS and GCP.

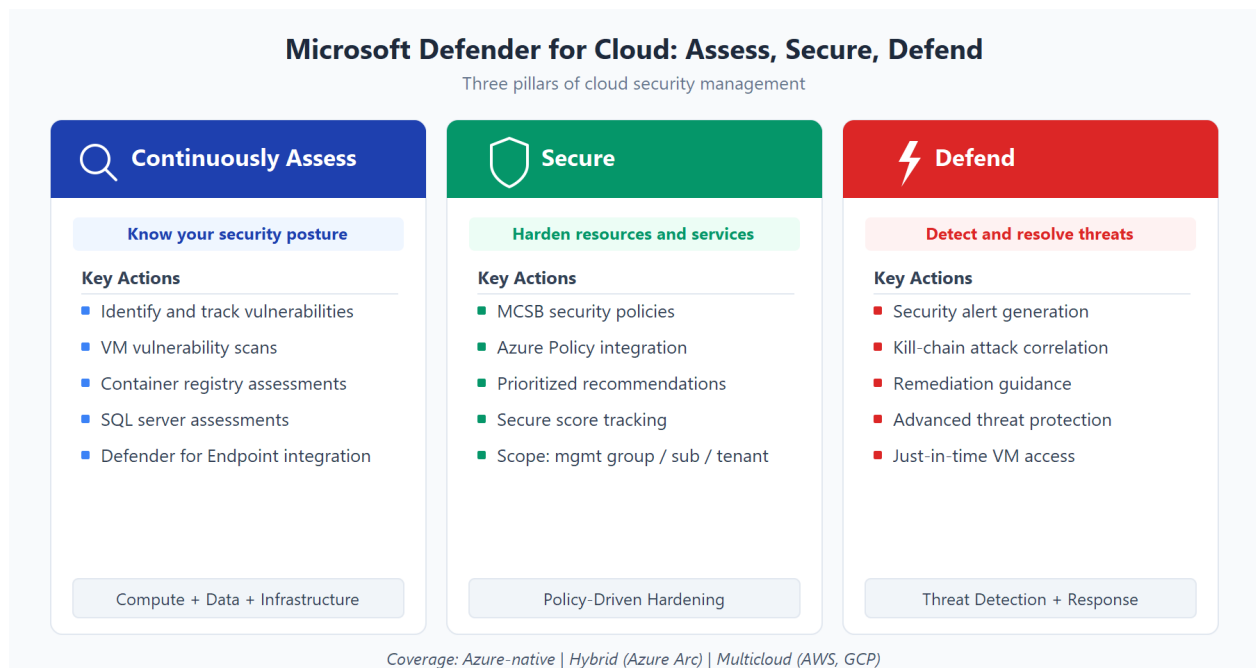
For connected AWS environments, Defender for Cloud can:

- Extend CSPM recommendations and compliance assessments to AWS resources.
- Extend Defender for Containers protections to Amazon EKS clusters.
- Extend Defender for Servers protections to EC2 instances.

Assess, Secure, and Defend

Defender for Cloud fills three vital needs as you manage the security of your resources and workloads in the cloud and on-premises:

- Continuously assess – Know your security posture. Identify and track vulnerabilities.
- Secure – Harden resources and services with Microsoft cloud security benchmark (MCSB).
- Defend – Detect and resolve threats to resources, workloads, and services.



Continuously assess

Defender for Cloud helps you continuously assess your environment. It includes vulnerability assessment for virtual machines, container registries, and SQL servers.

Microsoft Defender for servers includes automatic, native integration with Microsoft Defender for Endpoint. With this integration enabled, you'll have access to the vulnerability findings from Microsoft Defender Vulnerability Management.

Together, these tools provide regular vulnerability visibility across compute, data, and infrastructure from a unified experience.

Secure

To secure workloads, you need policies that align to your environment. Defender for Cloud builds on Azure Policy, so controls can be scoped at management group, subscription, or tenant levels.

As new resources are deployed, Defender for Cloud evaluates them against best practices and surfaces prioritized recommendations. These recommendations align to Microsoft cloud security benchmark (MCSB) guidance.

Defender for Cloud groups recommendations into security controls and calculates a secure score so teams can quickly understand posture and prioritize improvements.

Defend

The first two areas focused on assessing, monitoring, and maintaining your environment. Defender for Cloud also defends your environment by providing security alerts and advanced threat protection features.

Security alerts

When Defender for Cloud detects a threat in any area of your environment, it generates a security alert. Security alerts:

- Describe details of the affected resources
- Suggest remediation steps
- Provide, in some cases, an option to trigger a logic app in response

Whether an alert is generated by Defender for Cloud or received from an integrated security product, you can export it. Defender for Cloud also uses kill-chain analysis to automatically correlate related alerts, helping you see the full story of an attack — where it started, which resources were affected, and what impact it had.

Advanced threat protection

Defender for Cloud provides advanced threat protection for resources such as virtual machines, SQL databases, containers, web applications, and networks. Protections include just-in-time VM access and adaptive application controls.

11. Module assessment

Module assessment

Completed

12. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-azure-identity-access-security/11-summary>

Summary

Completed

In this module, you learned about Azure identity, access, and security services and tools. You covered authentication methods, including which ones are more secure. You learned about restricting access based on a role to help create a more secure environment. You also reviewed encryption concepts and key management options in Azure. And, you learned about the Defense In Depth and Zero Trust models.

Learning objectives

You should now be able to:

- Describe directory services in Azure, including Microsoft Entra ID and Microsoft Entra Domain Services.
- Describe authentication methods in Azure, including single sign-on (SSO), multifactor authentication (MFA), and passwordless.
- Describe external identities and guest access in Azure.
- Describe Microsoft Entra Conditional Access.
- Describe Azure Role Based Access Control (RBAC).
- Describe the concept of Zero Trust.
- Describe the purpose of the defense in depth model.
- Describe encryption concepts and key management options in Azure.
- Describe the purpose of Microsoft Defender for Cloud.

Additional resources

The following resources provide more information on topics in this module or related to this module. [Microsoft Certified: Security, Compliance, and Identity Fundamentals](#) is an entire certification, with associated training, dedicated to helping you better understand and manage Security, Compliance, and identity.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Use one end-to-end scenario to show how SSO, MFA, Conditional Access, and RBAC work together in a Zero Trust design."
- "Explain the difference between Microsoft Entra ID, Microsoft Entra Domain Services, and external identities with practical examples."
- "Simulate a security incident and show how encryption, key management, defense in depth, and Microsoft Defender for Cloud reduce risk."