

Part 2: Describe core Azure services

Explore Azure compute services

Introduction

Imagine that you work as a development lead at Tailwind Traders, a company that specializes in hardware manufacturing. Your management team tells you that the company's website has been having a difficult time keeping up with the application demands. The team wants you to investigate a solution. The front-end web servers are operating near capacity during peak periods of the day, and you need to get a solution in place quickly. But there's a problem. You don't have any free servers to scale out your application.



You could ask to buy new equipment, but your department's budget is tight. You want to make a good impression with leadership, but you don't know how many servers are necessary for this project, and you don't want to buy more hardware than you need. Even if you were able to procure several servers, you'd need to invest a lot of time to set them up and install software.

Ideally, you would obtain the resources you need to do the work without too much administration and configure them to do the work. You'd also pay only for the compute resources you need while you're using them.

This scenario is exactly what you can do in Azure. You can create compute resources, configure them to do the work that's needed, and pay for only what you use.

Learning objectives

After completing this module, you'll be able to describe the benefits and usage of:

- Azure Virtual Machines
- Azure App Service
- Azure Container Instances
- Azure Kubernetes Service
- Azure Functions
- Azure Virtual Desktop

Prerequisites








- You should be familiar with basic computing concepts and terminology.
- Familiarity with cloud computing is helpful but isn't necessary.

Overview of Azure compute services

Azure compute is an on-demand computing service for running cloud-based applications. It provides computing resources such as disks, processors, memory, networking, and operating systems. The resources are available on-demand and can typically be made available in minutes or even seconds. You pay only for the resources you use, and only for as long as you're using them.

Azure supports a wide range of computing solutions for development and testing, running applications, and extending your datacentre. The service supports Linux, Windows Server, SQL Server, Oracle, IBM, and SAP. Azure also has many services that can run virtual machines (VMs). Each service provides different options depending on your requirements. Some of the most prominent services are:

- Azure Virtual Machines
- Azure Container Instances
- Azure App Service
- Azure Functions (or *serverless computing*)

Everything	
General	
Compute	COMPUTE (28)
Networking	
Storage	
Web	
Mobile	
Containers	
Databases	
Analytics	
	 Virtual machines ★
	 Virtual machine scale sets ★
	 Function App ★
	 App Services ★
	 Kubernetes services ★
	 Availability sets ★
	 Disks ★

Virtual machines

Virtual machines are software emulations of physical computers. They include a virtual processor, memory, storage, and networking resources. VMs host an operating system, and you can install and run software just like a physical computer. When using a remote desktop client, you can use and control the VM as if you were sitting in front of it.



With [Azure Virtual Machines](#), you can create and use VMs in the cloud. Virtual Machines provides infrastructure as a service (IaaS) and can be used in different ways. When you need total control over an operating system and environment, VMs are an ideal choice. Just like a physical computer, you can customize all the software running on the VM. This ability is helpful when you're running custom software or custom hosting configurations.



Virtual machine scale sets

[Virtual machine scale sets](#) are an Azure compute resource that you can use to deploy and manage a set of identical VMs. With all VMs configured the same, virtual machine scale sets are designed to support true auto scale. No pre-provisioning of VMs is required. For this reason, it's easier to build large-scale services targeting big compute, big data, and containerized workloads. As demand goes up, more VM instances can be added. As demand goes down, VM instances can be removed. The process can be manual, automated, or a combination of both.

Containers and Kubernetes



[Container Instances](#) and [Azure Kubernetes Service](#) are Azure compute resources that you can use to deploy and manage containers. Containers are lightweight, virtualized application environments. They're designed to be quickly created, scaled out, and stopped dynamically. You can run multiple instances of a containerized application on a single host machine.

App Service



With [Azure App Service](#), you can quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. You can meet rigorous performance, scalability, security, and compliance requirements while using a fully managed platform to perform infrastructure maintenance. App Service is a platform as a service (PaaS) offering.

Functions



[Functions](#) are ideal when you're concerned only about the code running your service and not the underlying platform or infrastructure. They're commonly used when you need to perform work in response to an event (often via a REST request), timer, or message from another Azure service, and when that work can be completed quickly, within seconds or less.

Decide when to use Azure Virtual Machines

One possible solution to Tailwind Traders' lack of physical servers is through the use of virtual machines (VMs).

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

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

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



You can create and provision a VM in minutes when you select a preconfigured VM image. Selecting an image is one of the most important decisions you'll make when you create a VM. An image is a template used to create a VM. These templates already include an OS and often other software, like development tools or web hosting environments.

Examples of when to use VMs

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

an IaaS-based approach to disaster recovery. If a primary datacenter fails, you can create VMs running on Azure to run your critical applications and then shut them down when the primary datacenter becomes operational again.

Move to the cloud with VMs

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

Scale VMs in Azure

You can run single VMs for testing, development, or minor tasks. Or you can group VMs together to provide high availability, scalability, and redundancy. No matter what your uptime requirements are, Azure has several features that can meet them. These features include:

- Virtual machine scale sets
- Azure Batch

What are virtual machine scale sets?

Virtual machine scale sets let you create and manage a group of identical, load-balanced VMs. Imagine you're running a website that enables scientists to upload astronomy images that need to be processed. If you duplicated the VM, you'd normally need to configure an additional service to route requests between multiple instances of the website. Virtual machine scale sets could do that work for you.

Scale sets allow you to centrally manage, configure, and update a large number of VMs in minutes to provide highly available applications. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. With virtual machine scale sets, you can build large-scale services for areas such as compute, big data, and container workloads.

What is Azure Batch?

Azure Batch enables large-scale parallel and high-performance computing (HPC) batch jobs with the ability to scale to tens, hundreds, or thousands of VMs.

When you're ready to run a job, Batch does the following:

- Starts a pool of compute VMs for you.
- Installs applications and staging data.
- Runs jobs with as many tasks as you have.
- Identifies failures.
- Requeues work.
- Scales down the pool as work completes.

There might be situations in which you need raw computing power or supercomputer-level compute power. Azure provides these capabilities.

Decide when to use Azure App Service

In your research for Tailwind Traders, you've looked at different ways that you can virtualize your application. Another alternative is to deploy your application's front-end websites to Azure App Service, which makes it easy to respond to application demand.

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



This platform as a service (PaaS) environment allows you to focus on the website and API logic while Azure handles the infrastructure to run and scale your web applications.

Azure App Service costs

You pay for the Azure compute resources your app uses while it processes requests based on the App Service plan you choose. The App Service plan determines how much hardware is devoted to your host. For example, the plan determines whether it's dedicated or shared hardware and how much memory is reserved for it. There's even a *free* tier you can use to host small, low-traffic sites.

Types of app services

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

- Web apps
- API apps
- WebJobs
- Mobile apps

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

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

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

Web apps

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

API apps

Much like hosting a website, you can build REST-based web APIs by using your choice of language and framework. You get full Swagger support and the ability to package and publish your API in Azure Marketplace. The produced apps can be consumed from any HTTP- or HTTPS-based client.

WebJobs

You can use the WebJobs feature to run a program (.exe, Java, PHP, Python, or Node.js) or script (.cmd, .bat, PowerShell, or Bash) in the same context as a web app, API app, or mobile app. They can be scheduled or run by a trigger. WebJobs are often used to run background tasks as part of your application logic.

Mobile apps

Use the Mobile Apps feature of App Service to quickly build a back end for iOS and Android apps. With just a few clicks in the Azure portal, you can:

- Store mobile app data in a cloud-based SQL database.
- Authenticate customers against common social providers, such as MSA, Google, Twitter, and Facebook.
- Send push notifications.
- Execute custom back-end logic in C# or Node.js.

On the mobile app side, there's SDK support for native iOS and Android, Xamarin, and React native apps.

Decide when to use Azure Container Instances or Azure Kubernetes Service

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

What are containers?

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

Compare virtual machines to containers

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

Manage containers

Containers are managed through a container orchestrator, which can start, stop, and scale out application instances as needed. There are two ways to manage both Docker and Microsoft-based containers in Azure: Azure Container Instances and Azure Kubernetes Service (AKS).

Azure Container Instances



[Azure Container Instances](#) offers the fastest and simplest way to run a container in Azure without having to manage any virtual machines or adopt any additional services. It's a platform as a service (PaaS) offering that allows you to upload your containers, which it runs for you.



Azure Kubernetes Service

The task of automating, managing, and interacting with a large number of containers is known as orchestration. [Azure Kubernetes Service](#) is a complete orchestration service for containers with distributed architectures and large volumes of containers.

What is Kubernetes?

The following video discusses some important details about Kubernetes container orchestration.

Use containers in your solutions

Containers are often used to create solutions by using a *microservice architecture*. This architecture is where you break solutions into smaller, independent pieces. For example, you might split a website into a container hosting your front end, another hosting your back end, and a third for storage. This split allows you to separate portions of your app into logical sections that can be maintained, scaled, or updated independently.

Imagine your website back-end has reached capacity but the front end and storage aren't being stressed. You could:

- Scale the back end separately to improve performance.
- Decide to use a different storage service.
- Replace the storage container without affecting the rest of the application.

What is a microservice?

Decide when to use Azure Functions

After consulting with several of your fellow developers at Tailwind Traders, you've determined that some of your application logic is event driven. In other words, for a large amount of time, your application is waiting for a particular input before it performs any processing. To reduce your costs, you want to avoid having to pay for the time that your application is waiting for input. With that in mind, you've decided to investigate Azure Functions to see if it can help.

Serverless computing is the abstraction of servers, infrastructure, and operating systems. With serverless computing, Azure takes care of managing the server infrastructure and the allocation and deallocation of resources based on demand. Infrastructure isn't your responsibility. Scaling and performance are handled automatically. You're billed only for the exact resources you use. There's no need to even reserve capacity.

Serverless computing includes the abstraction of servers, an event-driven scale, and micro-billing:

- **Abstraction of servers:** Serverless computing abstracts the servers you run on. You never explicitly reserve server instances. The platform manages that for you. Each function execution can run on a different compute instance. This execution context is transparent to the code. With serverless architecture, you deploy your code, which then runs with high availability.
- **Event-driven scale:** Serverless computing is an excellent fit for workloads that respond to incoming events. Events include triggers by:
 - Timers, for example, if a function needs to run every day at 10:00 AM UTC.
 - HTTP, for example, API and webhook scenarios.
 - Queues, for example, with order processing.
 - And much more.

Instead of writing an entire application, the developer authors a function, which contains both code and metadata about its triggers and bindings. The platform automatically schedules the function to run and scales the number of compute instances based on the rate of incoming events. Triggers define how a function is invoked. Bindings provide a declarative way to connect to services from within the code.

- **Micro-billing:** Traditional computing bills for a block of time like paying a monthly or annual rate for website hosting. This method of billing is convenient but isn't always cost effective. Even if a customer's website gets only one hit a day, they still pay for a full day's worth of availability. With serverless computing, they pay only for the time their code runs. If no active function executions occur, they're not charged. For example, if the code runs once a day for two minutes, they're charged for one execution and two minutes of computing time.

Serverless computing in Azure

Azure has two implementations of serverless compute:

- **Azure Functions:** Functions can execute code in almost any modern language.
- **Azure Logic Apps:** Logic apps are designed in a web-based designer and can execute logic triggered by Azure services without writing any code.

Azure Functions

When you're concerned only about the code running your service, and not the underlying platform or infrastructure, using Azure Functions is ideal. 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're a solid choice when demand is variable. For example, you might receive messages from an IoT solution that's used to monitor a fleet of delivery vehicles. You'll likely have more data arriving during business hours.

Using a virtual machine-based approach, you'd incur costs even when the virtual machine is idle. With functions, Azure runs your code when it's triggered and automatically deallocates resources when the function is finished. In this model, you're only charged for the CPU time used while your function runs.

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

Functions are a key component of serverless computing. They're also a general compute platform for running any type of code. If the needs of the developer's app change, you can deploy the project in an environment that isn't serverless. This flexibility allows you to manage scaling, run on virtual networks, and even completely isolate the functions.

Azure Logic Apps

Logic apps are similar to functions. Both enable you to trigger logic based on an event. Where functions execute code, logic apps execute *workflows* that are designed to automate business scenarios and are built from predefined logic blocks.

Every Azure logic app workflow starts with a trigger, which fires when a specific event happens or when newly available data meets specific criteria. Many triggers include basic scheduling capabilities, so developers can specify how regularly their workloads will run. Each time the trigger fires, the Logic Apps engine creates a logic app instance that runs the actions in the workflow. These actions can also include data conversions and flow controls, such as conditional statements, switch statements, loops, and branching.

You create logic app workflows by using a visual designer on the Azure portal or in Visual Studio. The workflows are persisted as a JSON file with a known workflow schema.

Azure provides more than 200 different connectors and processing blocks to interact with different services. These resources include the most popular enterprise apps. You can also build custom connectors and workflow steps if the service you need to interact with isn't covered. You then use

the visual designer to link connectors and blocks together. You pass data through the workflow to do custom processing, often all without writing any code.

As an example, let's say a ticket arrives in Zendesk. You could:

- Detect the intent of the message with cognitive services.
- Create an item in SharePoint to track the issue.
- Add the customer to your Dynamics 365 CRM system if they aren't already in your database.
- Send a follow-up email to acknowledge their request.

All of those actions could be designed in a visual designer, which makes it easy to see the logic flow. For this reason, it's ideal for a business analyst role.

Functions vs. Logic Apps

Functions and Logic Apps can both create complex orchestrations. An orchestration is a collection of functions or steps that are executed to accomplish a complex task.

- With Functions, you write code to complete each step.
- With Logic Apps, you use a GUI to define the actions and how they relate to one another.

You can mix and match services when you build an orchestration, calling functions from logic apps and calling logic apps from functions. Here are some common differences between the two.

	Functions	Logic Apps
State	Normally stateless, but Durable Functions provide state.	Stateful.
Development	Code-first (imperative).	Designer-first (declarative).
Connectivity	About a dozen built-in binding types. Write code for custom bindings.	Large collection of connectors. Enterprise Integration Pack for B2B scenarios. Build custom connectors.
Actions	Each activity is an Azure function. Write code for activity functions.	Large collection of ready-made actions.
Monitoring	Azure Application Insights.	Azure portal, Log Analytics.
Management	REST API, Visual Studio.	Azure portal, REST API, PowerShell, Visual Studio.
Execution context	Can run locally or in the cloud.	Runs only in the cloud.

Decide when to use Azure Virtual Desktop

In addition to the challenges that Tailwind Traders has been facing with application scale, your manager has asked you to put together a new development team of remote workers.

This task would normally require setting up several new computers with all of the requisite development tools for your new team. Then you would need to ship them to the respective developers. The time to procure, set up, and ship each of these computers would be costly. Also, all of your new developers have their own computing devices that are running a mixture of Windows, Android, and macOS operating systems.

You want to find a way to expedite the deployment process for your remote workers. You also want to keep your management costs to a minimum. With that in mind, you want to see how Azure Virtual Desktop can help your organization.

What is Azure Virtual Desktop?

Azure Virtual Desktop is a desktop and application virtualization service that runs on the cloud. It enables your users to use a cloud-hosted version of Windows from any location. Azure Virtual Desktop works across devices like Windows, Mac, iOS, Android, and Linux. It works with apps that you can use to access remote desktops and apps. You can also use most modern browsers to access Azure Virtual Desktop-hosted experiences.

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

Why should you use Azure Virtual Desktop?

Provide the best user experience

Users have the freedom to connect to Azure Virtual Desktop with any device over the internet. They use a Azure Virtual Desktop client to connect to their published Windows desktop and applications. This client could either be a native application on the device or the Azure Virtual Desktop HTML5 web client.

You can make sure your session host virtual machines (VMs) run near apps and services that connect to your datacenter or the cloud. This way your users stay productive and don't encounter long load times.

User sign-in to Azure Virtual Desktop is fast because user profiles are containerized by using FSLogix. At sign-in, the user profile container is dynamically attached to the computing environment. The user profile is immediately available and appears in the system exactly like a native user profile.

You can provide individual ownership through personal (persistent) desktops. For example, you might want to provide personal remote desktops for members of an engineering team. Then they can add or remove programs without impacting other users on that remote desktop.

Enhance security

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

With Azure Virtual Desktop, the data and apps are separated from the local hardware. Azure Virtual Desktop runs them instead on a remote server. The risk of confidential data being left on a personal device is reduced.

User sessions are isolated in both single and multi-session environments.

Azure Virtual Desktop also improves security by using reverse connect technology. This connection type is more secure than the Remote Desktop Protocol. We don't open inbound ports to the session host VMs.

What are some key features of Azure Virtual Desktop?

Simplified management

Azure Virtual Desktop is an Azure service, so it will be familiar to Azure administrators. You use Azure AD and RBACs to manage access to resources. With Azure, you also get tools to automate VM deployments, manage VM updates, and provide disaster recovery. As with other Azure services, Azure Virtual Desktop uses Azure Monitor for monitoring and alerts. This standardization lets admins identify issues through a single interface.

Performance management

Azure Virtual Desktop gives you options to load balance users on your VM host pools. *Host pools* are collections of VMs with the same configuration assigned to multiple users. For the best performance, you can configure load balancing to occur as users sign in (breadth mode). With breadth mode, users are sequentially allocated across the host pool for your workload. To save costs, you can configure your VMs for depth mode load balancing where users are fully allocated on one VM before moving to the next. Azure Virtual Desktop provides tools to automatically provision additional VMs when incoming demand exceeds a specified threshold.

Multi-session Windows 10 deployment

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

How can you reduce costs with Azure Virtual Desktop?

Bring your own licenses

Azure Virtual Desktop is available to you at no additional cost if you have an eligible Microsoft 365 license. Just pay for the Azure resources used by Azure Virtual Desktop.

- Bring your eligible Windows or Microsoft 365 license to get Windows 10 Enterprise and Windows 7 Enterprise desktops and apps at no additional cost.
- If you're an eligible Microsoft Remote Desktop Services Client Access License customer, Windows Server Remote Desktop Services desktops and apps are available at no additional cost.

Save on compute costs

Buy one-year or three-year Azure Reserved Virtual Machine Instances to save you up to 72 percent versus pay-as-you-go pricing. You can pay for a reservation up front or monthly. Reservations provide a billing discount and don't affect the runtime state of your resources.

[A] Explore Azure networking services

In this module, you'll take a look at several of the core networking resources that are available in Azure. You'll learn about Azure Virtual Network, which you can configure into a customized network environment that meets your company's needs. You'll also learn how you can use Azure VPN Gateway and Azure ExpressRoute to create secure communication tunnels between your company's different locations.

Learning objectives

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

- Describe the core networking resources that are available in Azure.
- Describe the benefits and usage of Virtual Network, VPN Gateway, and ExpressRoute.

Introduction

Suppose your company, Tailwind Traders, has migrated some applications to the cloud and is architecting new ones. The servers that host Tailwind Traders' customer and product data are based in Silicon Valley. Your company also has several branch offices located in different geographic regions. As part of your migration strategy, your company needs to determine the correct approach to configure its network infrastructure.



To help save costs, you convince your team to move your website and several of your other networked resources to the cloud. With that in mind, you'll need to provide secure access to private company data for each of its branch locations. You want to know how Azure can help you manage your network more effectively. As it turns out, managing networks on Azure isn't entirely different from managing on-premises networks.

In this module, you'll learn about the different Azure networking options and the scenarios in which each is appropriate.

Learning objectives

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

- Describe the core networking resources that are available in Azure.
- Describe the benefits and usage of Azure Virtual Network, Azure VPN Gateway, and Azure ExpressRoute.

Prerequisites

- You should be familiar with basic network concepts and terminology.
- Familiarity with cloud computing is helpful but isn't necessary.

A.1 : Azure Virtual Network fundamentals

Tailwind Traders has an on-premises datacenter that you plan to keep, but you want to use Azure to offload peak traffic by using virtual machines (VMs) hosted in Azure. You want to keep your existing IP addressing scheme and network appliances while ensuring that any data transfer is secure.

Using Azure Virtual Network for your virtual networking can help you reach your goals.

What is Azure virtual networking?

Azure virtual networks 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 links 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

Network configurations for virtual machines:

Isolation and segmentation

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

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

Internet communications

A VM in Azure can connect to the internet by default. You can enable incoming connections from the internet by assigning a public IP address to the VM or by putting the VM behind a public load balancer. For VM management, you can connect via the Azure CLI, Remote Desktop Protocol, or Secure Shell.

Communicate between Azure resources

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

- **Virtual networks** 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** You can use service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts. This approach enables you to link multiple Azure resources to virtual networks to improve security and provide optimal routing between resources.

Communicate with on-premises resources

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

- **Point-to-site virtual private networks** The typical approach to a virtual private network (VPN) connection is from a computer outside your organization, back into your corporate network. In this case, the client computer initiates an encrypted VPN connection to connect that computer to the Azure virtual network.
- **Site-to-site virtual private networks** A site-to-site VPN links 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** For environments where you need greater bandwidth and even higher levels of security, Azure ExpressRoute is the best approach. ExpressRoute provides a dedicated private connectivity to Azure that doesn't travel over the internet. (You'll learn more about ExpressRoute in a separate unit later in this module.)

Route network traffic

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

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

Filter network traffic

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

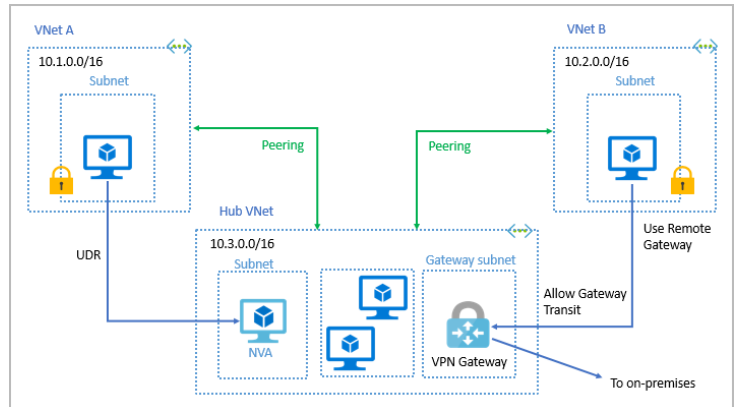
- **Network security groups** A network security group is an Azure resource that can contain multiple inbound and outbound security rules. You can define these rules to allow or block traffic, based on factors such as source and destination IP address, port, and protocol.

- **Network virtual appliances** A network virtual appliance is a specialized VM 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 enables resources in each virtual network to communicate with each other. These virtual networks can be in separate regions, which allows you to create a global interconnected network through Azure.

User-defined routes (UDR) are a significant update to Azure's Virtual Networks that allows for greater control over network traffic flow. This method allows network administrators to control the routing tables between subnets within a VNet, as well as between VNets.



A.2 : Azure Virtual Network settings

You can create and configure Azure virtual networks from the Azure portal, Azure PowerShell, Azure CLI, Azure Cloud Shell or an ARM template.

Create a virtual network

When you create an Azure virtual network, you configure a number of basic settings. You'll have the option to configure advanced settings, such as multiple subnets, distributed denial of service (DDoS) protection, Bastion host, Azure firewall, service endpoints, and use of a NAT gateway.

You'll configure the following settings for a basic virtual network:

[Home](#) > [Virtual networks](#) >

Create virtual network ...

[Basics](#) [IP Addresses](#) [Security](#) [Tags](#) [Review + create](#)

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

Project details

Subscription * ⓘ
 Resource group * ⓘ
[Create new](#)

Instance details

Name *
 Region *

[Review + create](#)

[< Previous](#)

[Next : IP Addresses >](#)

[Download a template for automation](#)

- **Subscription** This option only applies if you have multiple subscriptions to choose from.
- **Resource group** Like any other Azure resource, a virtual network needs to exist in a resource group. You can either select an existing resource group or create a new one.

- **Network name** The network name must be unique in your subscription, but it doesn't need to be globally unique. Make the name a descriptive one that's easy to remember and identified from other virtual networks.
- **Region** Select the region where you want the virtual network to exist.
- **Address space** When you set up a virtual network, you define the internal address space in Classless Interdomain Routing (CIDR) format. This address space needs to be unique within your subscription and any other networks that you connect to. Let's assume you choose an address space of 10.0.0.0/24 for your first virtual network. The addresses defined in this address space range from 10.0.0.1 to 10.0.0.254. You then create a second virtual network and choose an address space of 10.0.0.0/8. The addresses in this address space range from 10.0.0.1 to 10.255.255.254. Some of the addresses overlap and can't be used for the two virtual networks. But you can use 10.0.0.0/16, with addresses that range from 10.0.0.1 to 10.0.255.254, and 10.1.0.0/16, with addresses that range from 10.1.0.1 to 10.1.255.254. You can assign these address spaces to your virtual networks because there's no address overlap.

Home > Virtual networks >

Create virtual network

Basics IP Addresses Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space

10.0.0.0/16

☐ Add IPv6 address space

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

+ Add subnet - Remove subnet

Subnet name	Subnet address range	NAT gateway
default	10.0.0.0/24	-

i Use of a NAT gateway is recommended for outbound internet access from a subnet. You can deploy a NAT gateway and assign it to a subnet after you create the virtual network. [Learn more](#)

[Review + create](#) < Previous Next: Security > [Download a template for automation](#)

Edit subnet

Subnet name *

default

Subnet address range * ⓘ

10.0.0.0/24

10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

NAT GATEWAY

Simplify connectivity to the internet using a network address translation gateway. Outbound connectivity is possible without a load balancer or public IP addresses attached to your virtual machines. [Learn more](#)

NAT gateway

None

SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ

0 selected

[Save](#) [Cancel](#)

Note

You can add address spaces after you create the virtual network.

- **Subnet** Within each virtual network address range, you can create one or more subnets that partition the virtual network's address space. Routing between subnets will then depend on the default traffic routes. You also can define custom routes. Alternatively, you can define one subnet that encompasses all the virtual networks' address ranges.

Note

Subnet names must begin with a letter or number and end with a letter, number, or underscore. They may contain only letters, numbers, underscores, periods, or hyphens.

- **Service endpoints** Here, you enable service endpoints. Then you select from the list which Azure service endpoints you want to enable. Options include Azure Cosmos DB, Azure Service Bus, Azure Key Vault, and so on.

- **NAT gateway** A NAT gateway is a fully managed and highly resilient Network Address Translation (NAT) service. You can configure a subnet to use a static outbound IP address when accessing the internet. For more information about NAT gateway, see [Azure Virtual Network NAT overview](#)

[Home](#) > [Virtual networks](#) >

Create virtual network ...

Basics IP Addresses **Security** Tags Review + create

BastionHost ⓘ	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Bastion name *	BastionHost ✓
AzureBastionSubnet address space *	e.g. 10.0.0.0/24
Public IP address *	Choose public IP address ▼ Create new
DDoS Protection Standard ⓘ	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
I know my resource ID	<input type="checkbox"/>
DDoS protection plan *	No DDoS protection plan was found. ▼
Firewall ⓘ	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Firewall name *	AzureFirewall ✓
Firewall subnet address space *	10.0.1.0/24 ✓ <small>10.0.1.0 - 10.0.1.255 (256 addresses)</small>
Public IP address *	Choose public IP address ▼ Create new

[Review + create](#)

[< Previous](#)

[Next : Tags >](#)

[Download a template for automation](#)

- **BastionHost** You can select to enable or disable Azure Bastion in your virtual network. Azure Bastion service provides a secure and seamless RDP/SSH connectivity to your virtual machines directly in the Azure portal over SSL. For more information on Azure Bastion, see [Azure Bastion overview](#).
- **DDoS Protection Standard** You can select to enable or disable Standard DDoS protection. The Standard DDoS protection is a premium service. For more information on Standard DDoS protection, see [Azure DDoS protection Standard overview](#).
- **Firewall** You can enable or disable Azure Firewall. Azure Firewall service is managed cloud-based network security service that protects your Azure Virtual Network resources. For more information on Azure Firewall, see [Azure Firewall overview](#)

After you've configured these settings, select **Review + Create** and then select **Create** when validation is passed.

Define additional settings

After you create a virtual network, you can then define further settings. These include:

default

myVirtualNetwork

Name
default

Subnet address range * ⓘ
10.0.0.0/24
10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

☐ Add IPv6 address space ⓘ

NAT gateway ⓘ
None

Network security group
None

Route table
None

SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ
0 selected

SUBNET DELEGATION

Delegate subnet to a service ⓘ
None

Save Cancel

- **Network security group** Network security groups have security rules that enable you to filter the type of network traffic that can flow in and out of virtual network subnets and network interfaces. You create the network security group separately. Then you associate it with each subnet in the virtual network.
- **Route table** Azure automatically creates a route table for each subnet within an Azure virtual network and adds system default routes to the table. You can add custom route tables to modify traffic between subnets and virtual networks.
- **Subnet Delegation** You can designate the subnet to be used by a dedicate service.

You can also amend the service endpoints and NAT gateway configuration.

Configure virtual networks

After you've created a virtual network, you can change any further settings on the **Virtual network** pane in the Azure portal. Alternatively, you can use PowerShell commands or commands in Cloud Shell to make changes.

You can then review and change settings in further subpanes. These settings include:

Home > myVirtualNetwork Virtual network

Search (Ctrl+/) Refresh Move Delete Give feedback

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Address space
- Connected devices
- Subnets
- DDoS protection
- Firewall
- Security
- Network manager
- DNS servers
- Peerings
- Service endpoints
- Private endpoints
- Properties
- Locks

Essentials

Resource group (move) myVirtualNetworkRG Address space 10.0.0.0/16

Location West US DNS servers Azure provided DNS service

Subscription (move) Azure Subscription

Subscription ID abcdef12-3456-7890-abcd-ef1234567890

Tags (edit) Click here to add tags

Connected devices

Search connected devices

Device ↑↓	Type ↑↓	IP Address ↑↓	Subnet ↑↓
No results.			

- **Address spaces:** You can add additional address spaces to the initial definition.

- **Connected devices:** View a list of all connected host in the virtual network.
- **Subnets:** You can add additional subnets.
- **DDos protection:** You can enable or disable the Standard DDos protection plan.
- **Firewall:** Configure firewall settings with Azure Firewall service for the virtual network.
- **Security:** Provides security recommendation you can apply to your virtual network.
- **Network Manager:** View connectivity and security admin configuration the virtual network is associated to.
- **DNS servers:** Configure internal or external DNS servers the resources in the virtual network will use.
- **Peerings:** Link virtual networks in peering arrangements.
- **Service endpoints:** Enable service endpoints and apply them to multiple subnets.
- **Private endpoints:** View a list of private endpoints enabled in a subnet.

You can also monitor and view metrics to troubleshoot your virtual networks.

Virtual networks are powerful and highly configurable mechanisms for connecting entities in Azure. You can connect Azure resources to one another or to resources you have on-premises. You can isolate, filter, and route your network traffic. Azure allows you to increase security where you feel you need it.

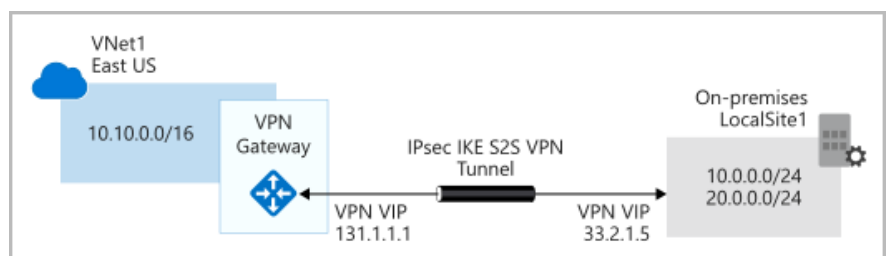
A 3 : Azure VPN Gateway fundamentals

VPNs use an encrypted tunnel within another network. They're 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.

For our Tailwind Traders scenario, VPNs can enable branch offices to share sensitive information between locations. For example, let's say that your offices on the East coast region of North America need to access your company's private customer data, which is stored on servers that are physically located in a West coast region. A VPN can connect your East coast offices to your West coast servers allowing your company to securely access your private customer data.

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, but you can use one gateway to connect to multiple locations, which includes other virtual networks or on-premises datacenters.

When you deploy a VPN gateway, you specify the VPN type: either *policy-based* or *route-based*. The main difference between these two types of VPNs is how traffic to be encrypted is specified. In Azure, both types of VPN gateways use a pre-shared key as the only method of authentication. Both types also rely on Internet Key Exchange (IKE) in either version 1 or version 2 and Internet Protocol Security (IPSec). IKE is used to set up a security association (an agreement of the encryption) between two endpoints. This association is then passed to the IPSec suite, which encrypts and decrypts data packets encapsulated in the VPN tunnel.

Policy-based VPNs

Policy-based VPN gateways specify statically the IP address of packets that should be encrypted through each tunnel. This type of device evaluates every data packet against those sets of IP addresses to choose the tunnel where that packet is going to be sent through.

Key features of policy-based VPN gateways in Azure include:

- Support for IKEv1 only.
- Use of *static routing*, where combinations of address prefixes from both networks control how traffic is encrypted and decrypted through the VPN tunnel. The source and destination of the tunneled networks are declared in the policy and don't need to be declared in routing tables.
- Policy-based VPNs must be used in specific scenarios that require them, such as for compatibility with legacy on-premises VPN devices.

Route-based VPNs

If defining which IP addresses are behind each tunnel is too cumbersome, route-based gateways can be used. With 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

Key features of route-based VPN gateways in Azure include:

- Supports IKEv2
- Uses any-to-any (wildcard) traffic selectors
- Can use *dynamic routing protocols*, where routing/forwarding tables direct traffic to different IPSec tunnels. In this case, the source and destination networks aren't statically defined as they are in policy-based VPNs or even in route-based VPNs with static routing. Instead, data packets

are encrypted based on network routing tables that are created dynamically using routing protocols such as Border Gateway Protocol (BGP).

VPN gateway sizes

The capabilities of your VPN gateway are determined by the SKU or size that you deploy. This table shows the main capabilities of each available SKU.

SKU	Site-to-site/Network-to-network tunnels	Aggregate throughput benchmark	Border Gateway Protocol support
Basic [See Note]	Maximum: 10	100 Mbps	Not supported
VpnGw1/Az	Maximum: 30	650 Mbps	Supported
VpnGw2/Az	Maximum: 30	1 Gbps	Supported
VpnGw3/Az	Maximum: 30	1.25 Gbps	Supported
VpnGw4/Az	Maximum: 100	5 Gbps Gbps	Supported
VpnGw5/Az	Maximum: 100	10 Gbps	Supported

Note

A Basic VPN gateway should only be used for Dev/Test workloads. In addition, it's unsupported to migrate from Basic to the VpnGW1/2/3/Az SKUs at a later time without having to remove the gateway and redeploy.

Deploy VPN gateways

Before you can deploy a VPN gateway, you'll need some Azure and on-premises resources.

Required Azure resources

You'll need these Azure resources before you can deploy an operational VPN gateway:

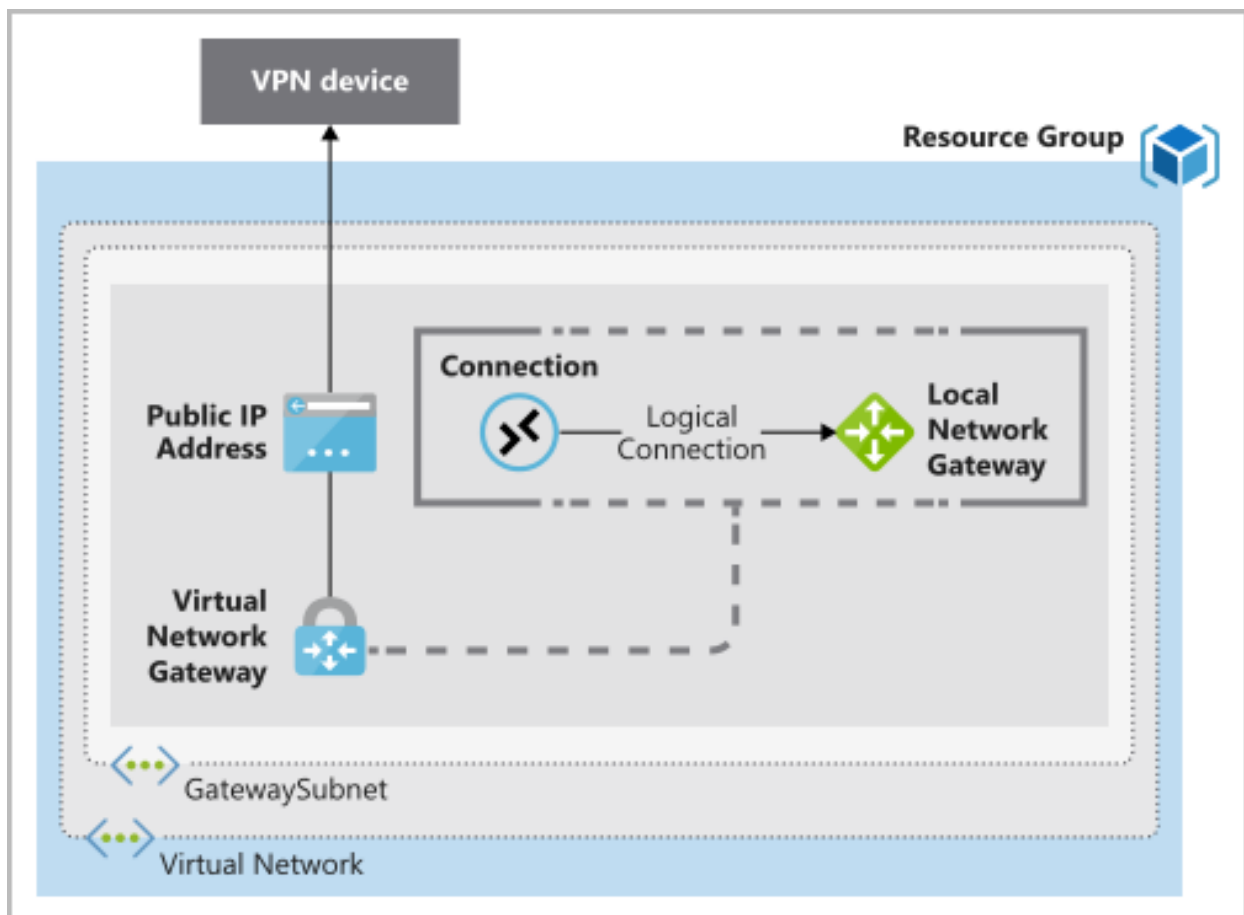
- **Virtual network.** Deploy a virtual network with enough address space for the additional subnet that you'll need for the VPN gateway. The address space for this virtual network must not overlap with the on-premises network that you'll be connecting to. You can deploy only one VPN gateway within a virtual network.
- **GatewaySubnet.** Deploy a subnet called GatewaySubnet for the VPN gateway. Use at least a **/27** address mask to make sure you have enough IP addresses in the subnet for future growth. You can't use this subnet for any other services.
- **Public IP address.** Create a Basic-SKU dynamic public IP address if you're using a non-zone-aware gateway. This address provides a public-routable IP address as the target for your on-premises VPN device. This IP address is dynamic, but it won't change unless you delete and re-create the VPN gateway.
- **Local network gateway.** Create a local network gateway to define the on-premises network's configuration, such as where the VPN gateway will connect and what it will

connect to. This configuration includes the on-premises VPN device's public IPv4 address and the on-premises routable networks. This information is used by the VPN gateway to route packets that are destined for on-premises networks through the IPsec tunnel.

- **Virtual network gateway.** Create the virtual network gateway to route traffic between the virtual network and the on-premises datacenter or other virtual networks. The virtual network gateway can be either a VPN or ExpressRoute gateway, but this unit only deals with VPN virtual network gateways. (You'll learn more about ExpressRoute in a separate unit later in this module.)
- **Connection.** Create a connection resource to create a logical connection between the VPN gateway and the local network gateway.
 - The connection is made to the on-premises VPN device's IPv4 address as defined by the local network gateway.
 - The connection is made from the virtual network gateway and its associated public IP address.

You can create multiple connections.

The following diagram shows this combination of resources and their relationships to help you better understand what's required to deploy a VPN gateway.



Required on-premises resources

To connect your datacenter to a VPN gateway, you'll need these on-premises resources:

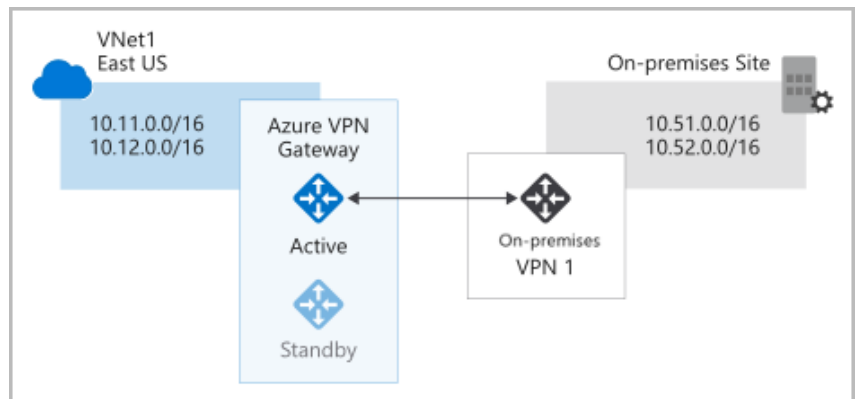
- A VPN device that supports policy-based or route-based VPN gateways
- A public-facing (internet-routable) IPv4 address

High-availability scenarios

There are several ways to ensure you have a fault-tolerant configuration.

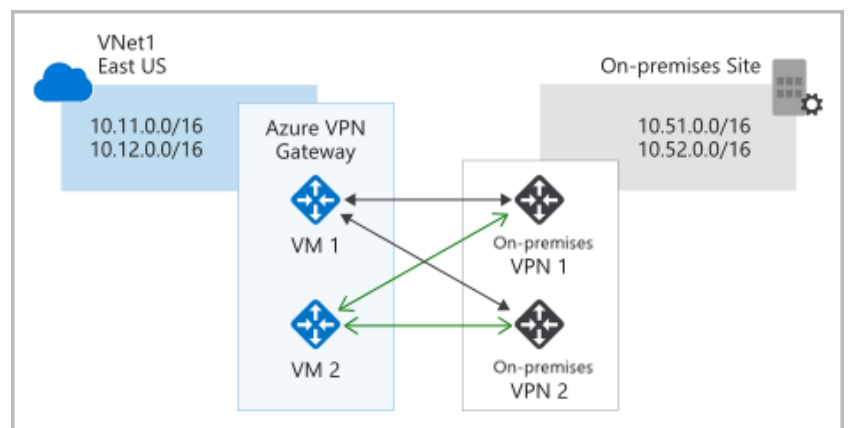
Active/standby

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. But 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.

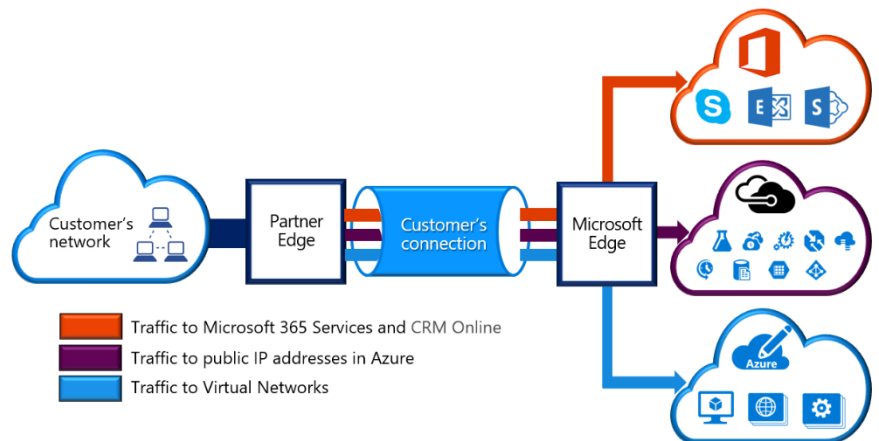
Zone-redundant gateways

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 SKUs and use Standard public IP addresses instead of Basic public IP addresses.

A.4 : Azure ExpressRoute fundamentals

ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure and Microsoft 365.



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

As part of your work for Tailwind Traders, you should understand what Azure ExpressRoute is and how it integrates with on-premises and Azure networks. In this unit, you'll learn about the benefits that ExpressRoute provides compared to other site-to-site connectivity options. As a result, you'll learn whether ExpressRoute can provide your company with the best possible network performance.

Throughout this unit, we'll focus on two different layers of the Open Systems Interconnection (OSI) model:

- **Layer 2 (L2):** This layer is the Data Link Layer, which provides node-to-node communication between two nodes on the same network.
- **Layer 3 (L3):** This layer is the Network Layer, which provides addressing and routing between nodes on a multi-node network.

Features and benefits of ExpressRoute

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

- Layer 3 connectivity between your on-premises network and the Microsoft Cloud through a connectivity provider. Connectivity can be from an any-to-any (IPVPN) network, a point-to-point Ethernet connection, or through a virtual cross-connection via an Ethernet exchange.
- Connectivity to Microsoft cloud services across all regions in the geopolitical region.
- Global connectivity to Microsoft services across all regions with the ExpressRoute premium add-on.
- Dynamic routing between your network and Microsoft via BGP.

- Built-in redundancy in every peering location for higher reliability.
- Connection uptime SLA.
- QoS support for Skype for Business.

Layer 3 connectivity

ExpressRoute provides Layer 3 (address-level) connectivity between your on-premises network and the Microsoft cloud through connectivity partners. These connections can be from a point-to-point or any-to-any network. They can also be virtual cross-connections through an exchange.

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. All redundant connections are configured with Layer 3 connectivity to meet service-level agreements.

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

Microsoft 365 was created to be accessed securely and reliably via the internet. For this reason, we recommend the use of ExpressRoute for specific scenarios. The "Learn more" section at the end of this module includes a link about using ExpressRoute to access Office 365.

Across on-premises connectivity with ExpressRoute Global Reach

You can enable ExpressRoute Global Reach to exchange data across your on-premises sites by connecting your ExpressRoute circuits. For example, assume that you have a private datacenter in California connected to ExpressRoute in Silicon Valley. You have another private datacenter in Texas connected to ExpressRoute in Dallas. With ExpressRoute Global Reach, you can connect your private datacenters through two ExpressRoute circuits. Your cross-datacenter traffic will travel through the Microsoft network.

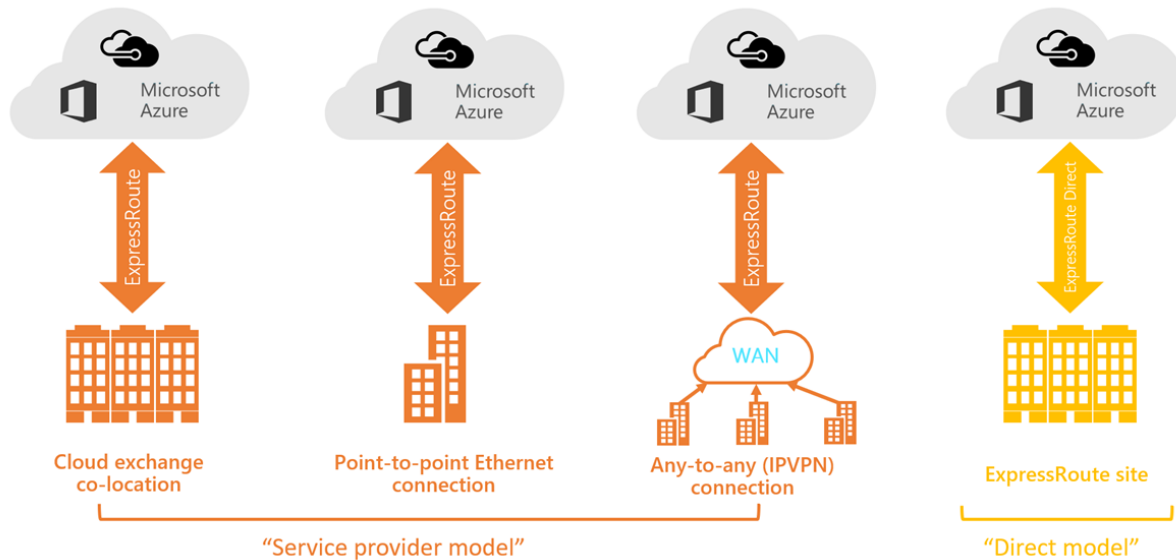
Dynamic routing

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

ExpressRoute connectivity models

ExpressRoute supports three models that you can use to connect your on-premises network to the Microsoft cloud:

- CloudExchange colocation
- Point-to-point Ethernet connection
- Any-to-any connection
- Directly from ExpressRoute sites



Colocation at a cloud exchange

Colocated providers can normally offer both Layer 2 and Layer 3 connections between your infrastructure, which might be located in the colocation facility, and the Microsoft cloud. For example, if your datacenter is colocated at a cloud exchange such as an ISP, you can request a virtual cross-connection to the Microsoft cloud.

Point-to-point Ethernet connection

Point-to-point connections provide Layer 2 and Layer 3 connectivity between your on-premises site and Azure. You can connect your offices or datacenters to Azure by using the point-to-point links. For example, if you have an on-premises datacenter, you can use a point-to-point Ethernet link to connect to Microsoft.

Any-to-any networks

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

With any-to-any connections, all WAN providers offer Layer 3 connectivity. For example, if you already use Multiprotocol Label Switching to connect to your branch offices or other sites in your

organization, an ExpressRoute connection to Microsoft behaves like any other location on your private WAN.

Directly from ExpressRoute sites

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

Security considerations

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

Summary

In this module, you used the Tailwind Traders scenario to learn about the core networking resources that are available in Azure. You learned about the benefits and usage of Azure Virtual Network, Azure VPN Gateway, and Azure ExpressRoute.

You can now apply this information to help your business use Azure's networking resources to configure its network infrastructure.

[B] Explore Azure Storage services

In this module, you'll learn about some of the different storage options that are available in Azure Storage services, and the scenarios in which each storage option is appropriate. As you complete the individual units in this module, you'll learn about Azure Blob Storage, Azure Disk Storage, Azure Files, and Blob access tiers.

Learning objectives

After completing this module, you'll be able to describe the benefits and usage of:

- Azure Blob Storage
- Azure Disk Storage
- Azure Files
- Azure Blob access tiers

Introduction



Suppose your company, Tailwind Traders, has a number of product brochures, datasheets, product images, and other files that are related to marketing, sales, and support. In the past, your company has been hosting these files on standalone web servers in your datacenter.

Your company is now in the process of migrating its applications to the cloud, and your development team is currently architecting new applications. Your Chief Technology Officer (CTO) wants to migrate all of your marketing, sales, and support files to the cloud in order to take advantage of geographic distribution of your files. This move also reduces the number of physical servers that your company maintains in your datacenter. As part of your migration strategy, you need to determine the correct approach for your cloud-based storage infrastructure.

In this module, you'll learn about the different Azure storage options and the scenarios in which each is appropriate.

Note: Azure storage isn't the same as Azure database services.

Learning objectives

After completing this module, you'll be able to describe the benefits and usage of:

- Azure Blob Storage
- Azure Disk Storage
- Azure Files Storage
- Azure Blob Access tiers

Prerequisites

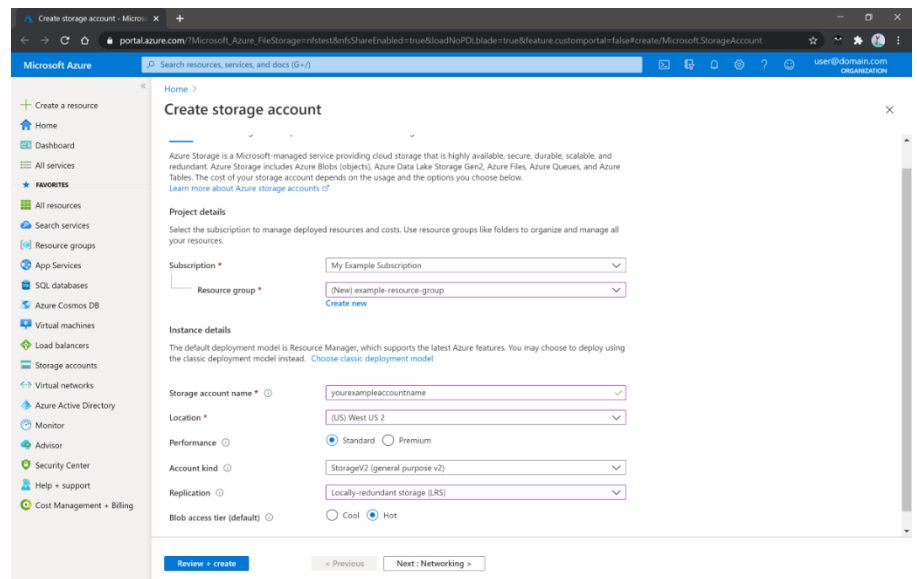
- You should be familiar with basic computing concepts and terminology

B.1 : Azure Storage account fundamentals

The Chief Technology Officer (CTO) for your company, Tailwind Traders, has tasked your team with migrating all of your files to the cloud. Your team has chosen [Azure Storage](#), which is a service that you can use to store files, messages, tables, and other types of information. Clients such as websites, mobile apps, desktop applications, and many other types of custom solutions can read data from and write data to Azure Storage. Azure Storage is also used by infrastructure as a service virtual machines, and platform as a service cloud services.

To begin using Azure Storage, you first create an Azure Storage account to store your data objects. You can create an Azure Storage account by using the Azure portal, PowerShell, or the Azure CLI.

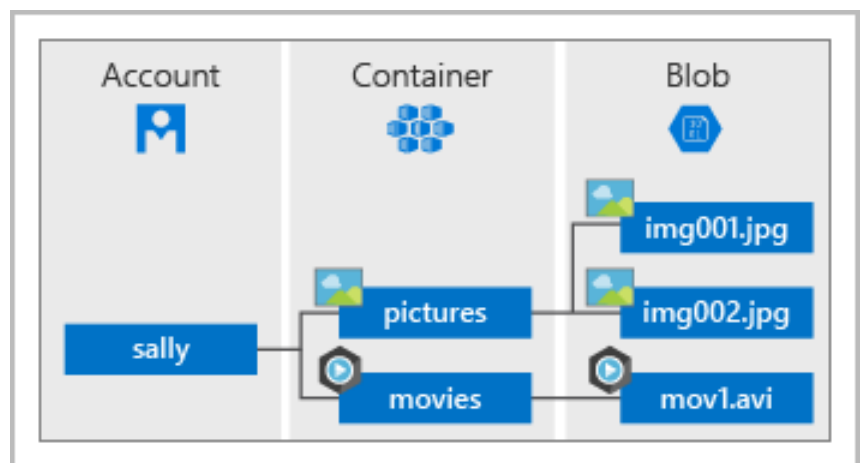
Your storage account will contain all of your Azure Storage data objects, such as blobs, files, and disks.



Note: Azure VMs use Azure Disk Storage to store virtual disks. However, you can't use Azure Disk Storage to store a disk outside of a virtual machine.

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.

For more information, see [Create a storage account](#).



B.1: Disk storage fundamentals

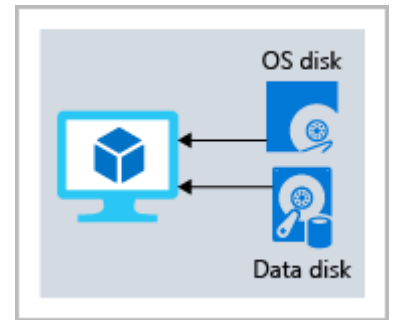
Disk Storage provides disks for Azure virtual machines. Applications and other services can access and use these disks as needed, similar to how they would in on-premises scenarios. Disk Storage allows data to be persistently stored and accessed from an attached virtual hard disk.

Disks come in many different sizes and performance levels, from solid-state drives (SSDs) to traditional spinning hard disk drives (HDDs), with varying performance tiers. You can use standard SSD and HDD disks for less critical workloads, premium SSD



disks for mission-critical production applications, and ultra disks for data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads. Azure has consistently delivered enterprise-grade durability for infrastructure as a service (IaaS) disks, with an industry-leading ZERO% annualized failure rate.

The following illustration shows an Azure virtual machine that uses separate disks to store different data.



B.2: Azure Blob storage fundamentals

Azure Blob Storage is an object storage solution for the cloud. It can store massive amounts of data, such as text or binary data. Azure Blob Storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold. Blob Storage can manage thousands of simultaneous uploads, massive amounts of video data, constantly growing log files, and can be reached from anywhere with an internet connection.



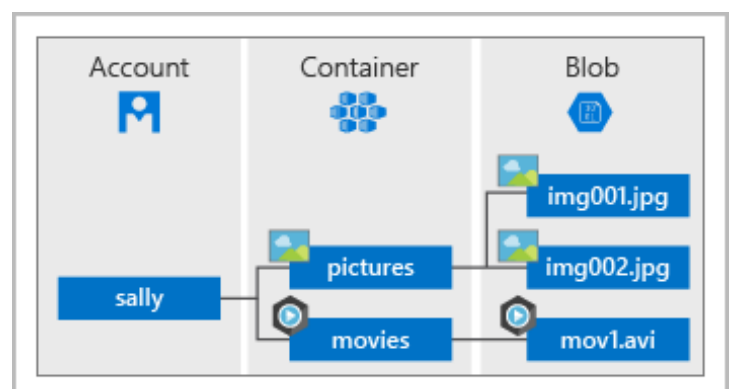
Blobs aren't limited to common file formats. A blob could contain gigabytes of binary data streamed from a scientific instrument, an encrypted message for another application, or data in a custom format for an app you're developing. One advantage of blob storage over disk storage is that it does not require developers to think about or manage disks; data is uploaded as blobs, and Azure takes care of the physical storage needs.

Blob Storage is ideal for:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Storing data for backup and restore, disaster recovery, and archiving.
- Storing data for analysis by an on-premises or Azure-hosted service.
- Storing up to 8 TB of data for virtual machines.

You store blobs in containers, which helps you organize your blobs depending on your business needs.

The following diagram illustrates how you might use Azure accounts, containers, and blobs.



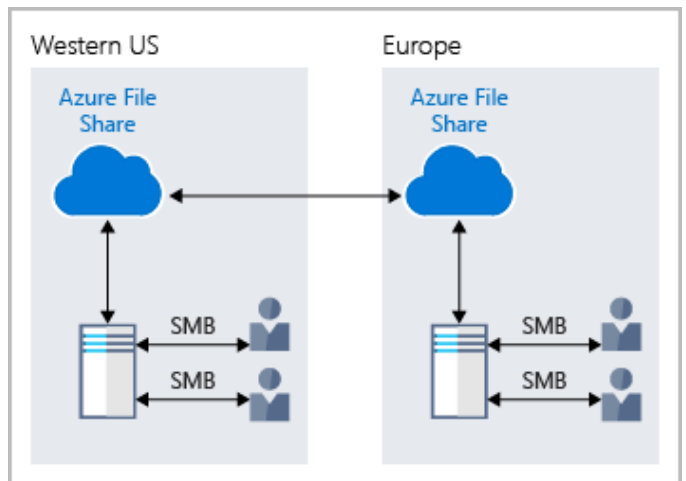
B.3: Azure Files fundamentals

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block and Network File System (preview) protocols. Azure file shares can be mounted concurrently by cloud or on-premises deployments of Windows, Linux, and macOS. Applications running in Azure virtual machines or cloud services can mount a file storage share to access file data, just as a desktop application would mount a typical SMB share. Any number of Azure virtual machines or roles can mount and access the file storage share simultaneously. Typical usage scenarios would be to share files anywhere in the world, diagnostic data, or application data sharing.



Use Azure Files for the following situations:

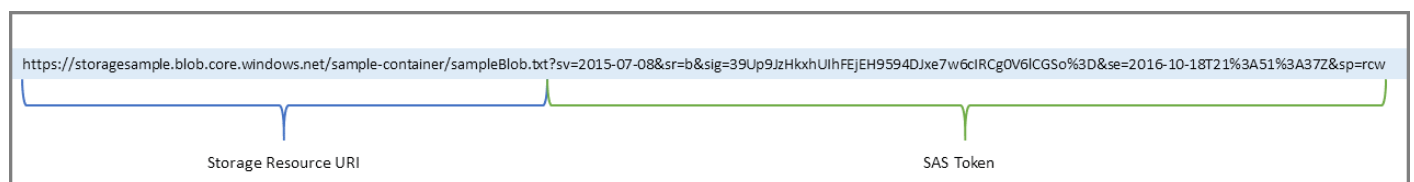
- Many on-premises applications use file shares. Azure Files makes it easier to migrate those applications that share data to Azure. If you mount the Azure file share to the same drive letter that the on-premises application uses, the part of your application that accesses the file share should work with minimal changes, if any.
- Store configuration files on a file share and access them from multiple VMs. Tools and utilities used by multiple developers in a group can be stored on a file share, ensuring that everybody can find them, and that they use the same version.
- Write data to a file share, and process or analyze the data later. For example, you might want to do this with diagnostic logs, metrics, and crash dumps.



The following illustration shows Azure Files being used to share data between two geographical locations. Azure Files ensures the data is encrypted at rest, and the SMB protocol ensures the data is encrypted in transit.

One thing that distinguishes Azure Files from files on a corporate file share is that you can access the files from anywhere in the world, by using a URL that points to the file. You can also use Shared Access Signature (SAS) tokens to allow access to a private asset for a specific amount of time.

Here's an example of a service SAS URI, showing the resource URI and the SAS token:



B.4: Understand Blob access tiers

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



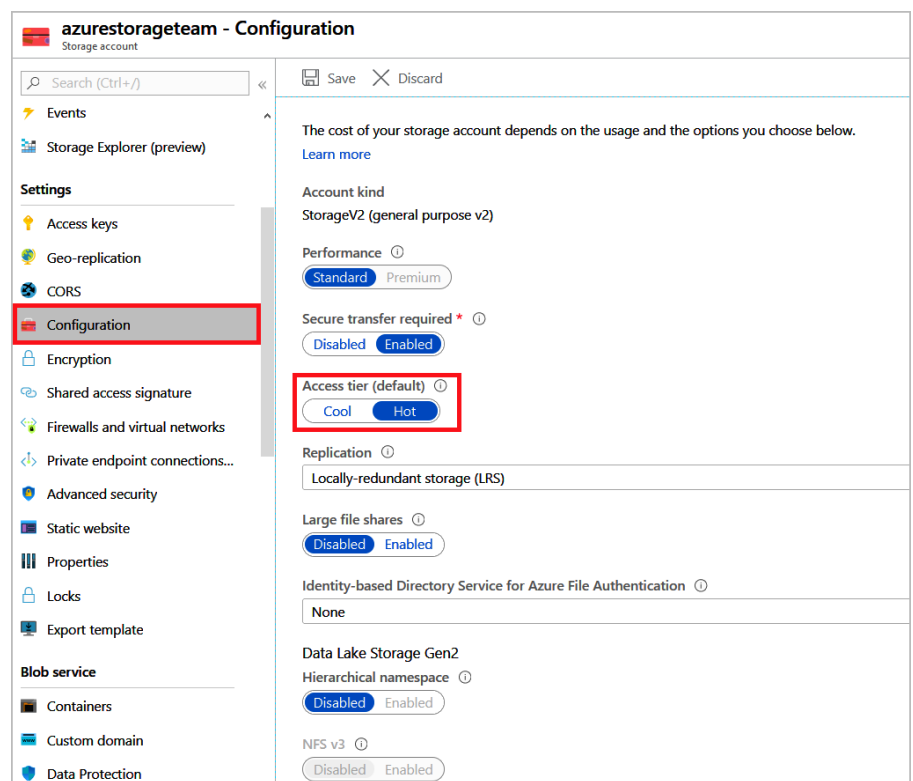
Azure Storage offers different access tiers for your blob storage, helping you store object data in the most cost-effective manner. The available access tiers include:

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

The following considerations apply to the different access tiers:

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

The following illustration demonstrates choosing between the hot and cool access tiers on a general purpose storage account.



[C] Explore Azure database and analytics services

In this module, you'll learn about several of the database services that are available on Microsoft Azure, such as Azure Cosmos DB, Azure SQL Database, Azure SQL Managed Instance, Azure Database for MySQL, and Azure Database for PostgreSQL. In addition, you'll learn about several of the big data and analysis services in Azure.

Learning objectives

After completing this module, you'll be able to describe the benefits and usage of:

- Azure Cosmos DB
- Azure SQL Database
- Azure SQL Managed Instance
- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Synapse Analytics
- Azure HDInsight
- Azure Databricks
- Azure Data Lake Analytics

Introduction

Due to a growing number of acquisitions over the last decade, Tailwind Traders use various database and analytics technologies. As the company begins to migrate existing data workloads and deploy new data workloads to Azure, it needs to understand which Azure technology will be appropriate for each workload. The company's Chief Technology Officer (CTO) has assigned you the task of researching the different database options that are available. This research will help your company choose the right options for each of your data scenarios.



Today's applications are required to be highly responsive and always online. To achieve low latency and high availability, instances of these applications need to be deployed in datacenters that are close to their users. Applications need to respond in real time to large changes in usage at peak hours, store ever-increasing volumes of data, and make this data available to users in milliseconds. To help your company reach its goals, Azure database services are globally distributed, and Azure supports many of the industry standard databases and APIs.

In this module, you'll learn more about several of the primary database services that are available on Azure, and you'll analyze some of the reasons why each of these services might be the right choice for your data needs.

Learning objectives

After completing this module, you'll be able to describe the benefits and usage of:

- Azure Cosmos DB
- Azure SQL Database
- Azure SQL Managed Instance

- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Synapse Analytics
- Azure HDInsight
- Azure Databricks
- Azure Data Lake Analytics

Prerequisites

- You should be familiar with basic computing concepts and terminology.
- You should be familiar with basic database concepts and terminology.

C.1: Explore Azure Cosmos DB

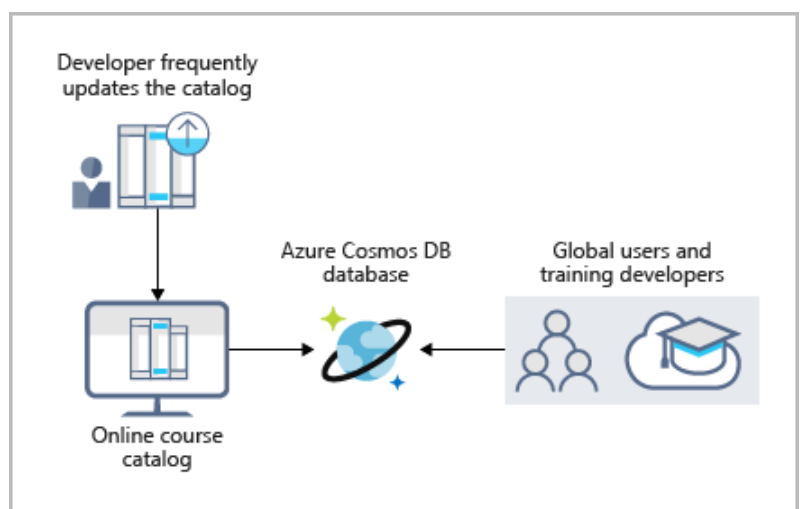
Over the years, Tailwind Traders has acquired several smaller companies. Each of these companies had teams of developers who used different database services and various APIs to work with their data. A long-term plan might be to eventually move all of the disparate data to a common database service. For now, though, you'd like to enable each of these teams to work with an environment where they can use their existing skills. Fortunately for you, Azure Cosmos DB can help out.



Azure Cosmos DB is a globally distributed, multi-model database service. You can elastically and independently scale throughput and storage across any number of Azure regions worldwide. You can take advantage of fast, single-digit-millisecond data access by using any one of several popular APIs. Azure Cosmos DB provides comprehensive service level agreements for throughput, latency, availability, and consistency guarantees.

Azure Cosmos DB supports schema-less data, which lets you build highly responsive and "Always On" applications to support constantly changing data. You can use this feature to store data that's updated and maintained by users around the world.

For example, Tailwind Traders provides a public training portal that is used by customers across the globe to learn about the different tools that Tailwind Traders creates. Tailwind Traders developers maintain and update the data. The following illustration shows a sample Azure Cosmos DB database that's used to store data for the Tailwind Traders training portal website.



Azure Cosmos DB is flexible. At the lowest level, Azure Cosmos DB stores data in atom-record-sequence (ARS) format. The data is then abstracted and projected as an API, which you specify when you're creating your database. Your choices include SQL, MongoDB, Cassandra, Tables, and Gremlin. This level of flexibility means that as you migrate your company's databases to Azure Cosmos DB, your developers can stick with the API that they're the most comfortable with.

C.2: Explore Azure SQL Database

Azure SQL Database is a relational database based on the latest stable version of the Microsoft SQL Server database engine. SQL Database is a high-performance, reliable, fully managed, and secure database. You can use it to build data-driven applications and websites in the programming language of your choice, without needing to manage infrastructure.



Features

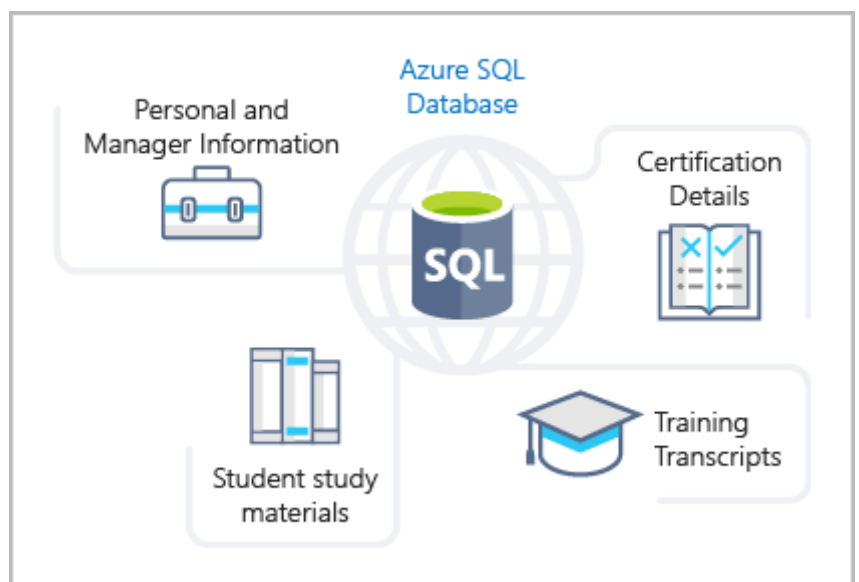
Azure SQL Database is a platform as a service (PaaS) database engine. It handles most of the database management functions, such as upgrading, patching, backups, and monitoring, without user involvement. SQL Database provides 99.99 percent availability. PaaS capabilities that are built into SQL Database enable you to focus on the domain-specific database administration and optimization activities that are critical for your business. SQL Database is a fully managed service that has built-in high availability, backups, and other common maintenance operations. Microsoft handles all updates to the SQL and operating system code. You don't have to manage the underlying infrastructure.

You can create a highly available and high-performance data storage layer for the applications and solutions in Azure. SQL Database can be the right choice for a variety of modern cloud applications because it enables you to process both relational data and non-relational structures, such as graphs, JSON, spatial, and XML.

You can use advanced query processing features, such as high-performance, in-memory technologies and intelligent query processing. In fact, the newest capabilities of SQL Server are released first to SQL Database, and then to SQL Server itself. You get the newest SQL Server capabilities, with no overhead for updates or upgrades, tested across millions of databases.

Migration

Tailwind Traders currently uses several on-premises servers running SQL Server, which provide data storage for your public-facing website (for example, customer data, order history, and product catalogs). In addition, your on-premises servers running SQL Server also provide data storage for your internal-only training portal website. Tailwind Traders uses the website for new employee training materials (such as study materials, certification details, and training transcripts). The following illustration shows the types of data that your company might store in the Azure SQL Database training portal website.



You can migrate your existing SQL Server databases with minimal downtime by using the Azure Database Migration Service. The Microsoft Data Migration Assistant can generate assessment reports that provide recommendations to help guide you through required changes prior to performing a migration. After you assess and resolve any remediation required, you're ready to begin the migration process. The Azure Database Migration Service performs all of the required steps. You just change the connection string in your apps.

C.3: Exercise - Create a SQL database

Tailwind Traders has chosen Azure SQL Database for part of its migration. You've been tasked with creating the database.

In this exercise, you'll create a SQL database in Azure and then query the data in that database.

Task 1: Create the database

In this task, you create a SQL database based on the *AdventureWorksLT* sample database.

1. Sign in to the [Azure portal](#).
2. Select **Create a resource > Databases > SQL database**. The **Create SQL Database** pane appears.
3. Enter the following values for each setting.

Setting	Value
Project details	
Subscription	Concierge Subscription
Resource group	[sandbox resource group name]
Database details	
Database name	db1
Server	Select Create new

4. The **Create SQL Database Server** pane appears.
5. Enter the following values for each setting.

Setting	Value
Server details	
Server name	sqlservernnnn (replace nnnn with letters and digits for a globally unique name)
Location	(US) East US
Authentication	
Authentication method	Use SQL Authentication

Server admin login	sqluser
Password	Pa\$\$w0rd1234

6. Select **OK**.
7. Complete the remaining fields for **Create SQL Database** using the following values.

Setting	Value
Want to use SQL elastic pool?	No (default)
Compute + storage	General Purpose (default)
Backup storage redundancy	
Backup storage redundancy	Geo-redundant backup storage

Create SQL Database

Microsoft

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Conclierge Subscription

Resource group *

learn-990a7123-0960-451f-adc5-2384b7f195f7

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name *

db1

Server *

(new) sqlserverja12 (East US)

Want to use SQL elastic pool? *

Yes

No

Compute + storage *

General Purpose

Gen5, 2 vCores, 32 GB storage, zone redundant disabled

Configure database

Backup storage redundancy

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy

Locally-redundant backup storage - Preview

Zone-redundant backup storage - Preview

Geo-redundant backup storage

Selected value for backup storage redundancy is Geo-redundant backup storage. Note that database backups will be geo-replicated to the paired region. [Learn more](#)

Your use of either of the Preview backup storage redundancy options (PDS and LRS) is governed by the agreement under which you obtained Microsoft Azure Services. By selecting a Preview redundancy option, you confirm that you agree to the preview terms in such agreement. [Microsoft Azure Legal Information](#); [Learn more](#)

Server details

Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.

Server name *

sqlserverja12

.database.windows.net

Location *

(US) East US

Authentication

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Azure AD authentication [Learn more](#) using an existing Azure AD user, group, or application as Azure AD admin [Learn more](#), or select both SQL and Azure AD authentication.

Authentication method

Use SQL Authentication

Use only Azure Active Directory (Azure AD) authentication

Use both SQL and Azure AD Authentication

Server admin login *

sqluser

Password *

Confirm password *

OK

Review + create

Next : Networking >

8.

9. Select **Next : Networking**, and configure the following settings (accept defaults for fields not specified).

Setting	Value
Network connectivity	
Connectivity method	Public endpoint

Create SQL Database ...

Microsoft

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. [Learn more](#)

- ☐ No access
- ☒ Public endpoint
- ☐ Private endpoint

Connectivity method * ⓘ

Firewall rules

Setting 'Allow Azure services and resources to access this server' to Yes allows communications from all resources inside the Azure boundary, that may or may not be part of your subscription. [Learn more](#)

Setting 'Add current client IP address' to Yes will add an entry for your client IP address to the server firewall.

Allow Azure services and resources to access this server *

☒ No ☐ Yes

Add current client IP address *

☒ No ☐ Yes

Connection policy

Configure how clients communicate with your SQL database server. [Learn more](#)

Connection policy ⓘ

- ☒ Default - Uses Redirect policy for all client connections originating inside of Azure and Proxy for all client connections originating outside Azure
- ☐ Proxy - All connections are proxied via the Azure SQL Database gateways
- ☐ Redirect - Clients establish connections directly to the node hosting the database

Encrypted connections

This server supports encrypted connections using Transport Layer Security (TLS). For information on TLS version and certificates, refer to connecting with TLS/SSL. [Learn more](#)

Minimum TLS version ⓘ

TLS 1.2

[Review + create](#)[< Previous](#)[Next : Security >](#)

- 10.
11. Select **Next : Security**, and for **Enable Azure Defender for SQL**, choose **Not now**. Leave the remaining settings as default (not configured).

Create SQL Database ...

Microsoft



Basics Networking Security Additional settings Tags Review + create

Azure Defender for SQL

Protect your data using Azure Defender for SQL, a unified security package including vulnerability assessment and advanced threat protection for your server. [Learn more](#)

Get started with a 30 day free trial period, and then 15 USD/server/month.

Enable Azure Defender for SQL * ⓘ

- ☐ Start free trial
- ☒ Not now

[Review + create](#)[< Previous](#)[Next : Additional settings >](#)

12. Select **Next : Additional settings**, and configure the following settings.

Setting	Value
Data source	
Use existing data	Sample
Database collation	
Collation	SQL_Latin1_General_CP1_CI_AS (default)

[Home](#) >

Create SQL Database

Microsoft

Basics Networking Security **Additional settings** Tags Review + create

Customize additional configuration parameters including collation & sample data.

Data source

Start with a blank database, restore from a backup or select sample data to populate your new database.

Use existing data *

None Backup **Sample**

AdventureWorksLT will be created as the sample database.

Database collation

Database collation defines the rules that sort and compare data, and cannot be changed after database creation. The default database collation is SQL_Latin1_General_CP1_CI_AS. [Learn more](#)

Collation ⓘ

SQL_Latin1_General_CP1_CI_AS

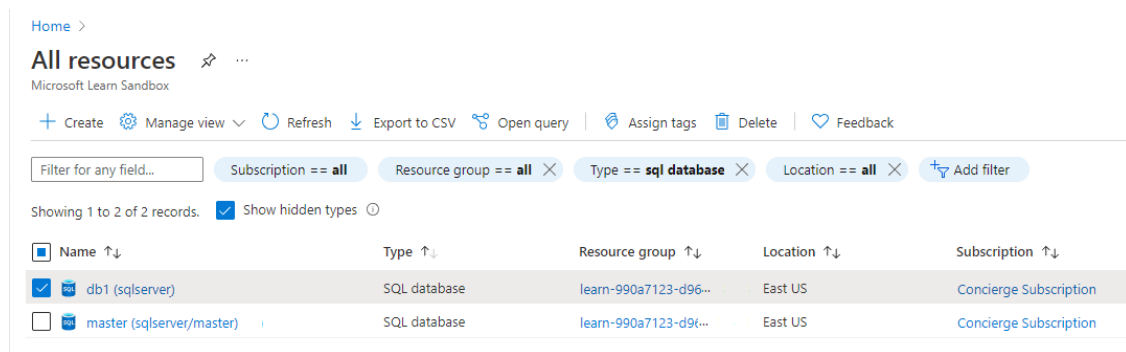
[Review + create](#) [< Previous](#) [Next : Tags >](#)

- 13.
14. Select **Review + create** to validate configuration entries.
15. Select **Create** to deploy the server and database. It can take approximately two to five minutes to create the server and deploy the sample database. The deployment pane shows the status, with updates for each resource that is created.
16. When deployment is complete, select **Go to resource**. The db1 SQL database Overview pane shows the essentials of the newly deployed database
17. In the command bar, select **Set server firewall**. The **Firewall settings** page appears.
18. Set **Allow Azure services and resources to access this server** to **Yes**, leaving other settings as default.
19. In the command bar, select **Save** to update firewall settings, and then close the Firewall settings pane.

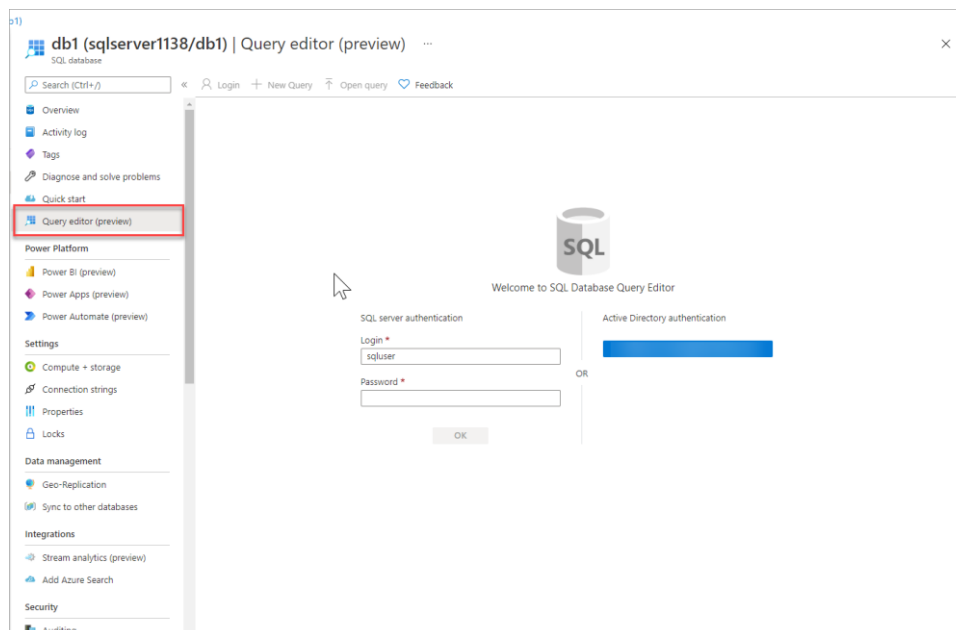
Task 2: Test the database

In this task, you configure the server and run a SQL query.

1. In Azure resources menu select **All resources**. Search for and select the **SQL database** resource Type, and ensure that your new database was created. You might need to refresh the page.



2. Select **db1**, the SQL database you created.
3. In the SQL database menu, select **Query editor (preview)**. The **Query editor (preview)** pane appears.



4. Sign in as **sqluser**, with the password **Pa\$\$w0rd1234**.

You will not be able to sign in because your IP address needs to be enabled in a firewall rule.

SQL

Welcome to SQL Database Query Editor

SQL server authentication

Login *

sqluser

Password *

..... ✓

Active Directory authentication

Continue as [redacted]

OR

✖ Cannot open server 'sqlserverxxx1' requested by the login. Client with IP address [redacted] is not allowed to access the server. To enable access, use the Windows Azure Management Portal or run `sp_set_firewall_rule` on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect.

[Set server firewall \(sqlserverxxx1\)](#)

5. In the Query editor menu, select **Overview** (your edits will be lost), and in the command bar, select **Set server firewall**. The **Firewall settings** page appears.
6. In the **Client IP address** section, your IP will be shown (verify that it is the same client IP address from the error you received in the previous step).
7. In the command bar select **Add client IP**. This will add a **Rule name** that contains your IP address in both the **Start IP** and **End IP** fields.
8. In the command bar, select **Save** to save this firewall rule.

All services > SQL databases > db1 (sqlserver1138/db1)

Firewall settings

sqlserver1138 (SQL server)

Save Discard Add client IP

Deny public network access

Yes No

Click here to create a new private endpoint.
Create private Endpoint

Minimum TLS version

1.0 1.1 1.2

Connection Policy

Default Proxy Redirect

Allow Azure services and resources to access this server

Yes No

Client IP address your IP address

Rule name	Start IP	End IP
ClientIPAddress_2021-3...	your IP address	your IP address

Virtual networks

+ Add existing virtual network + Create new virtual network

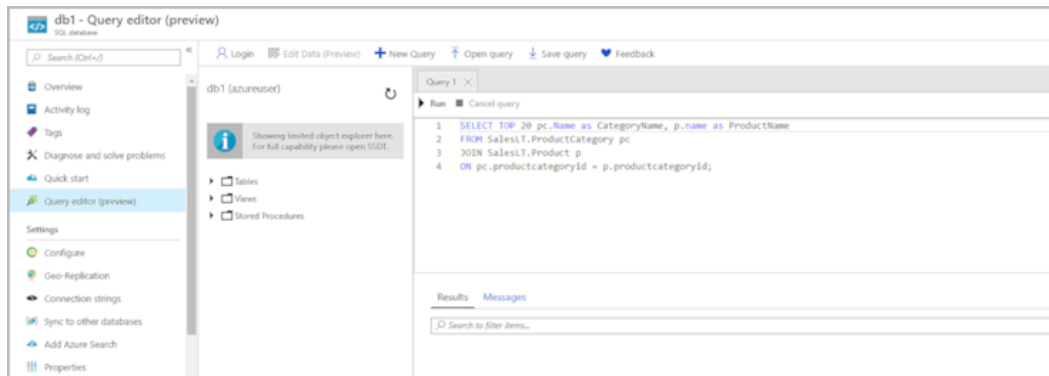
Rule name	Virtual network	Subnet	Address Range	Endpoint status
No vnet rules for this server.				

9. Select your db1 database in the breadcrumb at the top of the page to return to your SQL database, and then select **Query editor (preview)** from the menu. sign-in page.
10. Sign in again as **sqluser**, with the password **Pa\$\$w0rd1234**. This time you should succeed. It might take a couple of minutes for the new firewall rule to be deployed. If you still get an error, verify the client IP address in the error, and return to **Firewall settings** to add the correct client IP address.

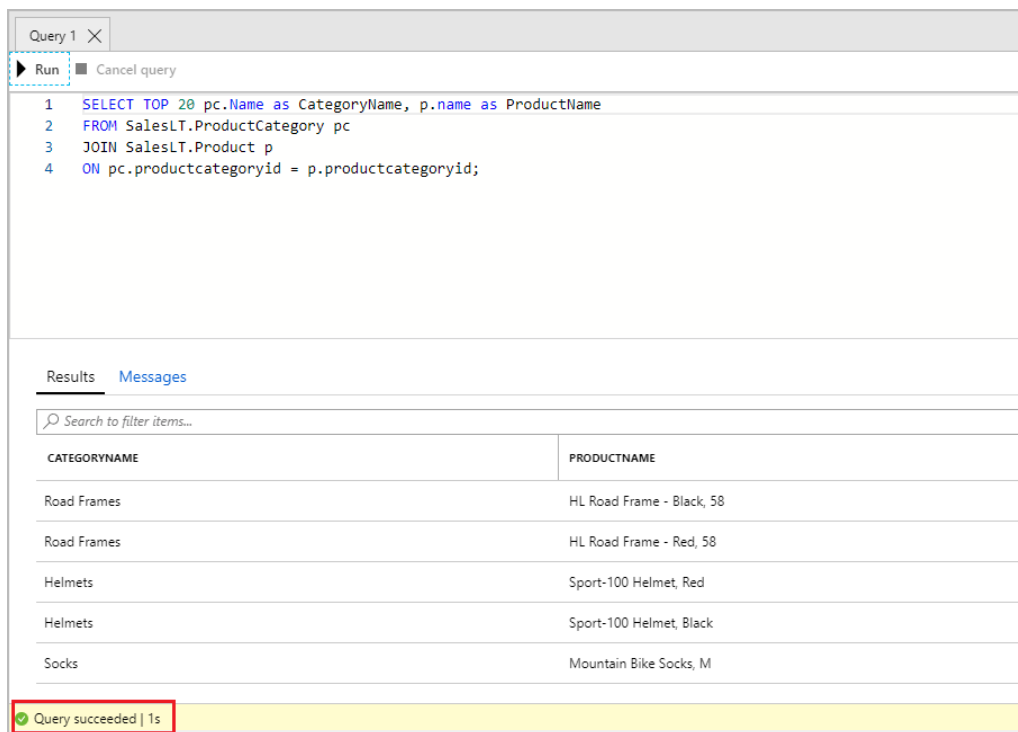
11. After you sign in successfully, the query pane appears. Enter the following SQL query into the editor pane.

SQLCopy

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
FROM SalesLT.ProductCategory pc
JOIN SalesLT.Product p
ON pc.productcategoryid = p.productcategoryid;
```



12. Select **Run**, and then review the query results in the **Results** pane. The query should run successfully.



Congratulations! You've created a SQL database in Azure and successfully queried the data in that database.

C.4: Explore Azure database for MySQL

Tailwind Traders currently manages several websites on-premises that use the LAMP stack (Linux, Apache, MySQL, PHP). As part of your planning for your migration strategy, the different teams at Tailwind Traders have been researching the available service offerings that Azure provides. You've already discovered that the Web Apps feature of Azure App Service provides built-in functionality to create web applications that use

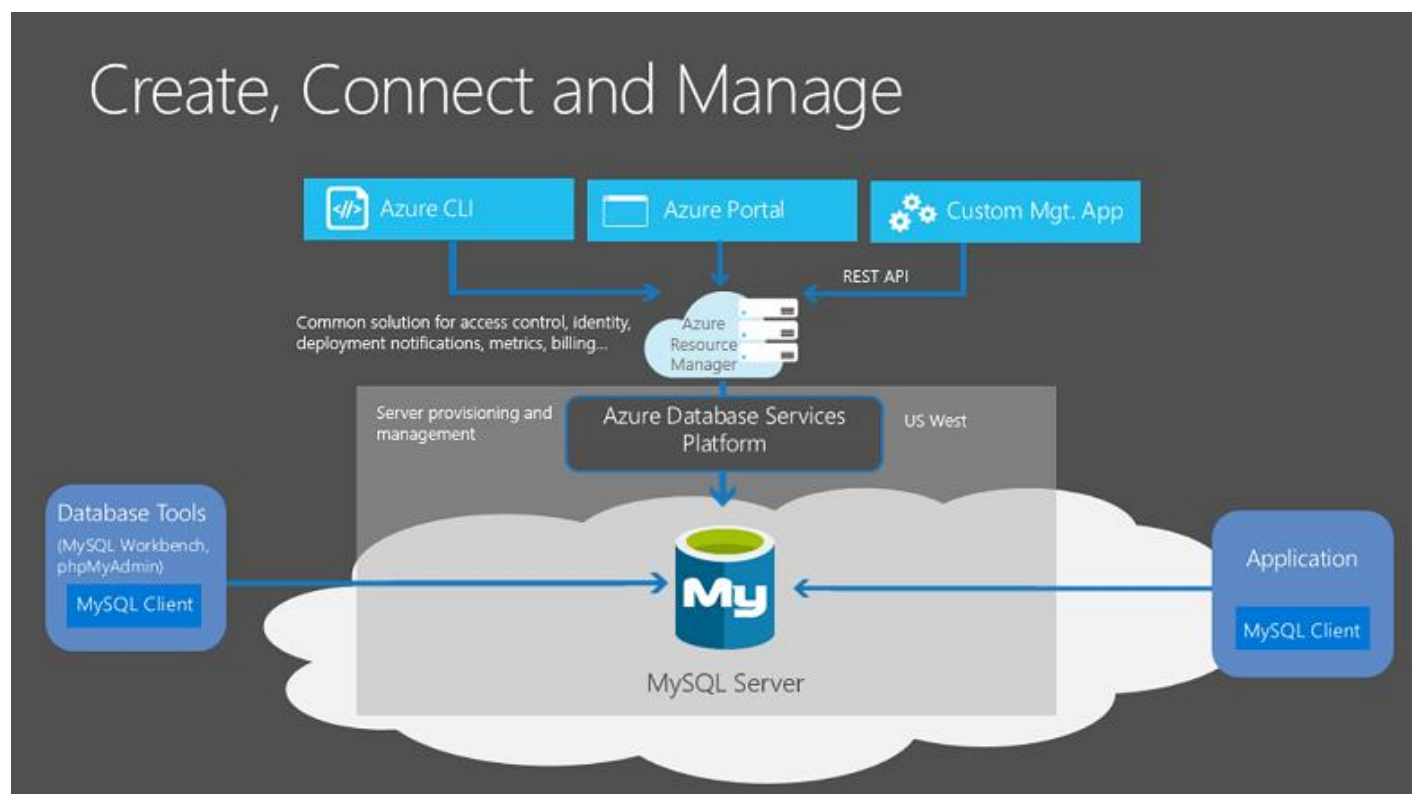
PHP on a Linux server running Apache. You've been tasked with investigating whether the database requirements for the web development team will continue to be met after the migration to Azure.

Azure Database for MySQL is a relational database service in the cloud, and it's based on the MySQL Community Edition database engine, versions 5.6, 5.7, and 8.0. With it, you have a 99.99 percent availability service level agreement from Azure, powered by a global network of Microsoft-managed datacenters. This helps keep your app running 24/7. With every Azure Database for MySQL server, you take advantage of built-in security, fault tolerance, and data protection that you would otherwise have to buy or design, build, and manage. With Azure Database for MySQL, you can use point-in-time restore to recover a server to an earlier state, as far back as 35 days.

Azure Database for MySQL delivers:

- Built-in high availability with no additional cost.
- Predictable performance and inclusive, pay-as-you-go pricing.
- Scale as needed, within seconds.
- Ability to protect sensitive data at-rest and in-motion.
- Automatic backups.
- Enterprise-grade security and compliance.

These capabilities require almost no administration, and all are provided at no additional cost. They allow you to focus on rapid app development and accelerating your time-to-market, rather than having to manage virtual machines and infrastructure. In addition, you can migrate your existing MySQL databases with minimal downtime by using the Azure Database Migration Service. After you have completed your migration, you can continue to develop your application with the open-source tools and platform of your choice. You don't have to learn new skills.



Azure Database for MySQL offers several service tiers, and each tier provides different performance and capabilities to support lightweight to heavyweight database workloads. You can build your first app on a small database for a few dollars a month, and then adjust the scale to meet the needs of

your solution. Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements. You only pay for the resources you need, and only when you need them.

C.5: Explore Azure Database for PostgreSQL

As part of its overall data strategy, Tailwind Traders have been using PostgreSQL for several years. You and your team probably already know the benefits of PostgreSQL. Part of your migration is to use Azure Database for PostgreSQL, and you want to make sure that you'll have access to the same benefits as your on-premises server before moving to the cloud.

Azure Database for PostgreSQL is a relational database service in the cloud. The server software is based on the community version of the open-source PostgreSQL database engine. Your familiarity with tools and expertise with PostgreSQL is applicable when you're using Azure Database for PostgreSQL.

Moreover, Azure Database for PostgreSQL delivers the following benefits:

- Built-in high availability compared to on-premises resources. There's no additional configuration, replication, or cost required to make sure your applications are always available.
- Simple and flexible pricing. You have predictable performance based on a selected pricing tier choice that includes software patching, automatic backups, monitoring, and security.
- Scale up or down as needed, within seconds. You can scale compute or storage independently as needed, to make sure you adapt your service to match usage.
- Adjustable automatic backups and point-in-time-restore for up to 35 days.
- Enterprise-grade security and compliance to protect sensitive data at-rest and in-motion. This security covers data encryption on disk and SSL encryption between client and server communication.

Azure Database for PostgreSQL is available in two deployment options: **Single Server** and **Hyperscale (Citrus)**.

Single Server

The Single Server deployment option delivers:

- Built-in high availability with no additional cost (99.99 percent SLA).
- Predictable performance and inclusive, pay-as-you-go pricing.
- Vertical scale as needed, within seconds.
- Monitoring and alerting to assess your server.
- Enterprise-grade security and compliance.
- Ability to protect sensitive data at-rest and in-motion.
- Automatic backups and point-in-time-restore for up to 35 days.

All those capabilities require almost no administration, and all are provided at no additional cost. You can focus on rapid application development and accelerating your time to market, rather than having to manage virtual machines and infrastructure. You can continue to develop your application with the open-source tools and platform of your choice, without having to learn new skills.

The Single Server deployment option offers three pricing tiers: Basic, General Purpose, and Memory Optimized. Each tier offers different resource capabilities to support your database workloads. You can build your first app on a small database for a few dollars a month, and then adjust the scale to meet the needs of your solution. Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements. You only pay for the resources you need, and only when you need them.

Hyperscale (Citus)

The Hyperscale (Citus) option horizontally scales queries across multiple machines by using sharding. Its query engine parallelizes incoming SQL queries across these servers for faster responses on large datasets. It serves applications that require greater scale and performance, generally workloads that are approaching, or already exceed, 100 GB of data.

The Hyperscale (Citus) deployment option supports multi-tenant applications, real-time operational analytics, and high throughput transactional workloads. Applications built for PostgreSQL can run distributed queries on Hyperscale (Citus) with standard connection libraries and minimal changes.

C.6: Explore Azure SQL Managed Instance

Azure SQL Managed Instance is a scalable cloud data service that provides the broadest SQL Server database engine compatibility with all the benefits of a fully managed platform as a service. Depending on your scenario, Azure SQL Managed Instance might offer more options for your database needs.

Features

Like Azure SQL Database, Azure SQL Managed Instance is a platform as a service (PaaS) database engine, which means that your company will be able to take advantage of the best features of moving your data to the cloud in a fully-managed environment. For example, your company will no longer need to purchase and manage expensive hardware, and you won't have to maintain the additional overhead of managing your on-premises infrastructure. On the other hand, your company will benefit from the quick provisioning and service scaling features of Azure, together with automated patching and version upgrades. In addition, you'll be able to rest assured that your data will always be there when you need it through built-in high availability features and a 99.99% uptime service level agreement (SLA). You'll also be able to protect your data with automated backups and a configurable backup retention period.

Azure SQL Database and Azure SQL Managed Instance offer many of the same features; however, Azure SQL Managed Instance provides several options that might not be available to Azure SQL Database. For example, Tailwind Traders currently uses several on-premises servers running SQL Server, and they would like to migrate their existing databases to a SQL database running in the cloud. However, several of their databases use Cyrillic characters for collation. In this scenario, Tailwind Traders should migrate their databases to an Azure SQL Managed Instance, since Azure SQL Database only uses the default SQL_Latin1_General_CP1_CI_AS server collation.

Note

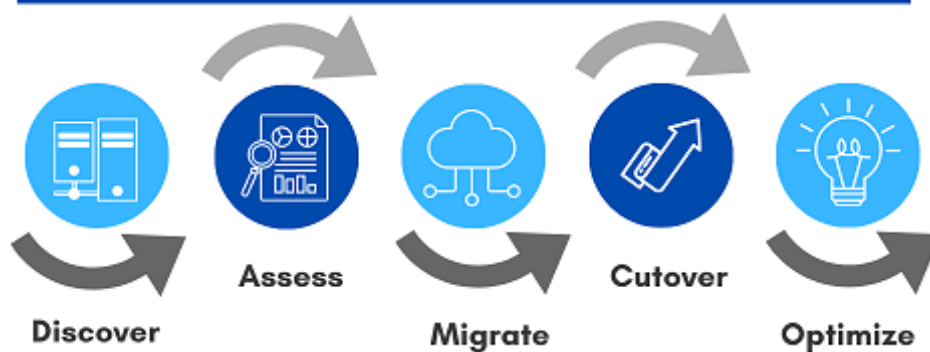
For a detailed list of the differences between Azure SQL Database and Azure SQL Managed Instance, see [Features comparison: Azure SQL Database and Azure SQL Managed Instance](#).

Migration

Azure SQL Managed Instance makes it easy to migrate your on-premises data on SQL Server to the cloud using the Azure Database Migration Service (DMS) or native backup and restore. After you have discovered all of the features that your company uses, you need to assess which on-premises SQL Server instances you can migrate to Azure SQL Managed Instance to see if you have any blocking issues. Once you have resolved any issues, you can migrate your data, then cutover from your on-premises SQL Server to your Azure SQL Managed Instance by changing the connection string in your applications.

Migration Process Flow

A step-by-step guide



Note

For a detailed description of the migration process, see [Migration guide: SQL Server to SQL Managed Instance](#)

C.7: Explore big data and analytics

Several years ago, Tailwind Traders rolled out a new GPS tracking system for all of its delivery vehicles. The new system provides real-time tracking data to your primary datacenter. Your CTO wants your team to look at several years of tracking data in order to determine trends. For example, an important trend might be a spike in deliveries around the holidays that would require hiring additional staff. Through an in-depth analysis of the tracking data that you've recorded, your CTO seeks to predict when changes are necessary, and then proactively take the necessary steps to appropriately manage spikes.

Data comes in all types of forms and formats. When we talk about big data, we're referring to large volumes of data. In this Tailwind Traders scenario, data is collected from the GPS sensors, which includes location information, data from weather systems, and many other sources that generate large amounts of data. This amount of data becomes increasingly hard to make sense of and to base decisions on. The volumes are so large that traditional forms of processing and analysis are no longer appropriate.

Open-source cluster technologies have been developed, over time, to try to deal with these large datasets. Microsoft Azure supports a broad range of technologies and services to provide big data and analytic solutions, including Azure Synapse Analytics, Azure HDInsight, Azure Databricks, and Azure Data Lake Analytics.

Azure Synapse Analytics

[Azure Synapse Analytics](#) (formerly Azure SQL Data Warehouse) is a limitless analytics service that brings together enterprise data warehousing and big data analytics. You can query data on your terms by using either serverless or provisioned resources at scale. You have a unified experience to ingest, prepare, manage, and serve data for immediate business intelligence and machine learning needs.



Azure HDInsight

[Azure HDInsight](#) is a fully managed, open-source analytics service for enterprises. It's a cloud service that makes it easier, faster, and more cost-effective to process massive amounts of data. You can run popular open-source frameworks and create cluster types such as [Apache Spark](#), [Apache Hadoop](#), [Apache Kafka](#), [Apache HBase](#), [Apache Storm](#), and [Machine Learning Services](#). HDInsight also supports a broad range of scenarios such as extraction, transformation, and loading (ETL), data warehousing, machine learning, and IoT.

Azure Databricks

[Azure Databricks](#) helps you unlock insights from all your data and build artificial intelligence solutions. You can set up your Apache Spark environment in minutes, and then autoscale and collaborate on shared projects in an interactive workspace. Azure Databricks supports Python, Scala, R, Java, and SQL, as well as data science frameworks and libraries including TensorFlow, PyTorch, and scikit-learn.



Azure Data Lake Analytics

[Azure Data Lake Analytics](#) is an on-demand analytics job service that simplifies big data. Instead of deploying, configuring, and tuning hardware, you write queries to transform your data and extract valuable insights. The analytics service can handle jobs of any scale instantly by setting the dial for how much power you need. You only pay for your job when it's running, making it more cost-effective.