100 XP

Introduction

3 minutes

Meet Tailwind Traders

Let's suppose you work as an Architect at Tailwind Traders. Tailwind Traders is a company that specializes in hardware manufacturing with online sales. Your management team tells you several development projects need to migrate to the cloud. There are also several new projects that should be optimized for the cloud.

You know the department's budget is tight. It will be important to select the right compute technology for each project. Ideally, you would like to create compute resources, configure the resources, and pay for only what you use.

Learning objectives

In this module, you'll learn how to:

- Choose a compute service.
- Design for Azure virtual machines solutions.
- Design for Azure Batch solutions.
- Design for Azure Function solutions.
- Design for Azure Logic App solutions.
- Design for Azure Container Instances solutions.
- Design for Azure App Services solutions.
- Design for Azure Kubernetes Service solutions.

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions.

Design for compute solutions

- Recommend an appropriately sized compute solution based on workload requirements.
- Recommend a Container-based compute solution.
- Recommend a Serverless-based compute solution
- Recommend a Virtual Machine-based compute solution

Prerequisites

- Conceptual knowledge of Azure compute solutions.
- Working experience with virtual machines, containers, and app service.

Next unit: Choose a compute service

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Choose a compute service

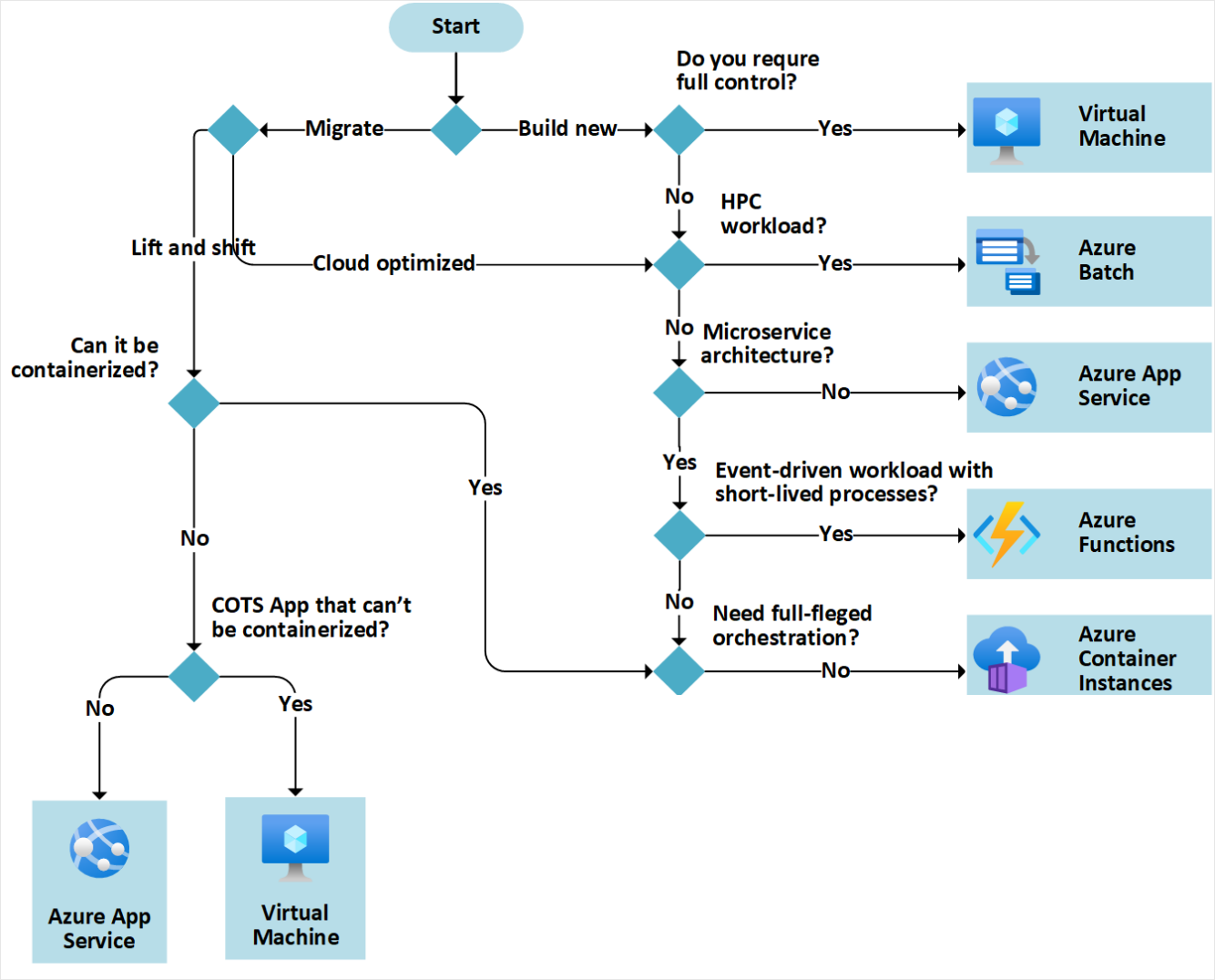
3 minutes

Compute refers to the hosting model for the computing resources that your applications run on. Azure offers several compute services, which we will cover in this module. Here's a short summary.

- **Virtual machines (IaaS)**. Deploy and manage VMs inside an Azure virtual network.
- **Azure Batch (PaaS)**. A managed service for running large-scale parallel and high-performance computing (HPC) applications.
- **Azure Functions (FaaS)**. A managed service for running code in the cloud, without worrying about the infrastructure.
- **Azure Logic Apps (PaaS)**. A cloud-based platform for creating and running automated workflows.
- **Container Instances (PaaS)**. A fast and simple way to run a container in Azure. You don't provision any virtual machines and don't need a higher-level service.
- **App Service (PaaS)**. A managed service for hosting web apps, mobile app back ends, RESTful APIs, or automated business processes.
- **Azure Kubernetes Service (PaaS).** A managed Kubernetes service for running containerized applications.
- **Azure Service Fabric.** A distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and containers.

This flowchart provides high-level guidance on when to select each compute option. You'll want to refer to this diagram as we go through the choices.

> **Note**
>
> This diagram has been edited to include only the services covered in this module.

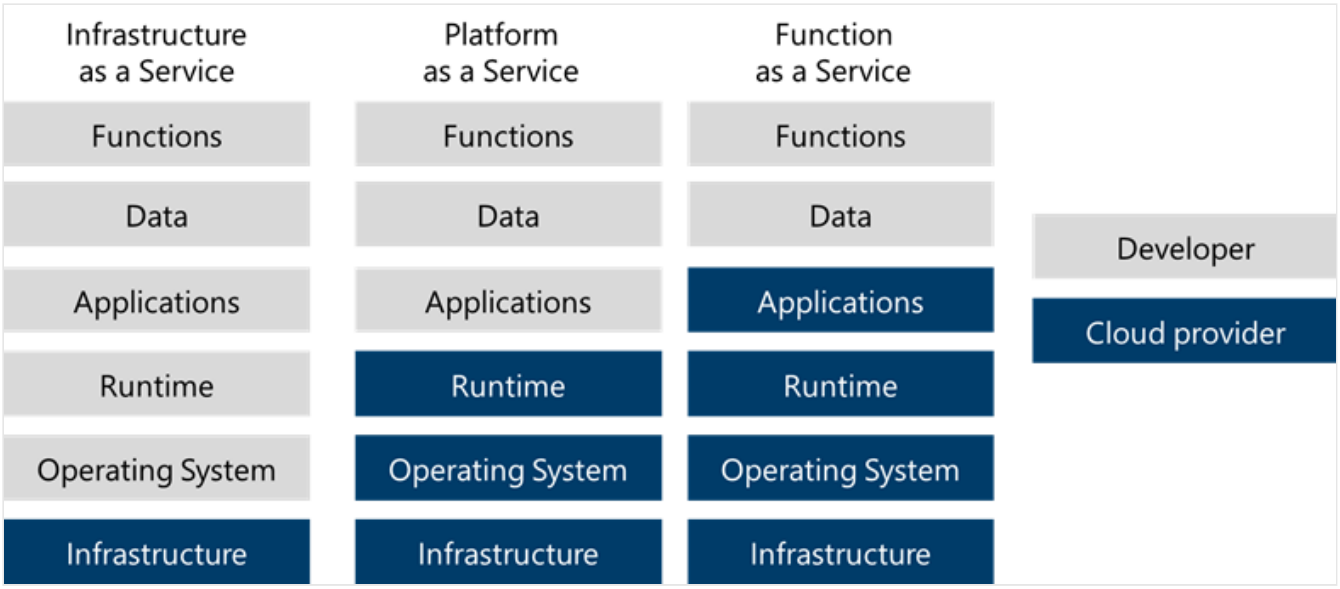


On the diagram, **Cloud optimized** is a strategy for migrating to the cloud. Cloud optimized refactors an application to take advantage of cloud-native features and capabilities. A **lift and shift** strategy migrates workloads without redesigning the application or making code changes. Lift-and-shift lets organizations keep running their applications with minimal changes and disruption.

> **Tip**
>
> The output from this flowchart is a **starting point** for consideration. You'll need to do a more detailed evaluation of the service to determine if it meets your needs. The next sections will help with this analysis.

Review the compute hosting options

The compute solution has three hosting options: Infrastructure as a Service, Platform as a Service, and Function as a Service? There's also Software-as-a-Service which isn't a compute solution. The hosting option determines the developer and cloud provider responsibilities. This hosting decision will influence your design.



- **Infrastructure-as-a-Service** (IaaS) lets you create individual VMs along with the associated networking and storage components. Then you deploy whatever software and applications you want onto those VMs. This model is the closest to a traditional on-premises environment, except that Microsoft manages the infrastructure. You still manage the individual VMs.
- **Platform-as-a-Service** (PaaS) provides a managed hosting environment, where you can deploy your application without needing to manage VMs or networking resources. Azure App Service is a PaaS service.
- **Functions-as-a-Service** (FaaS) goes even further in removing the need to worry about the hosting environment. In a FaaS model, you deploy your code, and the service automatically runs it. Azure Functions is a FaaS service.

Next unit: Design for Azure virtual machine solutions

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure virtual machine solutions

3 minutes

Whether you're building new or migrating using a lift and shift pattern, Azure virtual machines (VMs) might be a choice for you. Azure VMs are the basis of the Azure Infrastructure-as-a-Service (IaaS) model . Azure VMs can be used for the development, testing, deployment of applications in the cloud, or extension of your data center. Azure VMs provide a fast, scalable, flexible way to add more compute power to your enterprise.

There are two main scenarios for deciding to use virtual machines.

- **Build new** because demand for your application can fluctuate. It makes economic sense to run it on a VM in Azure.
- **Lift and shift (rehosting)** migration strategy that involves moving data and applications from an on-premises location to Azure-based virtual machines in the cloud.

Let's walk through a checklist of things to think about when designing for Azure VMs.

- Start with the network
- Name the VM
- Decide the location for the VM
- Determine the size of the VM
- Review the pricing model
- Review the storage options
- Select an operating system

Start with the network

The first thing to think about isn't the virtual machine at all - it's the network. So, spend some time thinking about your network configuration. Network addresses and subnets aren't trivial to change once you have them set up. If you have an on-premises network, you'll want to carefully consider the network topology before creating any virtual machines.

Name the virtual machine

One thing people don't put much thought into is the **name** of the VM. This name defines a manageable **Azure resource**, and it's also not easy to change. Choose names that are meaningful and consistent so you can easily identify what the VM does. For example, devusc-webvm01 might represent the first development web server hosted in the US South Central location.

Decide the location for the VM

Azure has data centers all over the world filled with servers and disks. These datacenters are grouped into geographic regions ('West US', 'North Europe', 'Southeast Asia' ...) to provide redundancy and availability.

Each virtual machine is in a region where you want the resources (CPU, storage ...) to be allocated. Regional location lets you place your VMs as close as possible to your users. This location can improve performance and ensure you meet any legal, compliance, or tax requirements.

Two other things to think about the location choice. First, the location can limit your available options. Each region has different hardware available, and some configurations aren't available in all regions. Second, there are price differences between locations. To find the most cost-effective choice, check for your required configuration in different regions.

Determine the size of the VM

Once you have the name and location set, you need to decide on the size of your VM. Azure offers different memory and storage options for different VM sizes.

The best way to determine the appropriate VM size is to consider the type of workload your VM needs to run. Based on the workload, you're able to choose from a subset of available VM sizes. Azure virtual machine workloads are classified as follows.

Option	Description

Option	Description
General purpose	General-purpose VMs are designed to have a balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute optimized	Compute optimized VMs are designed to have a high CPU-to-memory ratio. Suitable for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	Memory optimized VMs are designed to have a high memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	Storage optimized VMs are designed to have high disk throughput and IO. Ideal for VMs running databases.
GPU	GPU VMs are specialized virtual machines targeted for heavy graphics rendering and video editing. These VMs are ideal options for model training and inferencing with deep learning.
High performance computes	High performance compute is the fastest and most powerful CPU virtual machines with optional high-throughput network interfaces.

Tip

Try the **Virtual machines selector tool** to find other sizes that best fit your workload.

Review the pricing model

There are two separate costs the subscription will be charged for every VM: compute and storage. By separating these costs, you can scale them independently and only pay for what you need.

- **Compute costs** - Compute expenses are priced on a per-hour basis but billed on a per-minute basis. For example, you're only charged for 55 minutes of usage if the VM is deployed for 55 minutes. You're not charged for compute capacity if you stop and deallocate the VM. The hourly price varies based on the VM size and OS you select.

- **Storage costs** - You're charged separately for the storage the VM uses. The status of the VM has no relation to the storage charges that will be incurred. You are always charged for storage used by the disks.

Review the storage options

Managed disks handle Azure storage account creation and management in the background for you. You specify the disk size and the performance tier (Standard or Premium), and Azure creates and manages the disk. As you add disks or scale the VM up and down, you don't have to worry about the storage being used.

Select an operating system

Azure provides various OS images that you can install into the VM, including several versions of Windows and flavors of Linux. Azure bundles the cost of the OS license into the price.

If you're looking for more than just base OS images, you can search the Azure Marketplace . There are various install images that include not just the OS but popular software tools. For example, there is an image for WordPress. The image stack consists of a Linux server, Apache web server, a MySQL database, and PHP. So, instead of setting up and configuring each component, you can install a Marketplace image and get the entire stack all at once.

Lastly, if you can't find a suitable OS image, you can create your own disk image. Your disk image can then be uploaded to Azure storage and used to create an Azure VM. Keep in mind that Azure only supports 64-bit operating systems.

Important: There's a lot to think about when planning for virtual machines. Take a few minutes to think through what you have learned? Will you need virtual machines? If so, what decisions will you make on size, pricing, and operating systems?

Next unit: Design for Azure Batch solutions

Continue >

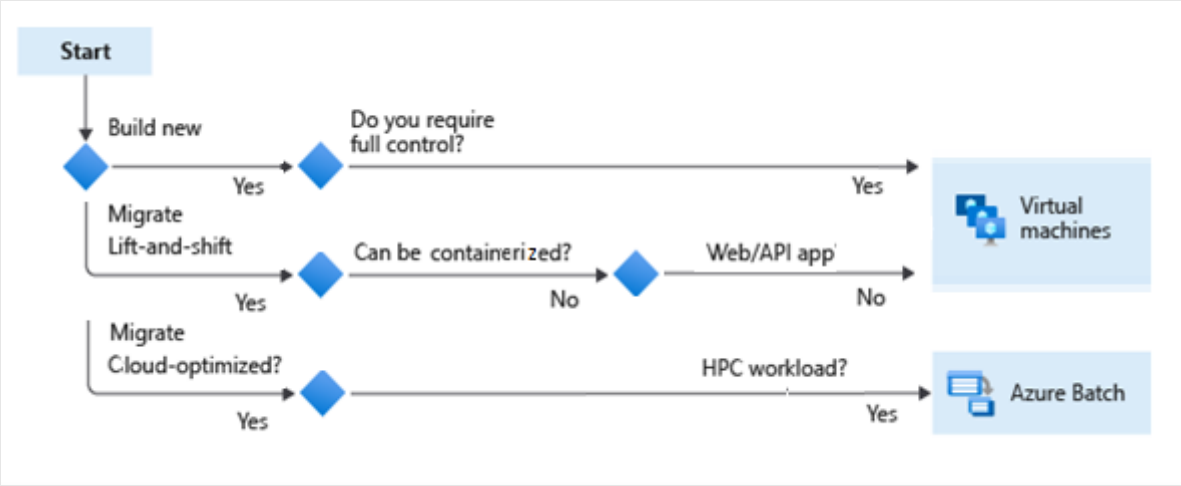
How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Batch solutions

3 minutes

Azure Batch runs large-scale applications efficiently in the cloud. You can schedule compute-intensive tasks and dynamically adjust resources for your solution without managing infrastructure. Azure Batch can create and manage a pool of compute nodes (virtual machines). Azure Batch can also install the application that you want to run, and schedule jobs to run on the compute nodes.



When to use Azure Batch

Azure Batch works well with applications that run independently (parallel workloads). Azure Batch is also effective for applications that need to communicate with each other (tightly coupled workloads). For example, you can use Batch to build a service that runs a Monte Carlo simulation for a financial services company or a service to process images.

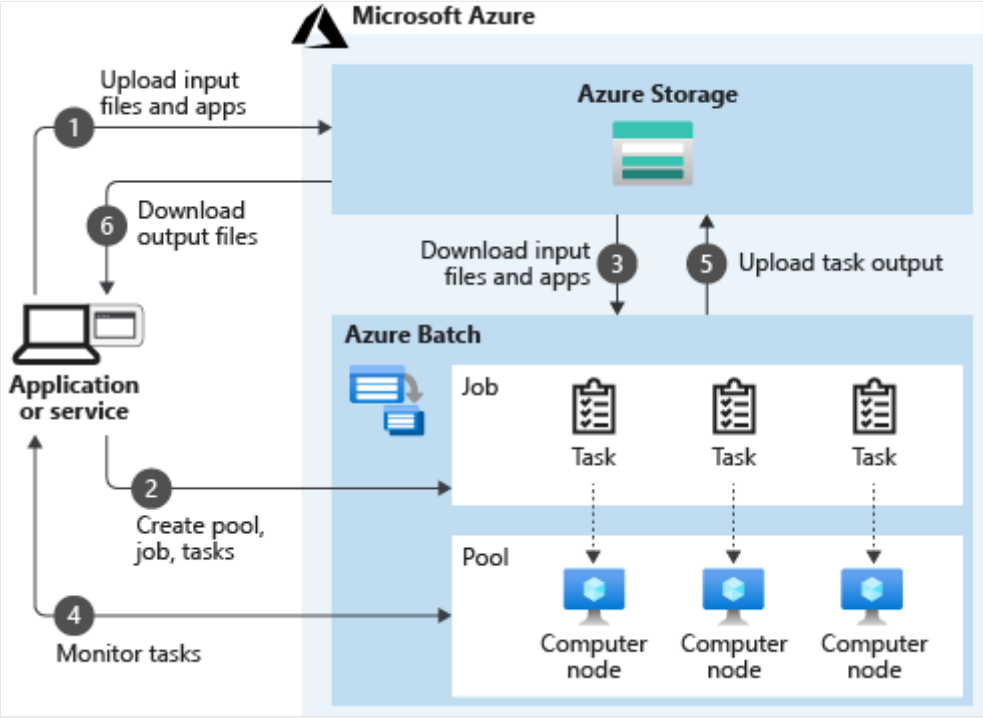
Azure Batch enables large-scale parallel and high-performance computing (HPC) batch jobs with the ability to scale to tens, hundreds, or thousands of VMs. When you're ready to run a job, Batch does the following.

- 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.

How Azure Batch works

As shown in the following diagram, a typical real-world scenario for Azure Batch requires data and application files. The Batch workflow begins with uploading the data and application files to an Azure storage account. Based on the demand, you create a Batch pool with as many Windows or Linux virtual compute nodes as needed. If the demand increases, compute nodes can be automatically scaled.



You can think of the diagram in two parts:

- **Your service** that uses Azure as the platform. The platform is for completing computationally intensive work and then retrieving results. You can also monitor jobs and task progress.
- **Batch as the compute platform behind your service.** Batch uses Azure Storage to fetch applications or data needed to complete a task. Azure Batch writes output to Azure storage. Behind the scenes, there are collections (pools) of virtual machines. Pools are the resources that jobs, and tasks are executed on.

Best practices and useful tips for using the Azure Batch service

Best practices for Azure Batch are grouped into pools, nodes, and jobs.

- **Pools.** If your jobs consist of short-running tasks, don't create a new pool for each job. The overhead to create new pools will diminish the run time of the job. Also, it's best to have your jobs use pools dynamically. If your jobs use the same pool for everything, there's a chance that jobs won't run if something goes wrong with the pool.
- **Nodes.** Individual nodes aren't guaranteed to always be available. If your Batch workload requires deterministic, guaranteed progress, you should allocate pools with multiple nodes. Consider using isolated VM sizes for workloads with compliance or regulatory requirements.
- **Jobs.** Uniquely name your jobs so you can accurately monitor and log the activity. Consider grouping your tasks into efficiently sized jobs. For example, it's more efficient to use a single job containing 1000 tasks rather than creating 100 jobs that contain 10 tasks each.

> **Tip**
>
> We've covered just a few best practices. Take a few minutes to read more about **Best practices - Azure Batch | Microsoft Docs**.

Next unit: Design for Azure App Services solutions

Continue >

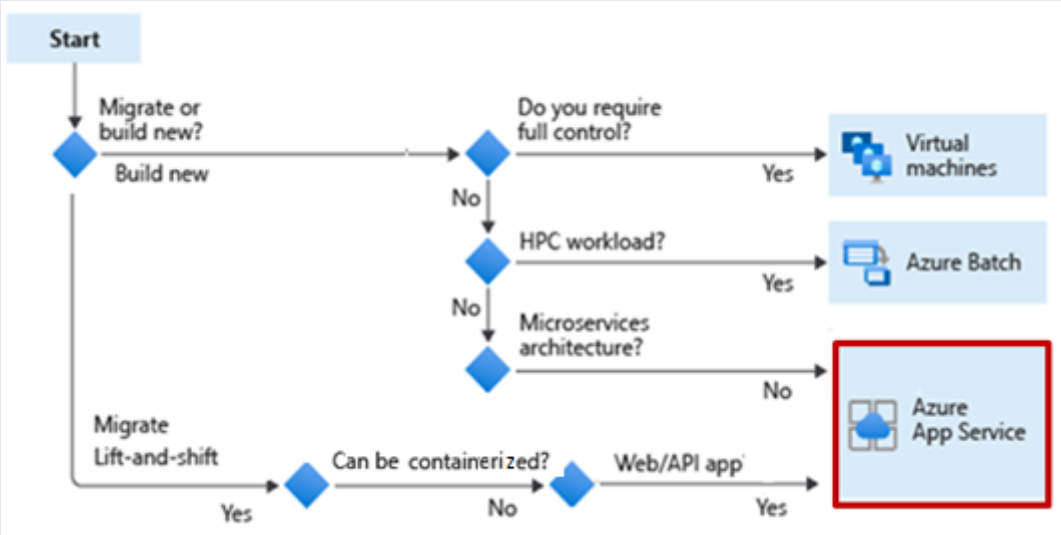
How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure App Services solutions

3 minutes

Azure App Service is an HTTP-based service that lets you build and host web apps, background jobs, mobile backends, and RESTful APIs. App Service lets you use the programming language of your choice. Azure App Services offers automatic scaling and high availability. App Service enables automated deployments from GitHub, Azure DevOps, or any Git repo.



Important: Azure App Service is platform as a service (PaaS) environment. You focus on the website development and API logic. Azure handles the infrastructure to run and scale your web applications.

Types of app services

With Azure App Service, all your apps share common benefits including:

- Development in multiple languages and frameworks.
- Integrated deployment and management with secured endpoints.
- Global scale with high availability.
- Built-in load balancing and traffic management.

These benefits make App Service the ideal choice for any hosted web application.

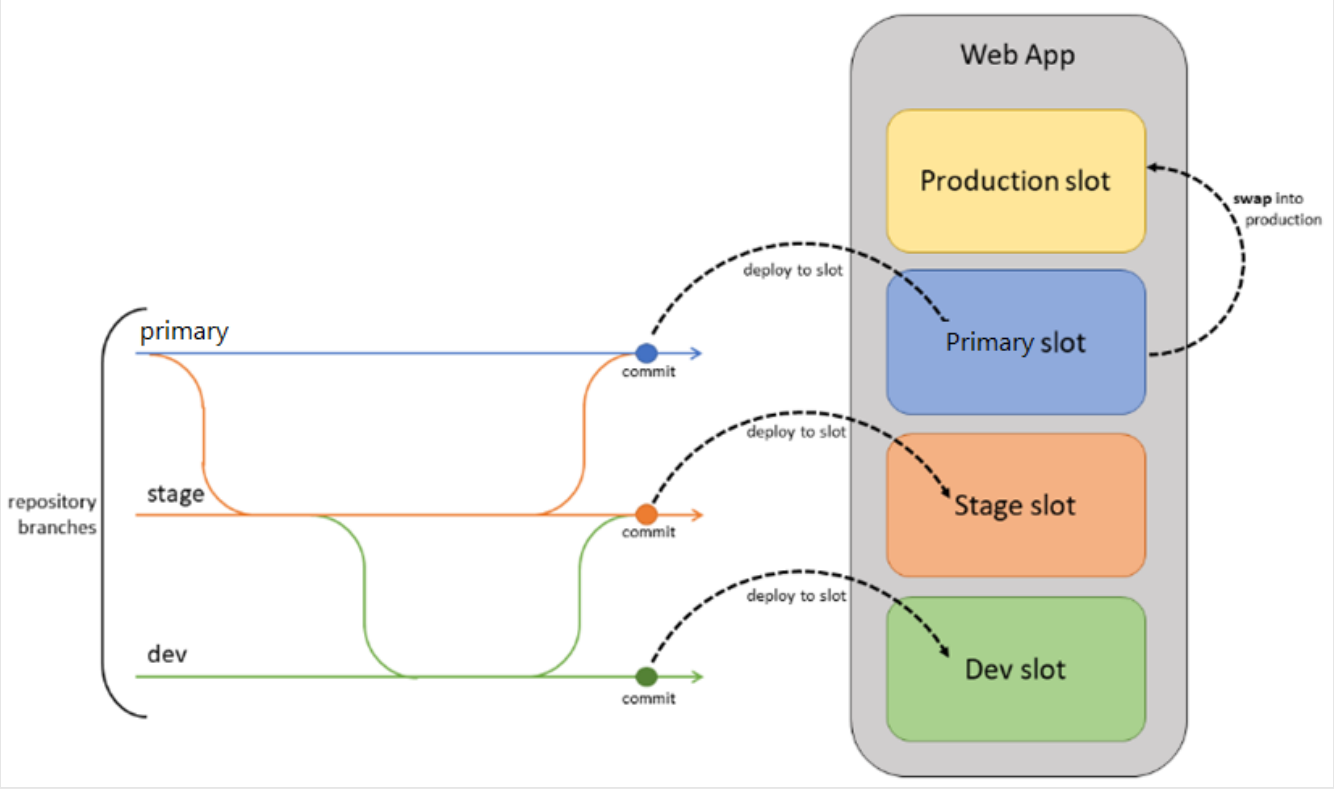
Azure App Service costs

You pay for the Azure compute resources your app uses while it processes requests. The cost is 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. You can have different app service plans for different apps.

Your App Service plan can be scaled up and down at any time. For example, you can start testing your web app in a Free App Service plan and pay nothing. When you want to add your custom DNS name to the web app, just scale your plan up to the Shared tier.

Use App Services deployment slots for continuous deployment

Azure DevOps provides developer services for support teams to plan work, collaborate on code development, and build and deploy applications. Whenever possible when continuously deploying your code, use deployment slots for a new production build.



When using a Standard App Service Plan tier or better, you can deploy your app to a staging environment, validate your changes, and do smoke tests. When you're ready, you can swap your staging and production slots. The swap operation warms up the necessary worker instances to match your production scale, thus eliminating downtime.

Consider authentication and authorization options

Implementing a secure solution for authentication (signing-in users) and authorization (providing access to secure data) can take significant effort. Azure App Service provides built-in authentication and authorization capabilities (sometimes referred to as "Easy Auth"). So, you can sign in users and access data by writing minimal or no code. Here are some benefits.

- Azure App Service provides built-in auth capabilities for your web app or API. You don't need to implement the authentication yourself.
- It's built directly into the platform. You don't need any language, SDK, security expertise, or even any code to utilize.
- You can integrate with multiple sign-in providers. For example, Azure AD, Facebook, Google, and Twitter.

> **Tip**
>
> The built-in authentication features for App Service is the same for Azure Functions.

When to use web apps

App Service supports web apps 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.

When to use 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.

When to use WebJobs

You can use the WebJobs feature to run a program or script. Program examples include Java, PHP, Python, or Node.js. Script examples include cmd, bat, PowerShell, or Bash. WebJobs can be scheduled or run by a trigger. WebJobs are often used to run background tasks as part of your application logic.

When to use 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 steps 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.

Next unit: Design for Azure Container Instances solutions

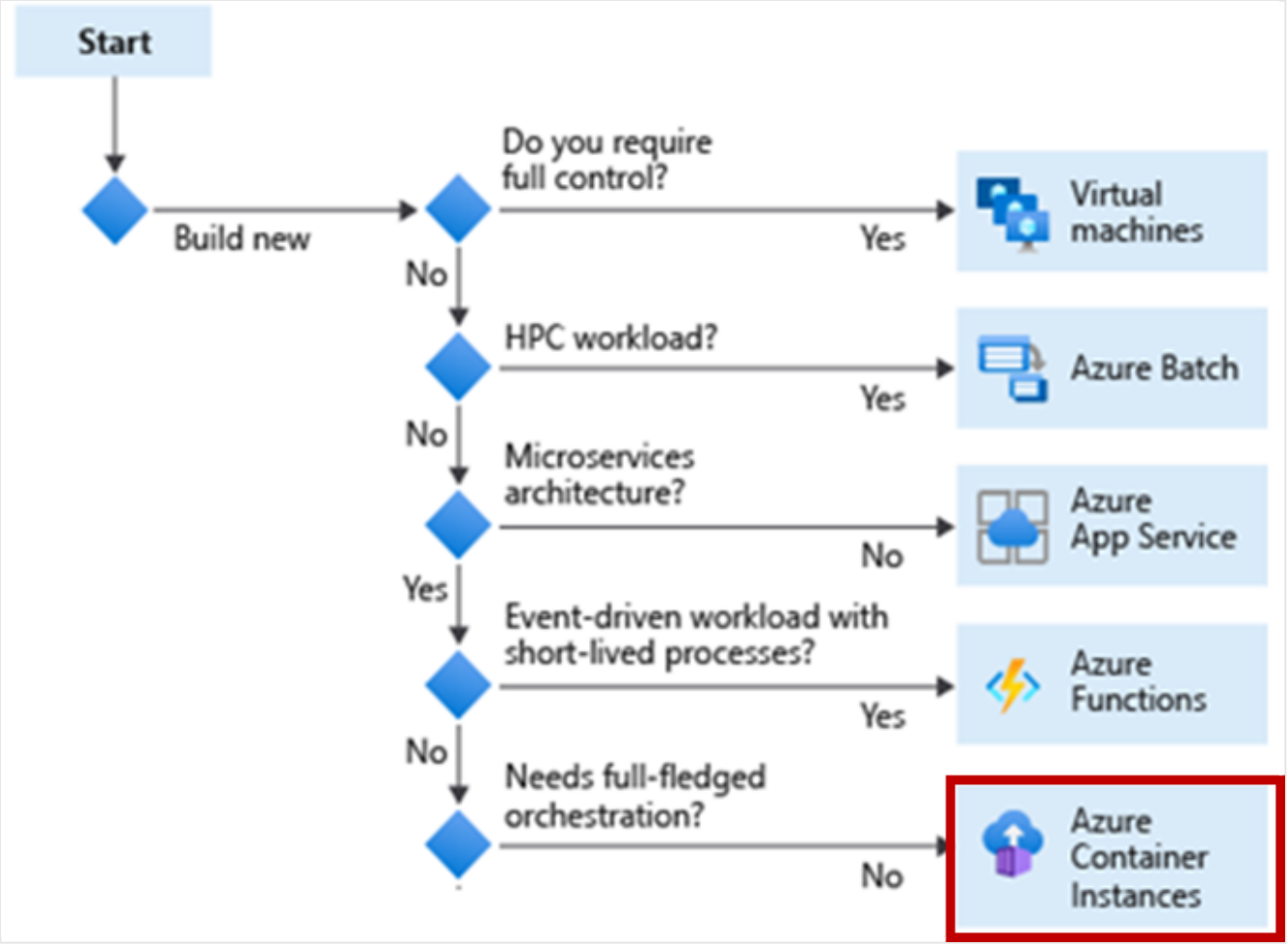
Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Container Instances solutions

3 minutes



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

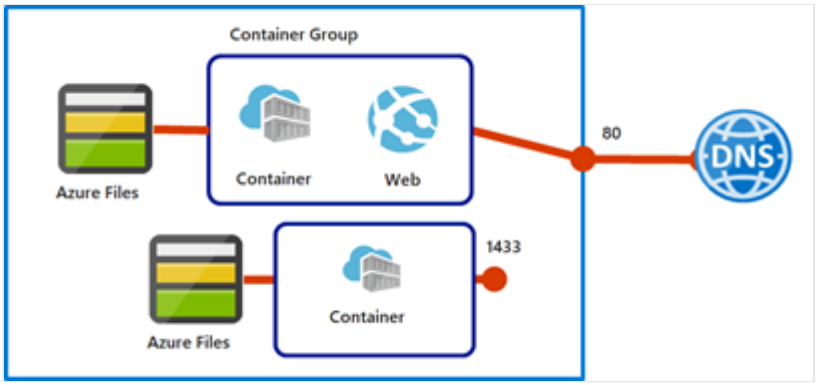
Azure Container Instances are a fast and simple way to run a container on Azure. Azure Container Instance scenarios include simple applications, task automation, and build jobs. Here are some benefits of containers.

- **Fast startup**. Launch containers in seconds.
- **Per second billing**. Incur costs only while the container is running.

- **Hypervisor-level security**. Isolate your application as completely as it would be in a VM.
- **Custom sizes**. Specify exact values for CPU cores and memory.
- **Persistent storage**. Mount Azure Files shares directly to a container to retrieve and persist state.
- **Linux and Windows**. Schedule both Windows and Linux containers using the same API.

What are container groups?

The top-level resource in Azure Container Instances is the container group. A container group is a collection of containers that get scheduled on the same host machine. The containers in a container group share a lifecycle, resources, local network, and storage volumes.



Multi-container groups are useful in cases where you want to divide a single functional task into several container images. These images can then be delivered by different teams and have separate resource requirements. Example usage could include:

- A container serving a web application and a container pulling the latest content from source control.
- An application container and a logging container. The logging container collects the logs and metrics output by the main application and writes them to long-term storage.
- An application container and a monitoring container. The monitoring container periodically makes a request to the application to ensure that it's running and responding correctly and raises an alert if it's not.
- A front-end container and a back-end container. The front end might serve a web application, with the back end running a service to retrieve data.

Security considerations for container instances

When working with container instances, consider these security best practices.

- **Use a private registry.** Containers are built from images that are stored in one or more repositories. These repositories can belong to a public registry or to a private registry. An example of a private registry is the Docker Trusted Registry , which can be installed on-premises or in a virtual private cloud. Another example is Azure Container Registry to build, store, and manage container images and artifacts.

- **Ensure the integrity of images throughout the lifecycle.** Part of managing security throughout the container lifecycle is to ensure the integrity of the container images. Images with vulnerabilities, even minor, shouldn't be allowed to run in a production environment. Keep the number of production images small to ensure that they can be managed effectively.

- **Monitor container resource activity**. Monitor your resource activity, like files, network, and other resources that your containers access. Monitoring resource activity and consumption is useful both for performance monitoring and as a security measure.

> **Tip**
>
> Read more about **Security considerations for container instances - Azure Container Instances | Microsoft Docs**

When to choose containers instead of virtual machines

Feature	Containers	Virtual Machines
Isolation	Typically provides lightweight isolation from the host and other containers but doesn't provide as strong a security boundary as a virtual machine.	Provides complete isolation from the host operating system and other VMs. Isolation is useful when a strong security boundary is critical, such as hosting apps from competing companies on the same server or cluster.
Operating system	Runs the user mode portion of an operating system and can be tailored to contain just the needed services for your app, using fewer system resources.	Runs a complete operating system. Typically, requires more system resources (CPU, memory, and storage).

Feature	Containers	Virtual Machines
Deployment	Deploy individual containers by using Docker via command line; deploy multiple containers by using an orchestrator such as Azure Kubernetes Service.	Deploy individual VMs by using Windows Admin Center or Hyper-V Manager; deploy multiple VMs by using PowerShell or System Center Virtual Machine Manager.
Persistent storage	Use Azure Disks for local storage for a single node, or Azure Files (SMB shares) for storage shared by multiple nodes or servers.	Use a virtual hard disk (VHD) for local storage for a single VM, or an SMB file share for storage shared by multiple servers.
Fault tolerance	If a cluster node fails, any containers running on it are rapidly recreated by the orchestrator on another cluster node.	VMs can fail over to another server in a cluster, with the VM's operating system restarting on the new server.

Next unit: Design for Azure Kubernetes solutions

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

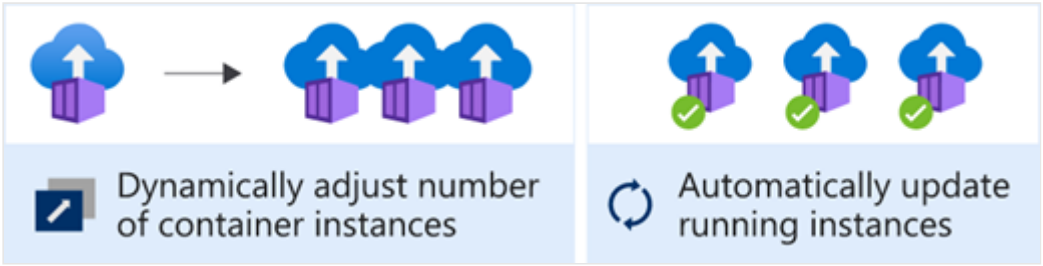
100 XP

Design for Azure Kubernetes solutions

3 minutes

Kubernetes is a portable, extensible open-source platform for automating deployment, scaling, and the management of containerized workloads. This orchestration platform gives us the same ease of use and flexibility as with Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) offerings. Kubernetes provides both container management and container orchestration.

- Container management is the process of organizing, adding, removing, or updating a significant number of containers. Most of these tasks are manual and error prone.
- Container orchestration is a system that automatically deploys and manages containerized apps. For example, the orchestrator can dynamically increase or decrease the deployed instances of the managed app. Or it can ensure all deployed container instances get updated if a new version of a service is released.



What is Azure Kubernetes Services (AKS)?

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment and makes it simple to deploy and manage containerized applications in Azure. Your AKS environment is enabled with features such as automated updates, self-healing, and easy scaling.

- The Kubernetes cluster is managed by Azure and is free. You manage the agent nodes in the cluster and only pay for the VMs on which your nodes run.

- When you create the cluster, you can use Resource Manager templates to automate cluster creation. With these templates, you specify features such as advanced networking, Azure Active Directory (AD) integration, and monitoring.
- With AKS, we get the benefits of open-source Kubernetes. You don’t have the complexity or operational overhead running your own custom Kubernetes cluster.

When to use Azure Kubernetes Service

Here, we'll discuss how you can decide whether Azure Kubernetes Service (AKS) is the right choice for you.

You'll either approach your decision from a green field or a lift-and-shift project point of view. A green fields project will allow you to evaluate AKS based on default features. A lift-and-shift project will require you to determine which features are best suited to support your migration. Here are a few factors to consider.

Factor	Things to consider
Identity and security management	Do you already use existing Azure resources and make use of Azure AD? If so, you can configure an AKS cluster to integrate with Azure AD and reuse existing identities and group membership.
Integrated logging and monitoring	Are you using Azure Monitor? If so, Azure Monitor provides performance visibility of the cluster.
Automatic cluster node and pod scaling	Do you need to scale up or down a large containerization environment? If so, AKS supports two auto cluster scaling options. The horizontal pod autoscaler watches the resource demand of pods and will increase pods to match demand. The cluster autoscaler component watches for pods that can't be scheduled because of node constraints. It will automatically scale cluster nodes to deploy scheduled pods.
Cluster node upgrades	Do you want to reduce the number of cluster management tasks? If so, AKS manages Kubernetes software upgrades and the process of cordoning off nodes and draining them.
Storage volume support	Does your application require persisted storage? If so, AKS supports both static and dynamic storage volumes. Pods can attach and reattach to these storage volumes as they're created or rescheduled on different nodes.

Factor	Things to consider
Virtual network support	Do you need pod to pod network communication or access to on-premises networks from your AKS cluster? If so, an AKS cluster can be deployed into an existing virtual network with ease.
Ingress with HTTP application routing support	Do you need to make your deployed applications publicly available? If so, the HTTP application routing add-on makes it easy to access AKS cluster deployed applications.
Docker image support	Do you already use Docker images for your containers? If so, AKS by default, supports the Docker file image format.
Private container registry	Do you need a private container registry? If so, AKS integrates with Azure Container Registry (ACR). You aren't limited to ACR though, you can use other container repositories, public, or private.

All the above features are configurable either when you create the cluster or following deployment.

> **Tip**
>
> Take a few minutes to read about how **Mercedes-Benz R&D is using AKS** .

Next unit: Design for Azure Functions solutions

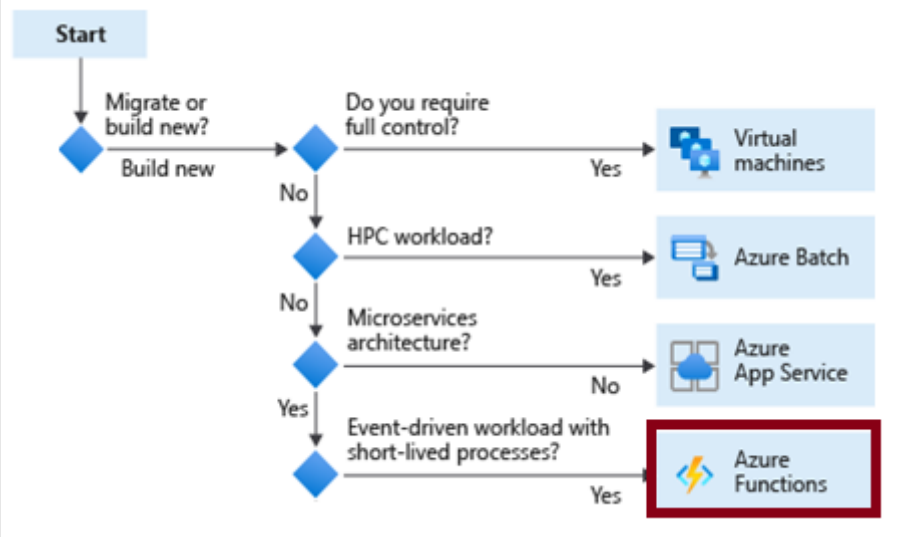
Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Functions solutions

3 minutes



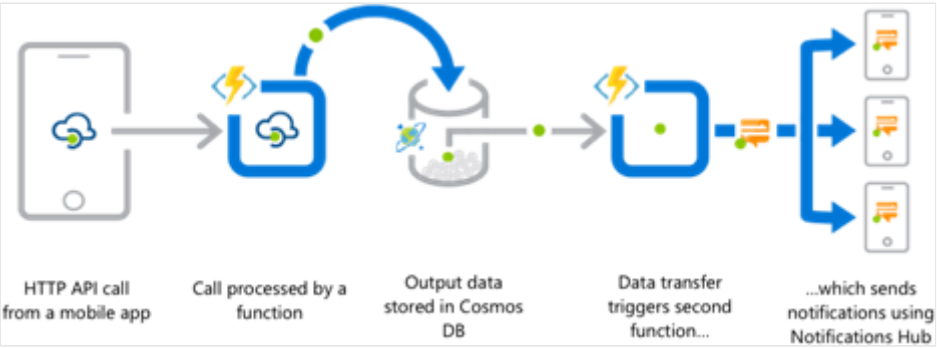
What are Azure Functions?

Azure Functions is a serverless application platform. Functions are used when you want to run a small piece of code in the cloud, without worrying about the infrastructure. Functions provide intrinsic scalability, and you're charged only for the resources used. You can write your function code in the language of your choice. Functions provide "compute on demand" in two significant ways.

- First, Azure Functions allows you to implement your system's logic into readily available blocks of code. These code blocks (functions) can run anytime you need to respond to critical events.
- Second, as requests increase, Azure Functions meets the demand with as many resources and function instances as necessary. As requests complete, any extra resources and application instances drop off automatically.

Scenarios for Azure Functions

Azure Functions are best when handling specific definable actions triggered by an event. For example, a function could process an API call and then store the processed data in Cosmos DB. Once the data transfer happens, another function could trigger a notification.

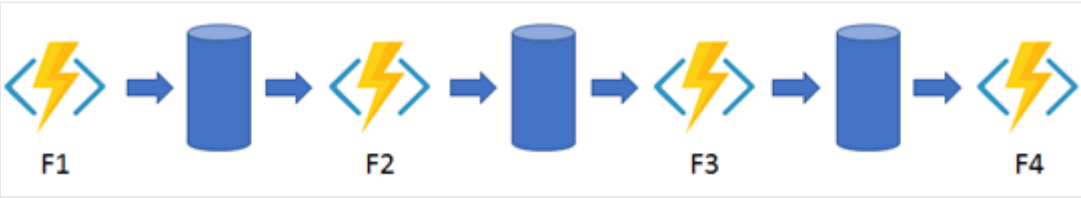


> **Tip**
>
> You can get some other ideas on how to use Azure Functions by visiting the **code samples** page.

Best practices and tips for using Azure Functions

- **Avoid long running functions.** Large, long-running functions can cause unexpected timeout issues. Whenever possible, refactor large functions into smaller function sets that work together and return responses faster. The default timeout is 300 seconds for Consumption Plan functions, 30 minutes for any other plan.

- **Know when to use durable functions**. Durable functions let you write stateful functions. So, behind the scenes, the function manages app state, checkpoints, and restarts. An example application pattern for durable functions is function chaining. Function chaining executes a sequence of functions in a specific order. The output of one function is applied to the input of another function. Do you understand how timeout issues can be overcome with durable functions and smaller function sets?



- **Organize functions for performance and scaling.** Consider how you want to group functions with different load profiles. For example, let's say you have two functions. One function processes many thousands of queued messages and has low memory requirements. The other function is only called occasionally but has high memory requirements. You might want to deploy separate function apps, so each function gets its own set of resources. Separate resources mean you can independently scale the functions.

- **Write defensive functions.** Design your functions assuming an exception could occur at any time. Downstream services, network outages, or memory limits could cause the function to fail. Plan out how you continue from a failure point.
- **Avoid sharing storage accounts.** When you create a function app, you must associate it with a storage account. To maximize performance, use a separate storage account for each function app. This is important if your function generates a high volume of storage transactions.

> **Tip**
>
> Take a few minutes to read about other **Azure Function best practices**.

Next unit: Design for Logic App solutions

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Logic App solutions

3 minutes

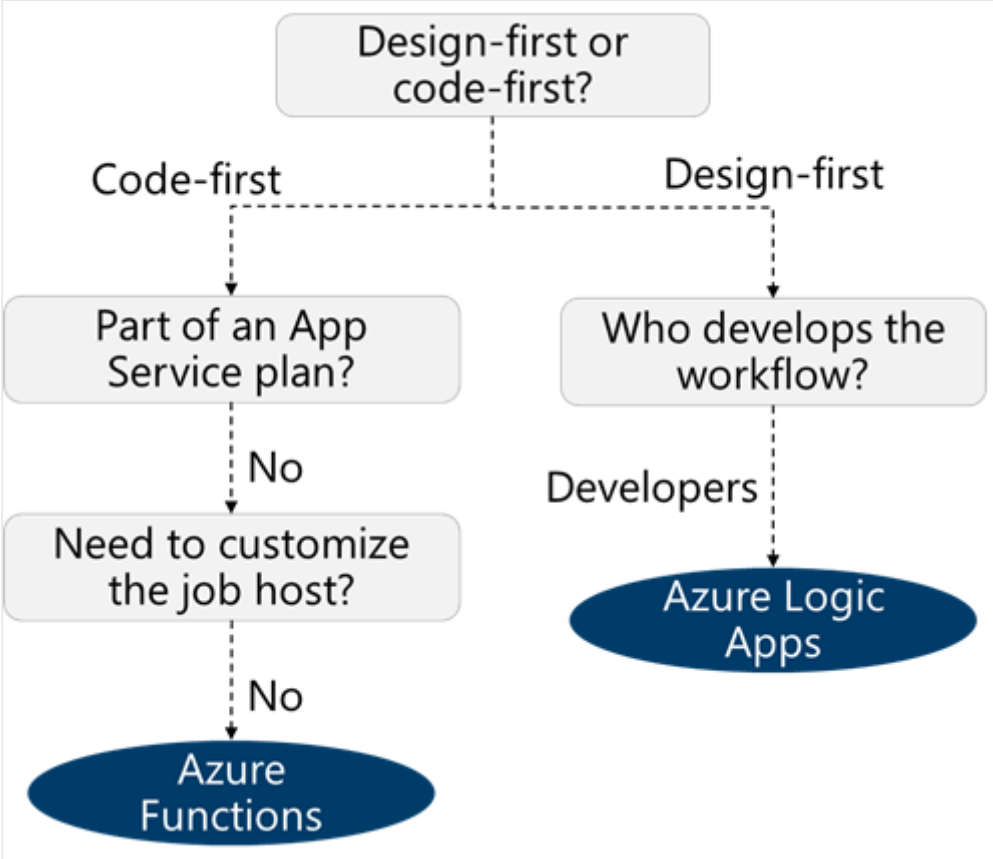
Azure Logic Apps is another type of serverless compute solution. Azure Logic Apps is a cloud-based platform for creating and running automated workflows. Workflows are step-by-step processes that integrate your apps, data, services, and systems. With Azure Logic Apps, you can quickly develop highly scalable integration solutions for your enterprise and business-to-business (B2B) scenarios.

Logic Apps is a member of Azure Integration Services. Logic Apps simplifies the way that you connect legacy, modern, and cutting-edge systems across cloud, on premises, and hybrid environments. The following list describes just a few example tasks, business processes, and workloads that you can automate using the Logic Apps service.

- Schedule and send email notifications using Office 365 when a specific event happens. For example, a new file is uploaded.
- Route and process customer orders across on-premises systems and cloud services.
- Move uploaded files from an SFTP or FTP server to Azure Storage.
- Monitor tweets, analyze the sentiment, and create alerts or tasks for items that need review.

How are Azure Logic Apps and Azure Functions different?

Azure Logic Apps and Azure Functions may seem similar but there are basic differences. Azure Functions is a code-first technology. Azure Logic Apps is a design-first technology.



Here are some other differences.

Comparison area	Durable Functions	Logic Apps
Development	Code-first	Designer-first
Method	Write code and use the durable functions extension	Create orchestrations by using a GUI or editing configuration files
Connectivity	Large selection of built-in binding types, write code for custom bindings	Large collection of connectors, Enterprise Integration Pack for B2B scenarios, build custom connectors
Monitoring	Azure Application Insights	Azure portal, Azure Monitor logs

Tip

You can mix and match services when you build an orchestration. You can call functions from logic apps and call logic apps from functions. Build each orchestration based on the services' capabilities or your personal preference.

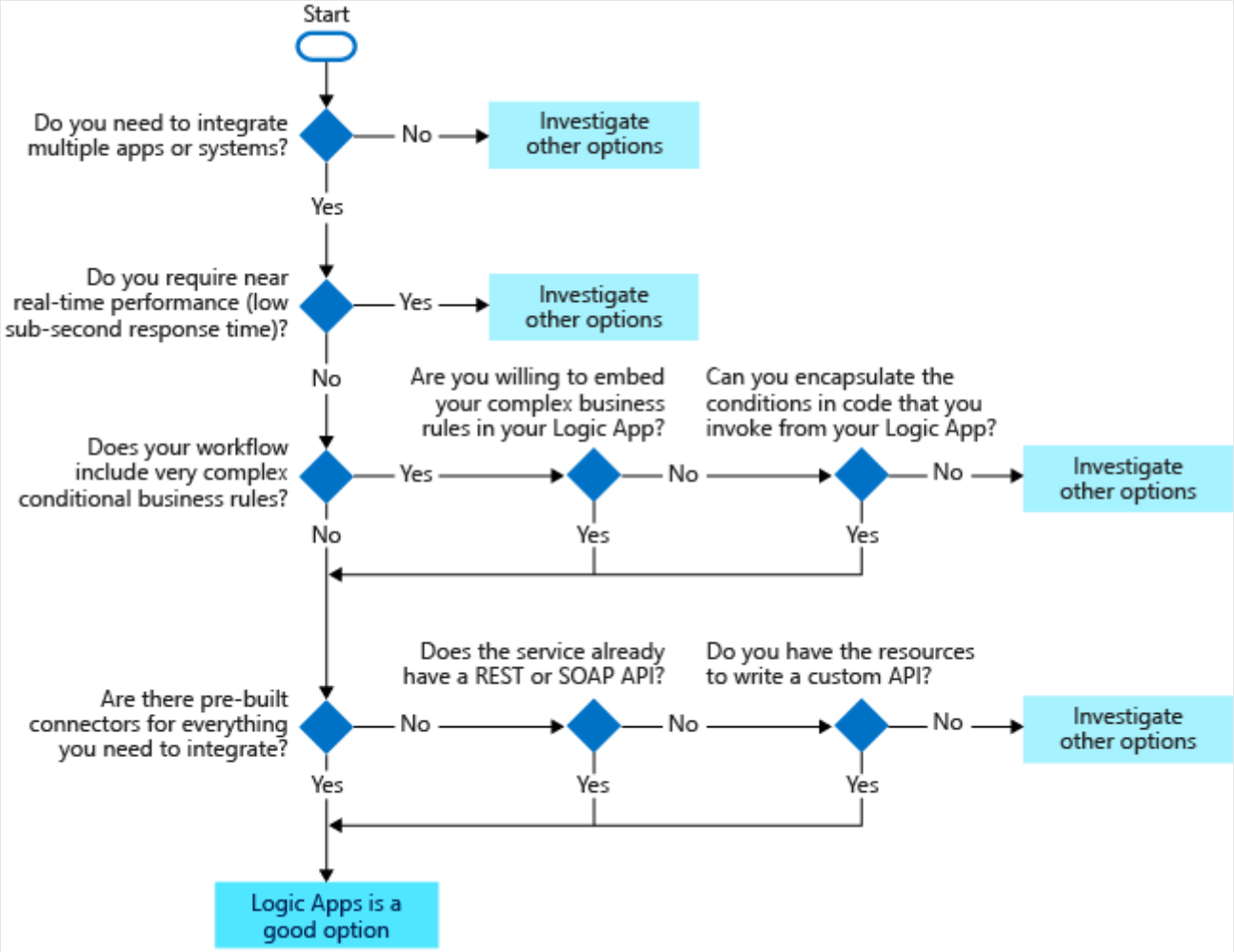
Decision criteria for Logic Apps

When designing for Logic Apps consider integration, performance, conditionals, and connectors.

- **Integration.** The key question to ask when you're considering Logic Apps is "do I need to integrate services?" Logic Apps work well when you need to get multiple applications and systems to work together. That's what they were designed to do. If you're building an app with no external connections, Logic Apps is probably not the best option.
- **Performance**. The next consideration is performance. The Logic Apps execution engine scales your apps automatically. Logic Apps can process large datasets in parallel to let you achieve high throughput. However, fast activation time is not always guaranteed, nor enforcement of real-time constraints on execution time.
- **Conditionals**. Logic Apps provides control constructs like Boolean expressions, switch statements, and loops so your apps can make decisions based on your data. You can build highly complex and deeply nested conditionals into your Logic Apps.
- **Connectors**. The last consideration is whether there are pre-built connectors for all the services you need to access. If so, then you're ready to go. If not, then you'll need to create a custom connector. If the service has an existing REST or SOAP API, you can make the custom connector in a few hours without writing any code. If not, then you'll need to create the API first before making the connector.

> **Tip**
>
> Knowing when not to use Logic Apps is also important. The cases where Logic Apps might not be the best option include real-time requirements, complex business rules, or use of non-standard services.

Summary of design criteria for logic apps



> **Note**
>
> Take a few minutes to learn about how **Cramo is using Logic Apps** in their new integration platform.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

200 XP

Knowledge check

3 minutes

Tailwind Traders has several development projects. It is important you select the right compute technology for each project. Ideally, you would like to create compute resources, configure them to do the work that's needed, and pay for only what you use. Here are the specific requirements.

- **Real-time inventory tracking.** The company website updates product availability only once a night. The company would like the inventory updated as products are ordered. This means updating the database and if necessary, sending reorder notifications. The current program is a Windows service that's written in C#.
- **Migrate the data center virtual machines**. The company's data center virtual machines host relational database servers. These machines are used for online orders. The company would like to move this capability to the cloud.
- **Host the data processing app.** The company has a small data processing app. The app ingests new product photos and writes the content to Azure blob storage. The app takes only a few seconds to run. The company would like to reduce costs and use the cloud.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. Which compute option is best to support the company's real-time inventory tracking requirement?

- ○ Azure Logic Apps
- ◉ Azure Functions

  ✔ **Correct. The Tailwind Traders developers' team has already written the logic in C#. It would make sense to copy the relevant C# code from the Windows service and port it to an Azure function. The developers would bind the function to trigger each time a new message appears on a specific queue.**
- ○ Azure virtual machines

2. Which type of virtual machine is best for the data center migration requirement?

- ○ General purpose

- ◉ Compute optimized

  ✗ **Incorrect. Compute optimized VMs are designed to have a high CPU-to-memory ratio. Suitable for medium traffic web servers, network appliances, batch processes, and application servers.**

- ○ Memory optimized

  ✓ **Correct. Memory optimized VMs are designed to have a high memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.**

3. What compute solution is best for hosting the company's data processing app?

- ◉ Azure Container Instances

  ✓ **Correct. This is an ideal use for containers. The company will achieve significant cost saving through per-second billing.**

- ○ Azure virtual machines
- ○ Azure functions.

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Introduction

3 minutes

The cloud is changing how applications are designed and secured. Instead of monoliths, applications are divided into smaller, decentralized services.

These services communicate through APIs or by using asynchronous messaging or events. The services scale horizontally, adding new instances as demand requires.

These design changes bring new challenges. Application states are distributed, and operations are done in parallel and asynchronously. Applications must:

- Communicate with each other effectively.
- Be able to be deployed rapidly.
- Be resilient when failures occur.
- Be able to integrate with other systems seamlessly.

Azure lets you create applications composed of various components:

- Website front ends
- Back-end services
- Triggered functions

Azure includes various communication strategies to let these various components pass data to each other.

After completing this module, you'll be able to evaluate and design an effective application architecture. This architecture provides the best Azure solutions for exchanging messages. It also helps automate deployment solutions for your applications and manage configurations. Azure also enables integration with APIs and provides appropriate caching.

Learning objectives

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

- Describe message and event scenarios.
- Design a messaging solution.
- Design an event hubs messaging solution.
- Design an event-driven solution.
- Design an automated app deployment solution.
- Design an API integration solution.
- Design an application configuration management solution.
- Design a caching solution.

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design Infrastructure solutions

- Design an Application Architecture.

Prerequisites

- Working experience with developing cloud applications.
- Conceptual knowledge of messaging, events, code deployments, configurations, API management, and app caching.

Next unit: Describe message and event scenarios

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Describe message and event scenarios

3 minutes

Imagine you're designing the architecture for a distributed music-sharing application. You want to ensure that the application is as reliable and scalable as possible. You want to use Azure technologies to build a robust communication infrastructure.

But before you can choose the Azure technology, you must understand how each individual component communicates with the other components of the application. For each communication, you can choose a different Azure technology.

Select messages or events for your application

The first thing you must understand is whether an app sends messages or events. Knowing the difference helps you choose the appropriate Azure service.

What is a message?

Messages have the following characteristics.

- Contains raw data, produced by one component, that will be consumed by another component.
- Contains the data itself, not just a reference to that data.

The sending component expects the message content to be processed in a certain way by the destination component. The integrity of the overall system may depend on both sender and receiver doing a specific job.

Example of a message

Let's suppose a user uploads a new song by using your mobile music-sharing app. The mobile app must send that song to the web API that runs in Azure. The song file must be sent, not just an alert that indicates that a new song has been added. The mobile app expects that the web API stores the new song in the database and makes it available to other users.

What is an event?

Events are lighter weight than messages and are most often used for broadcast communications. There are two components involved with events:

- Publishers, which send the event.
- Subscribers, which receive events.

With events, receiving components generally decide in which communications they're interested and subscribe to those events. The subscription is managed by an intermediary. The intermediary can be provided by services such as Azure Event Grid or Azure Event Hubs. When publishers send an event, the intermediary routes that event to interested. This pattern is known as a publish-subscribe architecture and is the most used.

Events have the following characteristics:

- Is a lightweight notification that indicates something occurred?
- May be sent to multiple receivers or to none.
- Is often intended to "fan out" or have many subscribers for each publisher.
- Publisher has no expectation about the action a receiving component takes.
- Is a discrete unit and unrelated to other events?
- Might be part of a related and ordered series.

When should you choose messages or events?

A single application is likely to use events for some purposes and messages for others. The following table describes when to use which:

Event or message	When to use
Event	More likely to be used for broadcasts and are often ephemeral. Ephemeral means the communication might not be handled by any receiver if none is currently subscribing.
Message	More likely to be used where the distributed application requires a guarantee that the communication will be processed.

For each communication, consider the following question: Does the sending component expect the communication to be processed in a particular way by the destination component?

- If yes, choose to use a message.
- If no, you might be able to use events.

Next unit: Design a messaging solution

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design a messaging solution

3 minutes

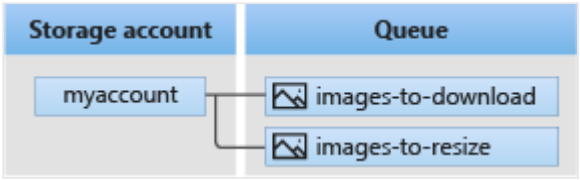
In this unit, you'll learn how to choose the best architecture for a message-based delivery system. Let's imagine you're planning the music-sharing application. You want to:

- Ensure that music files are uploaded to the web API reliably from the mobile app.
- Deliver the details about new songs directly to the app. For example, when an artist adds new music to their collection.

This scenario is a perfect use of a message-based system. Azure offers two message-based solutions Queue Storage and Azure Service Bus (queues and topics).

What is Azure Queue Storage?

Azure Queue storage is a service that uses Azure Storage to store large numbers of messages. These messages can be securely accessed from anywhere in the world using a simple REST-based interface. Queues can contain millions of messages. Azure Queue storage is limited only by the capacity of the storage account that owns it. Queues generally provide increased reliability, guaranteed message delivery, and transactional support.



What is Azure Service Bus?

Microsoft Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics. Service Bus is used to decouple applications and services from each other, providing the following benefits:

- Load-balancing work across competing workers.
- Safely routing and transferring data and control across service and application boundaries.

- Coordinating transactional work that requires a high degree of reliability.

What are Azure Service Bus Queues?

Azure Service Bus queues is a message broker system built on top of a dedicated messaging infrastructure. Like Azure queues, Service Bus holds messages until the target is ready to receive them.

Azure Service Bus is intended for enterprise applications. For example, an application that uses communication protocols and different data contracts.

What is an Azure Service Bus publish-subscribe topic?

Azure Service Bus topics are like queues but can have multiple subscribers. When a message is sent to a topic, multiple components can be triggered to perform a task.

For example, suppose a user is listening to a song using a music-sharing application. The mobile app might send a message to the *Listened* topic. That topic could have a subscription for *UpdateUserListenHistory*, and a different subscription for *UpdateArtistsFanList*. Each subscription receives its own copy of the message.

Which messaging service should I choose?

Each messaging product has a slightly different feature set. This means you can choose one or the other or use both. It depends on the problem you're solving.

Use Azure Queue storage if you need/have:

- A simple queue to organize messages.
- An audit trail of all messages that pass through the queue.

- Queue to exceed 80 GB in size.
- To track progress for processing a message inside of the queue.

Use Azure Service Bus queues if you need/have:

- An At-Most-Once delivery guarantee.
- At-Least-Once message processing (PeekLock receive mode)
- At-Most-Once message processing (ReceiveAndDelete receive mode)
- To group messages into transactions.
- To receive messages without polling the queue.
- To handle messages larger than 64 KB but less than 256 KB.
- Queue size will not grow larger than 80 GB.
- To publish and consume batches of messages.

Use Azure Service Bus topics if you need/have:

- Multiple receivers to handle each message.
- Multiple destinations for a single message but need queue-like behavior.

Next unit: Design an Event Hubs messaging solution

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design an Event Hubs messaging solution

3 minutes

There are certain applications that produce a massive number of events from almost as many sources. We often refer to this as Big Data. Big Data can require extensive infrastructure.

Let's imagine you work for Contoso Aircraft Engines. The engines your employer manufactures have hundreds of sensors. Before an aircraft can be flown, its engines are connected to a test harness and put through their paces. Additionally, cached in-flight data is streamed when the aircraft is connected to ground equipment.

In this situation, you might choose an event hubs-based messaging solution. Event hubs can receive and process millions of events per second. Data sent to an event hubs can be transformed in real-time and stored for later analysis.

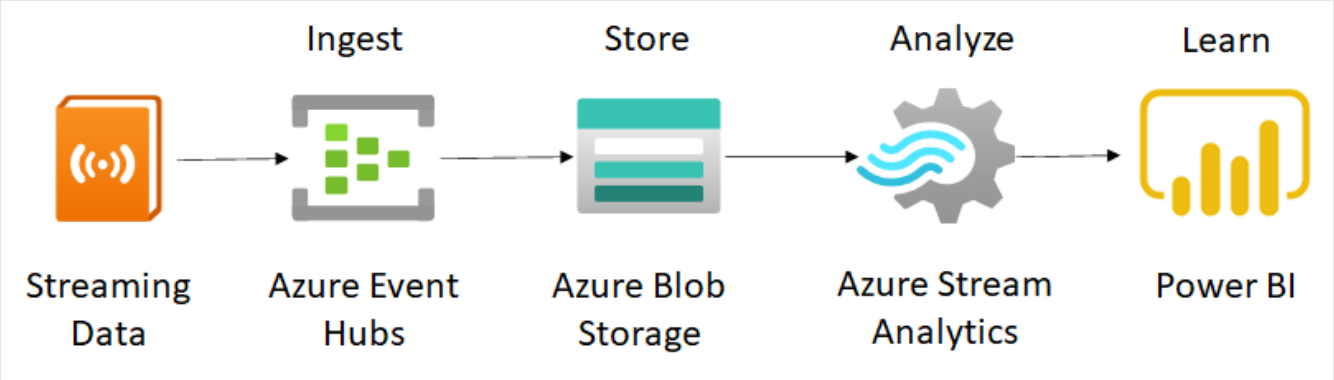
How Azure Event Hubs works

Azure Event Hubs is a fully managed, real time data ingestion. Event Hubs support real time data ingestion and microservices batching on the same stream. Here are some common scenarios for Event Hubs.

- Anomaly detection (fraud/outliers) and live dashboarding
- Analytics pipelines, such as clickstreams, and archiving data
- Transaction processing with real-time analysis

This diagram shows how Event Hubs could be used in the aircraft engine application.

- Event hubs captures streaming data from the testing equipment.
- The data is stored in Azure blob storage.
- Azure Stream Analytics to identify patterns in the sensor data.
- Power BI is used to make decisions on manufacturing improvements.



Considerations for Event Hubs

When selecting Event Hubs, consider the following guidance:

- **Expect language and framework integration**. You can send and receive events in many different languages. Messages can also be received from Event Hubs using Apache Storm.
- **Choose a tier and throughput**. Scaling of Event Hubs is controlled by how many throughput units or processing units you purchase . Other performance aspects depend on the pricing tier chosen. Basic, standard, premium, and dedicated pricing tiers are available. A single throughput unit equates to:
  - Ingress: Up to 1 MB per second or 1000 events per second (whichever comes first).
  - Egress: Up to 2 MB per second or 4096 events per second.
- **Remember Event Hubs uses a pull model**. The pull model used by Event Hubs differentiates it from some other messaging services, such as Azure Service Bus Queues. The pull model means that Event Hubs simply holds the message in its cache and allows it to be read. When a message is read from Event Hubs, it isn't deleted. The message remains for other consumers.
- **Account for data failures.** There's no built-in mechanism to handle messages that aren't processed as you expect them. For example, imagine your consumer malfunctions because of data format. Event Hubs won't detect this issue. The message remains until its time-to-live has expired.
- **Process the data stream.** Events received by Event Hubs are added to the end of its data stream. This data stream orders events by the time they are received, and consumers can seek along this stream using time offsets.

Next unit: Design an event-driven solution

100 XP

Design an event-driven solution

3 minutes

Event driven architecture enables you to connect to the core application without needing to modify the existing code. When an event occurs, you can react with your own code to these events. Event-driven applications use the send and forget principle. The event gets sent toward the next system, which can be another service, an event hub, a stream, or a message broker.

Let's consider that music-sharing application again, this time with a Web API that runs in Azure. When a user uploads a new song, you must notify all the mobile apps installed on user devices around the world that are interested in that genre of music. In this requirement. Azure Event Grid is a perfect solution for this sort of requirement.

- The publisher of the sound file doesn't need to know about any of the subscribers interested in the shared music.
- We want to have a one-to-many relationship where we can have multiple subscribers. Subscribers who can optionally decide whether they're interested in this new song.

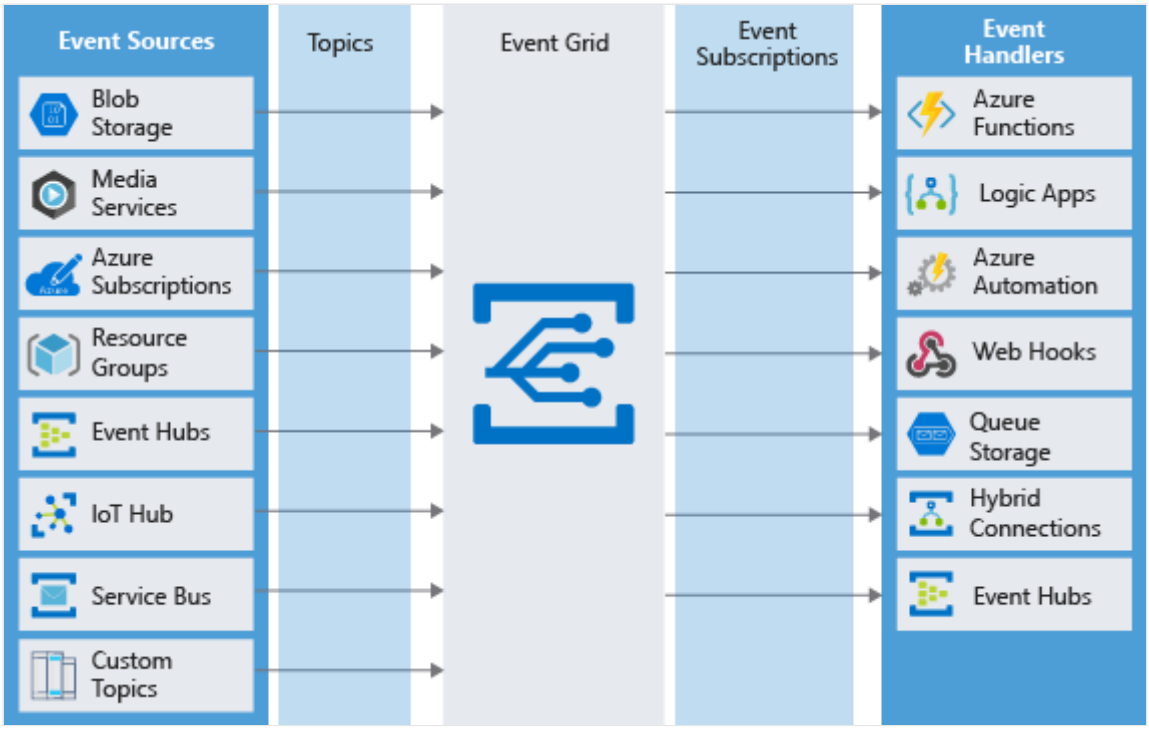
What is Azure Event Grid?

Azure Event Grid is a fully managed event routing service running on top of Azure Service Fabric. Event Grid distributes events from sources such as Azure blob storage accounts and Azure media services. These events are distributed to handlers such as Azure Functions and Webhooks. Event Grid exists to make it easier to build event-based and serverless applications on Azure.

- Aggregates all your events and provides routing from any source to any destination.
- Is a service that manages the routing and delivery of events from many sources and subscribers. This process eliminates the need for polling and results in minimized cost and latency.

The following illustration displays an Azure Event Grid positioned between multiple event sources and multiple event handlers. The event sources send events to the Event Grid and the Event Grid forwards relevant events to the subscribers. Event Grid use topics to decide which

events to send to which handlers. Events sources tag each event with one or more topics, and event handlers subscribe to the topics they're interested in.



Event Grid sends an event to indicate something has happened or changed. However, the actual object that was changed isn't part of the event data. Instead, a URL or identifier is often passed to reference the changed object.

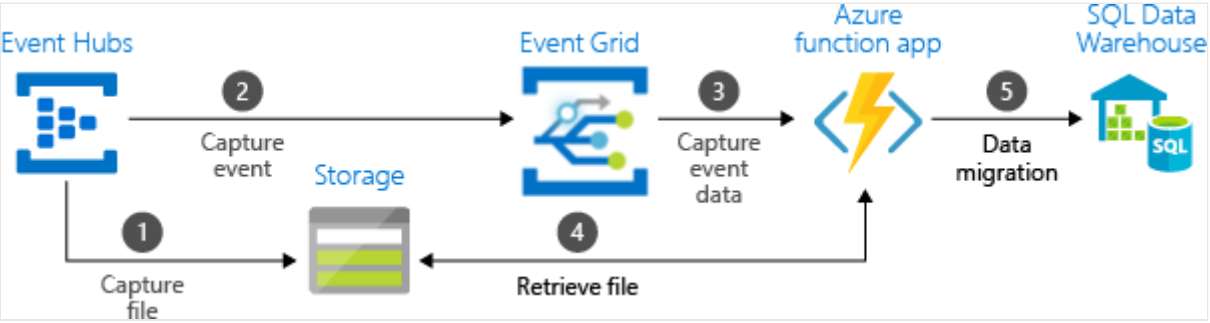
Comparison of services

Let's take a few minutes to review the message and event solutions we have covered.

Service	Purpose	Type	When to use
Event Grid	Reactive programming	Event distribution (discrete)	React to status changes
Event Hubs	Big data pipeline	Event streaming (series)	Telemetry and distributed data streaming
Service Bus	High-value enterprise messaging	Message	Order processing and financial transactions

Use the services together

In some cases, you use the services side by side to fulfill distinct roles. For example, an e-commerce site can use Service Bus to process the order, Event Hubs to capture site telemetry, and Event Grid to respond to events like an item was shipped. In other cases, you link them together to form an event and data pipeline. You use Event Grid to respond to events in the other services. The following image shows the workflow for streaming the data.



Next unit: Design a caching solution

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design a caching solution

3 minutes

Caching is a common technique that aims to improve the performance and scalability of a system. It does this by temporarily copying frequently accessed data to fast storage that's located close to the application. If this fast data storage is located closer to the application than the original source, then caching can significantly improve response times for client applications by serving data more quickly.

Caching is most effective when a client instance repeatedly reads the same data, especially if all the following conditions apply to the original data store:

- It remains relatively static.
- It's slow compared to the speed of the cache.
- It's subject to a high level of contention.
- It's far away when network latency can cause access to be slow.

In this unit, you'll learn how to use Azure Cache for Redis to manage caching requirements.

Recommend a caching solution for applications

Suppose you work at a Tailwind Traders that has launched a new game ,which will have real time leaderboards for gamers to check their scores. The leaderboard would show the user's rank in the game in real time while getting updated as soon as the events in the game changes. This requires in-memory fast read and writes for which you have been asked to design a caching solution.

What is Azure Cache for Redis?

Azure Cache for Redis provides an in-memory data store based on the Redis software. Redis improves the performance and scalability of an application that uses backend data stores heavily. It's able to process large volumes of application requests by keeping frequently accessed data in the server memory, which can be written to and read from quickly. Redis

brings a critical low-latency and high-throughput data storage solution to modern applications.

Azure Cache for Redis offers both:

- The Redis open source (OSS Redis)
- A commercial product from Redis Labs (Redis Enterprise) as a managed service.

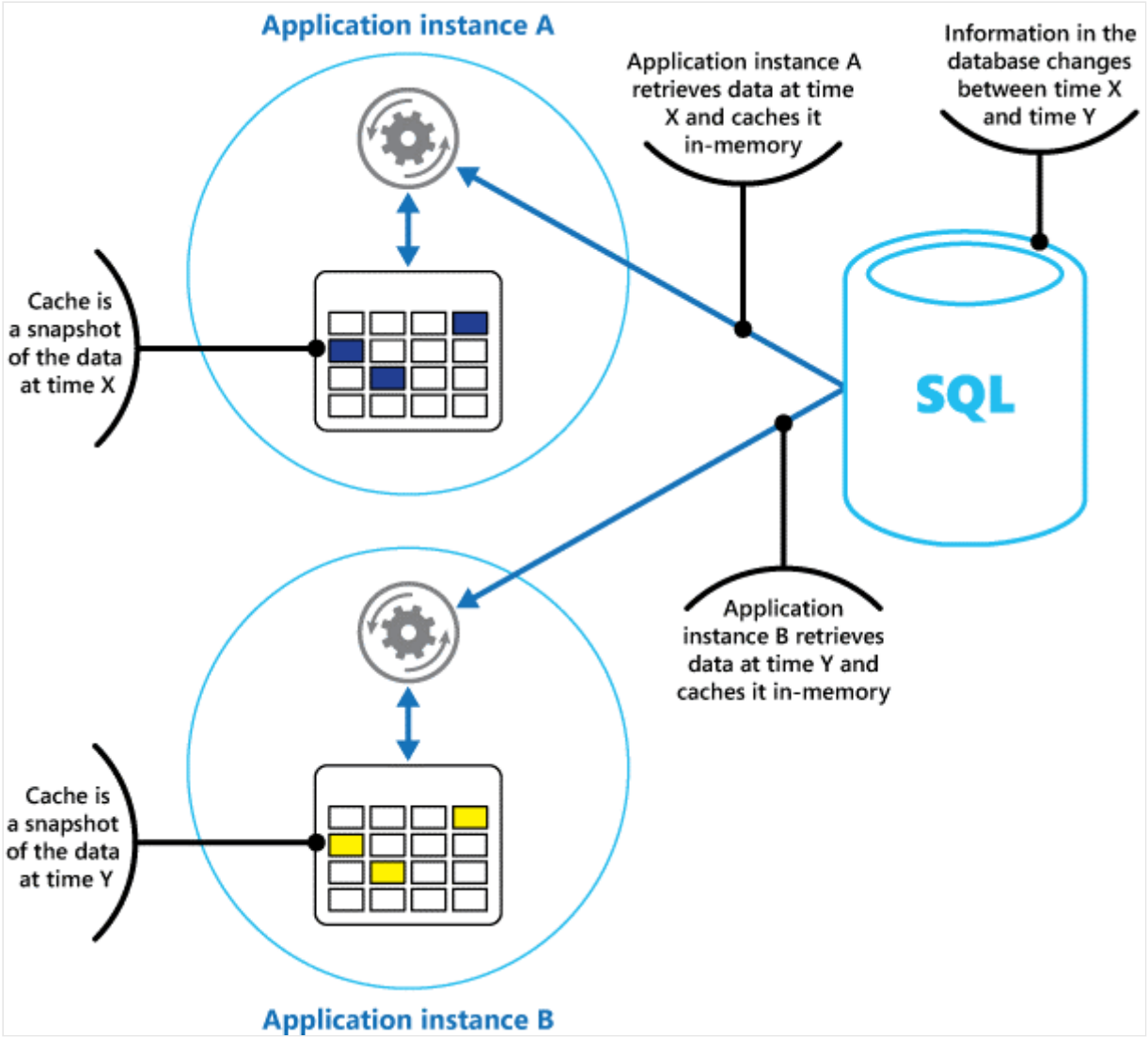
Azure Cache for Redis provides secure and dedicated Redis server instances and full Redis API compatibility. The service is operated by Microsoft, hosted on Azure, and usable by any application within or outside of Azure.

You can use Azure Cache for Redis for several purposes, including as a:

- Distributed data or content cache
- A session store
- A message broker

> **Tip**
>
> You can deploy Azure Cache for Redis as a standalone. Alternatively, you can deploy it with other Azure database services, such as Azure SQL or Cosmos DB.

The following diagram shows how cache works in applications.



When to use Azure Cache for Redis?

Azure Cache for Redis improves application performance by supporting common application architecture patterns. Some of the most common include the following patterns.

Audience	Azure Cache for Redis
Data cache	Databases are often too large to load directly into a cache. It's common to use the cache-aside pattern to load data into the cache only as needed. When the system makes changes to the data, the system can also update the cache, which is then distributed to other clients. Additionally, the system can set an expiration on data, or use an eviction policy to trigger data updates into the cache.

Audience	Azure Cache for Redis
Content cache	Many web pages are generated from templates that use static content such as headers, footers, banners. These static items shouldn't change often. Using an in-memory cache provides quick access to static content compared to backend datastores. This pattern reduces processing time and server load, allowing web servers to be more responsive. It can allow you to reduce the number of servers needed to handle loads. Azure Cache for Redis provides the Redis Output Cache Provider to support this pattern with ASP.NET.
Session store	This pattern is commonly used with shopping carts and other user history data that a web application might associate with user cookies. Storing too much in a cookie can have a negative effect on performance as the cookie size grows and is passed and validated with every request. A typical solution uses the cookie as a key to query the data in a database. Using an in-memory cache, like Azure Cache for Redis, to associate information with a user, is much faster than interacting with a full relational database.
Job and message queuing	Applications often add tasks to a queue when the operations associated with the request take time to execute. Longer running operations are queued to be processed in sequence, often by another server. This method of deferring work is called task queuing. Azure Cache for Redis provides a distributed queue to enable this pattern in your application.
Distributed transactions	Applications sometimes require a series of commands against a backend data-store to execute as a single atomic operation. All commands must succeed, or all must be rolled back to the initial state. Azure Cache for Redis supports executing a batch of commands as a single transaction.

Next unit: Design API integration

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design API integration

3 minutes

Publishing an API is a great way to increase market share, generate revenue, and foster innovation. However, maintaining even one API brings significant challenges, such as onboarding users, managing revisions, and implementing security.

How do you reduce the complexity inherent in having numerous APIs and their management? You need an API Management that acts as a front door for all your APIs. API Management provides tools for implementing security, managing revisions, and performing analytics.

In this unit, you'll learn about Azure API Management, and determine whether it's the correct solution to help reduce your API complexity.

Select an API management solution

Suppose you work at a Tailwind Traders that has acquired a food-delivery platform. The customers use a mobile app or a website to browse the menus of multiple restaurants. Customers then place an order for the food they want. Your new company delivers the food. The backbone of your platform is a large collection of APIs. Some of the APIs that you publish are used by:

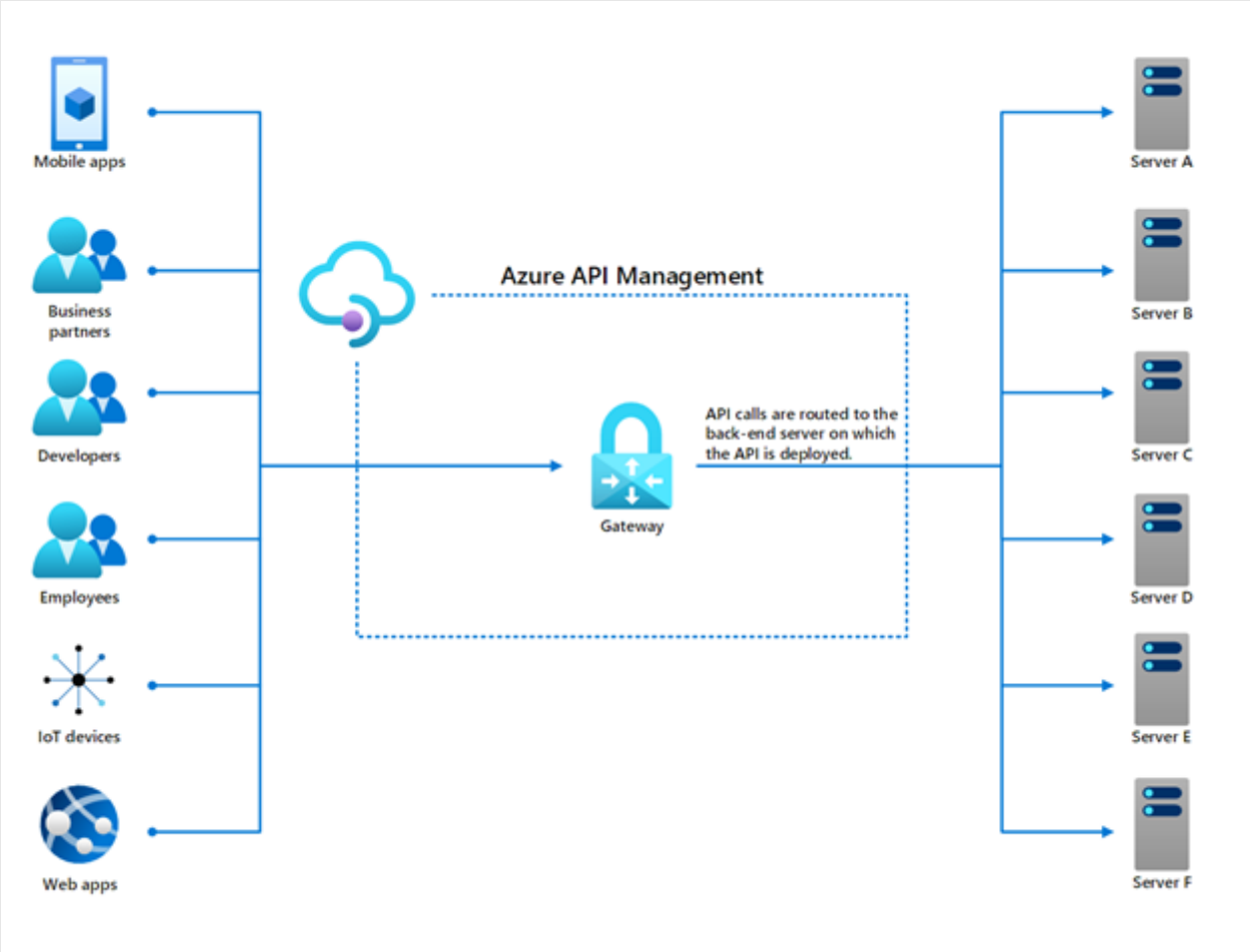
- Your mobile app
- Your web app
- Your partner restaurants
- The IoT devices on your delivery vehicles
- Your in-house development teams
- Your employees, such as business analysts

Each published API resides on a different server, has its own process for onboarding users, and has its own policies for security, revisions, analytics, and more. You've been tasked to find a way to reduce this complexity.

Let's learn how Azure API Management can standardize, centralize, and help secure all the aspects of publishing and maintaining APIs across the full API lifecycle.

What is Azure API Management?

Azure API Management is a cloud service platform that lets you publish, secure, maintain, and analyze all your company's APIs. The following diagram shows Azure API Management that acts as a 'front door' for all an organization's APIs, which are then routed to the server where the API is deployed.



ⓘ Important

Azure API Management does not host your actual APIs; your APIs remain where they were originally deployed. Instead, Azure API Management acts as a kind of façade or "front door" for your APIs. In this way, Azure API Management is said to decouple your APIs. This lets you set API policies and other management options in Azure, while leaving your deployed backend APIs untouched.

When to use Azure API Management?

Here are the criteria we'll use to help you decide whether Azure API Management is a suitable choice for managing and publishing your organization's inventory of APIs.

- Number of APIs
- Rate of API changes
- API administration load

When you have numerous deployed APIs that you revise frequently and that require significant administrative overhead. Azure API Management can help you administer and publish them. Azure API Management might not be the correct choice for use cases that typically involve small, static, or simple API deployments. Let's review the decision criteria in more detail.

Criteria	Analysis
Number of APIs	The key consideration when you're evaluating Azure API Management is the number of APIs that you manage. The more APIs you've deployed, the greater the need for deployment standardization, and centralization of API control.
Rate of API changes	The next consideration is the rate at which your organization implements API revisions and versions. The faster you create API revisions and publish new API versions, the greater the need for a robust, and flexible versioning control system.
API administration load	The last consideration is how much policy overhead you apply to your APIs. This includes usage quotas, call rate limits, request transformations, and request validation. The more configurations and options your APIs require, the greater the need for standardized, and centralized policy implementations.

Consider Azure API Management

There are many reasons to choose Azure API Management. Using the food delivery scenario above as an example, let’s investigate API lifecycle management with respect to standardizing APIs, centralizing API management, and enhancing API security. The following table describes these features.

Feature	Description

Feature	Description
Standardize all disparate APIs	Considering disparate APIs in the new company such as mobile, partner restaurants, and IoT devices. It's imperative that you use an API management solution. This solution will standardize API specs, help in creating documentation, and standardize the base URL for ease of use. API Management can provide consistent analytics across multiple APIs and ensure compliance across all APIs.
Centralize API operations	By bringing multiple APIs under a single administrative umbrella, Azure API Management enhances the centralization of all API operations. Without an API management service, each API is on its own in terms of administration, deployment, and developer access. A centralized model results in less duplicated effort and increases efficiency in the food delivery scenario.
Secure APIs	Azure API Management was designed with API security in mind. It manages permissions, access, protects the API from malicious usage and helps in achieving all corporate and government-related compliance.

Next unit: Design an automated app deployment solution

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design an automated app deployment solution

3 minutes

With the move to the cloud, many teams have adopted agile development methods. These teams must iterate quickly and repeatedly deploy their solutions to the cloud. Teams must be assured their infrastructure is in a reliable state. Application code must be managed through a unified process.

To meet these challenges, you can automate deployments and use the practice of infrastructure as code. In this unit, you'll learn how to evaluate different Azure solutions for deployment and automation for your applications. These solutions include Azure Resource Manager templates, and Azure Automation.

What are Azure Resource Manager templates?

Azure Resource Manager templates are files that define the infrastructure and configuration for your deployment. When you write a template, you take a declarative approach to your resource provisioning. These templates describe each resource in the deployment, but they don't describe how to deploy the resources.

There are many benefits to using templates for your resource provisioning. These benefits are described in the following table:

Function	Template benefit
Repeatable results	Templates are idempotent. Idempotent means you can repeatedly deploy the same template and get the same result.
Orchestration	When a template deployment is submitted to Azure Resource Manager, the resources in the template are deployed in parallel. This process allows deployments to finish faster.

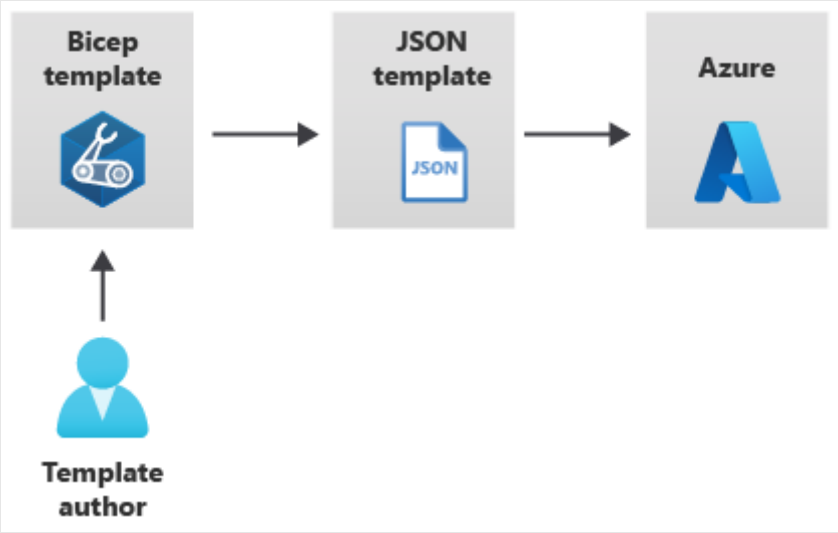
Function	Template benefit
Preview	The WhatIf parameter, available in PowerShell and Azure CLI, allows you to preview changes to your environment before template deployment. This parameter will detail any creations, modification, and deletions that will be made by your template.
Testing and Validation	Templates submitted to Resource Manager are validated before the deployment process. This validation alerts you to any errors in your template before resource provisioning.
Modularity	You can break up your templates into smaller components and link them together at deployment.
CI/CD integration	Your templates can be integrated into multiple CI/CD tools, like Azure DevOps and GitHub Actions.
Extensibility	With deployment scripts, you can run Bash or PowerShell scripts from within your templates. Through extensibility, you can use a single template to deploy a complete solution.

Note

Two types of templates are available for use today: JSON templates and Bicep templates. JavaScript Object Notation (JSON) is an open-standard file format that multiple languages can use. Bicep is a new domain-specific language that was recently developed for authoring templates by using an easier syntax. You can use either template format for your templates and resource deployments.

What are Bicep templates?

Bicep is an Azure Resource Manager template language that's used to declaratively deploy Azure resources. Bicep is a domain-specific language, which means that it's designed for a specific scenario or domain. Bicep is used to create Azure Resource Manager templates.



There are many reasons to choose Bicep as the main tool set for your infrastructure as code deployments. These benefits are described in the following table.

Feature	Description
Azure-native	Bicep is native to the Azure ecosystem. When new Azure resources are released or updated, Bicep will support those features on day one.
Azure integration	Templates, both JSON and Bicep, are fully integrated within the Azure platform. With Resource Manager-based deployments, you can monitor the progress of your deployment in the Azure portal.
Azure support	Bicep is a fully supported product with Microsoft Support.
No state or state files to manage	All state is stored in Azure. Users can collaborate and have confidence their updates are handled as expected.
Easy transition from JSON	If you're already using JSON templates as your declarative template language, it isn't difficult to transition to Bicep. You can use the Bicep CLI to decompile any template into a Bicep template.

What is Azure Automation?

Azure Automation delivers a cloud-based automation and configuration service that supports consistent management across your Azure and non-Azure environments. Automation gives

you complete control of process automation, configuration management, and update management.

Process	Description
Process Automation	Enables you to automate frequent, time-consuming, and error-prone cloud management tasks. This service helps you focus on work that adds business value. By reducing errors and boosting efficiency, it also helps to lower your operational costs. The service allows you to author runbooks graphically, in PowerShell, or using Python.
Configuration Management	Enables access to two features: Change Tracking and Inventory and Azure Automation State Configuration. The service supports change tracking across services, daemons, software, registry, and files in your environment to help you diagnose unwanted changes and raise alerts.
Update management	Includes the Update Management feature for Windows and Linux systems across hybrid environments. The feature allows you to create scheduled deployments that orchestrate the installation of updates within a defined maintenance window.

100 XP

Design an application configuration management solution

3 minutes

Traditionally, shipping a new application feature requires a complete redeployment of the application itself. Testing or deployment of a feature often requires multiple versions of the application. Each deployment may require different configurations, credentials, changing settings or parameters for testing.

Configuration management is a modern software-development practice that decouples configuration from code deployment and enables quick changes to feature availability on demand. Decoupling configuration as a service enables systems to dynamically administer the deployment lifecycle.

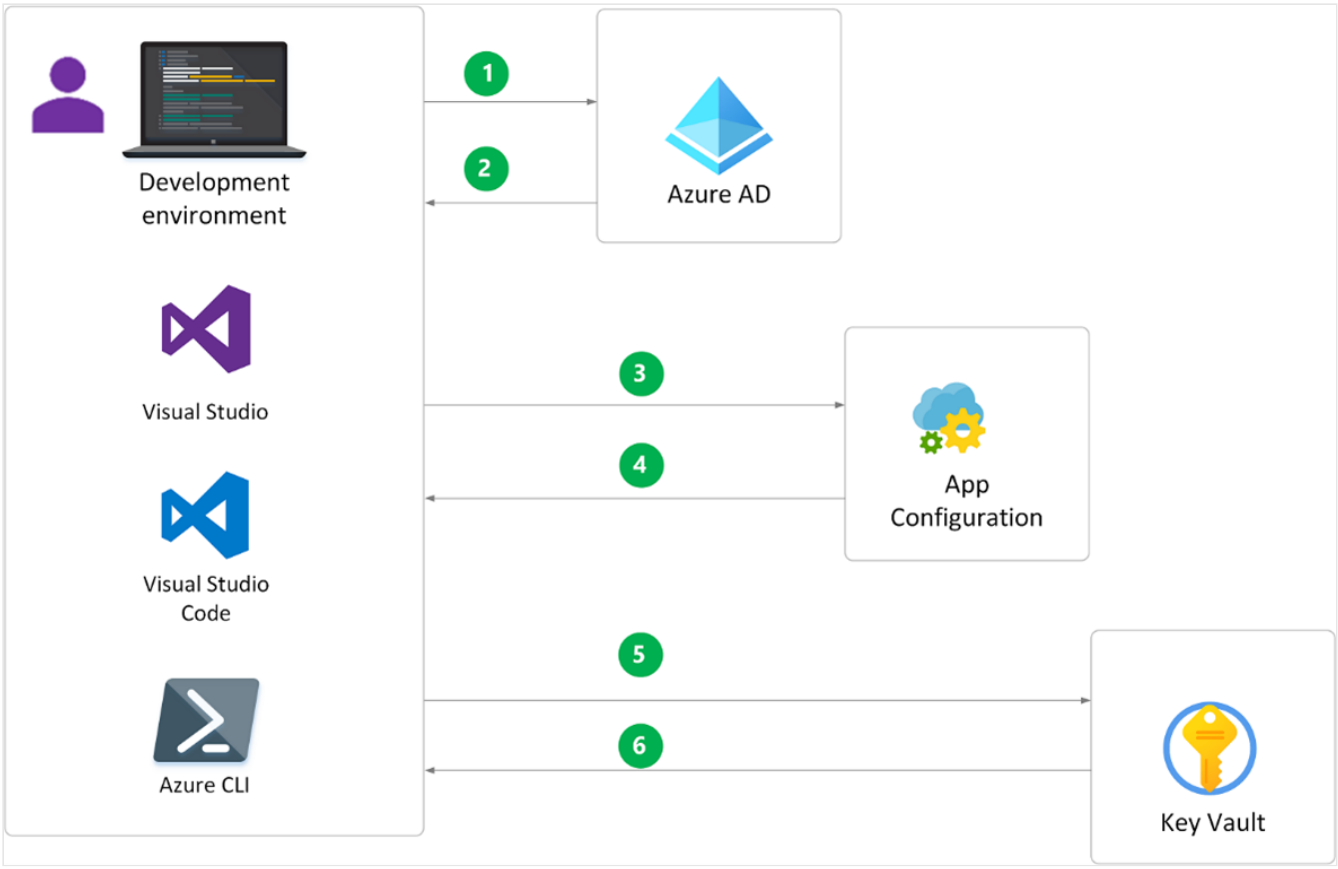
In this unit, you'll learn about Azure Configuration Management solutions that can help you address deployment issues.

What is Azure App Configuration?

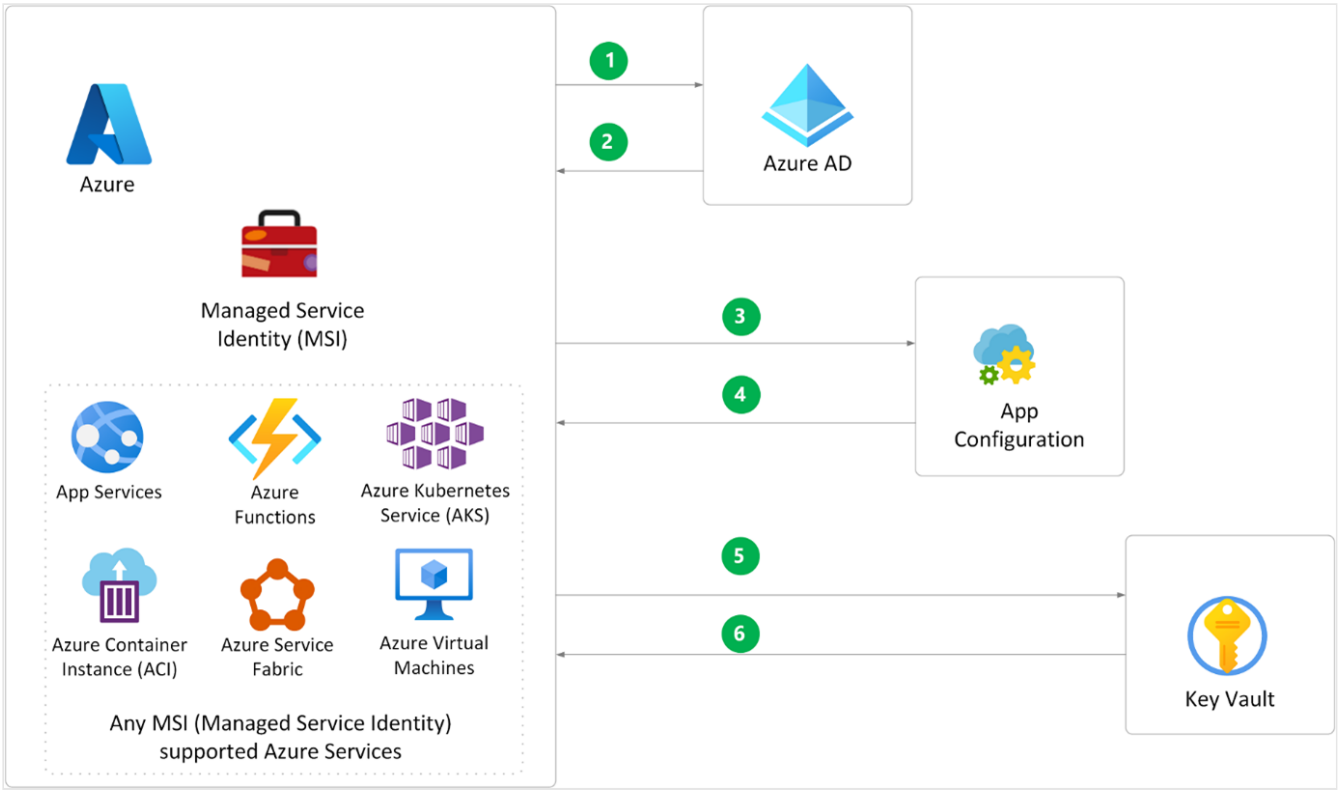
Azure App Configuration provides a service to centrally manage application settings and feature flags. Use App Configuration to store all the settings for your application and secure their accesses in one place.

The following two diagrams show how Azure App Configuration works in Development and Productions environments:

Development



Production



What are the benefits of App Configuration?

App Configuration offers the following benefits:

- A fully managed service that can be set up in minutes.
- Flexible key representations and mappings.
- Tagging with labels.
- Point-in-time replay of settings.
- Dedicated UI for feature flag management.
- Comparison of two sets of configurations on custom-defined dimensions.
- Enhanced security through Azure-managed identities.
- Encryption of sensitive information at rest and in transit.
- Native integration with popular frameworks.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

200 XP

Knowledge check

3 minutes

Tailwind Traders is continuing work on its new marketing game. With each purchase, the game will provide players a chance to win cash or prizes. There are several specific requirements.

- **Transaction processing**. When a customer makes a purchase, details should be grouped into a single transaction.
- **Update management.** Developers require scheduled software deployments and expect installation of updates within a defined maintenance window.
- **Event handling**. During game play millions of events are expected per second. Players will expect a low latency on responses. The event stream should be saved to blob storage.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. Which the following solutions should be used for the transaction processing requirement?

- ○ Azure Queue storage
- ○ Azure Service Bus queues

✔ **Correct. Azure Service Bus queues provide advanced message handing. For example, the ability to group messages into a transaction.**

- ◉ Azure Service Bus topics

✘ **Incorrect. Azure Service Bus topics is better for multiple receivers with multiple destinations.**

2. Which of the following solutions should be used for the update management requirement?

- ○ Azure Resource Manager templates
- ◉ Azure Bicep templates

✘ **Incorrect. Azure Bicep is a template language that's used to declaratively deploy Azure resources.**

- ○ Azure Automation

Azure Automation

✔ Correct. Azure Automation gives you complete control of update management.

3. Which of the following solutions should be used for the event handling requirement?

- ○ Azure Event Hubs

  ✔ Correct. Azure Event Hubs can handle millions of events with low latency. Azure Event Hubs can stream events to blob storage.

- ◉ Azure Event Grid

  ✘ Incorrect. Azure Event Grid distributes events from different sources to different handlers.

- ○ Azure IoT Hub

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Introduction

3 minutes

Meet Tailwind Traders

Tailwind Traders is a fictitious home improvement retailer. It operates retail hardware stores across the globe and online. Tailwind Traders specializes in competitive pricing, fast shipping, and a large range of items. It's looking at cloud technologies to improve business operations and support growth into new markets. By moving to the cloud, the company plans to enhance its shopping experience to further differentiate itself from competitors.

As the Tailwind Traders Enterprise IT team prepares to define the strategy to migrate some of company's workloads to Azure, it must identify the required networking components and design a network infrastructure necessary to support them. Considering the global scope of its operations, Tailwind Traders will be using multiple Azure regions to host its applications. Most of these applications have dependencies on infrastructure and data services, which will also reside in Azure. Internal applications migrated to Azure must remain accessible to Tailwind Traders users. Internet-facing applications migrated to Azure must remain accessible to any external customer.

Learning objectives

In this module, you will:

- Recommend a network architecture solution based on workload requirements
- Design for on-premises connectivity to Azure Virtual Networks
- Design for Azure network connectivity services
- Design for application delivery services
- Design for application protection services

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design Network Solutions

- Recommend a network architecture solution based on workload requirements
- Recommend a connectivity solution that connects Azure resources to the Internet
- Recommend a connectivity solution that connects Azure resources to on-premises networks
- Optimize network performance for applications
- Recommend a solution to optimize network security
- Recommend a load balancing and routing solution

Prerequisites

- Working experience with enterprise networking.
- Conceptual knowledge of software defined networking and hybrid connectivity.

Next unit: Recommend a network architecture solution based on workload requirements

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Recommend a network architecture solution based on workload requirements

3 minutes

Tailwind Traders currently runs its workloads on-premises, in its datacenter. As the Tailwind Traders Enterprise IT team prepares to define the strategy to migrate some of company's workloads to Azure, it must identify the required networking components and design a network infrastructure necessary to support them. Considering the global scope of its operations, Tailwind Traders will be using multiple Azure regions to host its applications. Most of these applications have dependencies on infrastructure and data services, which will also reside in Azure. Internal applications migrated to Azure must remain accessible to Tailwind Traders users. Internet-facing applications migrated to Azure must remain accessible to any external customer.

In this module you will learn how the networking services in Azure provide a variety of networking capabilities that can be used together or separately to meet your requirements. Take note of which services you think Tailwind Traders will need for their production environment in Azure.

- **Connectivity services**: Connect Azure resources and on-premises resources using any or a combination of these networking services in Azure - Virtual Network (VNet), Virtual WAN, ExpressRoute, VPN Gateway, Virtual network NAT Gateway, Azure DNS, Peering service, and Azure Bastion.
- **Application protection services**: Protect your applications using any or a combination of these networking services in Azure - Load Balancer, Private Link, DDoS protection, Firewall, Network Security Groups, Web Application Firewall, and Virtual Network Endpoints.
- **Application delivery services**: Deliver applications in the Azure network using any or a combination of these networking services in Azure - Content Delivery Network (CDN), Azure Front Door Service, Traffic Manager, Application Gateway, Internet Analyzer, and Load Balancer.

Gather Network Requirements

Naming

All Azure resources have a name. The name must be unique within a scope, that may vary for each resource type. For example, the name of a virtual network must be unique within a resource group, but can be duplicated within a subscription or Azure region. Defining a naming convention that you can use consistently when naming resources is helpful when managing several network resources over time.

Regions

All Azure resources are created in an Azure region and subscription. A resource can only be created in a virtual network that exists in the same region and subscription as the resource. You can however, connect virtual networks that exist in different subscriptions and regions. When deciding which region(s) to deploy resources in, consider where consumers of the resources are physically located:

- Consumers of resources typically want the lowest network latency to their resources.
- Do you have data residency, sovereignty, compliance, or resiliency requirements? If so, choosing the region that aligns to the requirements is critical.
- Do you require resiliency across Azure Availability Zones within the same Azure region for the resources you deploy? You can deploy resources, such as virtual machines (VM) to different availability zones within the same virtual network. Not all Azure regions support availability zones however.

Subscriptions

You can deploy as many virtual networks as required within each subscription, up to the limit. Some organizations have different subscriptions for different departments, for example.

Segmentation

You can create multiple virtual networks per subscription and per region. You can create multiple subnets within each virtual network. The considerations that follow help you determine how many virtual networks and subnets you require:

Virtual networks A virtual network is a virtual, isolated portion of the Azure public network. Each virtual network is dedicated to your subscription. Things to consider when deciding whether to create one virtual network, or multiple virtual networks in a subscription:

- Do any organizational security requirements exist for isolating traffic into separate virtual networks? You can choose to connect virtual networks or not. If you connect virtual

networks, you can implement a network virtual appliance, such as a firewall, to control the flow of traffic between the virtual networks.

- Do any organizational requirements exist for isolating virtual networks into separate subscriptions or regions?
- A network interface enables a VM to communicate with other resources. Each network interface has one or more private IP addresses assigned to it. How many network interfaces and private IP addresses do you require in a virtual network? There are limits to the number of network interfaces and private IP addresses that you can have within a virtual network.
- Do you want to connect the virtual network to another virtual network or on-premises network? You may choose to connect some virtual networks to each other or on-premises networks, but not others. Each virtual network that you connect to another virtual network, or on-premises network, must have a unique address space. Each virtual network has one or more public or private address ranges assigned to its address space. An address range is specified in classless internet domain routing (CIDR) format, such as 10.0.0.0/16. Learn more about address ranges for virtual networks.
- Do you have any organizational administration requirements for resources in different virtual networks? If so, you might separate resources into separate virtual network to simplify permission assignment to individuals in your organization or to assign different policies to different virtual networks.
- When you deploy some Azure service resources into a virtual network, they create their own virtual network.

Subnets A virtual network can be segmented into one or more subnets up to the limits. Things to consider when deciding whether to create one subnet, or multiple virtual networks in a subscription:

- Each subnet must have a unique address range, specified in CIDR format, within the address space of the virtual network. The address range cannot overlap with other subnets in the virtual network.
- If you plan to deploy some Azure service resources into a virtual network, they may require, or create, their own subnet, so there must be enough unallocated space for them to do so. For example, if you connect a virtual network to an on-premises network using an Azure VPN Gateway, the virtual network must have a dedicated subnet for the gateway.
- Azure routes network traffic between all subnets in a virtual network, by default. You can override Azure's default routing to prevent Azure routing between subnets, or to route traffic between subnets through a network virtual appliance, for example. If you require that traffic between resources in the same virtual network flow through a network virtual appliance (NVA), deploy the resources to different subnets. Learn more in security.

- You can limit access to Azure resources such as an Azure storage account or Azure SQL Database, to specific subnets with a virtual network service endpoint. Further, you can deny access to the resources from the internet. You may create multiple subnets, and enable a service endpoint for some subnets, but not others. Learn more about service endpoints, and the Azure resources you can enable them for.
- You can associate zero or one network security group to each subnet in a virtual network. You can associate the same, or a different, network security group to each subnet. Each network security group contains rules, which allow or deny traffic to and from sources and destinations. Learn more about network security groups.

Security

You can filter network traffic to and from resources in a virtual network using network security groups and network virtual appliances. You can control how Azure routes traffic from subnets. You can also limit who in your organization can work with resources in virtual networks.

Traffic filtering

- You can filter network traffic between resources in a virtual network using a network security group, an NVA that filters network traffic, or both. When using an NVA, you also create custom routes to route traffic from subnets to the NVA. Learn more about traffic routing.
- A network security group contains several default security rules that allow or deny traffic to or from resources. A network security group can be associated to a network interface, the subnet the network interface is in, or both. To simplify management of security rules, it's recommended that you associate a network security group to individual subnets, rather than individual network interfaces within the subnet, whenever possible.
- If different VMs within a subnet need different security rules applied to them, you can associate the network interface in the VM to one or more application security groups. A security rule can specify an application security group in its source, destination, or both. That rule then only applies to the network interfaces that are members of the application security group. Learn more about network security groups and application security groups.
- Azure creates several default security rules within each network security group. One default rule allows all traffic to flow between all resources in a virtual network. To override this behavior, use network security groups, custom routing to route traffic to an NVA, or both. It's recommended that you familiarize yourself with all of Azure's default security rules and understand how network security group rules are applied to a resource.

Traffic routing Azure creates several default routes for outbound traffic from a subnet. You can override Azure's default routing by creating a route table and associating it to a subnet. Common reasons for overriding Azure's default routing are:

- Because you want traffic between subnets to flow through an NVA. To learn more about how to configure route tables to force traffic through an NVA.
- Because you want to force all internet-bound traffic through an NVA, or on-premises, through an Azure VPN gateway. Forcing internet traffic on-premises for inspection and logging is often referred to as forced tunneling. Learn more about how to configure forced tunneling.

Best practice: Plan IP addressing

When you create virtual networks as part of your migration, it's important to plan out your virtual network IP address space.

You should assign an address space that isn't larger than a CIDR range of /16 for each virtual network. Virtual networks allow for the use of 65,536 IP addresses. Assigning a smaller prefix than /16, such as a /15, which has 131,072 addresses, will result in the excess IP addresses becoming unusable elsewhere. It's important not to waste IP addresses, even if they're in the private ranges defined by RFC 1918.

Other tips for planning are:

- The virtual network address space shouldn't overlap with on-premises network ranges.
- Overlapping addresses can cause networks that can't be connected, and routing that doesn't work properly.
- If networks overlap, you'll need to redesign the network.
- If you absolutely can't redesign the network, network address translation (NAT) can help but should be avoided or limited as much as possible.

Best practice: Implement a hub and spoke network topology

A hub and spoke network topology isolates workloads while sharing services, such as identity and security. The hub is an Azure virtual network that acts as a central point of connectivity. The spokes are virtual networks that connect to the hub virtual network by using peering. Shared services are deployed in the hub, while individual workloads are deployed as spokes.

Consider the following:

- Implementing a hub and spoke topology in Azure centralizes common services, such as connections to on-premises networks, firewalls, and isolation between virtual networks. The hub virtual network provides a central point of connectivity to on-premises networks, and a place to host services used by workloads hosted in spoke virtual networks.
- A hub and spoke configuration is typically used by larger enterprises. Smaller networks might consider a simpler design to save on costs and complexity.
- You can use spoke virtual networks to isolate workloads, with each spoke managed separately from other spokes. Each workload can include multiple tiers, and multiple subnets that are connected with Azure load balancers.
- You can implement hub and spoke virtual networks in different resource groups, and even in different subscriptions. When you peer virtual networks in different subscriptions, the subscriptions can be associated to the same, or different, Azure Active Directory (Azure AD) tenants. This allows for decentralized management of each workload, while sharing services maintained in the hub network.

Best practice: Design subnets

To provide isolation within a virtual network, you segment it into one or more subnets, and allocate a portion of the virtual network's address space to each subnet.

- You can create multiple subnets within each virtual network.
- By default, Azure routes network traffic between all subnets in a virtual network.
- Your subnet decisions are based on your technical and organizational requirements.
- You create subnets by using CIDR notation.

Example:

The table shows an example of a virtual network with an address space of 10.245.16.0/20 segmented into subnets, for a planned migration.

Subnet	CIDR	Addresses	Usage
DEV-FE-EUS2	10.245.16.0/22	1019	Front-end or web-tier VMs

Subnet	CIDR	Addresses	Usage
DEV-APP-EUS2	10.245.20.0/22	1019	Application-tier VMs
DEV-DB-EUS2	10.245.24.0/23	507	Database VMs

Next unit: Design for on-premises connectivity to Azure Virtual Networks

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for on-premises connectivity to Azure Virtual Networks

3 minutes

For a successful migration, it's critical to connect on-premises corporate networks to Azure. This creates an always-on connection known as a hybrid-cloud network, where services are provided from the Azure cloud to corporate users.

This section describes services that provide connectivity between Azure resources, connectivity from an on-premises network to Azure resources, and branch to branch connectivity in Azure - Virtual Network (VNet), ExpressRoute, VPN Gateway, Virtual WAN, Virtual network NAT Gateway, Azure DNS, Azure Peering service, and Azure Bastion.

Let's compare the options for connecting an on-premises network to an Azure Virtual Network (VNet). For each option, a more detailed reference architecture is available.

VPN connection

A VPN gateway is a type of virtual network gateway that sends encrypted traffic between an Azure virtual network and an on-premises location. The encrypted traffic goes over the public Internet. There are different configurations available for VPN Gateway connections, such as, site-to-site, point-to-site, or VNet-to-VNet.

This architecture is suitable for hybrid applications where the traffic between on-premises hardware and the cloud is likely to be light, or you are willing to trade slightly extended latency for the flexibility and processing power of the cloud.

Benefits

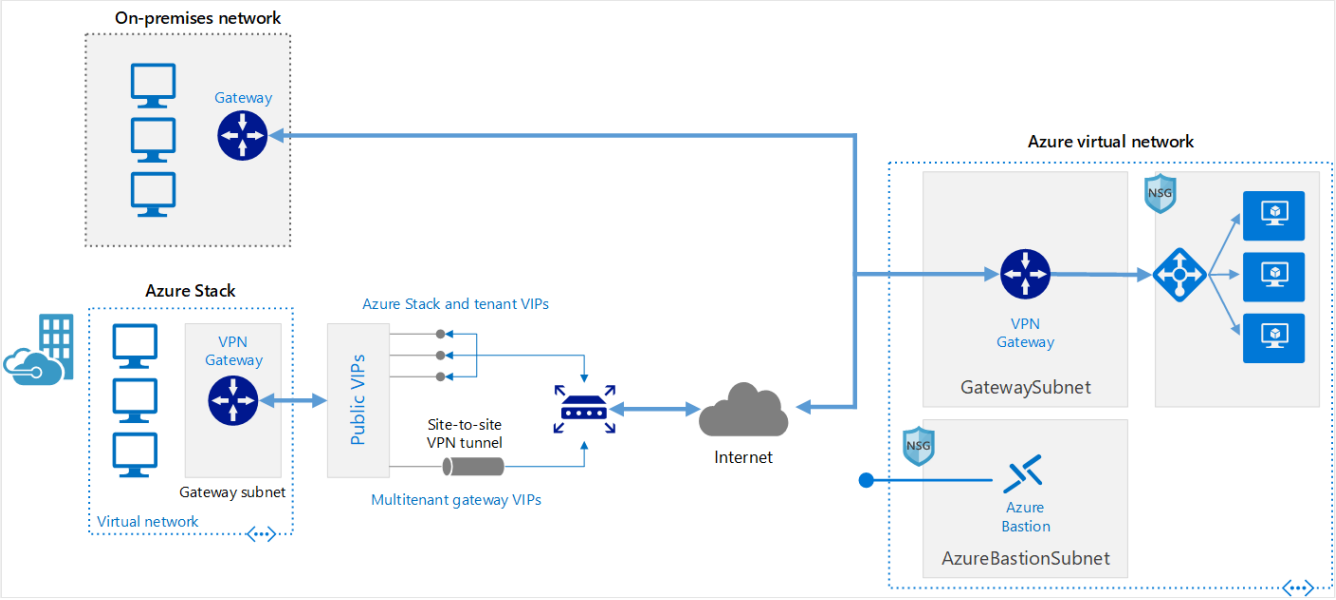
- Simple to configure.
- Much higher bandwidth available; up to 10 Gbps depending on the VPN Gateway SKU.

Challenges

- Requires an on-premises VPN device.

Reference architecture

Hybrid network with VPN gateway



Azure ExpressRoute connection

ExpressRoute connections use a private, dedicated connection through a third-party connectivity provider. This connection is private. Traffic does not go over the internet. The private connection extends your on-premises network into Azure.

This architecture is suitable for hybrid applications running large-scale, mission-critical workloads that require a high degree of scalability.

Benefits

- Much higher bandwidth available; up to 10 Gbps depending on the connectivity provider.
- Supports dynamic scaling of bandwidth to help reduce costs during periods of lower demand. However, not all connectivity providers have this option.
- May allow your organization direct access to national clouds, depending on the connectivity provider.

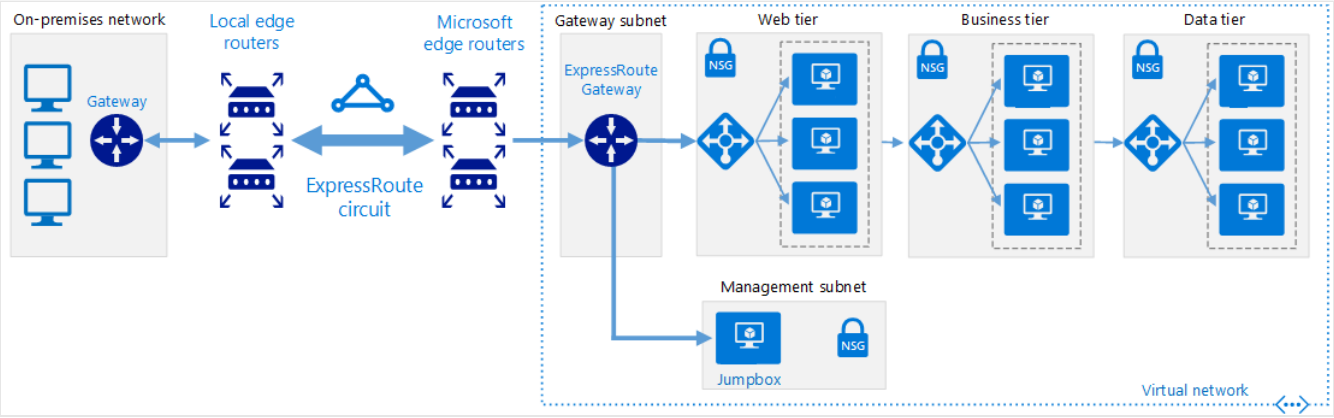
Challenges

- Can be complex to set up. Creating an ExpressRoute connection requires working with a third-party connectivity provider. The provider is responsible for provisioning the network connection.

- Requires high-bandwidth routers on-premises.

Reference architecture

- Hybrid network with ExpressRoute



ExpressRoute with VPN failover

This option combines the previous two, using ExpressRoute in normal conditions, but failing over to a VPN connection if there is a loss of connectivity in the ExpressRoute circuit.

This architecture is suitable for hybrid applications that need the higher bandwidth of ExpressRoute, and also require highly available network connectivity.

Benefits

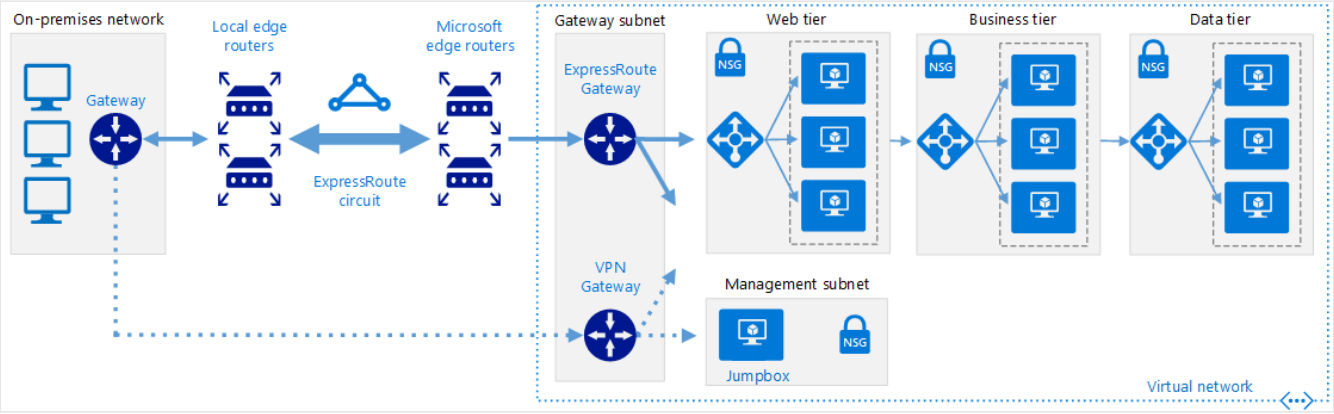
- High availability if the ExpressRoute circuit fails, although the fallback connection is on a lower bandwidth network.

Challenges

- Complex to configure. You need to set up both a VPN connection and an ExpressRoute circuit.
- Requires redundant hardware (VPN appliances), and a redundant Azure VPN Gateway connection for which you pay charges.

Reference architecture

- Hybrid network with ExpressRoute and VPN failover



Hub-spoke network topology

A hub-spoke network topology is a way to isolate workloads while sharing services such as identity and security. The hub is a virtual network (VNet) in Azure that acts as a central point of connectivity to your on-premises network. The spokes are VNets that peer with the hub. Shared services are deployed in the hub, while individual workloads are deployed as spokes.

Reference architectures

- Hub-spoke topology

Hub-spoke network topology with Azure Virtual WAN

A hub-spoke architecture can be achieved two ways: a customer-managed hub infrastructure, or a Microsoft-managed hub infrastructure. In either case, spokes are connected to the hub using virtual network peering.

The hub is a virtual network in Azure that acts as a central point of connectivity to your on-premises network. The spokes are virtual networks that peer with the hub and can be used to isolate workloads. Traffic flows between the on-premises data center(s) and the hub through an ExpressRoute or VPN gateway connection. The main differentiator of this approach is the use of Azure Virtual WAN (VWAN) to replace hubs as a managed service.

Azure Virtual WAN is a networking service that provides optimized and automated branch connectivity to, and through, Azure. Azure regions serve as hubs that you can choose to connect your branches to. You can leverage the Azure backbone to also connect branches and enjoy branch-to-VNet connectivity. Azure Virtual WAN brings together many Azure cloud connectivity services such as site-to-site VPN, ExpressRoute, point-to-site user VPN into a single operational interface. Connectivity to Azure VNets is established by using virtual network connections.

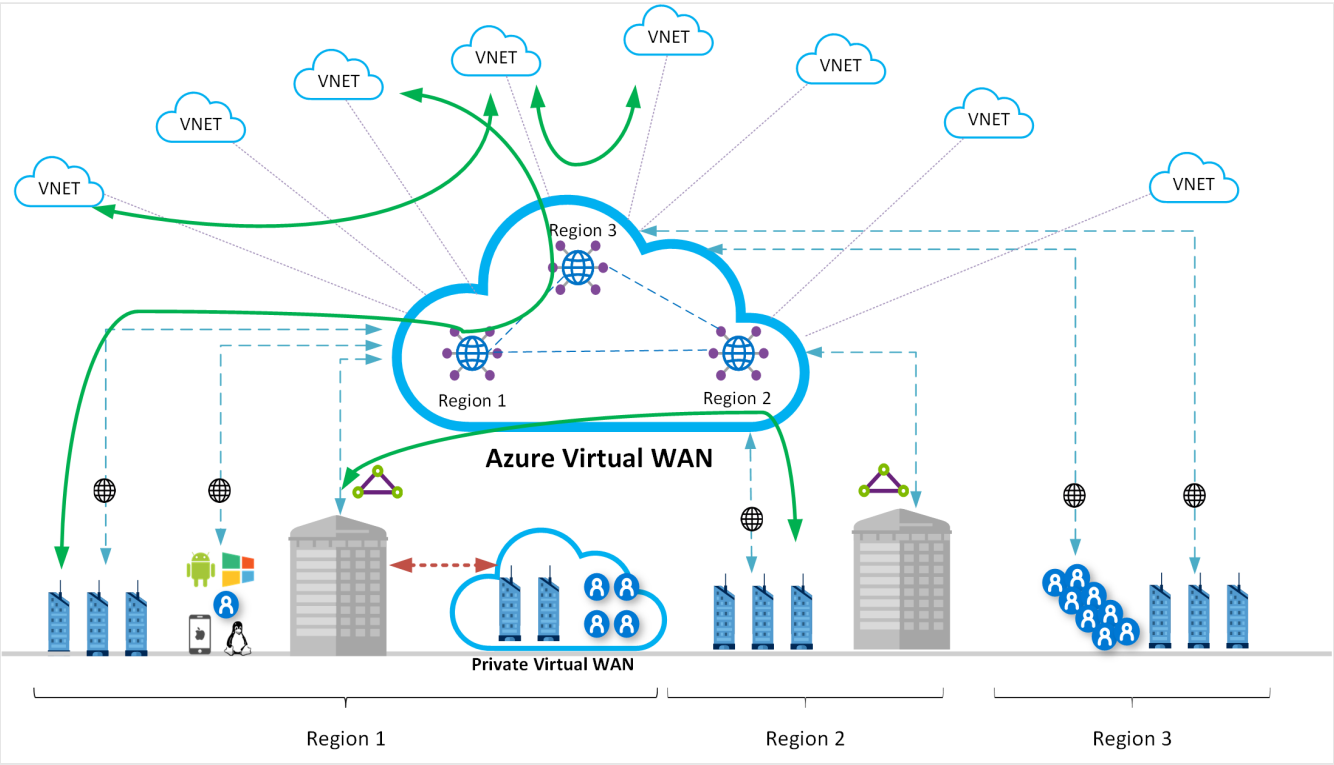
This architecture includes the benefits of standard hub-spoke network topology and introduces new benefits:

- **Less operational overhead** by replacing existing hubs with a fully managed VWAN service.
- **Cost savings** by using a managed service and removing the necessity of network virtual appliance.
- **Improved security** by introducing centrally managed secured Hubs with Azure Firewall and VWAN to minimize security risks related to misconfiguration.
- **Separation of concerns** between central IT (SecOps, InfraOps) and workloads (DevOps).

Typical uses for this architecture include cases in which:

- Connectivity among workloads requires central control and access to shared services.
- An enterprise requires central control over security aspects, such as a firewall, and requires segregated management for the workloads in each spoke.

Advantages



This diagram illustrates a few of the advantages that this architecture can provide:

- A full meshed hub among Azure Virtual Networks
- Branch to Azure connectivity

- Branch to Branch connectivity
- Mixed use of VPN and Express Route
- Mixed use of user VPN to the site
- VNET to VNET connectivity

Next unit: Design for Azure network connectivity services

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure network connectivity services

3 minutes

Virtual network

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. You can use a VNets to:

- **Communicate between Azure resources**: You can deploy VMs, and several other types of Azure resources to a virtual network, such as Azure App Service Environments, the Azure Kubernetes Service (AKS), and Azure Virtual Machine Scale Sets.
- **Communicate between each other**: You can connect virtual networks to each other, enabling resources in either virtual network to communicate with each other, using virtual network peering. The virtual networks you connect can be in the same, or different, Azure regions.
- **Communicate to the internet**: All resources in a VNet can communicate outbound to the internet, by default. You can communicate inbound to a resource by assigning a public IP address or a public Load Balancer. You can also use Public IP addresses or public Load Balancer to manage your outbound connections.
- **Communicate with on-premises networks**: You can connect your on-premises computers and networks to a virtual network using VPN Gateway or ExpressRoute.

When you design a network from bottom up, you gather some basic information. This information could be number of hosts, network devices, number of subnets, routing between subnets, isolation domains such as VLANs. This information helps in sizing the network and security devices as well creating the architecture to support applications and services.

When you plan to deploy your applications and services in Azure, you will start by creating a logical boundary in Azure, which is called a virtual network. This virtual network is akin to a physical network boundary. As it is a virtual network, you don't need physical gear but still have to plan for the logical entities such as IP addresses, IP subnets, routing, and policies.

When you create a virtual network in Azure, it's pre-configured with an IP range (10.0.0.0/16). This range isn't fixed, you can define your own IP range. You can define both IPv4 and IPv6 address ranges. IP ranges defined for the virtual network are not advertised to Internet. You can create multiple subnets from your IP range. These subnets will be used to assign IP addresses to virtual network interfaces (vNICs). Azure reserves the first four and last IP address for a total of 5 IP addresses within each subnet. There is no concept of VLANs in a public cloud. However, you can create isolation within a virtual network based on your defined subnets.

You can create one large subnet encompassing all the virtual network address space or choose to create multiple subnets.

A virtual network is a virtual, isolated portion of the Azure public network. Each virtual network is dedicated to your subscription. Things to consider when deciding whether to create one virtual network, or multiple virtual networks in a subscription:

- Do any organizational security requirements exist for isolating traffic into separate virtual networks? You can choose to connect virtual networks or not. If you connect virtual networks, you can implement a network virtual appliance, such as a firewall, to control the flow of traffic between the virtual networks. For more information, visit security and connectivity.
- Do any organizational requirements exist for isolating virtual networks into separate subscriptions or regions?
- A network interface enables a VM to communicate with other resources. Each network interface has one or more private IP addresses assigned to it. How many network interfaces and private IP addresses do you require in a virtual network? There are limits to the number of network interfaces and private IP addresses that you can have within a virtual network.

Here are more questions to consider when designing an Azure virtual network.

Design network segmentation

Segmentation is a model in which you take your networking footprint and create software defined perimeters using tools available in Microsoft Azure. You then set rules that govern the traffic from/to these perimeters so that you can have different security postures for various parts of your network. When you place different applications (or parts of a given application) into these perimeters, you can govern the communication between these segmented entities. If a part of your application stack is compromised, you'll be better able to contain the impact of this security breach and prevent it from laterally spreading through the rest of your

network. This ability is a key principle associated with the Zero Trust model published by Microsoft that aims to bring world-class security thinking to your organization.

When you operate on Azure, you have a wide and diverse set of segmentation options available to help you be protected.

1. **Subscription**: Subscriptions are a high-level construct, which provides platform powered separation between entities. It's intended to carve out boundaries between large organizations within a company. Communication between resources in different subscriptions needs to be explicitly provisioned.

2. **Virtual Network**: Virtual networks are created within a subscription in private address spaces. The networks provide network-level containment of resources, with no traffic allowed by default between any two virtual networks. Like subscriptions, any communication between virtual networks needs to be explicitly provisioned.

3. **Network Security Groups (NSG)**: NSGs are access control mechanisms for controlling traffic between resources within a virtual network. An NSG also controls traffic with external networks, such as the internet, other virtual networks, and so on. NSGs can take your segmentation strategy to a granular level by creating perimeters for a subnet, group of VMs, or even a single virtual machine.

4. **Application Security Groups (ASGs)**: An ASG allows you to group a set of VMs under an application tag. Once an ASG is created and VMs are assigned to it, the ASG can be used as a source or target in the NSG to simplify management.

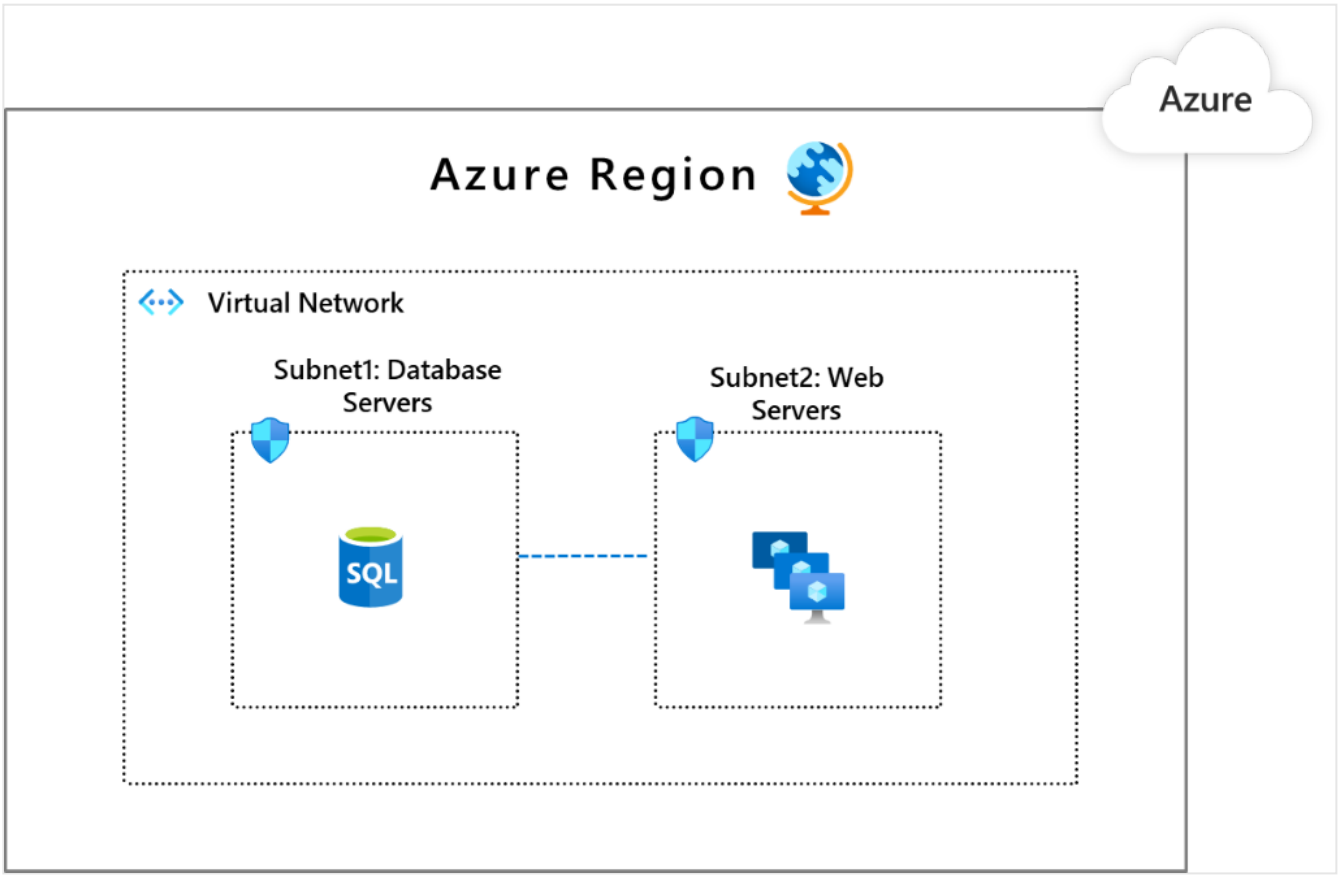
5. **Azure Firewall**: Azure Firewall is a cloud native stateful Firewall as a service. This firewall can be deployed in your virtual networks or in Azure Virtual WAN hub deployments for filtering traffic that flows between cloud resources, the Internet, and on-premise. You create rules or policies (using Azure Firewall or Azure Firewall Manager) specifying allow/deny traffic using layer 3 to layer 7 controls. You can also filter traffic that goes to the internet using both Azure Firewall and third parties. Direct some or all traffic through third-party security providers for advanced filtering and user protection.

The following three patterns are common when it comes to organizing your workload in Azure from a networking perspective. Each of these patterns provides a different type of isolation and connectivity. Choosing which model works best for your organization is a decision you should make based on your organization's needs. With each of these models, we describe how segmentation can be done using the above Azure Networking services.

Pattern 1: Single virtual network

In this pattern, all the components of your workload or, in some cases, your entire IT footprint is put inside a single virtual network. This pattern is possible if you're operating solely in a single region since a virtual network can't span multiple regions.

The entities you would most likely use for creating segments inside this virtual network are NSGs,potentially using ASGs to simplify administration. The image below is an example of a segmented virtual network.



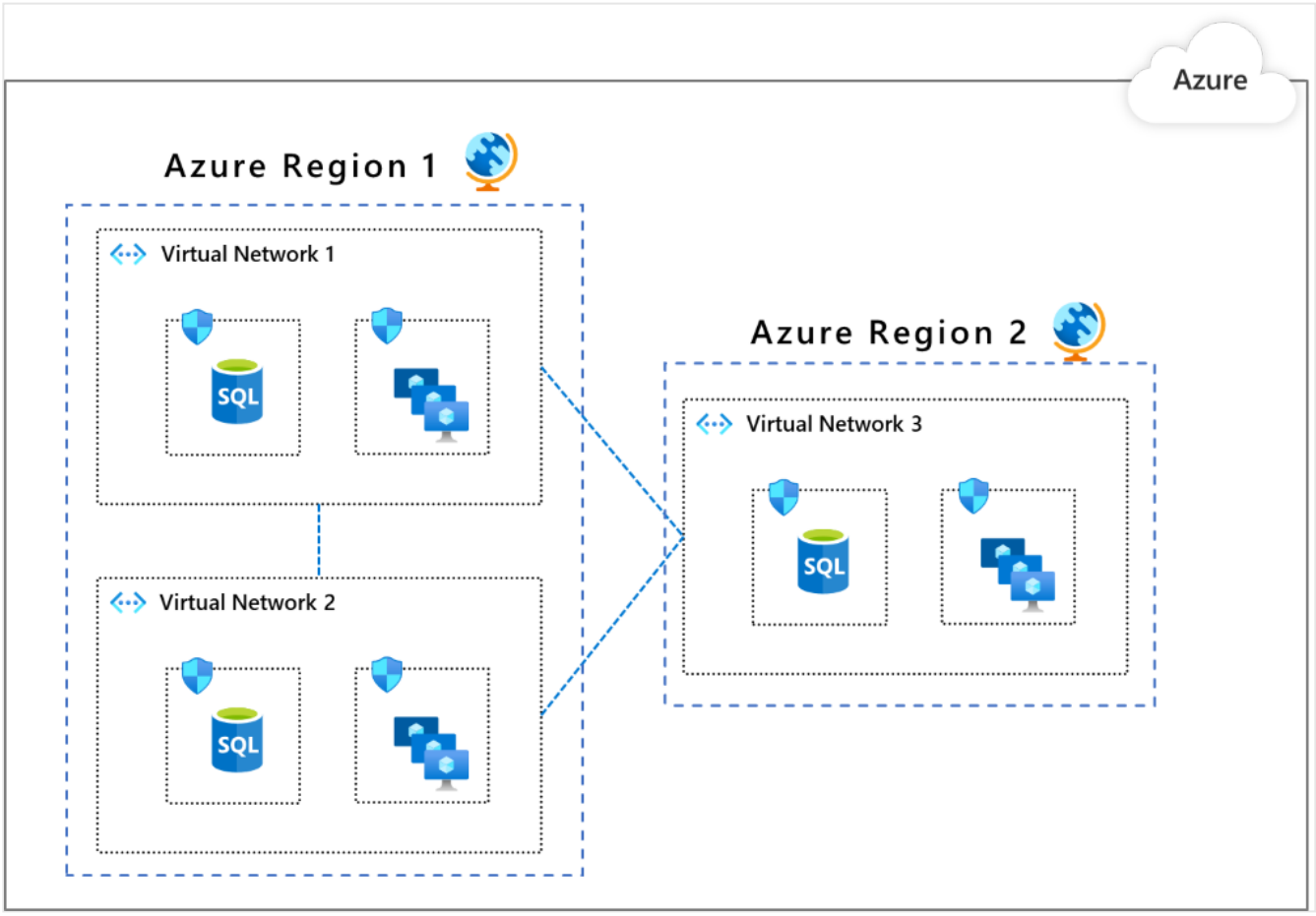
In this setup, you have Subnet1, where you placed your database workloads, and Subnet2, where you've placed your web workloads. You can put NSGs that specify that Subnet1 can talk only with Subnet2, and that Subnet2 can talk to the Internet. You can also take this concept further in the presence of many workloads. Carve out subnets that, for example, won't allow one workload to communicate to the backend of another workload.

Although we used NSGs to illustrate how subnet traffic can be governed, you can also enforce this segmentation by using a Network Virtualized Appliance from Azure Marketplace or Azure Firewall.

Pattern 2: Multiple virtual networks with peering in between them

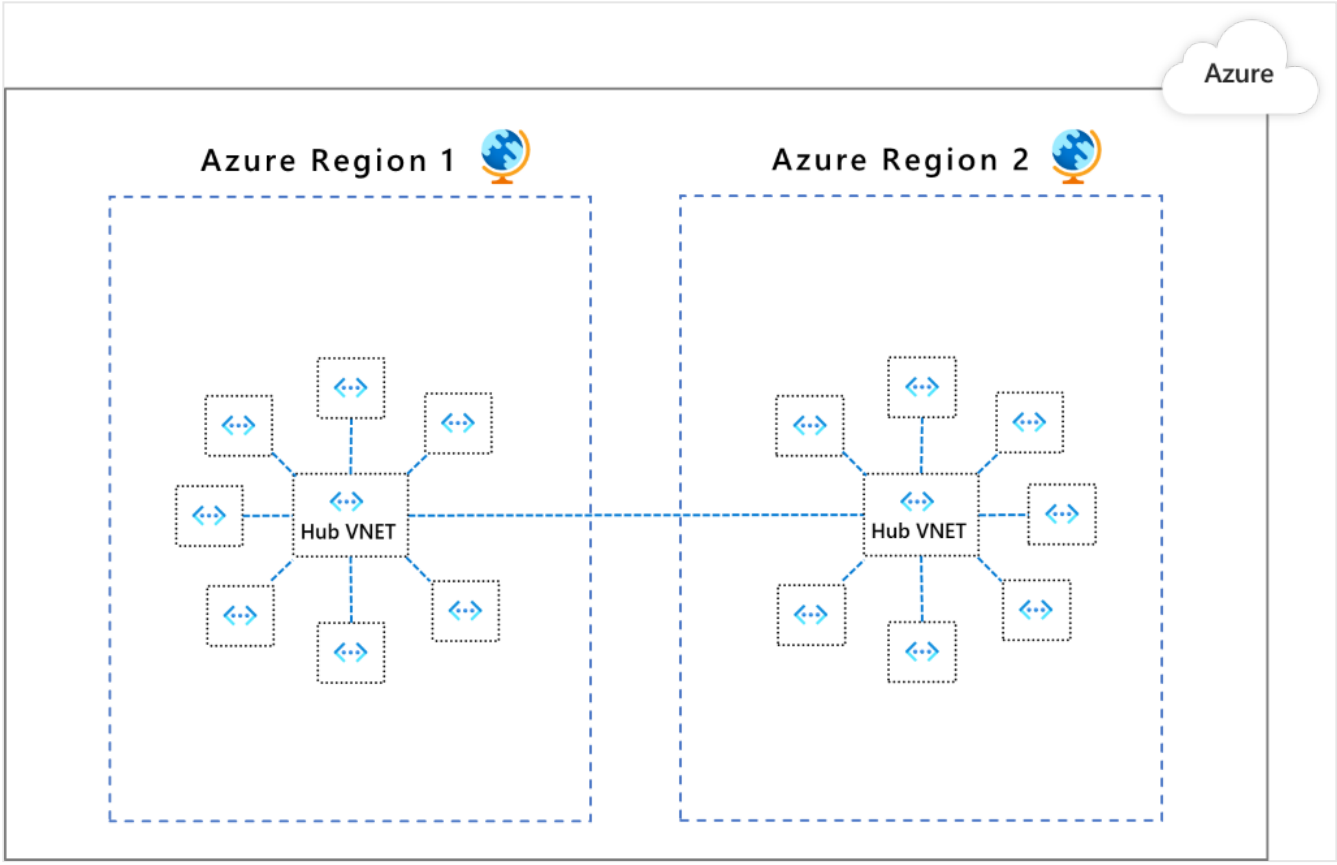
This pattern is the extension of the previous pattern where you have multiple virtual networks with potential peering connections. You might opt for this pattern to group applications into

separate virtual networks, or you might need presence in multiple Azure regions. You get built-in segmentation through virtual networks because you must explicitly peer a virtual network to another one for them to communicate. (Keep in mind that virtual network peering connectivity isn't transitive.) To further segment within a virtual network in a manner similar to pattern 1, use NSGs in the virtual networks.



Pattern 3: Multiple virtual networks in a hub & spoke model

This pattern is a more advanced virtual network organization where you choose a virtual network in a given region as the hub for all the other virtual networks in that region. The connectivity between the hub virtual network and its spoke virtual networks is achieved by using Azure virtual network peering. All traffic passes through the hub virtual network, and it can act as a gateway to other hubs in different regions. You set up your security posture at the hubs, so they get to segment and govern the traffic between the virtual networks in a scalable way. One benefit of this pattern is that. as your network topology grows, the security posture overhead doesn't grow (except when you expand to new regions).



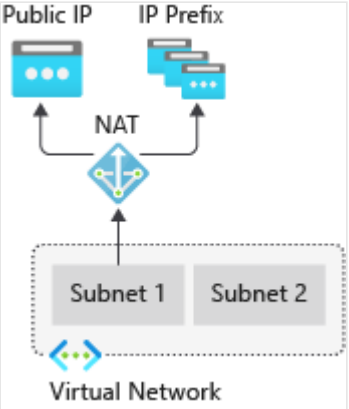
The recommended Azure cloud native segmentation control is Azure Firewall. Azure Firewall works across both Virtual Networks and subscriptions to govern traffic flows using layer 3 to layer 7 controls. You can define your communication rules (for example, virtual network X can't talk with virtual network Y but can talk with virtual network Z, no Internet for Virtual network X except for access to *.github.com, and so on) and apply it consistently. With Azure Firewall Manager, you can centrally manage policies across multiple Azure Firewalls and enable DevOps teams to further customize local policies.

Network capabilities	**Pattern 1**	**Pattern 2**	**Pattern 3**
Connectivity/Routing: how each segment communicates to each other	System routing provides default connectivity to any workload in any subnet	Same as a pattern 1	No default connectivity between spoke virtual networks. A layer 3 router, such as the Azure Firewall, in the hub virtual network is required to enable connectivity.
Network level traffic filtering	Traffic is allowed by default. NSG can be used for filtering this pattern.	Same as a pattern 1	Traffic between spoke virtual networks is denied by default. Azure Firewall configuration can enable selected traffic, such as windowsupdate.com.

Network capabilities	Pattern 1	Pattern 2	Pattern 3
Centralized logging	NSG logs for the virtual network	Aggregate NSG logs across all virtual networks	Azure Firewall logs to Azure Monitor all accepted/denied traffic that is sent via a hub
Unintended open public endpoints	DevOps can accidentally open a public endpoint via incorrect NSG rules.	Same as a pattern 1	Accidentally opened public endpoint in a spoke virtual network won't enable access. The return packet will be dropped via stateful firewall (asymmetric routing).
Application level protection	NSG provides network layer support only.	Same as a pattern 1	Azure Firewall supports FQDN filtering for HTTP/S and MSSQL for outbound traffic and across virtual networks.
Connectivity/Routing: how each segment communicates to each other	System routing provides default connectivity to any workload in any subnet	Same as a pattern 1	No default connectivity between spoke virtual networks. A layer 3 router such as the Azure Firewall in the hub virtual network is required to enable connectivity.

Virtual network NAT gateway

Virtual Network NAT (network address translation) simplifies outbound-only Internet connectivity for virtual networks. When configured on a subnet, all outbound connectivity uses your specified static public IP addresses. Outbound connectivity is possible without load balancer or public IP addresses directly attached to virtual machines. NAT is fully managed and highly resilient.



Choose Virtual Network NAT gateway when:

- You need on-demand outbound to internet connectivity without pre-allocation
- You need one or more static public IP addresses for scale
- You need configurable idle timeout
- You need TCP reset for unrecognized connections

Routing

When you create a virtual network, Azure creates a routing table for your network. This routing table contains following types of routes.

- System routes
- Subnet default routes
- Routes from other virtual networks
- BGP routes
- Service endpoint routes
- User Defined Routes (UDR)

When you create a virtual network for the first time without defining any subnets, Azure creates routing entries in the routing table. These routes are called system routes. System routes are defined at this location. You cannot modify these routes. However, you can override systems routes by configuring UDRs.

When you create one or multiple subnets inside a virtual network, Azure creates default entries in the routing table to enable communication between these subnets within a virtual network. These routes can be modified by using static routes, which are User Defined Routes (UDR) in Azure.

When you create a virtual network peering between two virtual networks, a route is added for each address range within the address space of each virtual network a peering is created for.

If your on-premises network gateway exchanges border gateway protocol (BGP) routes with an Azure virtual network gateway, a route is added for each route propagated from the on-premises network gateway. These routes appear in the routing table as BGP routes.

The public IP addresses for certain services are added to the route table by Azure when you enable a service endpoint to the service. Service endpoints are enabled for individual subnets within a virtual network. When you enable a service endpoint, route is only added to the route table of for the subnet that belongs to this service. Azure manages the addresses in the route table automatically when the addresses change.

User-defined routes are also called Custom routes. You create UDR in Azure to override Azure's default system routes, or to add additional routes to a subnet's route table.

When you have competing entries in a routing table, Azure selects the next hop based on the longest prefix match similar to traditional routers. However, if there are multiple next hop entries with the same address prefix then Azure selects the routes in following order.

1. User-defined routes (UDR)
2. BGP routes
3. System routes

Common reasons for overriding Azure's default routing are:

- Because you want traffic between subnets to flow through an NVA. To learn more about how to configure route tables to force traffic through an NVA.
- Because you want to force all internet-bound traffic through an NVA, or on-premises, through an Azure VPN gateway. Forcing internet traffic on-premises for inspection and logging is often referred to as forced tunneling. Learn more about how to configure forced tunneling.

System routes

- When you need traffic routed between VMs in the same virtual network or peered virtual networks
- You need communication between VMs using a VNet-to-VNet VPN
- You need site-to-site communication through ExpressRoute or a VPN gateway

User defined routes (UDRs)

- You want to enable filtering of Internet traffic via Azure Firewall or forced tunneling.
- You want traffic between subnets to flow though an NVA.
- You need to create routes to specify how packets should be routed in a virtual network.
- You need to create routes that control network traffic and specify the next hop in the traffic flow.

Next unit: Design for application delivery services

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for application delivery services

3 minutes

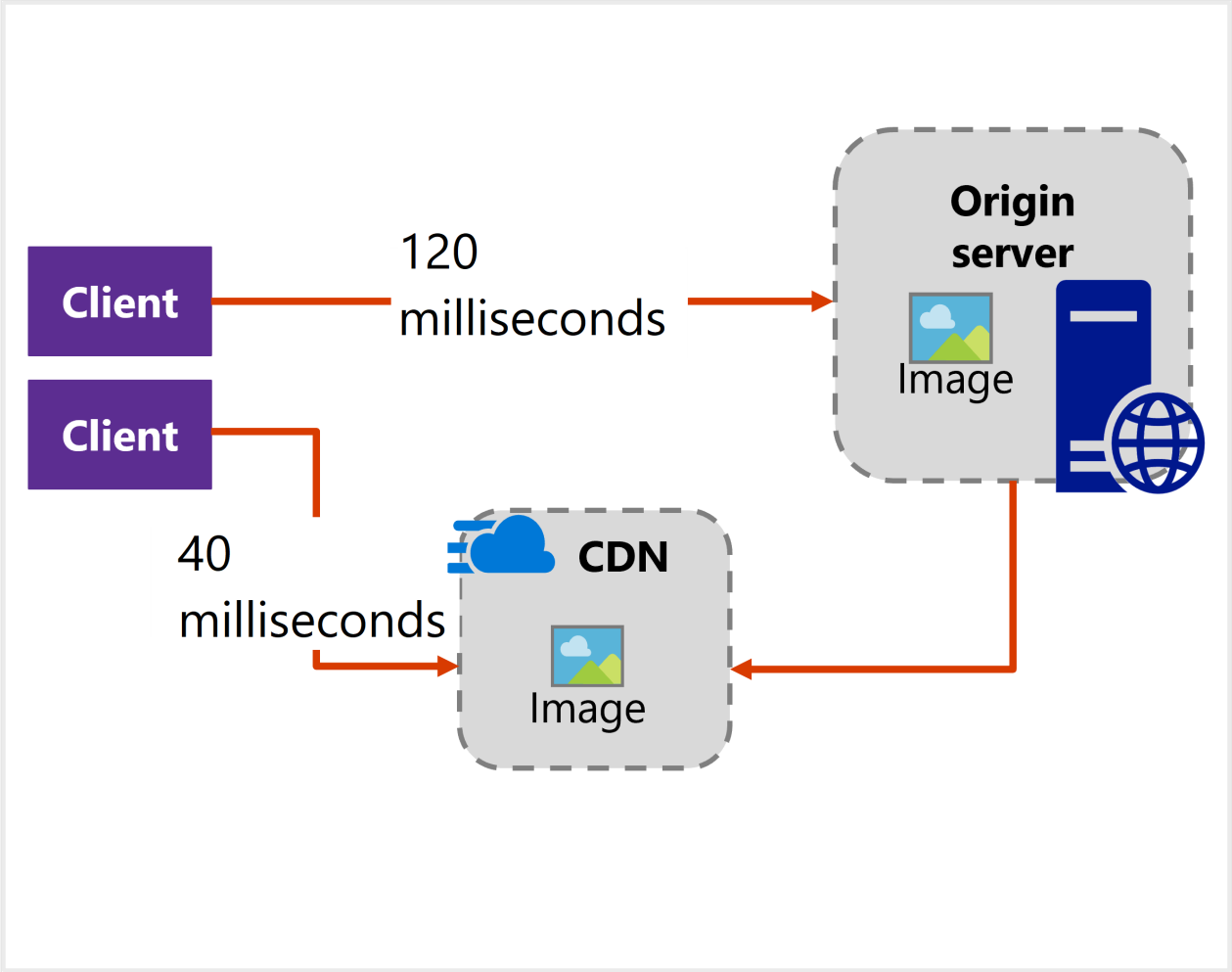
This section describes networking services in Azure that help deliver applications - Content Delivery Network, Azure Front Door Service, Traffic Manager, Load Balancer, and Application Gateway.

Content Delivery Network (CDN)

Azure Content Delivery Network (CDN) offers developers a global solution for rapidly delivering high-bandwidth content to users by caching their content at strategically placed physical nodes across the world.

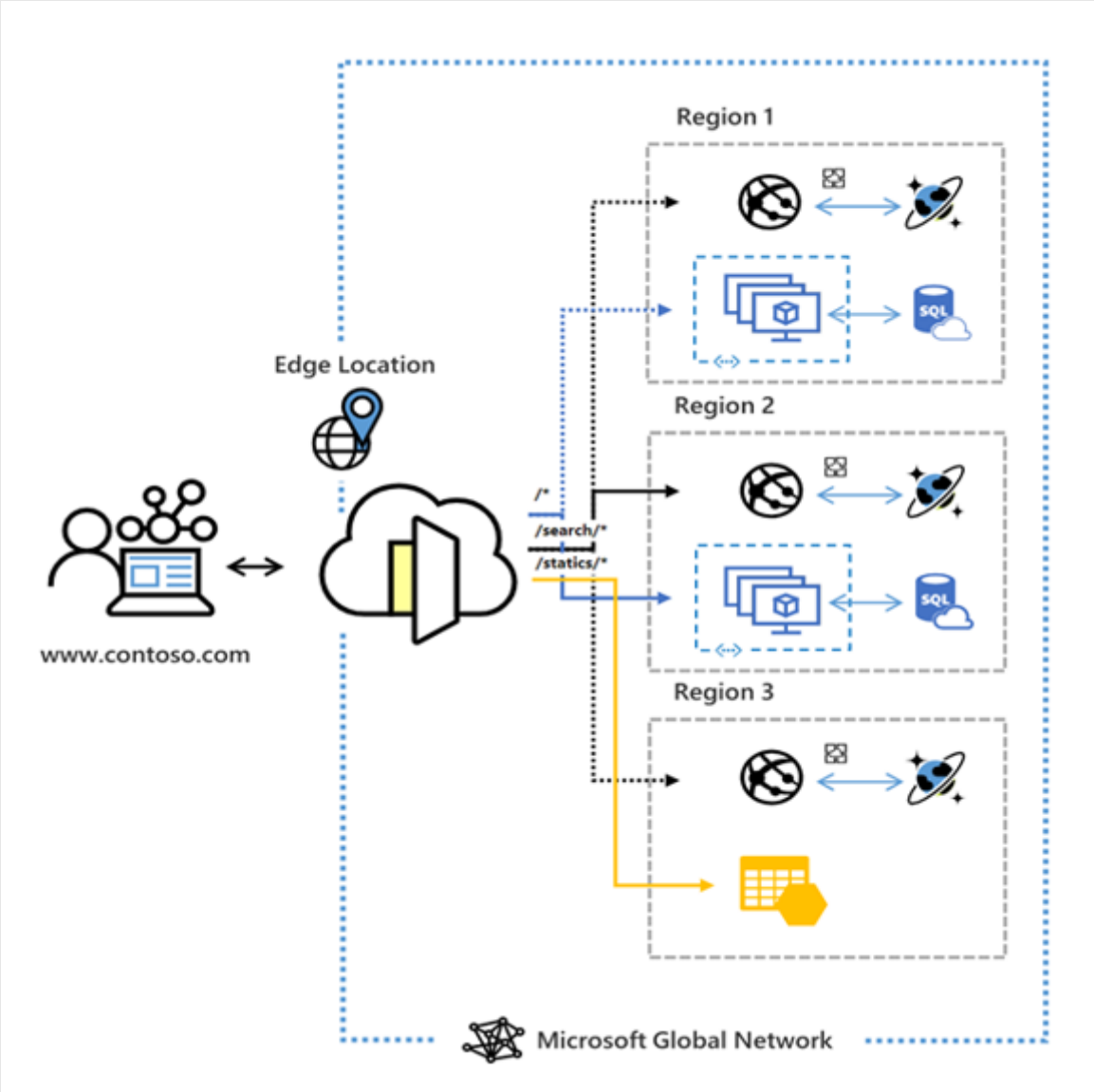
When to use a CDN:

- You want point-of-presence locations that are close to large clusters of users.
- You want to reduce latency - both the transmission delay and the number of router hops.
- You have networks in Microsoft, Akamai, or Verizon
- You want custom domains, file compression, caching, and geo-filtering



Azure Front Door Service

Azure Front Door Service enables you to define, manage, and monitor the global routing for your web traffic by optimizing for best performance and instant global failover for high availability. With Front Door, you can transform your global (multi-region) consumer and enterprise applications into robust, high-performance personalized modern applications, APIs, and content that reaches a global audience with Azure.



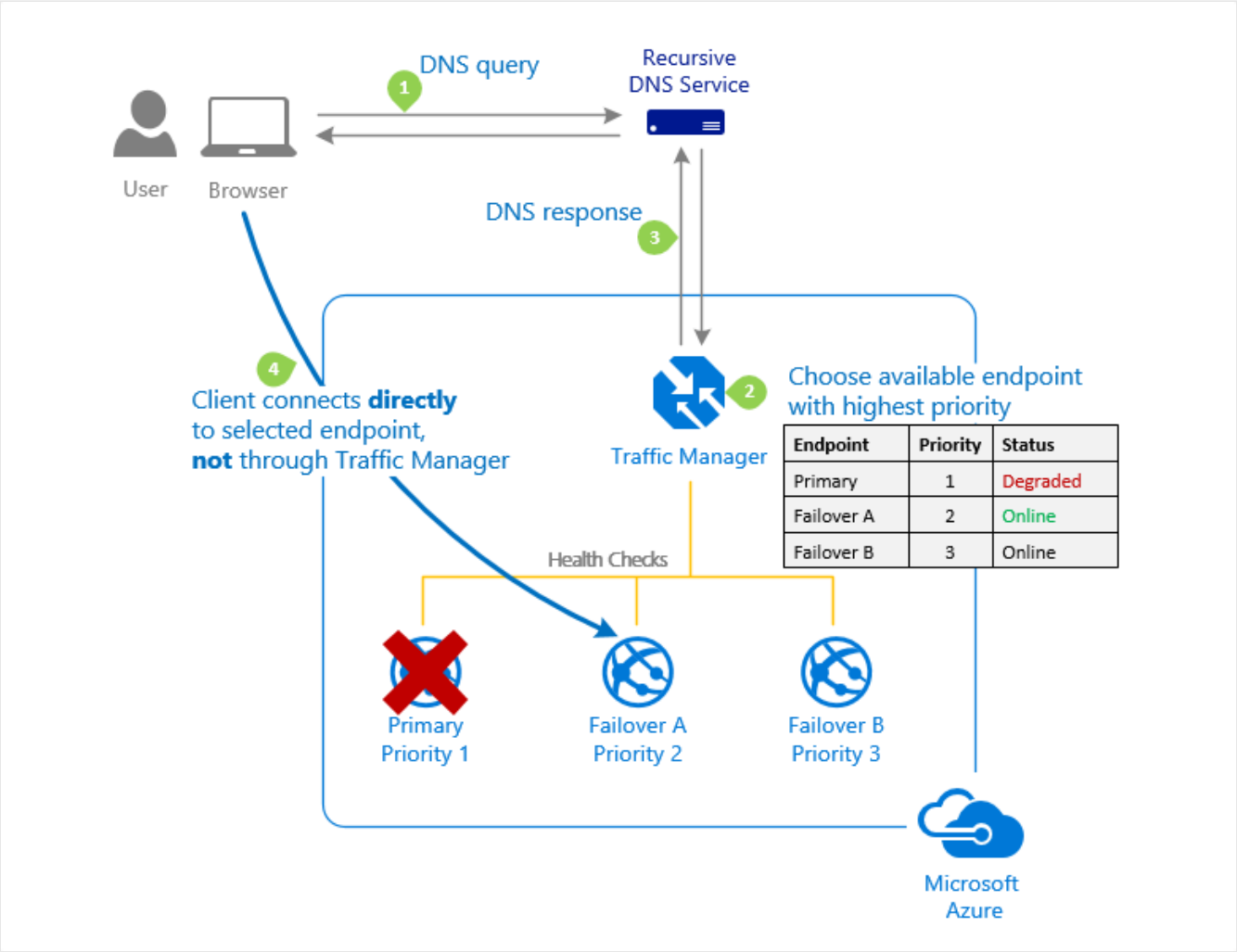
Choose front door when:

- You need to ensure that requests are sent to the lowest latency backends (low latency)
- You have primary and secondary backends (priority)
- You want to distribute traffic using weight coefficients (weighted)
- You want to ensure requests from the same end user gets sent to the same backend (affinity)
- Your traffic is HTTP(s) based and you need WAF and/or CDN integration

Traffic Manager

Azure Traffic Manager is a DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions, while providing high availability and responsiveness. Traffic Manager provides a range of traffic-routing methods to distribute traffic such as priority, weighted, performance, geographic, multi-value, or subnet.

The following diagram shows endpoint priority-based routing with Traffic Manager:



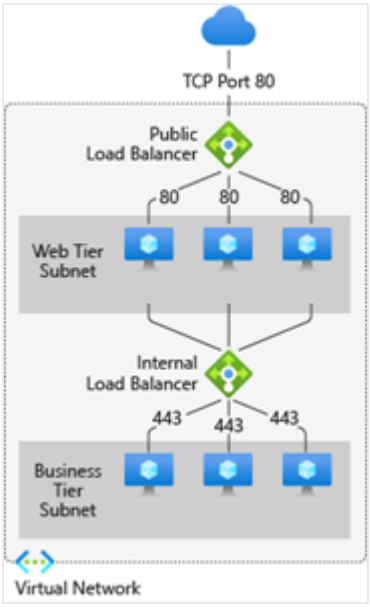
Choose Traffic Manager when:

- You need to increase application availability
- You need to improve application performance
- You need to combine hybrid applications
- You need to distribute traffic for complex deployments

Load balancer

The Azure Load Balancer provides high-performance, low-latency Layer 4 load-balancing for all UDP and TCP protocols. It manages inbound and outbound connections. You can configure public and internal load-balanced endpoints. You can define rules to map inbound connections to back-end pool destinations by using TCP and HTTP health-probing options to manage service availability.

The following picture shows an Internet-facing multi-tier application that utilizes both external and internal load balancers:

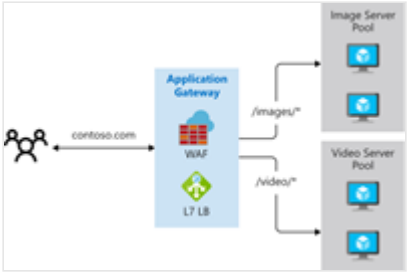


Application Gateway

Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications. It is an Application Delivery Controller (ADC) as a service, offering various layer 7 load-balancing capabilities for your applications.

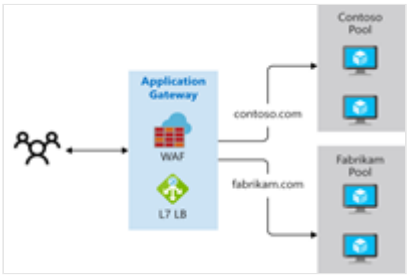
There are two primary methods of routing traffic, path-based routing, and multiple site routing.

Path-based routing



Use path-based routing to send requests with different URL paths to a different pool of backend servers

Multiple site routing



Use multiple-site routing for tenants with virtual machines or other resources hosting a web application

Choosing a load balancer solution

Azure provides various load-balancing services that you can use to distribute your workloads across multiple computing resources – Azure Front Door, Traffic Manager, Load Balancer, and Application Gateway.

This section describes how you can determine an appropriate load-balancing solution for your business needs.

Azure load-balancing services can be categorized along two dimensions: global versus regional, and HTTP(S) versus non-HTTP(S).

When selecting the load-balancing options, here are some factors that are considered when you select the **Help me choose** default tab in Azure load balancing:

Traffic type. Is it a web (HTTP/HTTPS) application? Is it public facing or a private application?

Global versus. regional. Do you need to load balance VMs or containers within a virtual network, or load balance scale unit/deployments across regions, or both?

Availability. What is the service SLA ?

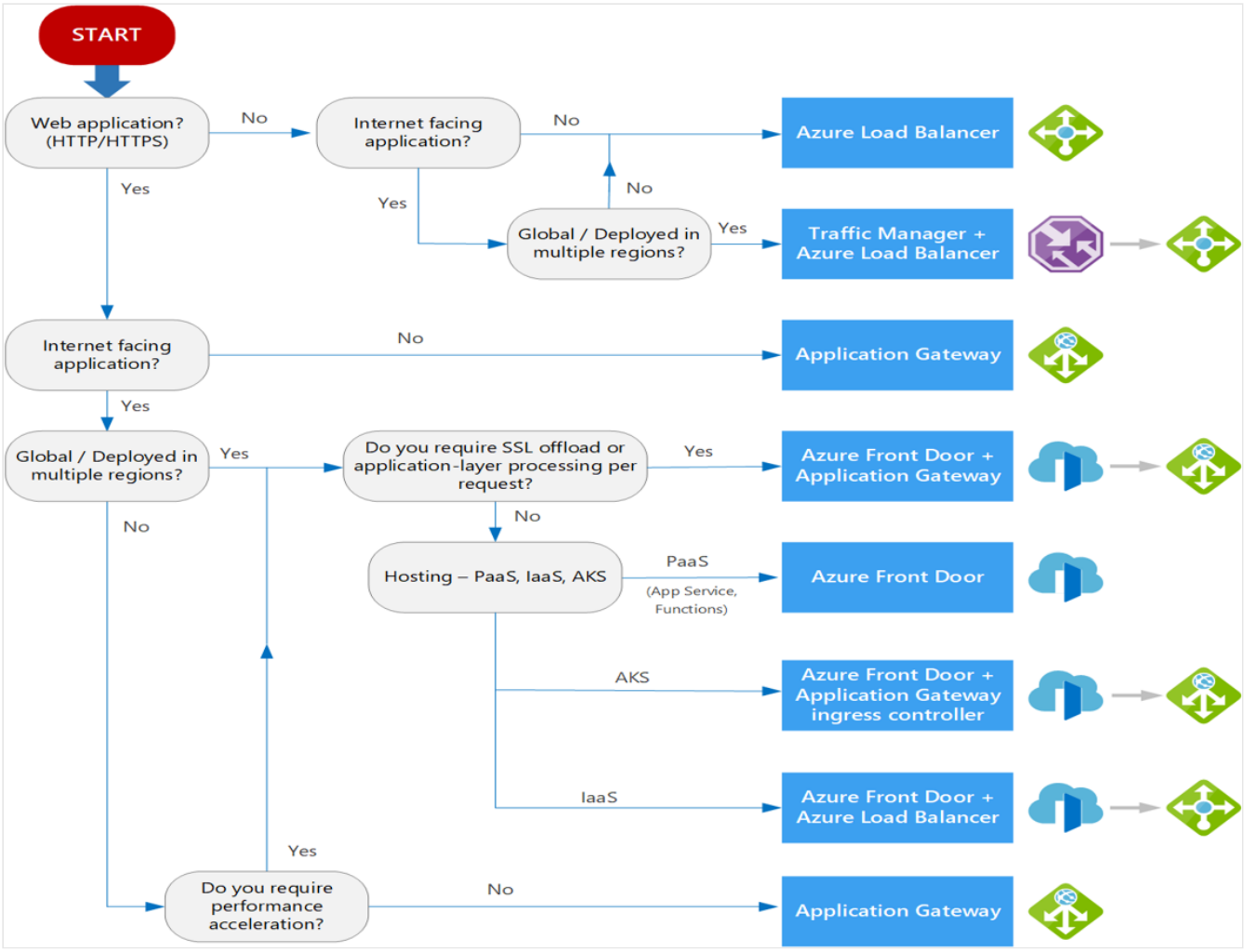
Cost. Visit Azure pricing . In addition to the cost of the service itself, consider the operations cost for managing a solution built on that service.

Features and limits. What are the overall limitations of each service? Visit Service limits.

The following flowchart will help you to choose a load-balancing solution for your application. The flowchart guides you through a set of key decision criteria to reach a recommendation.

Treat this flowchart as a starting point. Every application has unique requirements, so use the recommendation as a starting point. Then perform a more detailed evaluation.

If your application consists of multiple workloads, evaluate each workload separately. A complete solution may incorporate two or more load-balancing solutions.



Next unit: Design for application protection services

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

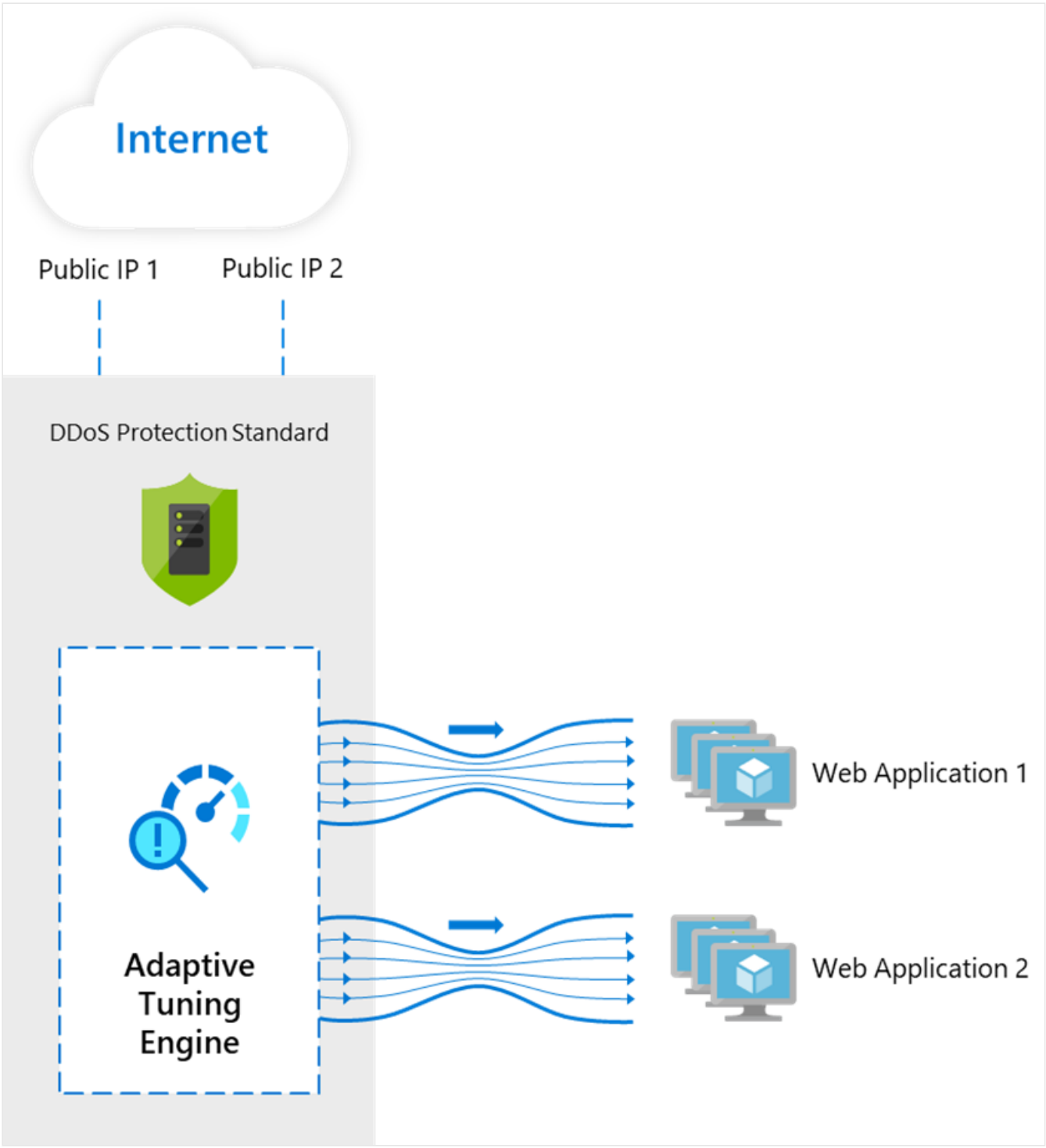
Design for application protection services

3 minutes

This section describes networking services in Azure that help protect your network resources - Protect your applications using any or a combination of these networking services in Azure - DDoS protection, Private Link, Firewall, Web Application Firewall, Network Security Groups, and Virtual Network Service Endpoints.

Distributed denial of service protection

Azure DDoS Protection provides countermeasures against the most sophisticated DDoS threats. The service provides enhanced DDoS mitigation capabilities for your application and resources deployed in your virtual networks. Additionally, customers using Azure DDoS Protection have access to DDoS Rapid Response support to engage DDoS experts during an active attack.



Use DDoS protection Standard when you need:

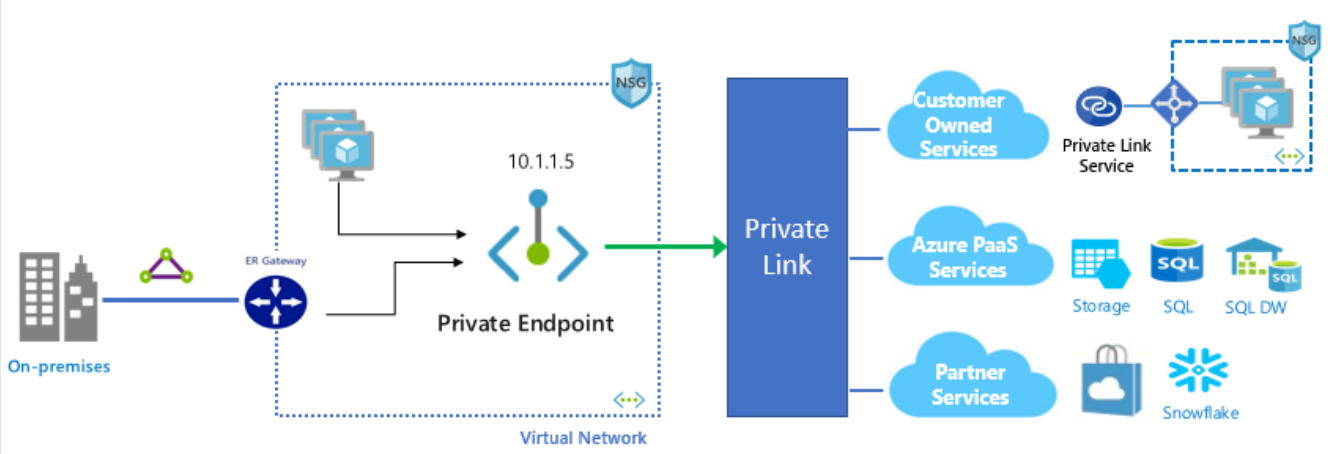
- Always-on traffic monitoring
- Adaptive tuning
- Multi-layered protection
- Mitigation scale
- Attack analytics and metrics
- Attack alerting
- DDoS rapid response team

Azure Private Link

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure hosted customer-owned/partner services over a private endpoint in your virtual network. Traffic between your virtual network and the service travels the Microsoft backbone network. Exposing your service to the public internet is no longer necessary. You can create your own private link service in your virtual network and deliver it to your customers. Private link is used to access PaaS services such as Azure Storage, Azure SQL, App Services and more as illustrated below.

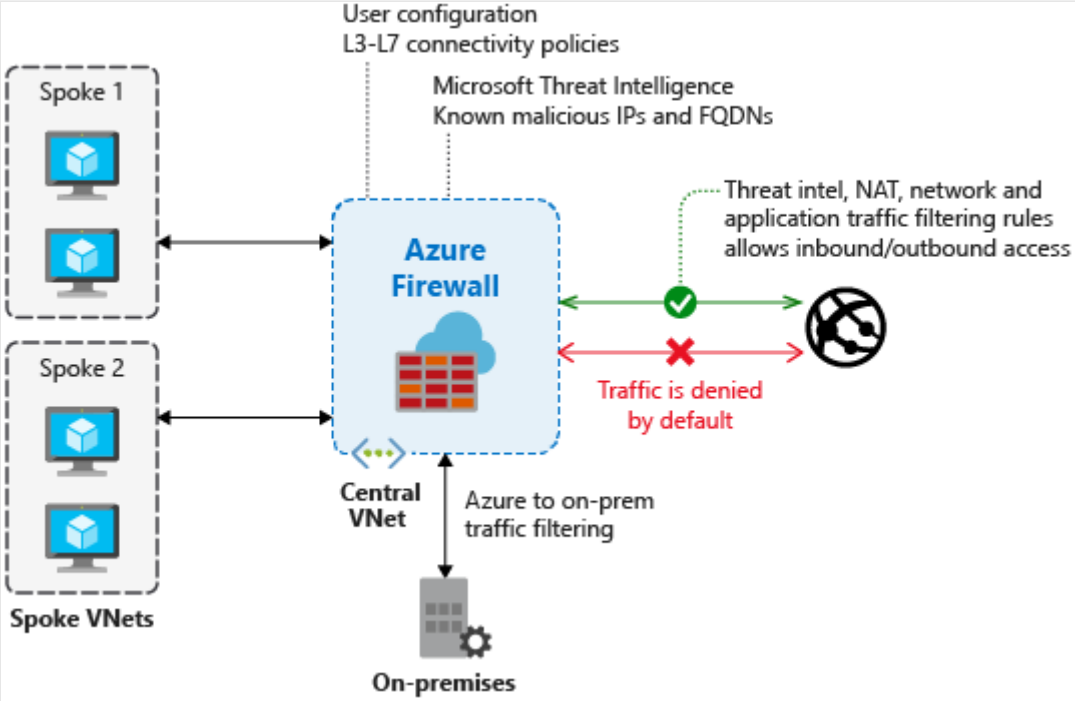
Recommend private link or private endpoints when:

- You need private connectivity to services on Azure
- You need integration with on-premises and peered networks
- You need traffic to remain on Microsoft network, with no public internet access



Azure Firewall

Azure Firewall is a managed, cloud-based network security service that protects your Azure Virtual Network resources. It's a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability. Using Azure Firewall, you can centrally create, enforce, and log application and network connectivity policies across subscriptions and virtual networks. Azure Firewall uses a static public IP address for your virtual network resources allowing outside firewalls to identify traffic originating from your virtual network. Azure Firewall provides inbound protection for non-HTTP/S protocols (for example, RDP, SSH, FTP), outbound network-level protection for all ports and protocols, and application-level protection for outbound HTTP/S.

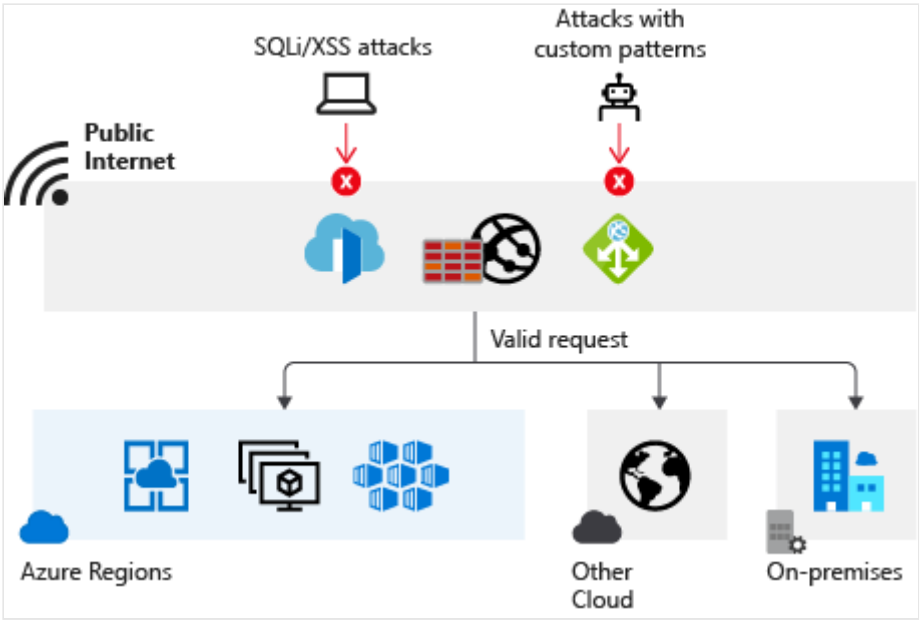


Web Application Firewall

Azure Web Application Firewall (WAF) provides protection to your web applications from common web exploits and vulnerabilities such as SQL injection, and cross site scripting. Azure WAF provides out of box protection from OWASP top 10 vulnerabilities via managed rules. Additionally, customers can also configure custom rules, which are customer managed rules to provide additional protection based on source IP range, and request attributes such as headers, cookies, form data fields or query string parameters. Preventing such attacks in application code is challenging. It can require rigorous maintenance, patching, and monitoring at multiple layers of the application topology. A centralized web application firewall helps make security management much simpler. A WAF also gives application administrators better assurance of protection against threats and intrusions.

A WAF solution can react to a security threat faster by centrally patching a known vulnerability, instead of securing each individual web application.

WAF can be deployed with Azure Application Gateway, Azure Front Door, and Azure Content Delivery Network (CDN) service from Microsoft. WAF on Azure CDN is currently under public preview. WAF has features that are customized for each specific service.



Network security groups

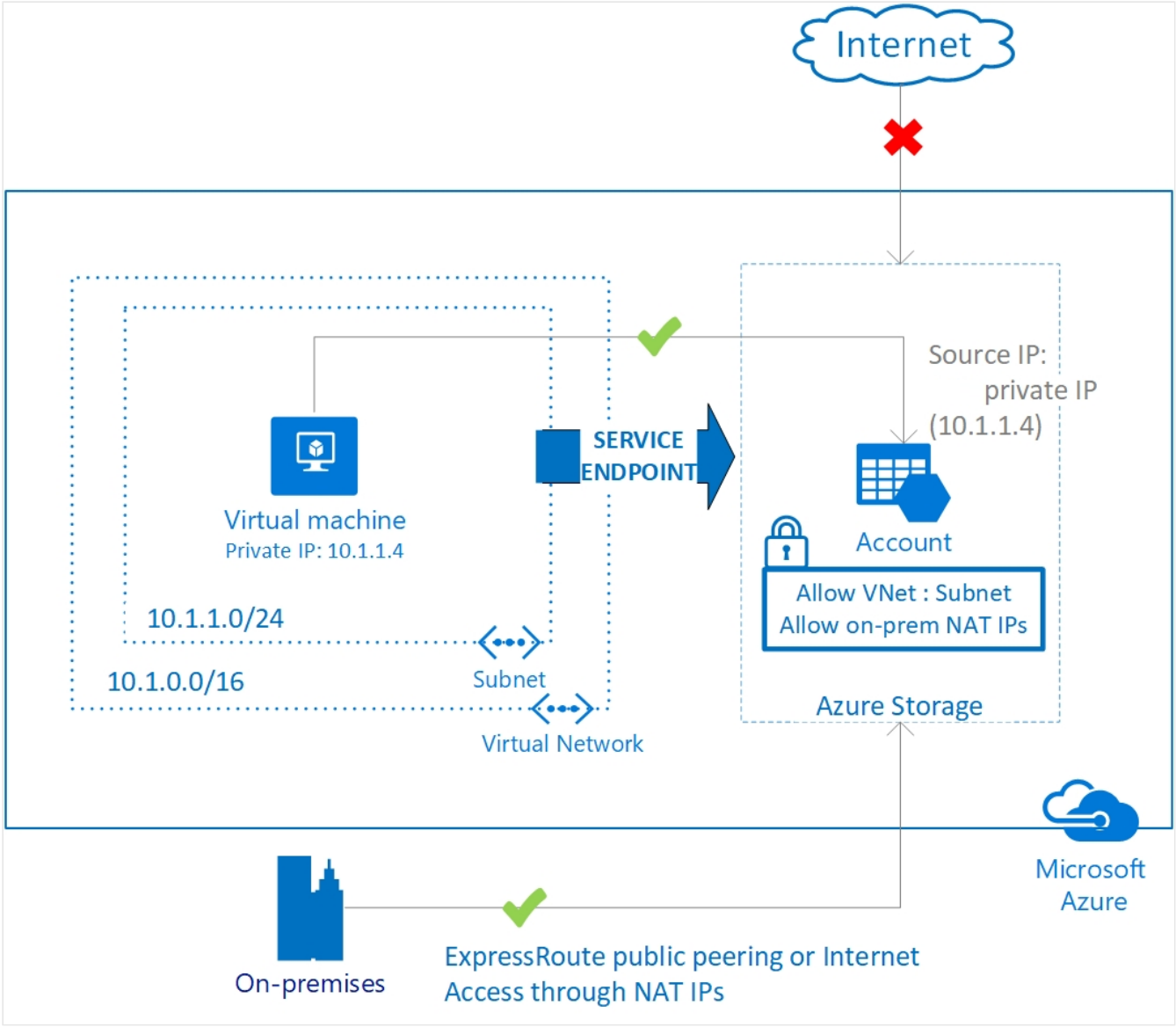
You can filter network traffic to and from Azure resources in an Azure virtual network with a network security group. You can also use network virtual appliances (NVA) such as Azure Firewall or firewalls from other vendors. You can control how Azure routes traffic from subnets. You can also limit who in your organization can work with resources in virtual networks.

A network security group (NSG) contains a list of Access Control List (ACL) rules that allow or deny network traffic to subnets, NICs, or both. NSGs can be associated with either subnets or individual NICs connected to a subnet. When an NSG is associated with a subnet, the ACL rules apply to all the VMs in that subnet. In addition, traffic to an individual NIC can be restricted by associating an NSG directly to a NIC.

NSGs contain two sets of rules: inbound and outbound. The priority for a rule must be unique within each set. Each rule has properties of protocol, source and destination port ranges, address prefixes, direction of traffic, priority, and access type. All NSGs contain a set of default rules. The default rules cannot be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create.

Service endpoints

Virtual Network (VNet) service endpoints extend your virtual network private address space and the identity of your VNet to the Azure services, over a direct connection. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Traffic from your VNet to the Azure service always remains on the Microsoft Azure backbone network.



Key Benefits:

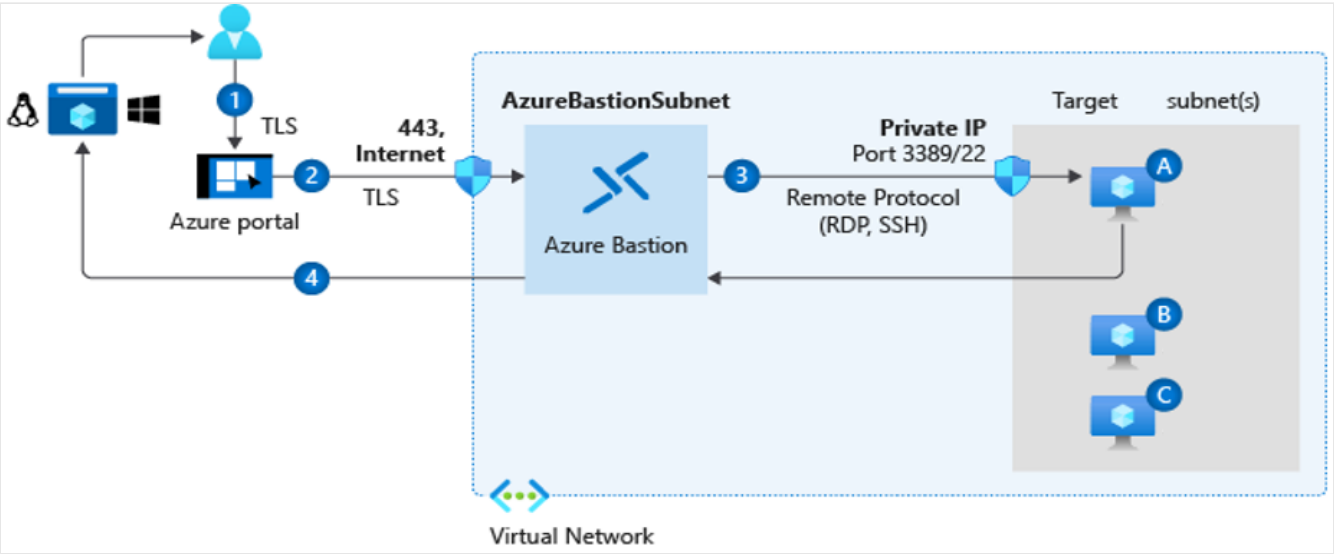
- Improved security for your Azure service resources
- Optimal routing for Azure service traffic from your virtual network
- Simple to set up with less management overhead

Azure Bastion

The Azure Bastion service is a new fully platform-managed PaaS service that you provision inside your virtual network. It provides secure and seamless RDP/SSH connectivity to your virtual machines directly in the Azure portal over TLS. When you connect via Azure Bastion, your virtual machines do not need a public IP address.

Recommend Azure Bastion when you need to:

- Secure remote connections from the Azure portal to Azure VMs
- Eliminate exposing RDP ports, SSH ports, or public IP addresses for your internal VMs



Key security features

- Traffic initiated from Azure Bastion to target virtual machines stays within the virtual network or between peered virtual networks.
- There's no need to apply NSGs to the Azure Bastion subnet, because it's hardened internally. For additional security, you can configure NSGs to allow only remote connections to the target virtual machines from the Azure Bastion host.
- Azure Bastion helps protect against port scanning. RDP ports, SSH ports, and public IP addresses aren't publicly exposed for your VMs.
- Azure Bastion helps protect against zero-day exploits. It sits at the perimeter of your virtual network. So you don't need to worry about hardening each of the virtual machines in your virtual network. The Azure platform keeps Azure Bastion up to date.
- The service integrates with native security appliances for an Azure virtual network, like Azure Firewall.
- You can use the service to monitor and manage remote connections.

Just in time (JIT) network access

With JIT, you can lock down the inbound traffic to your VMs, reducing exposure to attacks while providing easy access to connect to VMs when needed.

When you enable just-in-time VM access, you can select the ports on the VM to which inbound traffic will be blocked. This ensures "deny all inbound traffic" rules exist for your selected ports in the network security group (NSG) and Azure Firewall rules. These rules restrict access to your Azure VMs' management ports and defend them from attack.

If other rules already exist for the selected ports, then those existing rules take priority over the new "deny all inbound traffic" rules. If there are no existing rules on the selected ports, then the new rules take top priority in the NSG and Azure Firewall.

When a user requests access to a VM, Security Center checks that the user has Azure role-based access control (Azure RBAC) permissions for that VM. If the request is approved, NSGs and Azure Firewall allow inbound traffic to the selected ports from the relevant IP address (or range), for the amount of time that was specified. After the time has expired, the NSGs are returned to their previous states. Connections that are already established are not interrupted.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

200 XP

Knowledge check

3 minutes

Tailwind Traders has several requirements to meet for their production network environment. It's important that you select the right networking solutions to meet all of the requirements. Here are the specific requirements.

- A web application that isn't internet facing. They need to load balance incoming traffic to the web application, but it can't be internet facing.
- Network security. Filter HTTP(S) traffic from Azure to on premises. They also need to filter traffic outbound to the internet.
- Network architecture. The Network team intends to deploy resources across several Azure regions, requires global connectivity between VNets in these Azure regions and multiple on-premises locations, and wishes to centrally manage the networks and connections.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. Which load-balancing solution is recommended?

- ○ Application Gateway

  ✔ That's correct. An Application Gateway is the best choice for a web application that isn't internet facing.

- ◉ Front Door

  ✘ That's incorrect. Front Door isn't needed for non-internet facing applications.

- ○ Azure Load balancer

2. Which security solution is recommended?

- ◉ Azure Firewall

  ✔ That's correct. Azure Firewall can filter HTTP(S) traffic from Azure to on-premises and outbound to the internet

- ○ Web Application Firewall on Azure Application Gateway
- ○ Azure Bastion

3. What network topology is recommended based on the architecture requirement?

- ◉ Virtual WAN network topology

Virtual WAN network topology

✔ That's correct. A Virtual WAN topology meets the requirements for deploying resources across several Azure regions and enables global connectivity between VNets in these Azure regions and multiple on-premises locations.

- ○ Traditional network topology
- ○ Single Isolated Virtual Network

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Introduction

3 minutes

Meet Tailwind Traders

You work for Tailwind Traders, a home improvement retailer. Tailwind Traders currently manages on-premises datacenters that host the company's retail website. These datacenters also store all the data and streaming video for its applications.

The IT department is currently responsible for all the management tasks for its computing hardware and software. The IT team handles the procurement process to buy new hardware, installs and configures software, and deploys everything throughout the datacenter.

These management responsibilities create some obstacles for delivering applications to Tailwind Trader's users and customers in a timely fashion. As a result, you've been tasked with reviewing available migration options. You must also select the appropriate options to use to support the planned migrations. The planned workloads include virtual machines (VMs), databases, and applications.

After completing this module, you'll be able to assess and select a suitable migration strategy to support migration of your on-premises workloads.

Learning objectives

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

- Evaluate migration with the Cloud Adoption Framework.
- Describe the Azure Migration Framework.
- Assess your on-premises workloads.
- Select a migration tool.
- Migrate your databases.

- Select an online storage migration tool.
- Migrate offline data.

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions.

Design Infrastructure

- Design Migrations

Prerequisites

- Conceptual knowledge of migrating compute, database, and storage workloads.
- Working experience with planning migrations, assessing workloads, determining migration requirements, and deploying workloads.

Next unit: Evaluate migration with the Cloud Adoption Framework

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Evaluate migration with the Cloud Adoption Framework

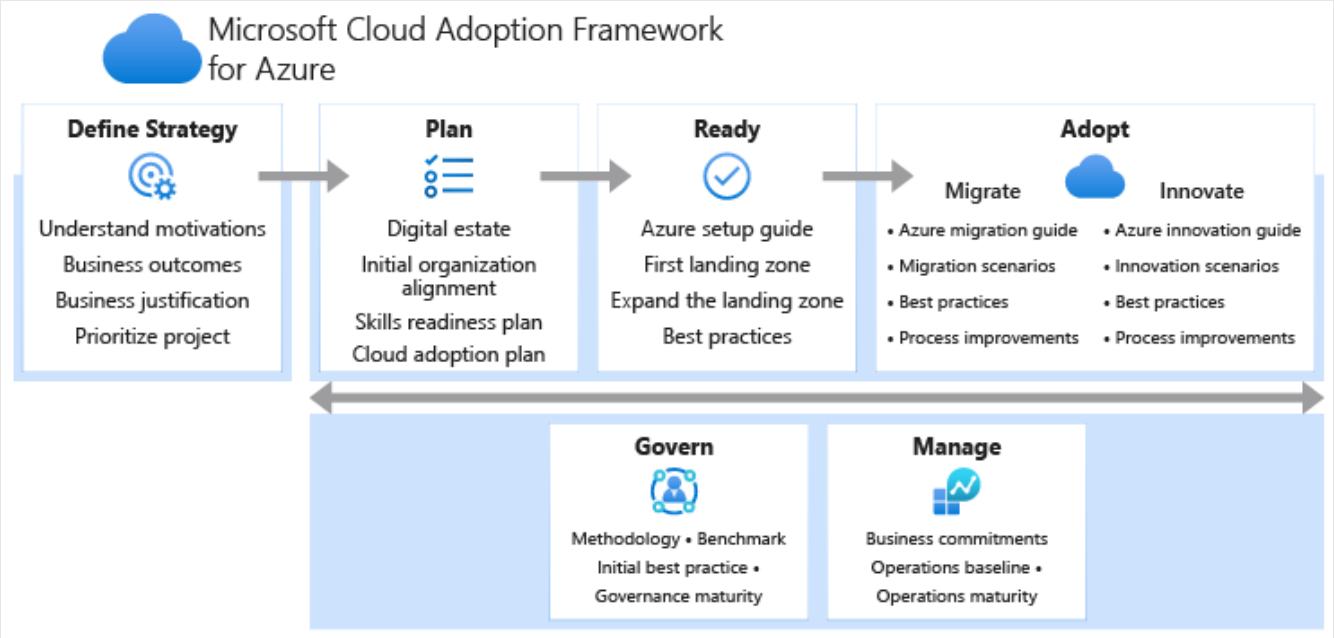
3 minutes

Understand Cloud migration in the Cloud Adoption Framework

The Microsoft Cloud Adoption Framework for Azure is provided to help you drive adoption of Azure in your organization. It provides recommendations, best practice guidance, documentation, and tools. The framework supports several methodologies:

- **Define strategy**: Define business justification and expected outcomes of adoption.
- **Plan**: Align actionable adoption plans to business outcomes.
- **Ready**: Prepare the cloud environment for the planned changes.
- **Migrate**: Migrate and modernize existing workloads.
- **Innovate**: Develop new cloud-native or hybrid solutions.
- **Govern**: Govern the environment and workloads.
- **Manage**: Operations management for cloud and hybrid solutions.
- **Organize**: Align the teams and roles supporting your organization's cloud adoption efforts.

Each of the methodologies is a phase within a cloud adoption lifecycle.



Tailwind Traders shouldn't undertake cloud adoption without considerable planning. This is especially true with the migrate phase in the cloud adoption lifecycle. To prepare you for this phase, you should review the following documentation:

- Azure migration guide overview: Review the Azure migration guide to learn about Azure native tools and a relevant approach to migration.
- The One Migrate approach to migrating the IT portfolio. Review the scenarios captured in this Migrate methodology: They demonstrate the same set of consistent guidelines and processes for migrating both Microsoft and third-party technologies.
- Azure cloud migration best practices checklist: Review this document to learn how best to address common migration needs through the application of consistent best practices.
- Cloud Adoption Framework migration model: Review this document to understand mitigation. Migration can be process intensive activity. As you increase migration effort, review these process improvements to help to optimize aspects of your migration.

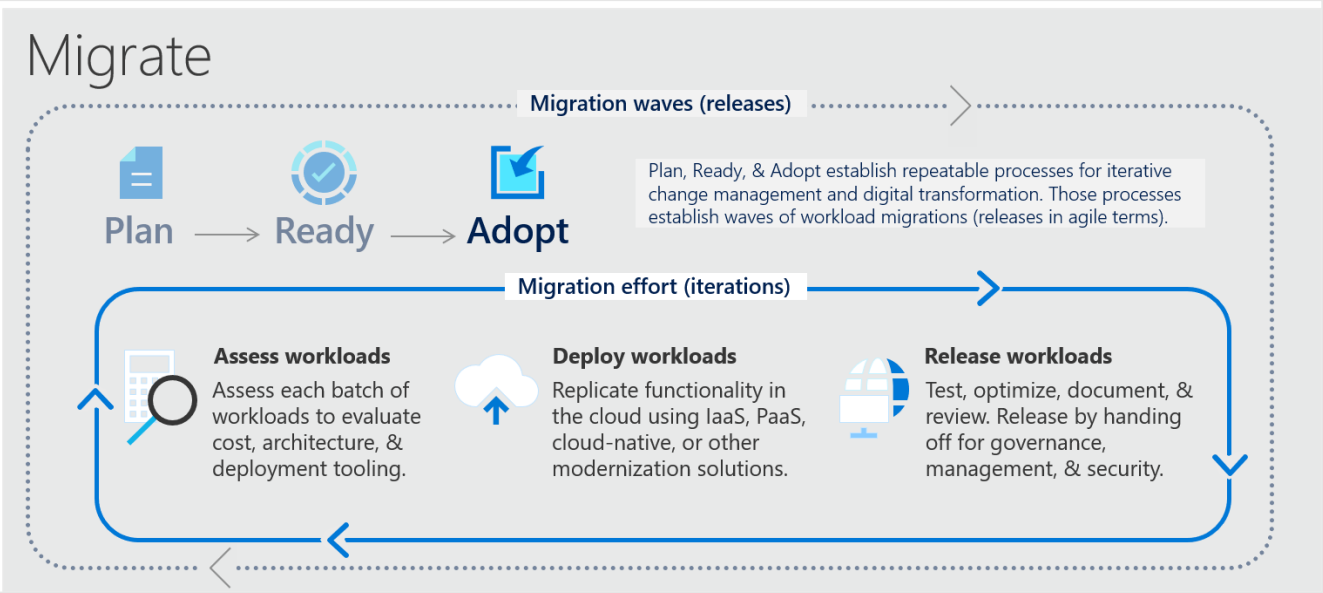
Understand the migration effort

The actions required to migrate Tailwind Traders' workloads will almost certainly fall into three efforts (or phases) for each workload:

- Assess
- Deploy
- Release

This section of the Cloud Adoption Framework explains how to maximize the return from each of the phases required to migrate a workload to production. The following table provides an overview of the phases of this process, as displayed in the diagram below:

Phase	Explanation
Assess	Assess your workloads to determine costs, modernization, and required deployment tools.
Deploy	After workloads are assessed, the existing functionality of those workloads is replicated (or improved) in the cloud.
Release	After workloads are replicated to the cloud, you can test, optimize, and document your migrated workloads. When satisfied, you can release these workloads to users. During this phase, ensure that you hand off the workloads to governance, operations management, and security teams for ongoing support of those workloads.



Next unit: Describe the Azure migration framework

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Describe the Azure migration framework

3 minutes

What is the Azure migration framework?

Before you can start migrating Tailwind Traders' on-premises workloads to Azure, you should consider creating a migration plan. This plan should help you to identify the workloads that you want to migrate. The plan should also help you identify and select the appropriate service or tools to use during the migration.

Ideally, your plan should also include details about how to optimize the migrated services. The Azure migration framework can help you work through your plan and the migration it addresses.

The Azure migration framework consists of four stages:

- Assess
- Migrate
- Optimize
- Monitor

Assess your on-premises environment

The best place to start is with an assessment of Tailwind Traders' current on-premises environment. During the assessment, you should:

- Identify apps, and their related servers, services, and data, that is within scope for migration.
- Start to involve stakeholders, such as the IT department and relevant business groups.
- Create a full inventory and a dependency map of servers, services, and apps that you're selecting for migration.
- Estimate cost savings by using the Azure Total Cost of Ownership Calculator.

- Identify appropriate tools and services you can use to perform the assessment, migration, optimization, and monitoring stages.

Strategies for migration to the cloud fall into four broad patterns: rehost, refactor, rearchitect, or rebuild. The strategy you adopt depends on your business drivers and migration goals. You might even adopt multiple patterns. For example, you might choose to rehost simple apps or apps that aren't critical to your business, but rearchitect apps that are more complex and business critical.

The following table describes the four patterns.

Pattern	Definition	When to use
Rehost	Often referred to as a lift and shift migration, this option doesn't require code changes, and allows you to migrate your existing workloads to Azure quickly. Each workload is migrated as is, without the risk and cost associated with code changes.	When you need to move workloads quickly to the cloud, when you want to move a workload without modifying it, when your apps are designed so that they can take advantage of Azure IaaS scalability after migration, and when workloads are important to your business, but you don't need immediate changes to app capabilities.
Refactor	Often referred to as repackaging, refactoring requires minimal changes to apps so they can connect to Azure platform as a service (PaaS) and use cloud offerings. For example, you could migrate existing apps to Azure App Service or Azure Kubernetes Service (AKS). Alternatively, you could refactor relational and non-relational databases into options such as Azure SQL Database Managed Instance, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Cosmos DB, but only if your app can easily be repackaged to work in Azure.	If you want to apply innovative DevOps practices provided by Azure, or if you're thinking about DevOps using a container strategy for workloads. For refactoring, you need to think about the portability of your existing code base and available development skills.

Pattern	Definition	When to use
Rearchitect	Rearchitecting for migration focuses on modifying and extending app functionality and the code base to optimize the app architecture for cloud scalability. For example, you could break down a monolithic application into a group of microservices that work together and scale easily. Alternatively, you could rearchitect relational and nonrelational databases to a fully managed database solution such as Azure SQL Database Managed Instance, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Cosmos DB.	When your apps need major revisions to incorporate new capabilities, or to work effectively on a cloud platform. When you want to use existing application investments, meet scalability requirements, apply innovative DevOps practices, and minimize use of VMs.
Rebuild	Rebuild takes things a step further by completely rebuilding an app using Azure cloud technologies. For example, you could build green-field apps with cloud-native technologies such as Azure Functions, Azure AI, Azure SQL Database Managed Instance, and Azure Cosmos DB.	When you want rapid development, and existing apps have limited functionality and lifespan. When you're ready to expedite business innovation by using Azure DevOps practices. When are building new applications using cloud-native technologies, like Azure Blockchain. When you are rebuilding legacy apps as no code or low apps in the cloud.

Choosing which strategy to use depends on what you’re trying to accomplish.

Migrate your workloads

After you complete the assessment, you can begin the process of migrating your targeted apps and their related services and data. The migration stage typically consists of the following elements:

Deploy cloud infrastructure targets: Before you can migrate Tailwind Traders' workloads, you'll need to create the required cloud infrastructure targets. Depending on the tools you use to perform the migration, you might need to create the required Azure resources before you begin the migration. Some tools, such as Azure Migrate and Azure Database Migration Service, can create the target Azure resources for you.

Migrate workloads: It's a good idea to pilot your workload migration, and to choose a non-critical app for the pilot. This approach enables you to:

- Become familiar with tools
- Gain experience with processes and procedures
- Reduce risk when migrating large or complex workloads

Depending on the workload you plan to migrate, the steps used to perform the migration will vary.

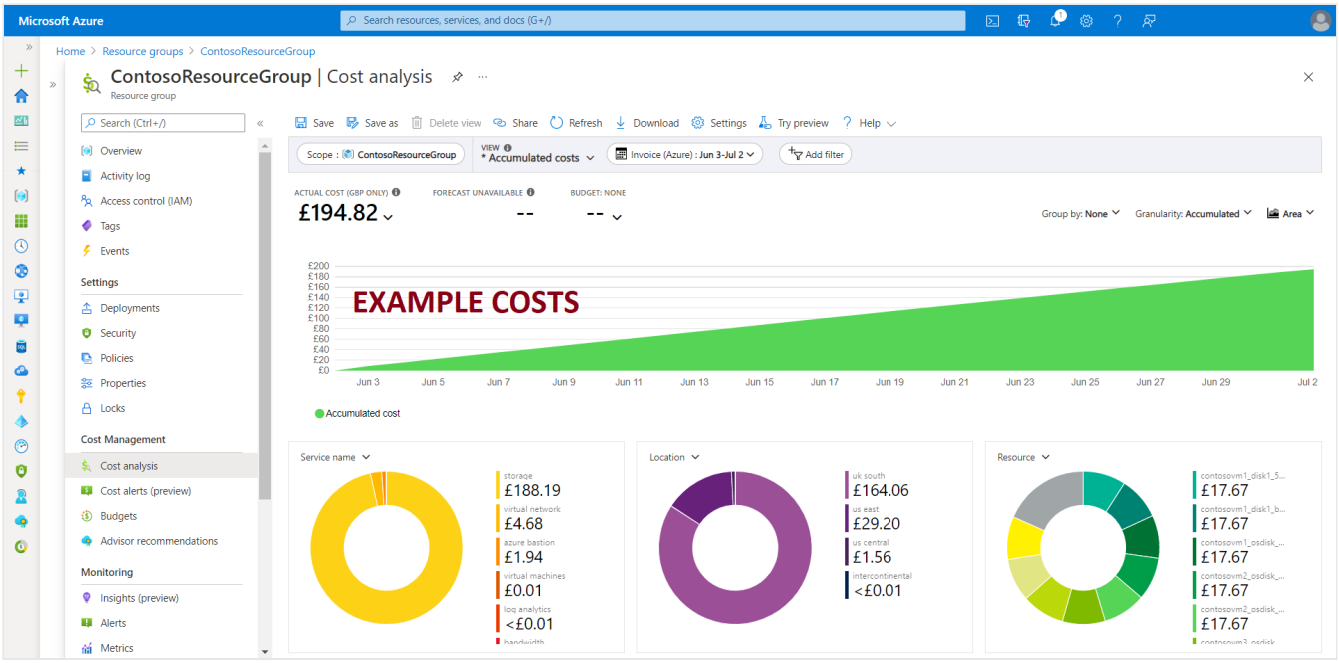
Decommission on-premises infrastructure: After you're satisfied that your source apps and databases are migrated successfully, you must decommission those source workloads. Consider retaining the source workload backups and archived data. This data might prove useful as it provides a historical archive. You can store these backups and archives in Azure Blob storage.

Optimize the migrated workloads

During the optimization stage, you should:

- Analyze the costs of the workload migration
- Identify ways to reduce costs
- Seek to improve workload performance

To analyze costs, you can use Cost Management in the Azure portal. Select the Azure resource group in which you're interested (the one which contains the migrated workloads), and then, in the navigation pane, select **Costs analysis** in the **Cost Management** section. The following screenshot shows the cost analysis for the last billable period for a resource group called ContosoResourceGroup. The results display the costs according to service name, region, and resource, although you can customize the results to your needs.



To help to reduce costs, select the **Advisor recommendations** link in the navigation pane. Recommendations, if there are any, are displayed on the details pane.

Monitor your workloads

You can use Azure Monitor to capture health and performance information from your Azure VMs. However, you must first install a Log Analytics agent on target VMs. After you've installed the agent, you can then set up alerting and reporting.

> **Tip**
>
> You can install the agent on machines running either Windows or Linux.

You can set up alerts based on a range of data sources, such as:

- Specific metric values like CPU usage
- Specific text in log files
- Health metrics
- An Autoscale metric

Next unit: Assess your on-premises workloads

Continue >

100 XP

Assess your on-premises workloads

3 minutes

Before you begin the migration project at Tailwind Traders, you should be familiar with the available tools and possible procedures that you can use during the migration. This unit discusses those tools and procedures.

Describe migration tools

There are a range of tools and services that can help you plan and complete the four stages of your migration. However, in some migrations, you might need to use only one or two of these services or tools.

Service or tool	Stage	Description
Service Map	Assess	Maps communication between app components on Windows or Linux. Can help you to identify dependencies when determining what to migrate. Service map requires an additional agent to be installed on the source environment VMs.
Azure TCO Calculator	Assess	Provides an estimate of your monthly running costs in Azure, which enables a comparison with on-premises costs.
Azure Migrate	Assess and migrate	Performs assessment and migration to Azure of VMs (Hyper-V and VMware), cloud based VMs, physical servers, databases, data, virtual desktop infrastructure, and web applications.
Data Migration Assistant (DMA)	Assess and Migrate	Performs assessment and migration specifically for Azure SQL database.

Service or tool	Stage	Description
Azure Database Migration Service	Assess and Migrate	Performs assessment and migration for several different databases, not just Azure SQL database.
Data Migration Tool	Migrate	Migrates your existing databases to Azure Cosmos DB.
Microsoft Cost Management	Optimize	Helps you monitor, optimize, and control your ongoing Azure costs.
Azure Advisor	Optimize	Helps optimize your Azure resources for reliability, performance, cost, security, and operational excellence.
Azure Monitor	Monitor	Collects monitoring telemetry from both on-premises and Azure resources. Enables you to analyze data, setup alerts, and identify problems.
Microsoft Sentinel	Monitor	Provides intelligent security analytics for your applications enabling you to collect, detect, investigate, and respond to incidents.

Assess your environment

You can use several assessment tools to help in Tailwind Traders' migration to Azure. These are:

- Service Map
- Azure TCO Calculator
- Azure Migrate

Use Azure Service Map

You use Service Map to automatically discover apps and their components in Tailwind Traders' on-premises environment. By doing so, you gain an invaluable insight into the app structure at

Tailwind Traders. This insight lets you effectively plan and perform your migration. Service Map supports discovery on both Windows and Linux platforms.

By using Service Map, you can review the Tailwind Traders' server environment as a collection of interconnected systems. Service Map displays:

- Connections between servers
- Server processes
- Inbound and outbound connection latency
- TCP or UDP ports across any connected architecture

You'll need the following to use Service Map:

- A Log Analytics workspace
- The Log Analytics agent installed on the Windows or Linux computers. This agent collects events and performance data from the computer and delivers it to the Log Analytics workspace.
- The Dependency agent installed on the Windows or Linux computers. This agent collects discovered data about processes running on the computer, and its external process dependencies.

Service Map provides the following functions:

- Discovery, which helps:
  - Build a common reference of dependencies of your servers and their processes.
  - Review the discovered information as an intuitive graphical map.
  - Identify failed network connections.
- Incident management, which helps:
  - Eliminate guesswork around problem isolation.
  - Identify misconfigured systems and components.
- Migration assurance, which helps:
  - Plan, accelerate, and validate your Azure migrations.
  - Ensure that nothing is left behind and unexpected outages don't occur.

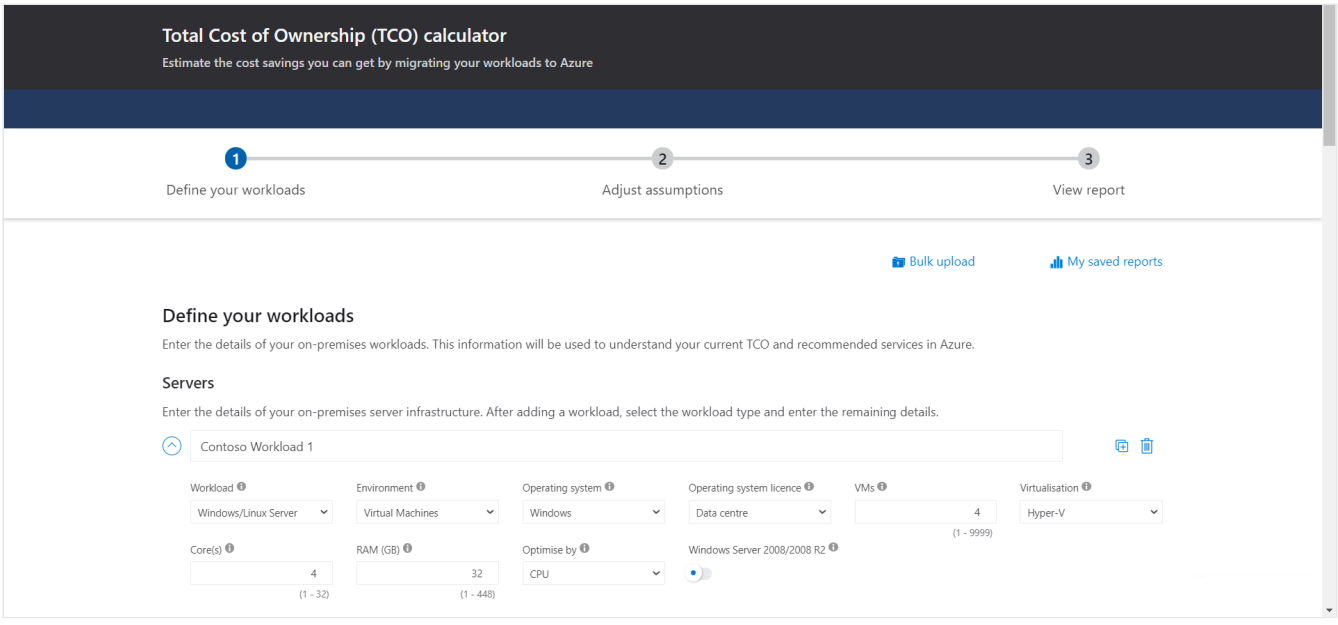
- Business continuity, which helps:
  - Identify how your systems rely on each other, thereby helping to ensure that your recovery plan is reliable.
  - Identify, which front-end systems you must recover after a server is restored and available once more.
- Patch management, which helps:
  - Identify, which other teams and servers depend on a service being patched.
  - Enable you to notify teams in advance before you take down your systems for patching.

Use Azure TCO Calculator

You can use the Azure TCO Calculator to estimate and optimize Tailwind Traders expected Azure costs following your migration. You start by defining the characteristics of Tailwind Traders' existing workloads, including:

- Servers
- Databases
- Storage
- Networking

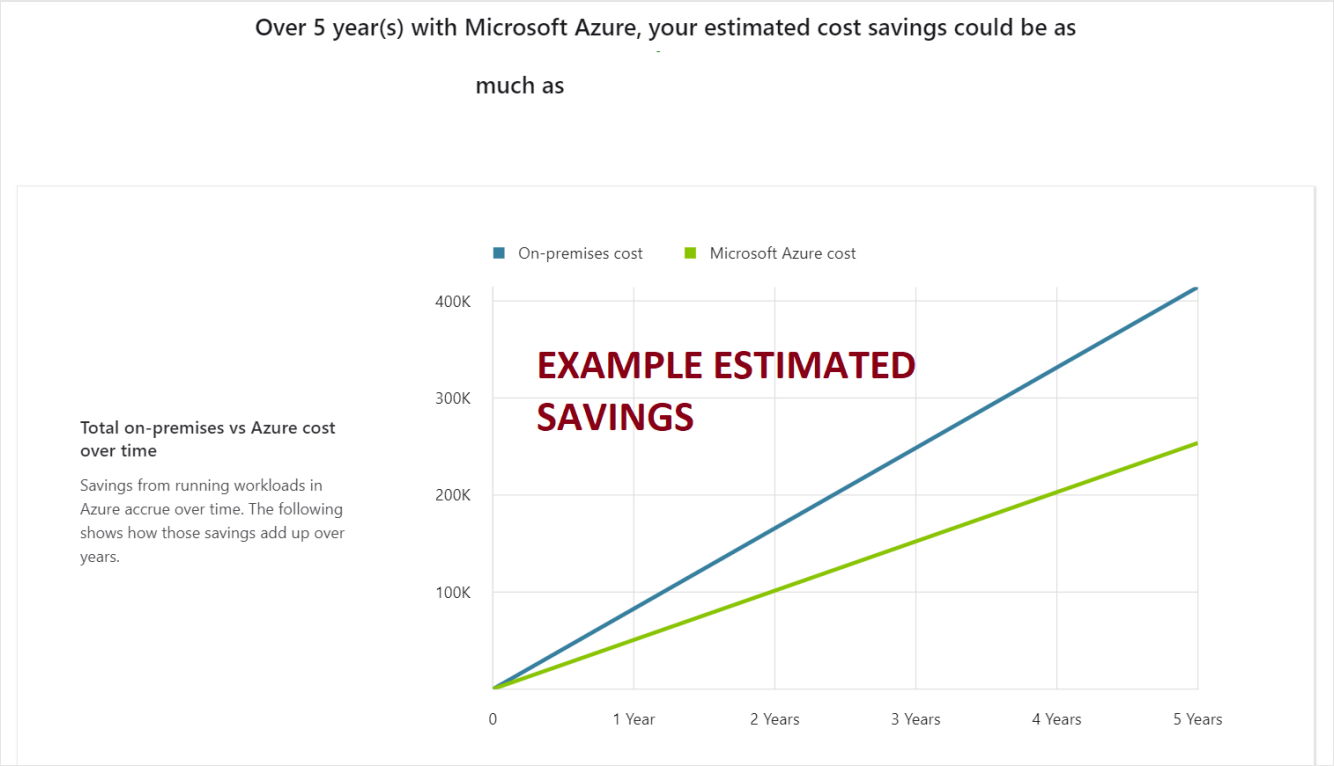
The following screenshot displays an example workload for servers:



Next, you can adjust assumptions, including:

- Software Assurance coverage
- Geo-redundant storage
- Virtual machine costs
- Electricity costs
- Storage costs
- IT labor costs
- Other assumptions, including hardware and software costs, virtualization costs, datacenter costs, and so on.

Making these adjustments enables you to fine tune your workloads to reflect their actual cost to Tailwind Traders as closely as possible. Finally, on the View report page, you can review information about possible cost savings, as displayed in the following screenshot.



The information available from the report can be useful in helping identify the benefits in moving from an on-premises content to one based in Azure.

Use Azure Migrate

Using Azure Migrate, you can perform an agentless environment discovery or use agents to perform a dependency analysis. The Azure portal helps you:

- Assess your current on-premises workloads.

- Makes recommendations for the size of VM you'll need to provision.

You want to assess readiness for the move to Azure. You also want to identify estimated costs for the resources that those machines will consume, so the management team can set the budgets.

Azure Migrate helps with performance-based sizing calculations (VM sizing, compute/storage) for the machines that you'll migrate and estimate the ongoing cost of running these machines in Azure. Azure Migrate can assess both Hyper-V and VMware-based virtual machines, as well as physical servers.

Azure Migrate also supports the visualization of dependencies for those machines. It helps you create groups of machines that can be assessed together and ultimately migrated to Azure at the same time.

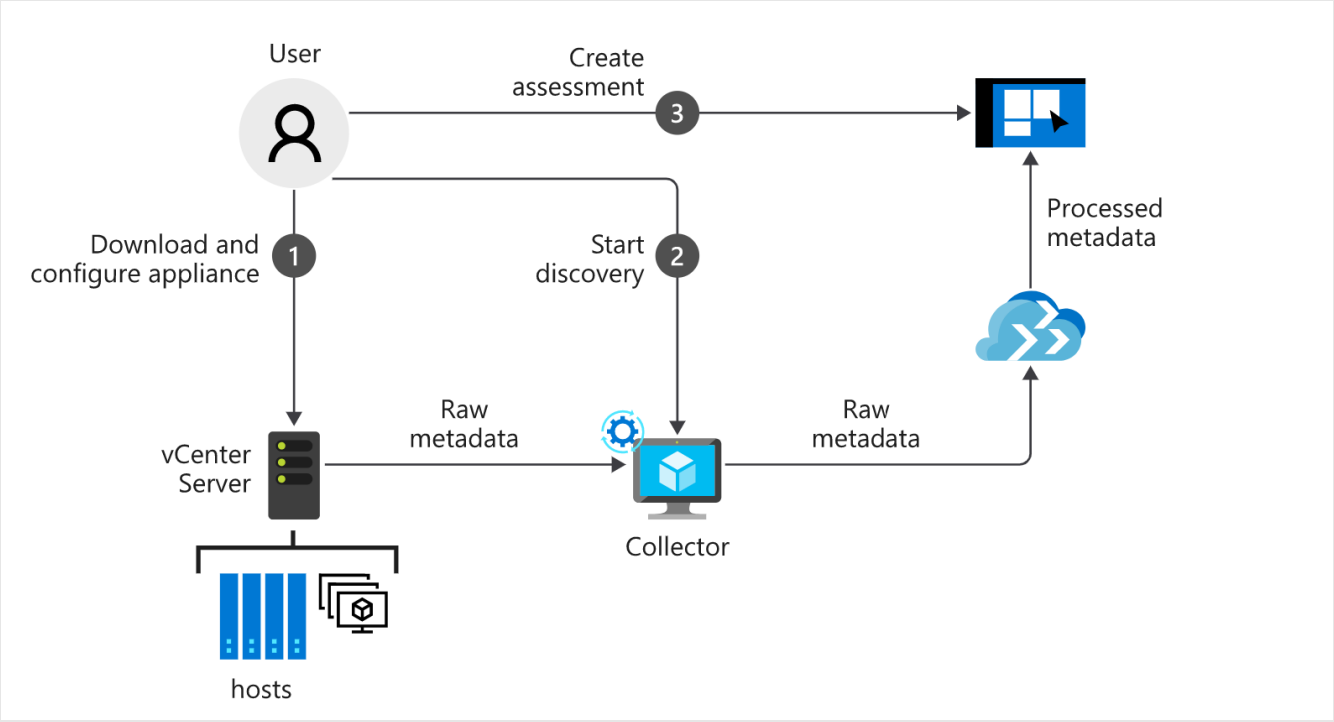
To perform an agentless discovery, the Azure Migrate: Server Assessment tool:

- Guides you through downloading a lightweight collector appliance. The appliance carries out the discovery of systems in your environment.
- Uses data collected by the appliance to identify data about VM cores, memory, disk sizes, and network adapters. Where applicable, the collector also gathers performance data like CPU and memory usage, disk IOPS, disk throughput, and network output.

After the data collection is complete, it's pushed to your Azure Migrate project. On the Azure portal, you can now view all the discovered systems or download a report to review.

The process can be visualized as follows:

1. Download and configure the appliance
2. Start discovery
3. Create an assessment
4. Review the assessment



You'll examine the role of Azure Migrate in more detail during the next unit.

Next unit: Select a migration tool

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Select a migration tool

3 minutes

After you've assessed Tailwind Traders' on-premises workloads, you should begin to consider how to migrate those workloads to Azure. You can use several tools in Azure Migrate to migrate your workloads, depending on your needs.

Describe Azure Migrate

Azure Migrate is a set of features located in a centralized hub that you can use to assess and migrate different workloads to Azure. You can use Azure Migrate to perform the migration of workloads, including apps and VMs. Workloads that can be migrated to Microsoft Azure include on-premises servers, infrastructure, applications, and data.

Azure Migrate components include:

- Unified migration platform: A single portal where you can perform migration to Azure and track the migration status.
- Assessment and migration tools: Azure migration tools consist of multiple assessment and migration tools, including Azure Migrate: Server Assessment and Azure Migrate: Server Migration and other independent software vendor (ISV) tools.
- Assessment and migration of different workloads: There are several different workloads that you can migrate with Azure Migrate hub, including:
  - Servers: On-premises servers are assessed and migrated to Azure VMs.
  - Databases: On-premises databases are assessed and migrated to Azure SQL Database or to an Azure SQL Database–managed instance.
  - Web applications: On-premises web applications are assessed and migrated Azure App Service by using the Azure App Service Migration Assistant.
  - Virtual desktops: On-premises virtual desktop infrastructure (VDI) is assessed and migrated to Azure Virtual Desktop.
  - Data: Large volumes of data are migrated to Azure by using Azure Data Box products.

Azure Migrate hub includes these tools:

Tool	Migration scenario
Azure Migrate: Discovery and assessment: Server Assessment	Discover and assess servers including SQL and web apps.
Azure Migrate: Server Migration	Migrate servers.
Data Migration Assistant	Assess SQL Server databases for migration to Azure SQL Database, Azure SQL Managed Instance, or Azure VMs running SQL Server.
Azure Database Migration Service	Migrate on-premises databases to Azure VMs running SQL Server, Azure SQL Database, or SQL Managed Instances.
Movere	Assess servers.
Web app migration assistant	Assess on-premises web apps and migrate them to Azure.
Azure Data Box	Migrate offline data.

What can you do with Azure Migrate?

Azure Migrate can help with several migration scenarios. The one that you select depends on what you're trying to achieve. The six major migration scenarios are:

- Windows Server workloads
- SQL Server workloads
- Linux workloads
- Windows apps, Java apps and PHP apps
- SAP HANA
- Specialized compute

Migrate web apps to Azure

Azure Migrate uses the Azure App Service Migration Assistant to assess and migrate your web apps. The Azure App Service Migration Assistant enables you to assess and migrate your on-premises Windows ASP.NET web apps to Azure. By using this assistant, you can:

- Determine whether your app is a suitable migration candidate.
- Run readiness checks to perform a general assessment of the app's configuration settings.
- Migrate the app to the Azure App Service.

The Migration Assistant uses an agent that you install locally, and then use to perform a detailed analysis of your apps. You can then use the tool to migrate those apps to Azure. After the initial assessment of your app is complete, you're guided through the migration process using a graphical wizard-driven interface.

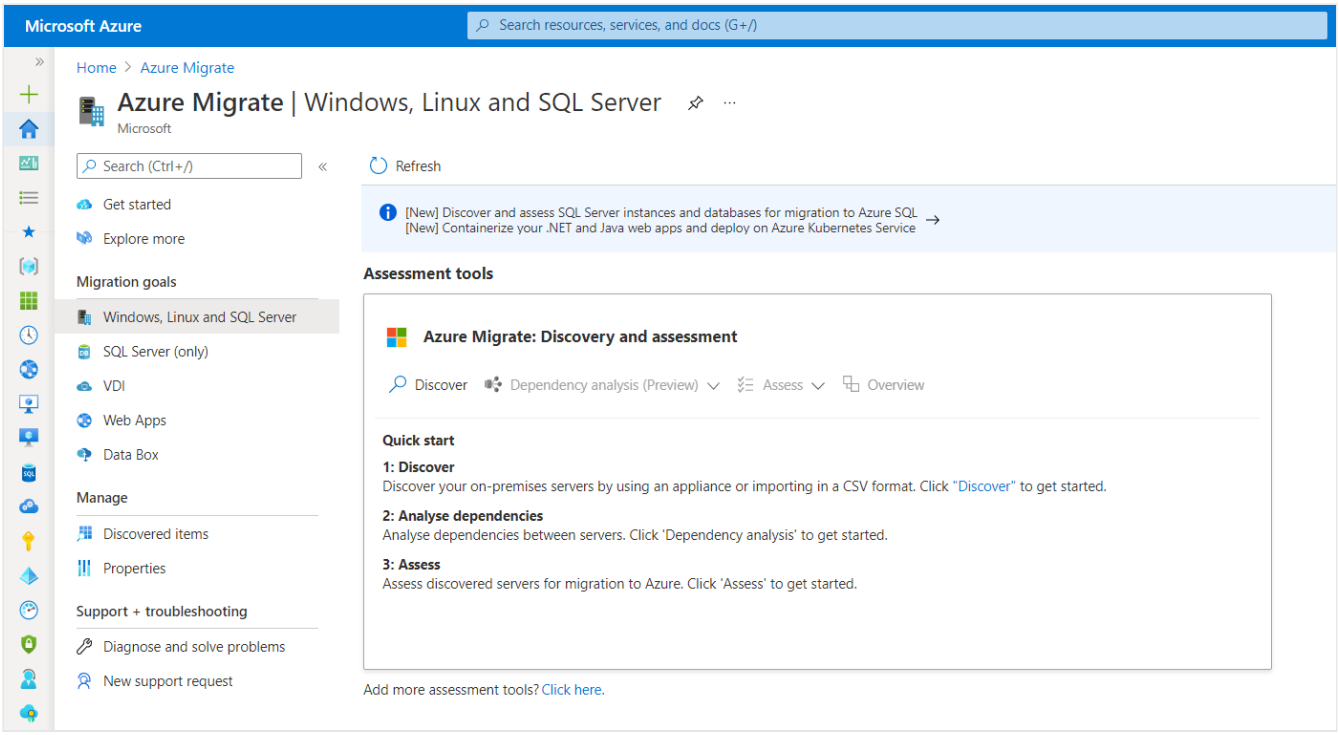
After moving the app to Azure, you may also consider migrating any connected databases.

> **ⓘ Important**
>
> The Migration Assistant migrates your web application and its associated configurations, but does not migrate any backend databases connected to the app. You can use the SQL Migration Tool to complete the migration of your database.

Migrate VMs with Azure Migrate

After you've selected the appropriate server workloads, you're ready to begin the migration. There are four main technical implementation steps involved in moving a server workload to an Azure VM workload using Azure Migrate. These are:

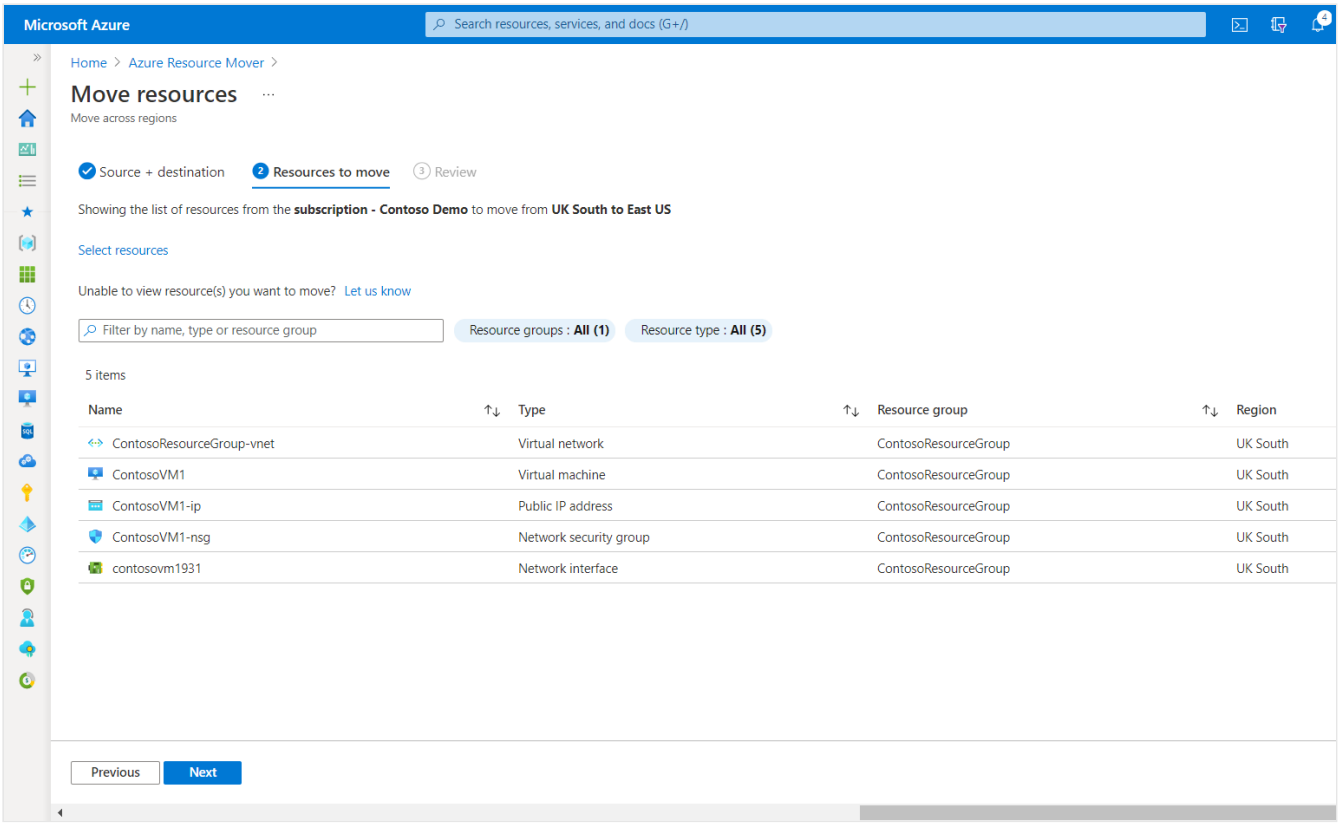
1. Prepare Azure for the Azure Migrate: Server Migration tool.
2. Prepare the on-premises VMs for migration.
3. Replicate the on-premises VMs.
4. Migrate the VMs.



Describe Azure Resource Mover

Azure Resource Mover is a tool that helps move your Azure resources between subscriptions, resource groups, and regions. The tool can be used:

- Before you migrate, to organize your resources.
- After you migrate, to optimize your resource organization.



Resource Mover provides:

- A single location for moving resources.
- Simplicity and speed in moving resources.
- A consistent interface and procedure for moving different types of Azure resources.
- A way to identify dependencies across resources that you want to move.
- Automatic clean-up of resources in the source region.
- The ability to test a move operation before you commit it.

Consider using Azure Resource Mover when, after migration to Azure, and you need to move any of your recently migrated resources across subscriptions, regions or resource groups.

Next unit: Migrate your databases

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Migrate your databases

3 minutes

Most applications use a database to store the data used by an application. It's important that you know how to migrate databases to Azure to properly support Tailwind Traders' move to the cloud.

What is the Azure Database Migration Service?

The Azure Database Migration Service is part of Azure Migrate. You can use the Database Migration Service to migrate your on-premises databases. This includes:

- Azure VMs running SQL Server
- Azure SQL Database (Database Migration Assistant)
- SQL Managed Instances
- Cosmos DB
- Azure DB for MySQL
- Azure DB for PostgreSQL

The Database Migration Service is a fully managed service. The migration service provides two different ways to migrate SQL Server databases:

- Online migration: An online migration uses a continuous synchronization of live data, allowing a cut over to the Azure replica database at any time. Online migration minimizes downtime.
- Offline migration: An offline migration requires shutting down the server at the start of the migration, which means downtime for the service.

Overview of database migration

When you begin the migration process, it's the Data Migration Assistant that guides you through the process. This process consists of three main elements:

1. Assess the databases you want to migrate
2. Migrate the schema: Separate the schema from the databases, and then recreate the schema in the target Azure SQL Database instances
3. Copy the databases data to the target instances and then verify the migrated databases

Prerequisites

Both online and offline migrations have the same prerequisite tasks:

- Download the Data Migration Assistant.
- Create an Azure Virtual Network instance.
- Configure the network security group.
- Configure the Windows Firewall.
- Configure credentials.
- Provision your target database in Azure.

Size the target database appropriately for the migrated workload.

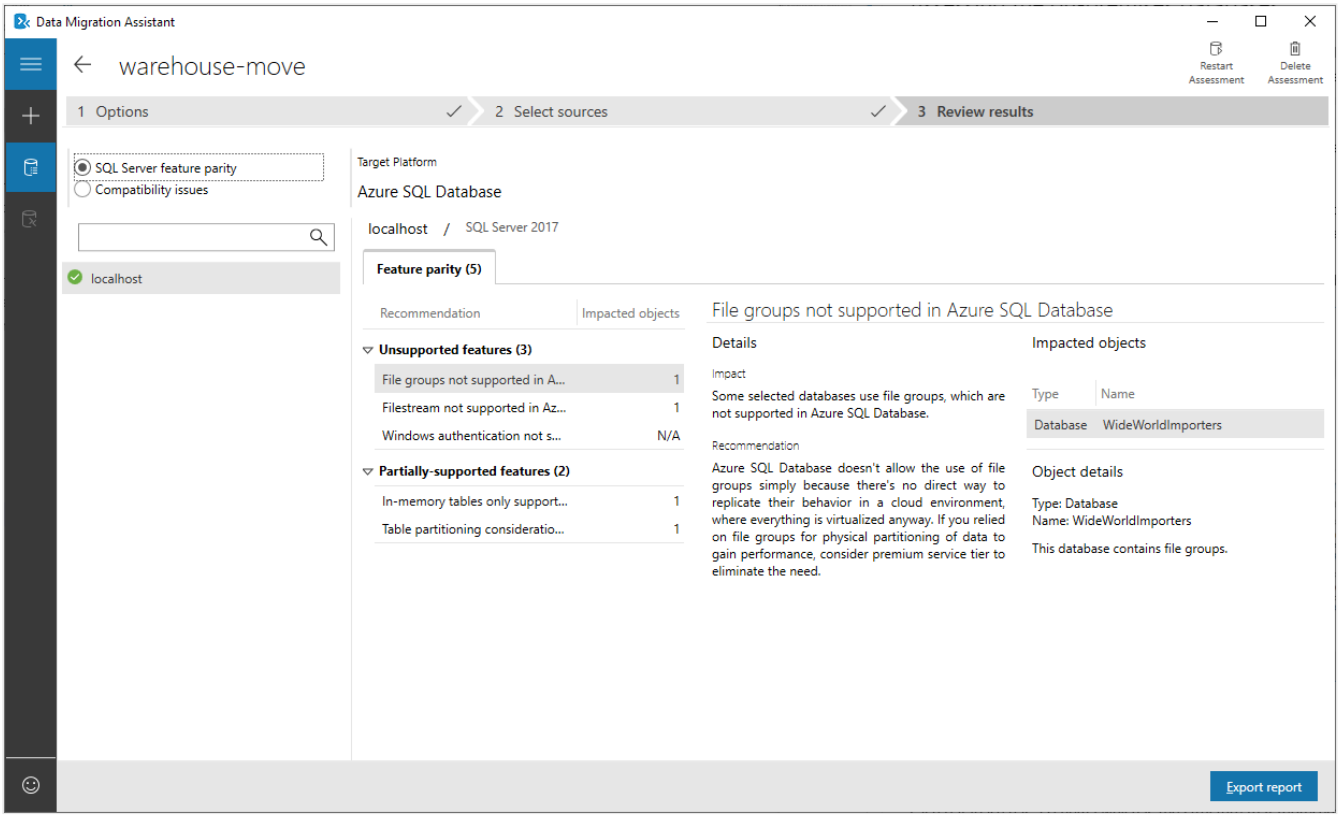
1. Assess the on-premises databases

After you've verified all the prerequisites are met, you're ready to begin the migration. This starts with the assessment of your on-premises environment.

You'll use the Data Migration Assistant to conduct the assessment. The assessment generates a report after completion, including a set of recommendations and alternative approaches that could be taken for the migration.

You'll be able to review any compatibility issues between the source and destination databases that could cause the migration to fail. Address the issues in the report, running it as many times as you need to make sure that the issues have been fixed.

The following screenshot displays a typical Data Migration Assistant report.



2. Migrate the schema by using the Data Migration Assistant

Each database has a schema that represents its entire structure. The schema defines the rules for how the data is organized and the relationships between data elements.

You migrate the schema before you migrate all the data in the database. Doing so:

- Creates an empty structure on the new Azure SQL database. This structure matches that of the on-premises source database.
- Validates connectivity before you do the full data migration.The Data Migration Assistant creates and runs a script to take the required actions.

When the script is complete, check the target server to make sure the database has been configured correctly.

3. Migrate your data with Database Migration Service

After you've conducted your assessment, and created the schema, you can migrate the data.

When these steps are complete, your schema and data have been migrated to the Azure SQL Database instance. You can then shut down and decommission your on-premises databases and servers.

---

100 XP

Select an online storage migration tool

3 minutes

In addition to migrating apps, VMs, and databases, it's often necessary to migrate unstructured data. This data might be stored in several locations and be liable to frequent changes. Therefore, migration of storage containing unstructured data can be challenging.

When considering how to migrate online on-premises unstructured data, consider the following options:

- The Windows Server Storage Migration Service
- Azure File Sync

The Windows Server Storage Migration Service is part of Windows Admin Center.

Overview of the Windows Storage Migration Service

Consider using the Storage Migration Service if you have one or more servers that you want to migrate to newer hardware or VMs. By using the Storage Migration Service, you can:

- Conduct an inventory of your servers and their data.
- Rapidly transfer files, file shares, and security configuration from the source servers.
- Take over the identity of the source servers (known as cutting over). This means that users and apps don't have to change anything to access existing data.
- Manage one or multiple migrations from the Windows Admin Center interface.

Migrate data with the Storage Migration Service

Migration consists of the following three steps:

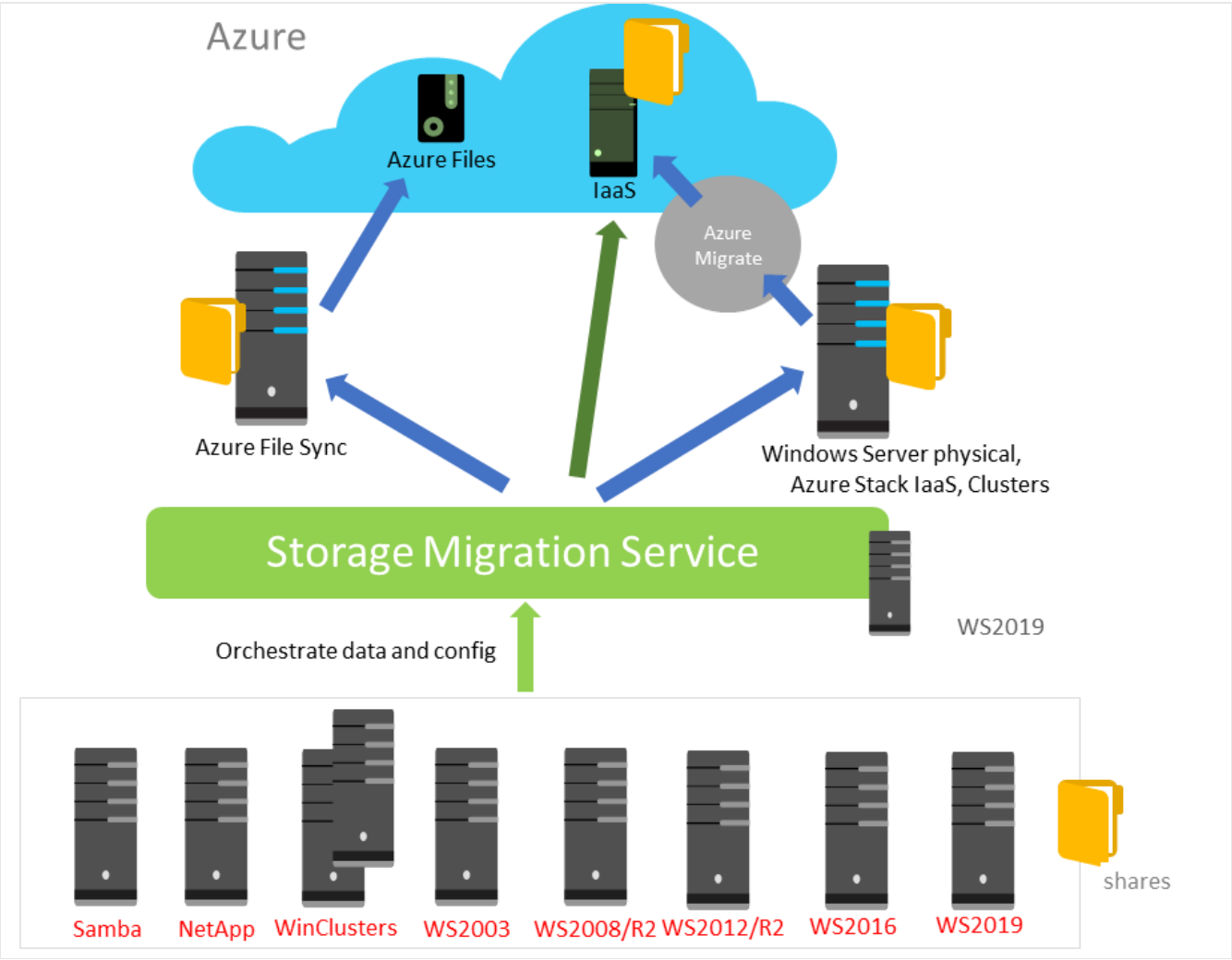
1. Inventory servers to gather information about their files and configuration.
2. Transfer data from the source to the destination servers.
3. Optionally, cut over to the new servers.

> **Important**
>
> The destination servers assume the source servers' former identities so that apps and users don't have to change anything.

After migration, the source servers enter a maintenance state. While in this state, the source servers still contain their original files, but are unavailable to users and apps.

> **Tip**
>
> Don't remove files from the source servers until you're ready to completely decommission the servers at your convenience.

As displayed in the following graphic, you can use the Storage Migration Service to migrate data stored in on-premises file servers via Azure File Sync and Azure Migrate to Azure Files and Azure hosted VMs.



Requirements

To use Storage Migration Service, you require:

- A source server, or failover cluster, containing the data you want to migrate.
- A destination server, or failover cluster, to which you want to migrate the data.
- An Orchestrator server to manage the migration.
- A PC or server running Windows Admin Center to run the Storage Migration Service user interface.

> **Note**
>
> There are additional requirements in terms of security, the Storage Migration Service proxy service, and required firewall port settings.

Use Azure File Sync

Azure File Sync is a feature of Azure Files. Azure Files is an Azure service that provides the functionality of an on-premises file share with the benefits of a platform as a service (PaaS) cloud service. You can use Azure Files in several common scenarios as described in the following table.

Usage	Description
Replace or supplement on-premises file servers	Virtually all companies use file servers. Azure Files can completely replace or supplement traditional on-premises file servers or Network Attached Storage (NAS) devices. With Azure file shares and AD DS authentication, you can migrate data to Azure Files and utilize high availability and scalability while minimizing client changes.
Lift and shift (rehome)	Azure Files makes it easy to lift-and-shift applications that expect a file share to store application or user data to the cloud.
Backup and disaster recovery	You can use Azure file shares as storage for backups, or for disaster recovery to improve business continuity. You can use Azure file shares to back up your data from existing file servers while preserving configured Windows discretionary access control lists (ACLs). Data that's stored on Azure file shares isn't affected by disasters that might affect on-premises locations.

Usage	Description
Azure File Sync	With Azure File Sync, Azure file shares can replicate to Windows Server, either on-premises or in the cloud, for performance and distributed caching of data where it's being used.

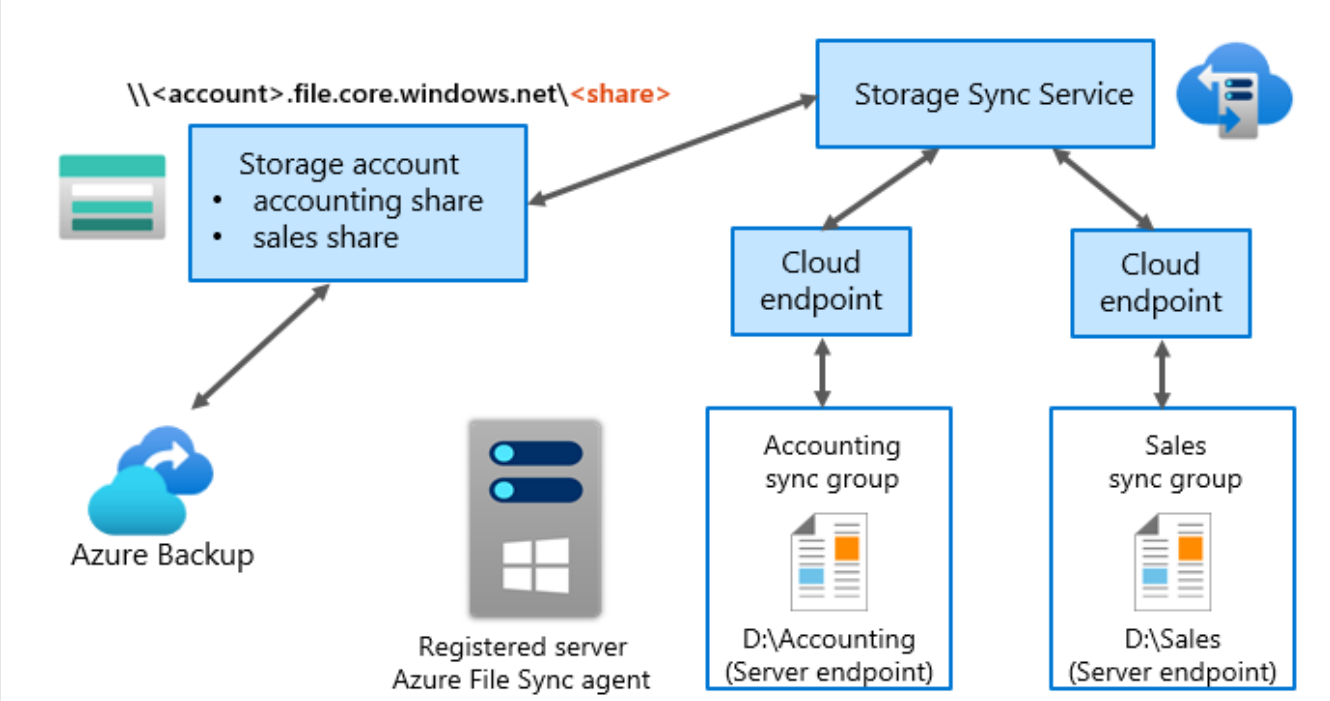
What is Azure File Sync?

Azure File Sync enables you to centralize your organization's file shares in Azure Files, while keeping the flexibility, performance, and compatibility of an on-premises file server. You can also use Azure File Sync to cache Azure file shares on Windows Server computers for fast access close to where the data is accessed. You can use any protocol that's available on Windows Server to access your data locally, including SMB, NFS, and FTPS.

In addition to using Azure Disks as back-end storage, you can utilize both Azure Files and a file server that's hosted in Azure VMs by installing Azure File Sync on a file server that's hosted on a cloud VM. If the Azure file share is in the same region as your file server, you can enable cloud tiering and set the volume of free space percentage to maximum (99%). This ensures minimal duplication of data. You also can use any applications you want with your file servers, such as applications that require NFS protocol support.

Azure File Sync terminology

If you want to understand how File Sync works, you must understand the terms that relate to it. The following diagram uses this terminology to depict how Azure File Sync works.



- The server running Windows Server in this diagram has the Azure File Sync agent and is registered with Azure File Sync.
- Next to this server are two sync groups: Accounting and Sales.
- The Accounting sync group has D:\Accounting as the server endpoint and the Sales sync group has D:\Sales as the server endpoint.
- Each sync group has a two-way interaction with the cloud endpoint, which means that the server endpoint syncs its content with the cloud endpoint content (the Azure file share is the cloud endpoint).
- Both cloud endpoints have a two-way interaction with the same Storage Sync Service.
- Azure File Sync uses the Storage Sync Service.
- Storage Sync Service has a two-way interaction with the Azure storage account, which symbolizes that the cloud endpoints (Azure file shares) are created in the Azure storage account.
- The storage account has two-way interaction with Azure Backup, which means the Azure storage account can be backed up by using Backup.

After you've configured Azure File Sync, data on the configured on-premises server endpoints is synchronized to Azure Files.

Consider using Azure File Sync when you want to migrate shared folder content to Azure. This is especially useful as a means for replacing the Distributed File System on your Windows Servers in your on-premises datacenters.

Next unit: Migrate offline data

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Migrate offline data

3 minutes

There are two choices for migrating offline data to Azure. We'll first cover Azure Import/Export. Then we'll cover Azure Data Box. A short comparison table is provided at the end.

Overview of Azure Import/Export

Azure Import/Export is an Azure service that's used to migrate large quantities of data between an on-premises location and an Azure Storage account. By using the service, you send and receive physical disks that contain your data between your on-premises location and an Azure datacenter. You ship data that's stored on your own disk drives. These disk drives can be Serial ATA (SATA) hard-disk drives (HDDs) or solid-state drives (SSDs).

When to use Azure Import/Export

The Azure Import/Export service is ideally suited to situations where you must upload or download large amounts of data, but your network backbone doesn't have sufficient capacity or reliability to support large-scale transfers. You typically use this service to:

- Migrate large amounts of data from on-premises to Azure, as a one-time task
- Back up your data on-premises in Azure Storage
- Recover large amounts of data that you previously stored in Azure Storage
- Distribute data from Azure Storage to customer sites

How Azure Import/Export works

To use Azure Import/Export, you create a job that specifies the data that you want to import or export. You then prepare the disks to use to transfer the data. For an import job, you write your data to these disks and ship them to an Azure datacenter. Microsoft uploads the data for you. For an export job, you prepare a set of blank disks and ship them to an Azure datacenter. Microsoft copies the data to these disks and ships them back to you. Here are a few other things to know.

- You can use the Import/Export service to export data from Azure Blob storage only. You can't export data that's stored in Azure Files.
- BitLocker must be enabled on the Windows system.
- You'll need an active shipping carrier account like FedEx or DHL for shipping drives to an Azure datacenter.
- If you're exporting, you'll need a set of disks that you can send to an Azure datacenter. The data center will use these disks to copy the data from Azure Storage.

Overview of Azure Data Box

Azure Data Box provides a quick, reliable, and inexpensive method for moving large volumes of data to Azure. By using Microsoft Azure Data Box, you can send terabytes of data into and out of Azure. The solution is based on a secure storage device which is shipped to your organization. Your Data Box might include disks, ruggedized server chassis, or mobile disk.

There are several products to fit different scenarios: Data Box, Data Box Disk, and Data Box Heavy .The configuration process is basically the same across all the products. After you receive your storage device, you can quickly set it up using the local web-based management interface. Assuming you're exporting data to Azure, copy the required data to the storage device, and then return it to Azure.

> ⓘ **Note**
>
> You will define your encryption keys for the storage device. The entire process is tracked end-to-end by the Data Box service in the Azure portal.

When should you use Data Box?

Data Box is ideally suited to transfer data sizes larger than 40 TBs. It's especially useful in scenarios with limited internet connectivity. You could consider using Data Box in the following situations.

Scenario	Explanation
One time migration	When you want to migrate a large amount of on-premises data to Azure. For example: moving a media library from offline tapes into Azure to create an online media library; migrating your VM farm, SQL server, and applications to Azure; moving historical data to Azure for in-depth analysis and reporting using HDInsight.

Scenario	Explanation
Initial bulk transfer	You perform an initial bulk transfer with Data Box and follow it with incremental transfers over the network. For example: you move large volumes of historical backup to Azure. After this data is added, you continue to maintain the archive with incremental data is transfers via network to Azure storage.
Periodic uploads	When you have large volumes of data which is generated periodically. You want to move this data to Azure. For example, data generated by sensors from customer connected IoT devices.

What are the Data Box components?

The Data Box includes the following components:

Component	Description
Data Box device	A physical device that provides primary storage, manages communication with cloud storage, and helps to ensure the security and confidentiality of all data stored on the device. The Data Box device has a usable storage capacity of 80 TB.
Data Box service	An extension of the Azure portal that lets you manage a Data Box device using a web interface that you can access from different geographical locations. Use the Data Box service to perform daily administration of your Data Box device. The service tasks include how to create and manage orders, view, and manage alerts, and manage shares.
Data Box local web-based user interface	A web-based UI that is used to configure the device so that it can connect to the local network, and then register the device with the Data Box service. Use the local web UI also to shut down and restart the Data Box device, view copy logs, and contact Microsoft Support to file a service request.

How to select between Azure Import/Export and Azure Data Box

Capacity	Azure Import/Export	Azure Data Box
Form factor	Internal SATA HDDs or SDDs	Secure, tamper-proof, single hardware appliance

Capacity	Azure Import/Export	Azure Data Box
Microsoft manages shipping logistics	No	Yes
Integrates with partner products	No	Yes
Custom appliance	No	Yes

Tip

If you're looking to import or export more moderate volumes of data to and from Azure Blob storage, consider using other tools like AzCopy or Azure Storage Explorer.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Tailwind Traders is assessing their migration to Azure. They have asked you to make recommendations and answer questions based on these requirements.

- **Large data transfers.** The company needs to transfer large amounts of data between on-premises storage and an Azure Storage account. They don't want to tie up network bandwidth. It's also desired that Microsoft handle the shipping and logistics.
- **Migrate to new hardware.** The company needs to conduct an inventory of their servers and data. Then they need to transfer files and security configuration from the source servers to new hardware.
- **Estimate migration costs.** The CTO is very interested in monthly running costs in Azure. The CTO would like to compare estimated Azure cost to on-premises costs.
- **Migrate web apps to Azure.** The IT department has several test and production web apps they would like to migrate to Azure. These are Windows ASP.NET web apps.
- **Migrate on-premises SQL Server**. The corporate data center has several virtual machines running SQL Server. These machines provide database backend support for the online sales portal. The IT department would like to move this workload to Azure.

1. What product would be best for the large data transfer requirement?

- ○ Azure Import/Export service
- ◉ Azure Data Box

  ✔ **Correct. For Azure Data Box, Microsoft provides the hardware appliance and handles the shipping.**
- ○ Azure Migrate

2. What product would be best for the migrating to new hardware requirements?

- ○ Azure Storage Migration service

  ✔ **Correct. The Storage Migration service is used if you've one or more servers that you want to migrate to newer hardware or virtual machines. The product provides an inventory and then can rapidly transfer files.**
- ◉ Azure Migrate

  ✘ **Incorrect. Azure Migrate is an assessment and migration tool.**
- ○ Azure File Sync

3. What product would be best to estimate monthly costs before migrating?

- ○ Microsoft Cost Management
- ◉ Azure Advisor

✗ **Incorrect. Azure Advisor helps optimize your Azure resources for high availability, performance, and cost.**

- ○ Azure TCO Calculator

✓ **Correct. The Azure TCO Calculator provides an estimate of monthly running costs in Azure, which enables a comparison with on-premises costs.**

4. What product would be best to migrate the web apps to Azure?

- ○ Azure App Service Migration Assistant

✓ **Correct. The Azure App Service Migration Assistant will help assess and migrate your web apps.**

- ○ Azure Resource Mover
- ◉ Data Migration Assistant

✗ **Incorrect. The Data Migration Assistant helps detects compatibility issues that can impact database functionality in a new version of SQL Server or Azure SQL Database.**

5. What product would be best to migrate the on-premises SQL Server databases?

- ○ Azure Resource Mover
- ◉ Data Migration Assistant

✓ **Correct. The Data Migration Assistant helps detects compatibility issues that can impact database functionality in a new version of SQL Server or Azure SQL Database.**

- ○ Azure Storage Migration service

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆