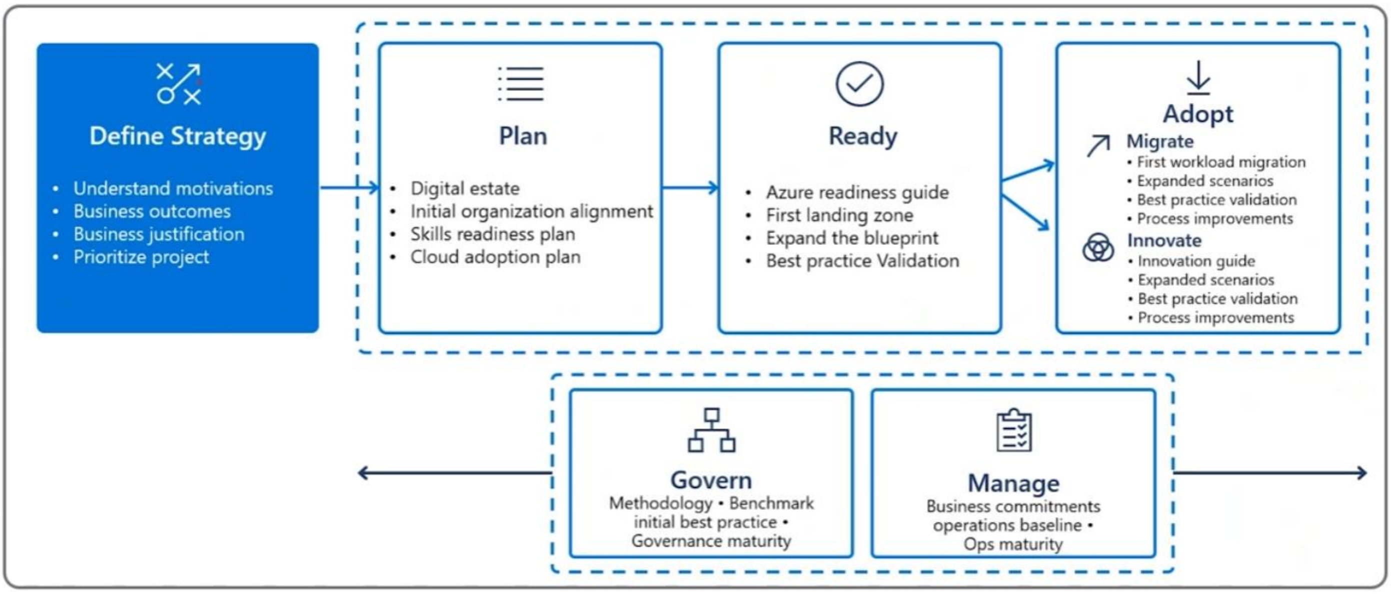
Summary

Review the Cloud Adoption Framework

The Microsoft Cloud Adoption Framework for Azure is a full lifecycle framework that enables cloud architects, IT professionals, and business decision makers to achieve their cloud adoption goals.

It provides best practices, documentation, and tools that help you create and implement business and technology strategies for the cloud.

Following best practices for the Cloud Adoption Framework allows your organization to better align business and technical strategies and ensure success.



Define Strategy
- Understand motivations
- Business outcomes
- Business justification
- Prioritize project

Plan
- Digital estate
- Initial organization alignment
- Skills readiness plan
- Cloud adoption plan

Ready
- Azure readiness guide
- First landing zone
- Expand the blueprint
- Best practice Validation

Adopt

Migrate
- First workload migration
- Expanded scenarios
- Best practice validation
- Process improvements

Innovate
- Innovation guide
- Expanded scenarios
- Best practice validation
- Process improvements

Govern

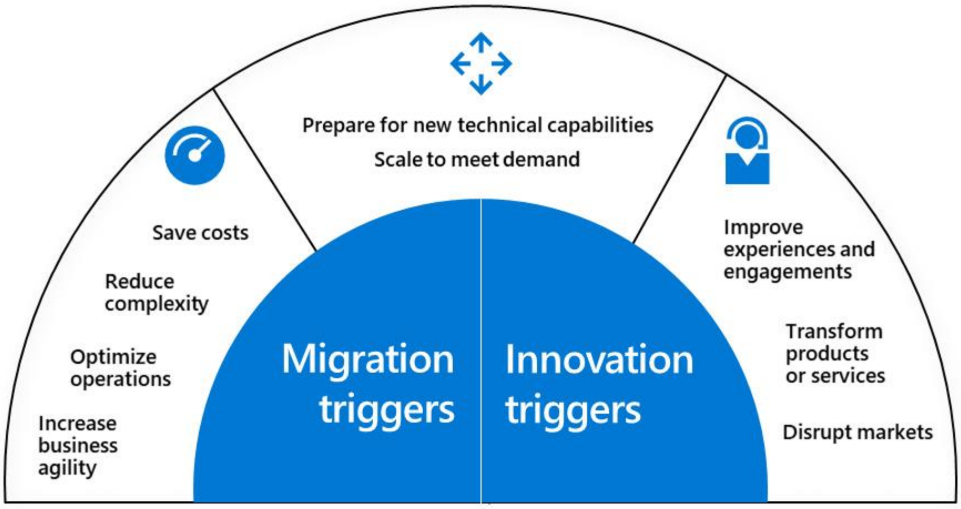
Methodology • Benchmark initial best practice • Governance maturity

Manage

Business commitments operations baseline • Ops maturity

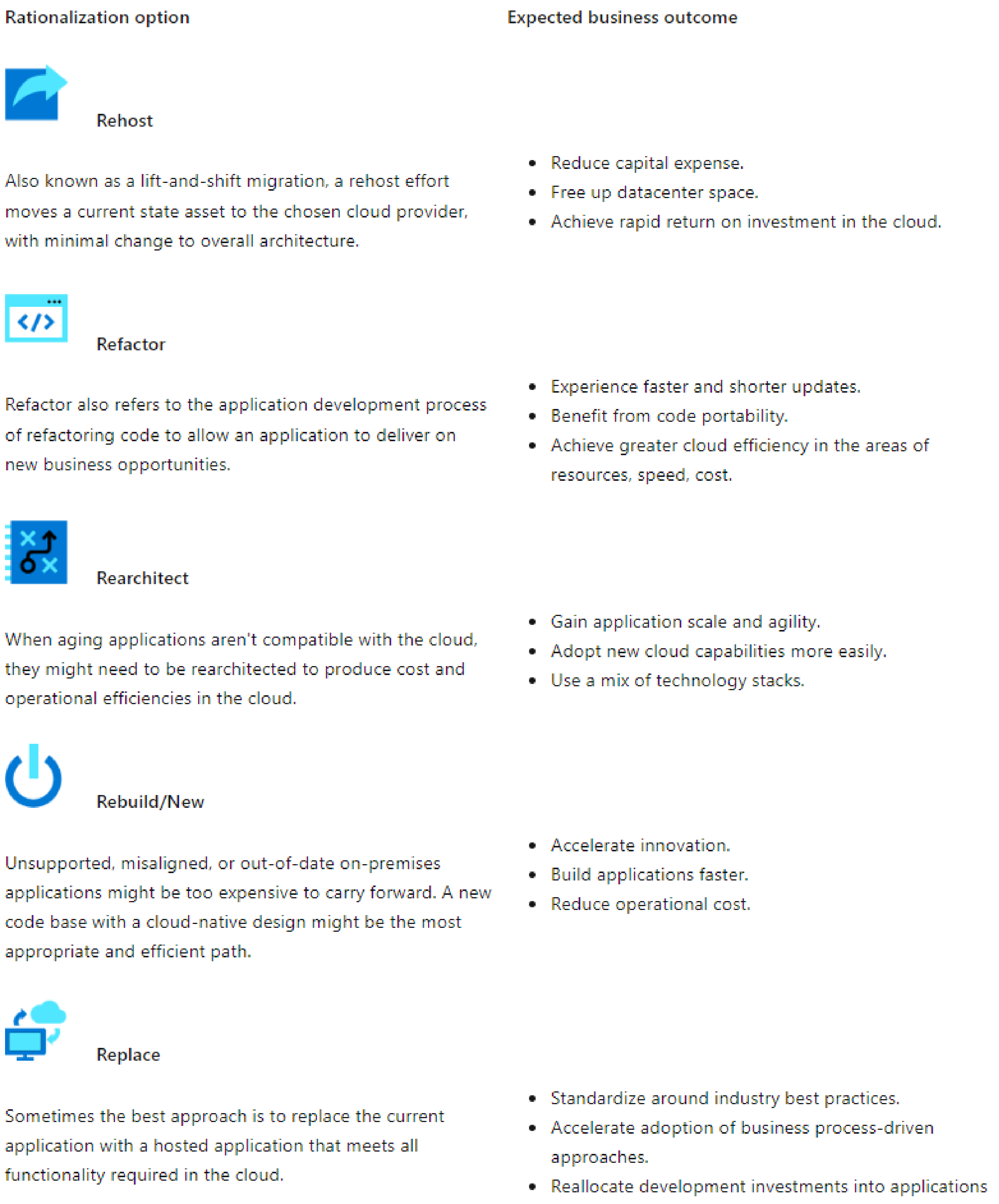
Define Strategy

- **Define and document your motivations**: Meet with key stakeholders and executives to document the motivations behind cloud adoption.
- **Document business outcomes**: Engage motivated stakeholders and executives to document specific business outcomes.
- **Evaluate financial considerations**: Learn how to use the cloud to make your IT cost structure more flexible. Then, build a business case to adopt the cloud.
- **Understand technical considerations**: Discover the technical flexibility, efficiencies, and capabilities that help you build a business case to adopt the cloud.


Prepare for new technical capabilities
Scale to meet demand
Save costs
Reduce complexity
Optimize operations
Increase business agility
Migration triggers
Innovation triggers
Improve experiences and engagements
Transform products or services
Disrupt markets

Plan

- **Digital estate**: Inventory and rationalize your digital estate based on assumptions that align your organization's motivations and business outcomes.
- **Initial organizational alignment**: Establish a plan for initial organizational alignment to support the adoption plan.
- **Skills readiness plan**: Create a plan for addressing skills readiness gaps within your organization.
- **Cloud adoption plan**: Develop a cloud adoption plan to manage change across skills, the digital estate, and your organization.



Rationalization option	Expected business outcome
Rehost Also known as a lift-and-shift migration, a rehost effort moves a current state asset to the chosen cloud provider, with minimal change to overall architecture.	• Reduce capital expense. • Free up datacenter space. • Achieve rapid return on investment in the cloud.
Refactor Refactor also refers to the application development process of refactoring code to allow an application to deliver on new business opportunities.	• Experience faster and shorter updates. • Benefit from code portability. • Achieve greater cloud efficiency in the areas of resources, speed, cost.
Rearchitect When aging applications aren't compatible with the cloud, they might need to be rearchitected to produce cost and operational efficiencies in the cloud.	• Gain application scale and agility. • Adopt new cloud capabilities more easily. • Use a mix of technology stacks.
Rebuild/New Unsupported, misaligned, or out-of-date on-premises applications might be too expensive to carry forward. A new code base with a cloud-native design might be the most appropriate and efficient path.	• Accelerate innovation. • Build applications faster. • Reduce operational cost.
Replace Sometimes the best approach is to replace the current application with a hosted application that meets all functionality required in the cloud.	• Standardize around industry best practices. • Accelerate adoption of business process-driven approaches. • Reallocate development investments into applications

Ready

- **Define skills and support readiness**: Create and implement a skills-readiness plan to:
    - Address current gaps.
    - Ensure that IT and business people are ready for the change and the new technologies.
    - Define support needs.
- **Create your landing zone**: Set up a migration target in the cloud to handle prioritized applications.
    - Before you begin to build and deploy solutions with Azure services, make sure your environment is ready.
    - The term landing zone is used to describe an environment that's provisioned and prepared to host workloads in a cloud environment, such as Azure.

Migrate

- **Migration preparation**: Establish a rough migration backlog, based largely on the current state and desired outcomes.
    - Business outcomes: The key business objectives that drive this migration. They're defined in the Plan phase.
    - Digital estate estimate: A rough estimate of the number and condition of workloads to be migrated. It's defined in the Plan phase.
    - Roles and responsibilities: A clear definition of the team structure, separation of responsibilities, and access requirements. They're defined in the Ready phase.
    - Change management requirements: The cadence, processes, and documentation required to review and approve changes. They're defined in the Ready phase.
- **Migrate your first workload**: Use the Azure migration guide to become familiar with the Azure native tools and approach to migration.
- **Migration scenarios**: Use other migration tools and approaches to act on other migration scenarios.
- **Best practices**: Address common migration needs through the application of consistent best practices.
- **Process improvements**: Migration is a process heavy activity. As migration efforts scale, use these process improvements to evaluate and mature various aspects of migration.

Innovate

- **Create hypothesis with business value consensus**: Before you decide on technical solutions, identify how new innovations can drive business value and come up with a hypothesis about customer needs.
- **Build your first MVP**: Once you have a hypothesis with enough value potential to integrate it into your application, start the build process. Development sprints should be as quick as possible. Quick sprints let teams quickly verify or reject a hypothesis, or fine tune how required functionality integrates with the application.
- **Measure & Learn from your MVP**: You want to verify the accuracy of your hypothesis as soon as possible. A minimum viable product (MVP) is a preliminary version of the new feature that offers enough functionality to gather feedback and confirm if you're moving in the right direction.
- **Expand digital innovation**: To refine your hypothesis using the innovation disciplines or the digital inventions that include:
    - Democratize data
    - Engage via applications
    - Empower adoption
    - Interact with devices
    - Predict and influence

Secure

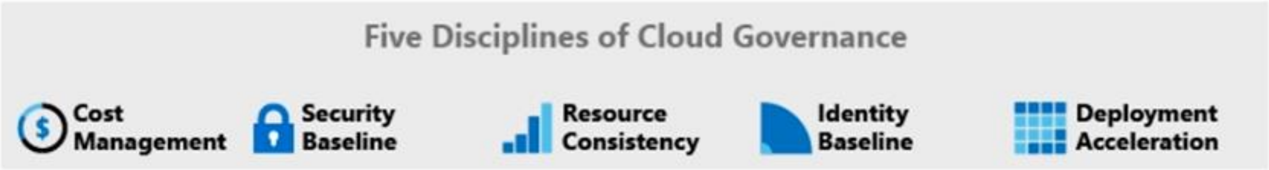
- Security is a standalone organizational discipline and an attribute that's integrated or overlaid on other disciplines.
- **Zero Trust**: Microsoft believes all security disciplines should follow the Zero Trust principles, which are assume breach, verify explicitly, and use least privilege access. These principles underpin any sound security strategy and must be balanced with business enablement goals. The first and most visible part of Zero Trust is in access control, so it's highlighted in the description of access control security discipline.

Manage

- **Define business commitments**: Document supported workloads to establish operational commitments with the business and agree on cloud management investments for each workload.
- **Establish a management baseline**: Define the criticality classifications, cloud management tools, and processes required to deliver your minimum commitment to operations management.
- **Expand the management baseline**: Based on business commitments and operations decisions, make use of the included best practices to implement the required cloud management tooling.
- **Advanced operations and design principles**: Platforms or workloads that require a higher level of business commitment might require a deeper architecture review to deliver on resiliency and reliability commitments.

Govern

- **Establish your methodology**: Establish a basic understanding of the methodology that drives cloud governance in the Cloud Adoption Framework to begin thinking through the end state solution.
- **Use the governance benchmark tool**: Assess your current state and future state to establish a vision for applying the framework.
- **Establish an initial governance foundation**: Begin your governance journey with a small, easily implemented set of governance tools. This initial governance foundation is called a minimum viable product (MVP).
- **Improve your initial governance foundation**: Throughout implementation of the cloud adoption plan, iteratively add governance controls to address tangible risks as you progress toward the end state.



Organize

- **Structure type**: Define the type of organizational structure that best fits your operating model.
- **Cloud functions**: Understand the cloud functionality required to adopt and operate the cloud.
- **Mature team structures**: Define the teams that can provide various cloud functions.
- **RACI matrix**: Use the provided RACI matrix to map roles to each team for functions of the cloud operating model. This matrix includes responsibility, accountability, consulted, and informed roles.

Microsoft Azure Well-Architected Framework

The Azure Well-Architected Framework is a set of guiding tenets that you can use to improve the quality of a workload. The framework consists of five pillars of architectural excellence:

Pillar	Description
Reliability	The ability of a system to recover from failures and continue to function.
Security	Protecting applications and data from threats.
Cost optimization	Managing costs to maximize the value delivered.
Operational excellence	Operations processes that keep a system running in production.
Performance efficiency	The ability of a system to adapt to changes in load.

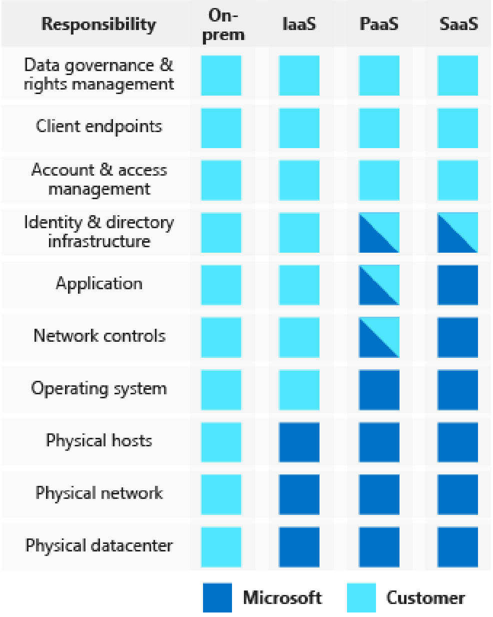
Reliability

- A reliable workload is both resilient and available.
- Resiliency is the ability of the system to recover from failures and continue to function.
- The goal of resiliency is to return the application to a fully functioning state after a failure occurs.
- Availability is whether your users can access your workload when they need to.
- Cost implications are inevitable when introducing greater reliability and high availability. This trade-off should be carefully considered.
- By monitoring the operation of an application relative to a healthy state, you can detect and predict reliability issues. Monitoring allows you to take swift and remedial action.
- One of the leading causes of application downtime is human error due to the deployment of insufficiently tested software or through misconfiguration. To minimize the possibility and consequence of human errors, it's vital to strive for automation in all aspects of a cloud solution.
- Self-healing describes the ability of a system to deal with failures automatically. Handling failures happens through pre-defined remediation protocols. These protocols connect to failure modes within the solution. It's an advanced concept that requires a high level of system maturity with monitoring and automation. From inception, self-healing should be an aspiration to maximize reliability.
- Scale-out is a concept that focuses on the ability of a system to respond to demand through horizontal growth. As traffic grows, more resource units are added in parallel, instead of increasing the size of the existing resources. Through scale units, a system can handle expected and unexpected traffic increases, essential to overall reliability. Scale units further reduce the effects of a single resource failure.
- **Recovery point objective (RPO)**: The maximum duration of acceptable data loss. RPO is measured in units of time, not volume. Examples are "30 minutes of data," "four hours of data," and so on. RPO is about limiting and recovering from data loss, not data theft.
- **Recovery time objective (RTO)**: The maximum duration of acceptable downtime, where your specification defines "downtime". For example, if the acceptable downtime duration is eight hours if there's a disaster, then your RTO is eight hours.

Security

- Think about security throughout the entire lifecycle of an application, from design and implementation to deployment and operations.
- The Azure platform provides protections against various threats, such as network intrusion and DDoS attacks. You still need to build security into your application and into your DevOps processes.
- Identity management: Consider using Azure Active Directory (Azure AD) to authenticate and authorize users. Azure AD is a fully managed identity and access management service.

- Protect your infrastructure: Use Azure role-based access control (Azure RBAC role) to grant users within your organization the correct permissions to Azure resources. Grant access by assigning Azure roles to users or groups at a certain scope.
- Application security: In general, the security best practices for application development still apply in the cloud. Best practices include:
  - Encrypt data in-transit with the latest supported TLS versions
  - Protect against CSRF and XSS attacks
  - Prevent SQL injection attacks
- Cloud applications often use managed services that have access keys. Never check these keys into source control. Consider storing application secrets in Azure Key Vault.
- Data sovereignty: Make sure that your data remains in the correct geopolitical zone when using Azure data services.
- Encryption: Use Key Vault to safeguard cryptographic keys and secrets. By using Key Vault, you can encrypt keys and secrets by using keys that are protected by hardware security modules (HSMs). Many Azure storage and DB services support data encryption at rest.
- Protect from common attacks:
  - **Data layer**: Exposing an encryption key or using weak encryption can leave your data vulnerable if unauthorized access occurs.
  - **Application layer**: Malicious code injection and execution are the hallmarks of application-layer attacks. Common attacks include SQL injection and cross-site scripting (XSS).
  - **VM/compute layer**: Malware is a common method of attacking an environment, which involves executing malicious code to compromise a system. After malware is present on a system, further attacks can occur that lead to credential exposure and lateral movement throughout the environment.
  - **Networking layer**: Taking advantage of unnecessary open ports to the internet is a common method of attack. Open ports might also include leaving the SSH or RDP protocols open to virtual machines. When these protocols are open, they can allow brute-force attacks against your systems as attackers attempt to gain access.
  - **Perimeter layer**: Denial-of-service (DoS) attacks often happen at this layer. These attacks try to overwhelm network resources, forcing them to go offline or making them incapable of responding to legitimate requests.
  - **Policies and access layer**: This layer is where authentication occurs for your application. The exposure of credentials is a risk at this layer, and it's important to limit the permissions of identities. You also want to have monitoring in place to look for possible compromised accounts, such as logins coming from unusual places.
  - **Physical layer**: Unauthorized access to facilities through methods, such as door drafting and theft of security badges, can happen at this layer.



Responsibility	On-prem	IaaS	PaaS	SaaS
Data governance & rights management	□	□	□	□
Client endpoints	□	□	□	□
Account & access management	□	□	□	□
Identity & directory infrastructure	□	□	◩	◩
Application	□	□	◩	■
Network controls	□	□	◩	■
Operating system	□	□	■	■
Physical hosts	□	■	■	■
Physical network	□	■	■	■
Physical datacenter	□	■	■	■

■ Microsoft □ Customer

Cost optimization

- When you design a cloud solution, focus on generating incremental value early.
- Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies.
- Use the pay-as-you-go strategy for your architecture, and invest in scaling out, rather than delivering a large investment-first version.
- Consider opportunity costs in your architecture, and the balance between first mover advantage versus fast follow.
- Use the cost calculators to estimate the initial cost and operation costs.
- Finally, establish policies, budgets, and controls that set cost limits for your solution.

Operational excellence

- Operational excellence covers the operations and processes that keep an application running in production.
- Deployments must be reliable and predictable.
- Automate deployments to reduce the chance of human error.
- Fast and routine deployment processes don't slow down the release of new features or bug fixes.
- A good testing strategy helps you identify issues in your application before it's deployed, and ensure that dependent services can properly communicate with your application.
- Equally important, you must be able to quickly roll back or roll forward if an update has problems.

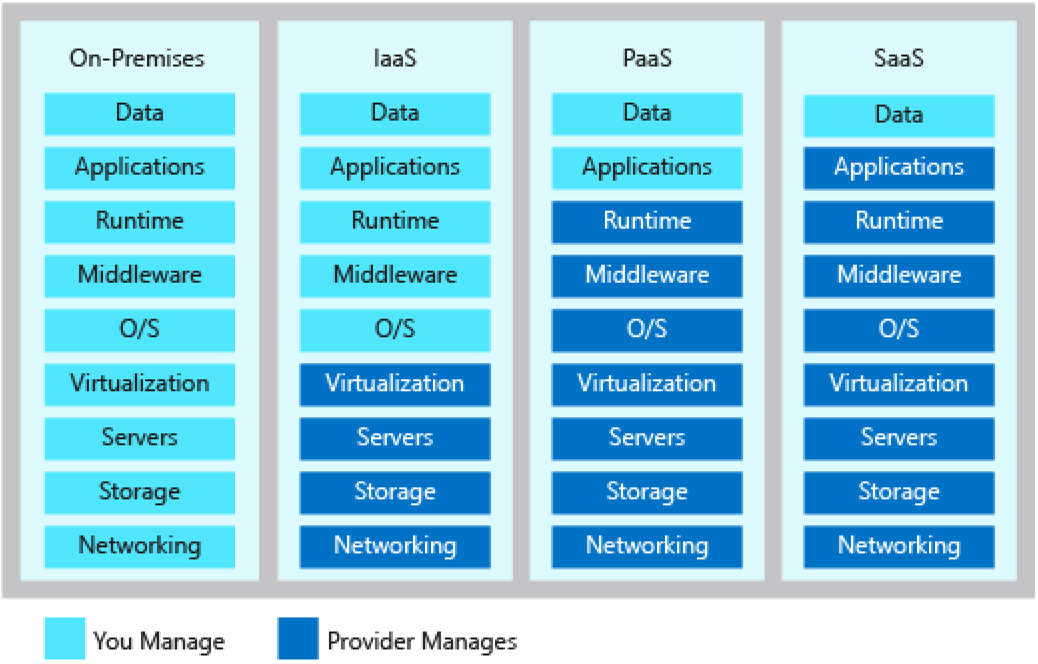
Performance efficiency

- Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner.
- Scaling up is adding more resources to a single instance. Also known as vertical scaling.
- Scaling out is adding more instances. Also known as horizontal scaling.
- The advantage of scaling out is that you can conceivably scale out forever if you have more machines to add to the architecture. Scaling out requires some type of load distribution.
- Autoscaling is the process of dynamically allocating resources to match performance requirements. As the volume of work grows, an application might need more resources to maintain the desired performance levels and satisfy service-level agreements (SLAs). As demand slackens and the added resources are no longer needed, they can be deallocated to minimize costs.
- The main ways to achieve performance efficiency include using scaling appropriately and implementing PaaS offerings that have scaling built in.
- Performance efficiency depends on the system's ability to handle load increases without impacting performance, or for available resources to be readily increased.
- Performance efficiency concerns not just compute instances, but other elements such as data storage, messaging infrastructure, and application architecture.
- When you're optimizing for performance, you look at network and storage performance to ensure that their levels are within acceptable limits.
- Partitioning can help improve scalability, reduce contention, and optimize performance.
- Use caching in your architecture to help improve performance. Caching is a mechanism to store frequently used data or assets (webpages, images) for faster retrieval.
- Best practices:
  - Autoscaling
  - Background jobs
  - Caching
  - CDN
  - Data partitioning

General design principles

- **Enable architectural evolution**: No architecture is static. Allow for the evolution of your architecture by taking advantage of new services, tools, and technologies when they're available.
- **Use data to make decisions**: Collect data, analyze it, and use it to make decisions surrounding your architecture. From cost data, to performance, to user load, using data can guide you to make the right choices in your environment.
- **Educate and enable**: Cloud technology evolves quickly. Educate your development, operations, and business teams to help them make the right decisions and build solutions to solve business problems. Document and share configurations, decisions, and best practices within your organization.
- **Automate**: Automation of manual activities reduces operational costs, minimizes error introduced by manual steps, and provides consistency between environments.

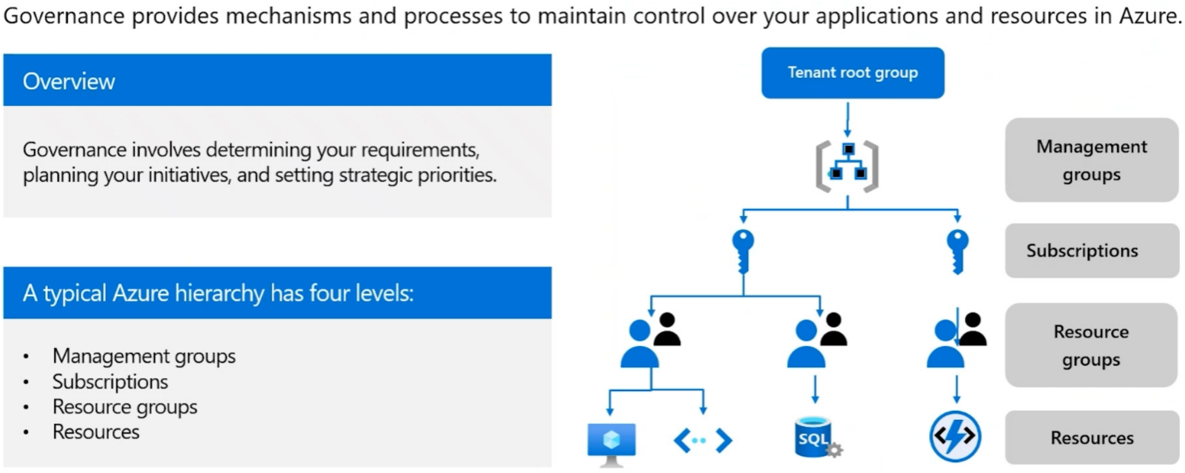
Shared responsibility



On-Premises	IaaS	PaaS	SaaS
◻ Data	◻ Data	◻ Data	◻ Data
◻ Applications	◻ Applications	◻ Applications	■ Applications
◻ Runtime	◻ Runtime	■ Runtime	■ Runtime
◻ Middleware	◻ Middleware	■ Middleware	■ Middleware
◻ O/S	◻ O/S	■ O/S	■ O/S
◻ Virtualization	■ Virtualization	■ Virtualization	■ Virtualization
◻ Servers	■ Servers	■ Servers	■ Servers
◻ Storage	■ Storage	■ Storage	■ Storage
◻ Networking	■ Networking	■ Networking	■ Networking

◻ You Manage ■ Provider Manages

Govern resource in Azure



Governance provides mechanisms and processes to maintain control over your applications and resources in Azure.

Overview

Governance involves determining your requirements, planning your initiatives, and setting strategic priorities.

A typical Azure hierarchy has four levels:

- Management groups
- Subscriptions
- Resource groups
- Resources

Management Groups

Management groups are containers that help you manage access, policy, and compliance across multiple subscriptions. You can use management groups to:

- Limit the regions where virtual machines can be created, across subscriptions.
- Provide user access to multiple subscriptions by creating one role assignment that's inherited by other subscriptions.
- Monitor and audit role and policy assignments, across subscriptions.

Management groups characteristics:

- Management groups can be used to aggregate policy and initiative assignments via Azure Policy.
- A management group tree can support **up to six levels of depth**. This limit doesn't include the tenant root level or the subscription level.
- Azure role-based access control authorization for management group operations isn't enabled by default.
- By default, all new subscriptions are placed under the root management group.

Things to consider when creating management groups:

- **Design management groups with governance in mind**. Use Azure policies at the management group level for all workloads that require the same security, compliance, connectivity, and feature settings.
- **Keep the management group hierarchy reasonably flat**. Plan the hierarchy to have no more than three or four levels of management groups. A hierarchy that's too flat doesn't provide flexibility and complexity for large organizations. A hierarchy with too many levels can be difficult to manage.
- **Consider a top-level management group**. Implement a top-level management group to support common platform policy and Azure role assignments across the entire organization. A management group can be a top-level management group for all organizational-wide policies.
- **Consider an organizational or departmental structure**. Design your management groups based on the organizational structure, to make it easy to understand. Separate the management groups for each department like Sales, Corporate, and IT.
- **Consider a geographical structure.** Build your management groups by using a geographical structure to allow for compliance policies in different regions.
- **Consider a production management group**. Institute a production management group to create policies that apply to all corporate products. A production management group can provide product-specific policies for corporate applications.
- **Consider a sandbox management group**. Offer a sandbox management group for users to experiment with Azure. The sandbox provides isolation from your development, test, and production environments. Users can experiment with resources that might not yet be allowed in official production environments.
- **Consider isolating sensitive information in a separate management group**. Secure sensitive data by using a corporate management group. The separate management group provides both standard and enhanced compliance policies for the main office.

Subscriptions

Teams often start their Azure governance strategy at the subscription level. There are three main aspects to consider when you create and manage subscriptions:

- **Billing:** You can create one billing report per subscription. If you have multiple departments and need to do a "chargeback" of cloud costs, one possible solution is to organize subscriptions by department or by project.
- **Access control:** Every subscription is associated with an Azure Active Directory tenant. Each tenant provides administrators the ability to set granular access through defined roles by using Azure role-based access control. With separate subscriptions, you can control access to each one separately and isolate their resources from one another.
- **Subscription limits:** Subscriptions also have some resource limitations. For example, the maximum number of network Azure ExpressRoute circuits per subscription is 10. Those limits should be considered during your design phase. If you'll need to exceed those limits, you might need to add more subscriptions. If you hit a hard limit maximum, there's no flexibility to increase it.

Things to consider when creating subscriptions:

- **Treat subscriptions as a democratized unit of management**. Align your subscriptions to meet specific business needs and priorities.
- **Group subscriptions together under management groups**. Group together subscriptions that have the same set of policies and Azure role assignments to inherit these settings from the same management group. Both the West and East subscriptions can inherit policy settings from the Sales management group.
- **Consider a dedicated shared services subscription**. Use a shared services subscription to ensure all common network resources are billed together and isolated from other workloads. Examples of shared services subscriptions include Azure ExpressRoute and Virtual WAN.
- **Consider subscription scale limits**. Subscriptions serve as a scale unit for component workloads. Large, specialized workloads like high-performance computing, IoT, and SAP are all better suited to use separate subscriptions. By having separate subscriptions for different groups or tasks, you can avoid resource limits (such as a limit of 50 Azure Data Factory integrations).
- **Consider administrative management**. Subscriptions provide a management boundary, which allows for a clear separation of concerns. Will each subscription need a separate administrator, or can you combine subscriptions? The Corporate management group could have a single subscription for both the HR and Legal departments.
- **Consider how to assign Azure policies**. Both management groups and subscriptions serve as a boundary for the assignment of Azure policies. Workloads like those for the Payment Card Industry (PCI) typically require extra policies to achieve compliance. Rather than using a management group to group workloads that require PCI compliance, you can achieve the same isolation with a subscription. These types of decisions ensure you don't have too many management groups with only a few subscriptions.
- **Consider network topologies**. Virtual networks can't be shared across subscriptions. Resources can connect across subscriptions with different technologies, such as virtual network peering or Virtual Private Networks (VPNs). Consider which workloads must communicate with each other when you decide if a new subscription is required.
- **Consider making subscription owners aware of their roles and responsibilities**. Conduct a quarterly or biannual access review by using Azure AD Privileged Identity Management. Access reviews ensure privileges don't proliferate as users move within the customer organization.

Resource Groups

Resource groups are logical containers into which Azure resources are deployed and managed. These resources can include web apps, databases, and storage accounts.

You can use resource groups to:

- Place resources of similar usage, type, or location in logical groups.
- Organize resources by life cycle so all the resources can be created or deleted at the same time.
- Apply role permissions to a group of resources or give a group access to administer a group of resources.
- Use resource locks to protect individual resources from deletion or change.

Things to know about resource groups

- Resource groups have their own location (region) assigned. This region is where the metadata is stored.
- If the resource group's region is temporarily unavailable, you can't update resources in the resource group because the metadata is unavailable. The resources in other regions still function as expected, but you can't update them.
- Resources in the resource group can be in different regions.
- A resource can connect to resources in other resource groups. You can have a web application that connects to a database in a different resource group.
- Resources can be moved between resource groups with some exceptions.
- You can add a resource to or remove a resource from a resource group at any time.
- Resource groups can't be nested.
- Each resource must be in one, and only one, resource group.
- Resource groups can't be renamed.

Things to consider when creating resource groups

- **Consider group by type**. Group resources by type for on-demand services that aren't associated with an app. For Tailwind Traders, you can have a resource group for the SQL databases (SQL-RG) and a separate resource group (WEB-RG) for the web services.
- **Consider group by app**. Group resources by app when all resources have the same policies and life cycle. This method can also be applied to test or prototype environments. For Tailwind Traders, App1 and App2 can have separate resource groups. Each group can have all the resources for the specific application.
- **Consider group by department, group by location (region), and group by billing (cost center)**. Review other grouping strategies that aren't common but might be useful in your situation.
- **Consider a combination of organizational strategies**. Don't restrict your Tailwind Traders strategy to using only a single resource group option. A combination of options is best.
- **Consider resource life cycle**. Design your resource groups according to life cycle requirements. Do you want to deploy, update, and delete certain resources at the same time? If so, place these resources in the same resource group.
- **Consider administration overhead**. Include overhead planning in your strategy. How many resource groups would you like to manage? Does Tailwind Traders have centralized or decentralized Azure administrators?
- **Consider resource access control**. Implement access control for your resource groups. At the resource group level, you can assign Azure policies, Azure roles, and resource locks. Resource locks prevent unexpected changes to critical resources.
- **Consider compliance requirements**. Plan to build in support for compliance in your Tailwind Traders strategy. Do you need to ensure your resource group metadata is stored in a particular region?

Resource tags

Resource tags are another way to organize resources. Tags provide extra information, or metadata, about your resources.

Things to know about resource tags:

- A resource tag consists of a name-value pair. For example, env = production or env = dev, test.
- You can assign one or more tags to each Azure resource, resource group, or subscription.
- Resource tags can be added, modified, and deleted. These actions can be done with PowerShell, the Azure CLI, Azure Resource Manager (ARM) templates, the REST API, or the Azure portal.
- Tags can be applied to a resource group. However, tags applied to a resource group aren't inherited by the resources in the group.

This metadata is useful for:

- **Resource management**: Tags enable you to locate and act on resources that are associated with specific workloads, environments, business units, and owners.
- **Cost management and optimization**: Tags enable you to group resources so that you can report on costs, allocate internal cost centers, track budgets, and forecast estimated cost.
- **Operations management**: Tags enable you to group resources according to how critical their availability is to your business. This grouping helps you formulate service-level agreements (SLAs). An SLA is an uptime or performance guarantee between you and your users.
- **Security**: Tags enable you to classify data by its security level, such as public or confidential.
- **Governance and regulatory compliance**: Tags enable you to identify resources that align with governance or regulatory compliance requirements, such as ISO 27001. Tags can also be part of your standards enforcement efforts. For example, you might require that all resources be tagged with an owner or department name.
- **Workload optimization and automation**: Tags can help you visualize all of the resources that participate in complex deployments. For example, you might tag a resource with its associated workload or application name and use software such as Azure DevOps to perform automated tasks on those resources.

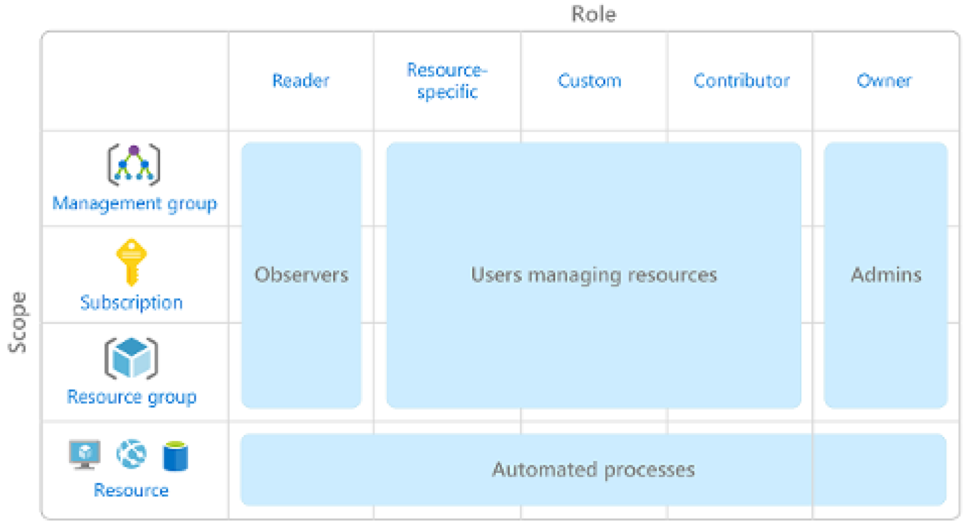
Things to consider when creating resource tags

- **Consider your organization's taxonomy**. Align your resource tags with accepted department nomenclature to make it easier to understand. Are there recognized terms for compliance or cost reporting for the organization? Add tags for office locations, confidentiality levels, or other defined policies.
- **Consider whether you need IT-aligned or business-aligned tagging**. Implement IT-aligned tagging or business-aligned tagging, or a combination of these approaches to be most effective.
  - The IT-aligned option is useful for tracking workload, application, function, or environment criteria. This option can reduce the complexity of monitoring assets.
  - The Business-aligned option helps to focus on accounting, business ownership, cost responsibility, and business criticality. This option provides improved accounting for costs and value of IT assets to the overall business. You can use Business-aligned tagging to shift the focus from an asset's operational cost to an asset's business value.
- **Consider the type of tagging required**. Plan to use different types of resource tags to support the organization.
  - Functional: Functional tags categorize resources according to their purpose within a workload. This tag shows the deployed environment for a resource, or other functionality and operational details.
  - Classification: Classification tags identify a resource by how it's used and what policies apply to it.
  - Accounting: Accounting tags allow a resource to be associated with specific groups within an organization for billing purposes.

    - Partnership: Partnership tags provide information about the people (other than IT members) who are associated with a resource, or otherwise affected by the resource.
    - Purpose: Purpose tags align resources to business functions to better support investment decisions.
- **Consider starting with a few tags and then scale out**. The resource tagging approach you choose can be simple or complex. Rather than identify all the possible tags required by the organization, prototype with just a few important or critical tags. Determine how effective the tagging scheme is before you add more resource tags.
- **Consider using Azure policy to apply tags and enforce tagging rules and conventions**. Resource tagging is only effective if it's used consistently across an organization. You can use Azure policy to require that certain tags be added to new resources as they're created. You can also define rules that reapply tags that have been removed.
- **Consider which resources require tagging**. Keep in mind that you don't need to enforce that a specific tag is present on all resources. You might decide that only mission-critical resources have the Impact tag. All non-tagged resources would then not be considered as mission critical.

Azure role-based access control (RBAC)

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



Things to know about Azure RBAC

- Allow one user to manage virtual machines in a subscription, and allow another user to manage virtual networks.
- Allow members of a database administrator group to manage SQL databases in a subscription.
- Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets.
- Allow an application to access all resources in a resource group.

Things to consider when using Azure RBAC

- **Consider the highest scope level for each requirement**. Your first step is to accurately define each role definition and its permissions. Next, assign the roles to specific users, groups, and service principals. Lastly, scope the roles to management groups, subscriptions, resource groups, and resources. Assign each role at the highest scope level that meets the requirements.

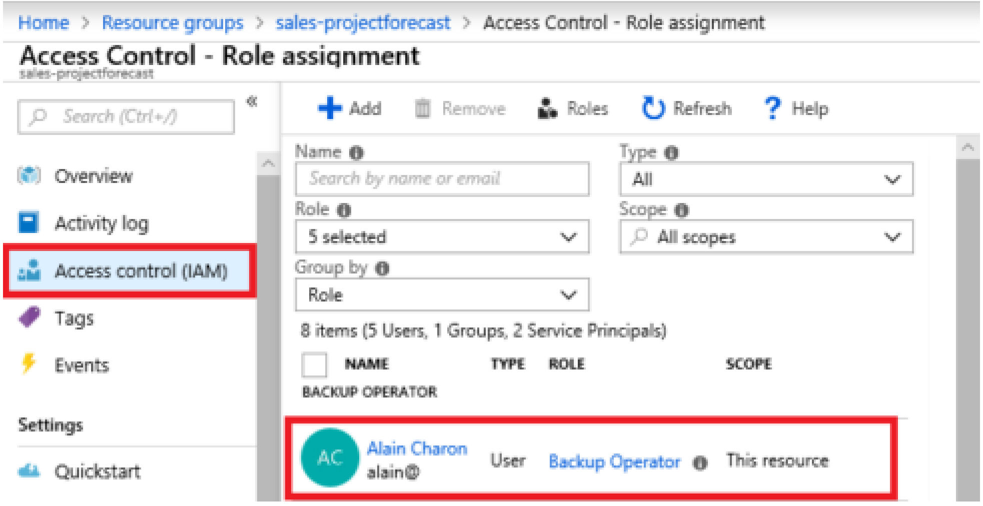
- **Consider the access needs for each user**. As you plan your access control strategy, it's a best practice to grant users the least privilege they need to get their work done. This method makes it easier to separate team member responsibilities. By limiting roles and scopes, you limit what resources are at risk if a security principal is ever compromised. You can create a diagram like the following example to help plan your Azure RBAC roles for Tailwind Traders.
- **Consider assigning roles to groups, and not users**. To make role assignments more manageable, avoid assigning roles directly to users. Instead, assign roles to groups. Assigning roles to groups helps minimize the number of role assignments.
- **Consider when to use Azure policies**. Azure policies are used to focus on resource properties. During deployment, an Azure policy can be used to ensure users can deploy only certain virtual machines in a resource group. By using a combination of Azure policies and Azure RBAC, you can provide effective access control in your Tailwind Traders solution. The following table compares these access models.
- **Consider when to create a custom role**. Sometimes, the built-in roles don't grant the precise level of access you need. Custom roles allow you to define roles that meet the specific needs of your organization. Custom roles can be shared between subscriptions that trust the same Azure Active Directory.
- **Consider how to resolve overlapping role assignments**. Azure RBAC is an additive model, so your effective permissions are the sum of your role assignments. Consider a user is granted the Contributor role at the subscription scope and the Reader role on a resource group. The sum of the Contributor permissions and the Reader permissions is effectively the Contributor role for the subscription. Therefore, in this case, the Reader role assignment has no impact.

When you grant access at a parent scope, those permissions are inherited by all child scopes. For example:

- When you assign the **Owner role** to a user at the management group scope, that user can manage everything in all subscriptions within the management group.
- When you assign the **Reader role** to a group at the subscription scope, the members of that group can view every resource group and resource within the subscription.
- When you assign the **Contributor role** to an application at the resource group scope, the application can manage resources of all types within that resource group, but not other resource groups within the subscription.

How do I manage Azure RBAC permissions?

You manage access permissions on the Access control (IAM) pane in the Azure portal.



Home > Resource groups > sales-projectforecast > Access Control - Role assignment

Access Control - Role assignment
sales-projectforecast

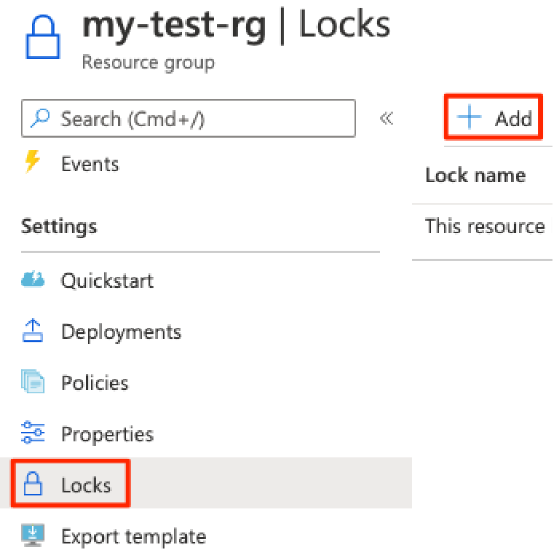
Name: Search by name or email
Type: All
Role: 5 selected
Scope: All scopes
Group by: Role

8 items (5 Users, 1 Groups, 2 Service Principals)

NAME	TYPE	ROLE	SCOPE
BACKUP OPERATOR			
AC Alain Charon alain@	User	Backup Operator	This resource

Prevent accidental changes by using resource locks

A resource lock prevents resources from being accidentally deleted or changed.



You can apply locks to a subscription, a resource group, or an individual resource. You can set the lock level to CanNotDelete or ReadOnly.

- **CanNotDelete** means authorized people can still read and modify a resource, but they can't delete the resource without first removing the lock.
- **ReadOnly** means authorized people can read a resource, but they can't delete or change the resource. Applying this lock is like restricting all authorized users to the permissions granted by the Reader role in Azure RBAC.

To modify a locked resource, you must first remove the lock. After you remove the lock, you can apply any action you have permissions to perform. This additional step allows the action to be taken, but it helps protect your administrators from doing something they might not have intended to do.

To make the protection process more robust, you can combine resource locks with Azure Blueprints. Azure Blueprints enables you to define the set of standard Azure resources that your organization requires. For example, you can define a blueprint that specifies that a certain resource lock must exist. Azure Blueprints can automatically replace the resource lock if that lock is removed.

Control and audit your resources by using Azure Policy

Overview

Azure Policy is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources. These policies enforce different rules across all of your resource configurations so that those configurations stay compliant with corporate standards.

Azure Policy enables you to define both individual policies and groups of related policies, known as initiatives.

Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies you've created.

Azure Policy can also prevent noncompliant resources from being created.

Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. For example, if you define a policy that allows only a certain SKU (stock-keeping unit) size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment.

Things to know about Azure Policy

- Azure Policy lets you define both individual policies and groups of related policies, called initiatives. Azure Policy comes with many built-in policy and initiative definitions.
- Azure policies are inherited down the hierarchy.
- You can scope and enforce Azure policies at different levels in the organizational hierarchy.
- Azure Policy evaluates all resources in Azure and Arc-enabled resources (specific resource types that are hosted outside of Azure).
- Azure Policy highlights resources that aren't compliant with the current policies.
- Use Azure Policy to prevent noncompliant resources from being created, and automatically remediate noncompliant resources.
- Azure Policy integrates with Azure DevOps by applying pre-deployment and post-deployment policies.

Azure Policy in action

- **Create a policy definition**
  - A policy definition expresses what to evaluate and what action to take.
  - For example, you could prevent VMs from being deployed in certain Azure regions. You also could audit your storage accounts to verify that they only accept connections from allowed networks.
- **Assign the definition to resources**
  - To implement your policy definitions, you assign definitions to resources.
  - A policy assignment is a policy definition that takes place within a specific scope. This scope could be a management group (a collection of multiple subscriptions), a single subscription, or a resource group.
  - Policy assignments are inherited by all child resources within that scope.
- **Review the evaluation results**
  - When a condition is evaluated against your existing resources, each resource is marked as compliant or noncompliant. You can review the noncompliant policy results and take any action that's needed.
  - Policy evaluation happens about once per hour.

What are Azure Policy initiatives?

An Azure Policy initiative is a way of grouping related policies together. The initiative definition contains all of the policy definitions to help track your compliance state for a larger goal.

For example, Azure Policy includes an initiative named Enable Monitoring in Azure Security Center. Its goal is to monitor all of the available security recommendations for all Azure resource types in Azure Security Center.

Under this initiative, the following policy definitions are included:

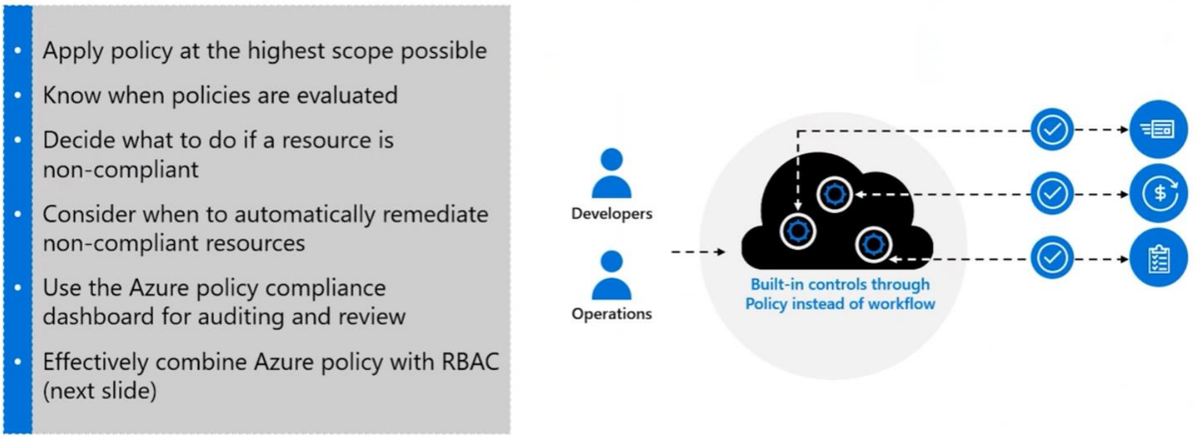
- **Monitor unencrypted SQL Database in Security Center**: This policy monitors for unencrypted SQL databases and servers.
- **Monitor OS vulnerabilities in Security Center**: This policy monitors servers that don't satisfy the configured OS vulnerability baseline.
- **Monitor missing Endpoint Protection in Security Center**: This policy monitors for servers that don't have an installed endpoint protection agent.

In fact, the Enable Monitoring in Azure Security Center initiative contains over 100 separate policy definitions.

You define initiatives by using the Azure portal or command-line tools. From the Azure portal, you can search the list of built-in initiatives that are built into Azure. You also can create your own custom policy definition.

Like a policy assignment, an initiative assignment is an initiative definition that's assigned to a specific scope of a management group, a subscription, or a resource group.

Considerations for Azure Policy



- Apply policy at the highest scope possible
- Know when policies are evaluated
- Decide what to do if a resource is non-compliant
- Consider when to automatically remediate non-compliant resources
- Use the Azure policy compliance dashboard for auditing and review
- Effectively combine Azure policy with RBAC (next slide)

Azure Active Directory (Azure AD)

Azure AD provides services such as:

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

Things to know about Azure AD identity management

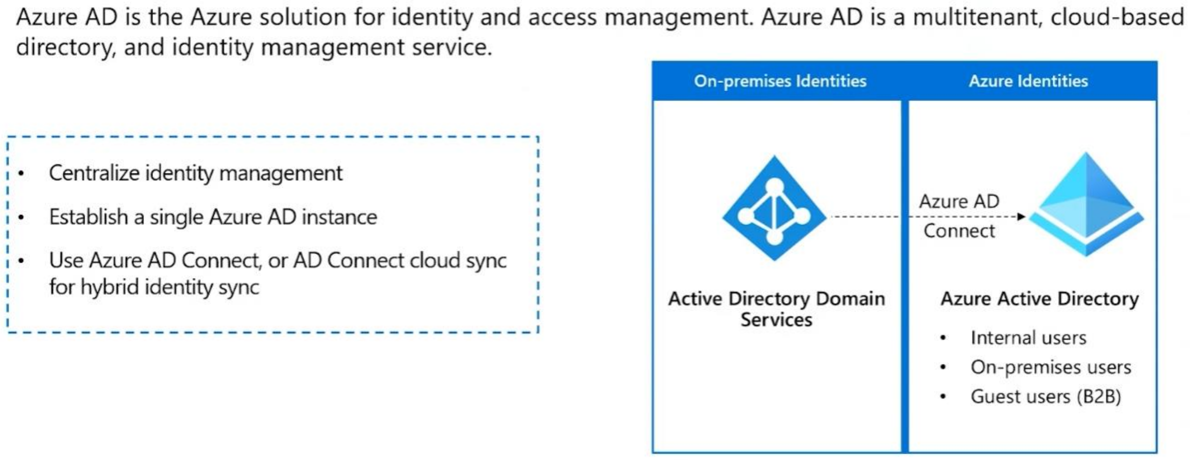
- You can implement Azure AD as a **cloud-only identity solution** for all your employee user accounts.
- The cloud-only identity solution provides both identity management and protection for your accounts, including role-based access control (RBAC), conditional access, and access reviews.
- Azure AD also offers a **hybrid identity solution** for identity management in Tailwind Traders hybrid environments.
- In hybrid environments, Azure AD extends on-premises Active Directory to the cloud.

- With Azure AD Connect or Azure AD Connect cloud sync, you can bring on-premises identities into Azure AD. After the on-premises accounts are in Azure AD, they'll get the benefits of easy management and identity protection.

Things to consider when using Azure AD identity management

- **Consider benefits of centralized identity management**. (Microsoft recommended) Integrate your on-premises and cloud directories when you're working in a hybrid identity scenario. Integration enables your Tailwind Traders IT team to manage accounts from one location, whenever an account is created. Centralized integration also helps your users be more productive by providing a common identity for accessing both cloud and on-premises resources.
- **Consider using a single Azure AD instance**. Use a single authoritative source and consistency to increase clarity and reduce security risks from human errors and configuration complexity. Designate a single Azure AD directory as the authoritative source for Tailwind Traders corporate and organizational accounts.
- **Consider limiting account synchronization**. Don't synchronize accounts to Active Directory that have high privileges in your existing Azure AD Tailwind Traders instance. By default, Azure AD Connect filters out these high privileged accounts. This configuration mitigates the risk of adversaries pivoting from cloud to on-premises assets (which could result in a major incident).
- **Consider password hash synchronization**. Enable password hash synchronization to sync user password hashes from an on-premises Azure AD instance to a cloud-based Azure AD instance. This sync helps to protect Tailwind Traders against leaked credentials being replayed from previous sign-ins.
- **Consider single sign-on (SSO)**. Enable SSO to reduce the need for multiple passwords. Multiple passwords increase the likelihood of users reusing passwords or using weak passwords. With SSO, users provide their primary work or school account for their domain-joined devices and company resources. Their application access can be automatically provisioned (or deprovisioned) based on their Tailwind Traders organization group memberships and their status as an employee.
- **Consider overhead of managing separate identities**. Calculate the overhead of not integrating the Tailwind Traders on-premises identity with their cloud identity. Separate identities can result in extra account management. This overhead increases the likelihood of mistakes and security breaches.

When to use Azure Active Directory?



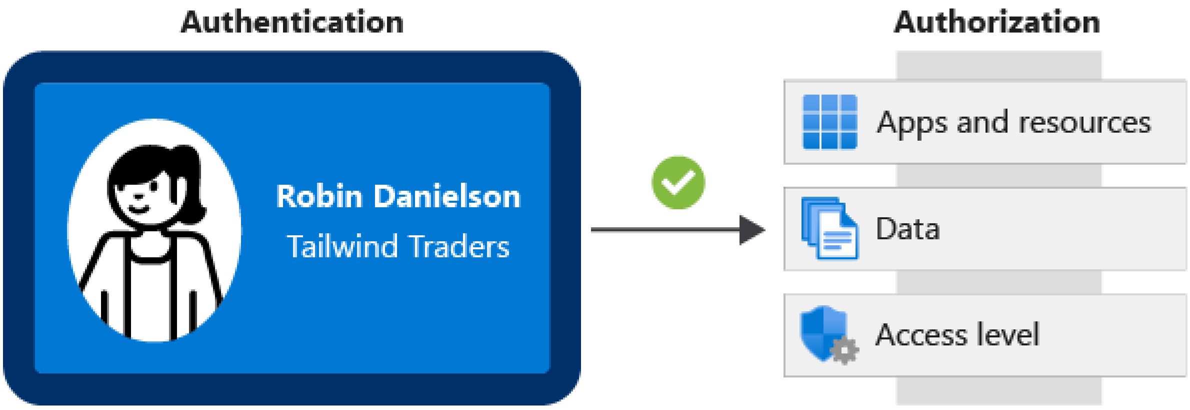
Azure AD is the Azure solution for identity and access management. Azure AD is a multitenant, cloud-based directory, and identity management service.

- Centralize identity management
- Establish a single Azure AD instance
- Use Azure AD Connect, or AD Connect cloud sync for hybrid identity sync

Compare authentication and authorization

Authentication is the process of establishing the identity of a person or service that wants to access a resource. It involves the act of challenging a party for legitimate credentials and provides the basis for creating a security principal for identity and access control. It establishes whether the user is who they say they are.

Authorization is the process of establishing what level of access an authenticated person or service has. It specifies what data they're allowed to access and what they can do with it.



Identity and Access Management (IAM)

To implement authentication and authorization, Azure Architects design identity and access management (IAM) solutions. These solutions must work for all users, applications, and devices.

Things to know about IAM

- **Unified identity management**. Manage all your identities and access to all your applications in a central location, whether they're in the cloud or on-premises, to improve visibility and control.
- **Seamless user experience**. Provide an easy, fast sign-in experience to keep your users productive, reduce time spent managing passwords, and increase end-user productivity.
- **Secure adaptive access**. Protect access to resources and data by using strong authentication and risk-based adaptive access policies without compromising the user experience.
- **Simplified identity governance**. Control access to applications and data for all users and admins. Automated identity governance ensures only authorized users have access.

Things to consider when using IAM

- **Consider using Azure Active Directory**. Develop with Azure AD for a solution that combines core directory services, application access management, and identity protection. Azure AD provides an identity and access management system for employees that can operate in a cloud or hybrid environment.
- **Consider your business-to-business (B2B) requirements**. Support collaboration for guest users and external business partners, such as suppliers and vendors. Build your solution with Azure AD B2B (business-to-business) to support business-to-business operations.
- **Consider your business-to-customer (B2C) scenarios**. Control how customers sign up, sign in, and manage their profiles when they use your apps. Use Azure AD B2C (business-to-customer) to develop an Azure AD solution that supports customer-focused operations.

What's Single Sign-On (SSO)

Single sign-on enables a user to sign in one time and use that credential to access multiple resources and applications from different providers.

More identities mean more passwords to remember and change. Password policies can vary among applications. As complexity requirements increase, it becomes increasingly difficult for users to remember them.

The more passwords a user has to manage, the greater the risk of a credential-related security incident.

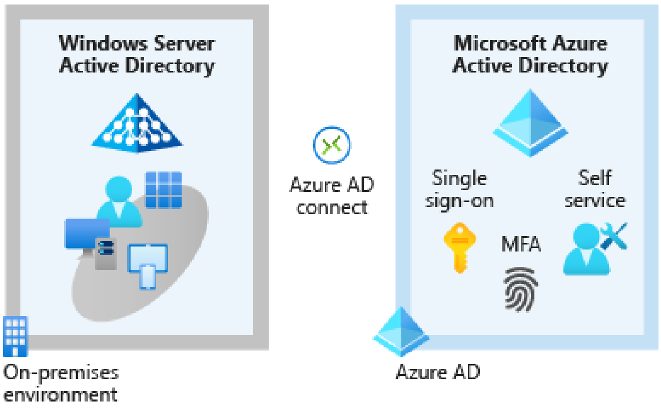
With SSO, you need to remember only one ID and one password. Access across applications is granted to a single identity that's tied to the user, which simplifies the security model.

How can I connect Active Directory with Azure AD?

Connecting Active Directory with Azure AD enables you to provide a consistent identity experience to your users.

There are a few ways to connect your existing Active Directory installation with Azure AD. Perhaps the most popular method is to use Azure AD Connect.

Azure AD Connect synchronizes user identities between on-premises Active Directory and Azure AD. Azure AD Connect synchronizes changes between both identity systems, so you can use features like SSO, multifactor authentication, and self-service password reset under both systems. Self-service password reset prevents users from using known compromised passwords.



What are multifactor authentication

Multifactor authentication is a process where a user is prompted during the sign-in process for an additional form of identification. Examples include a code on their mobile phone or a fingerprint scan.

Multifactor authentication provides additional security for your identities by requiring two or more elements to fully authenticate.

These elements fall into three categories:

- **Something the user knows:** This might be an email address and password.
- **Something the user has:** This might be a code that's sent to the user's mobile phone.
- **Something the user is:** This is typically some sort of biometric property, such as a fingerprint or face scan that's used on many mobile devices.

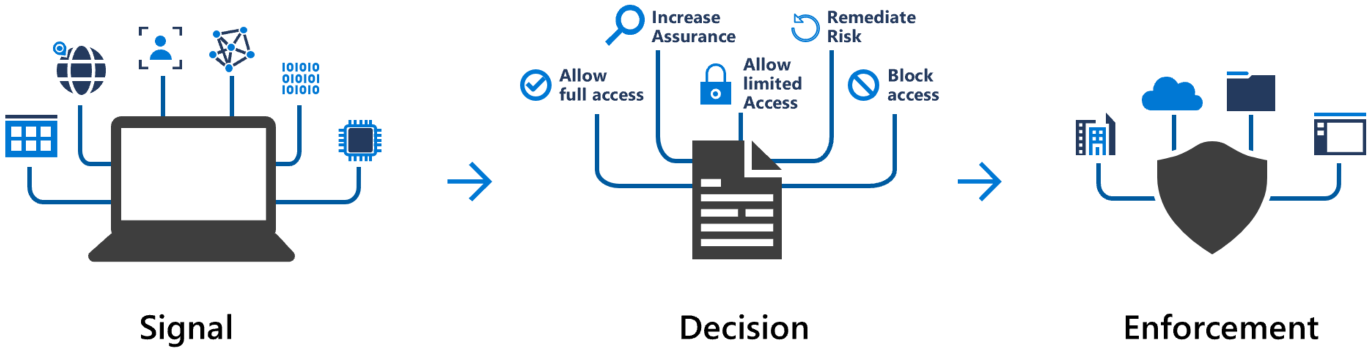
With multifactor authentication enabled, an attacker who has a user's password would also need to have possession of their phone or their fingerprint to fully authenticate.

What's Conditional Access

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

Conditional Access also provides a more granular multifactor authentication experience for users. For example, a user might not be challenged for second authentication factor if they're at a known location.

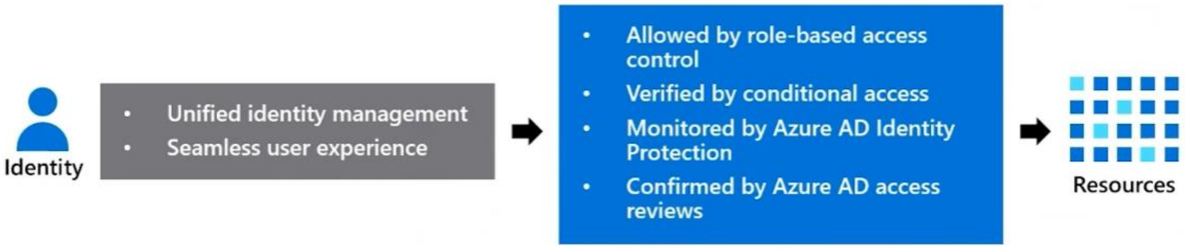
However, they might be challenged for a second authentication factor if their sign-in signals are unusual or they're at an unexpected location.

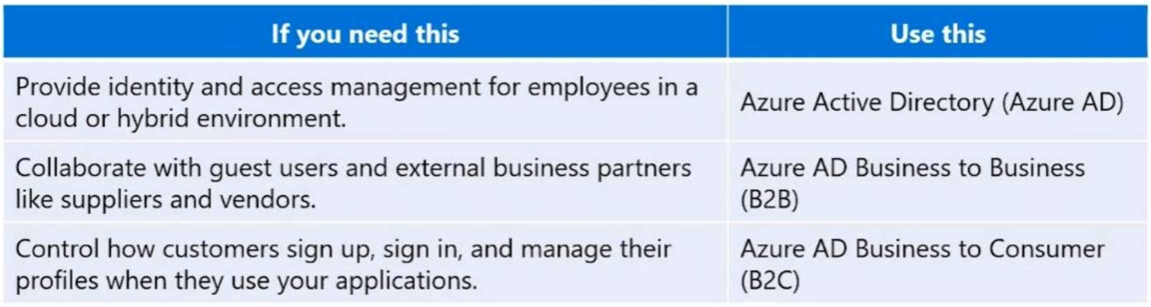


Conditional Access is useful when you need to:

- **Require multifactor authentication to access an application**.
    - You can configure whether all users require multifactor authentication or only certain users, such as administrators.
    - You can also configure whether multifactor authentication applies to access from all networks or only untrusted networks.
- **Require access to services only through approved client applications**.
    - For example, you might want to allow users to access Office 365 services from a mobile device as long as they use approved client apps, like the Outlook mobile app.
- **Require users to access your application only from managed devices**.
    - A managed device is a device that meets your standards for security and compliance.
- **Block access from untrusted sources**, such as access from unknown or unexpected locations.

What is identity and access management?


Identity
Unified identity management
Seamless user experience
Allowed by role-based access control
Verified by conditional access
Monitored by Azure AD Identity Protection
Confirmed by Azure AD access reviews
Resources

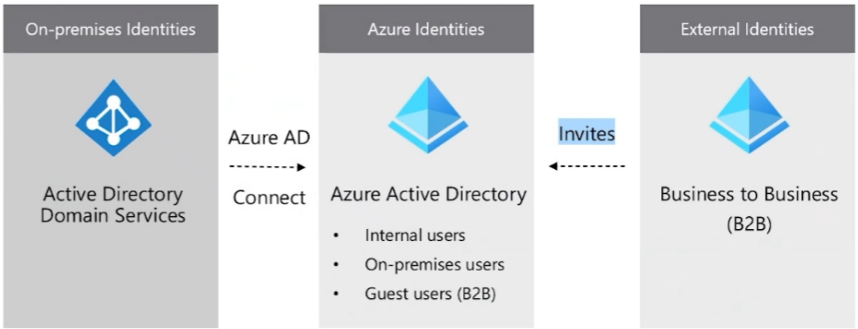


If you need this	Use this
Provide identity and access management for employees in a cloud or hybrid environment.	Azure Active Directory (Azure AD)
Collaborate with guest users and external business partners like suppliers and vendors.	Azure AD Business to Business (B2B)
Control how customers sign up, sign in, and manage their profiles when they use your applications.	Azure AD Business to Consumer (B2C)

When to use Azure AD Business to Business (B2B)?

Azure AD B2B enables you to securely collaborate with external partners.

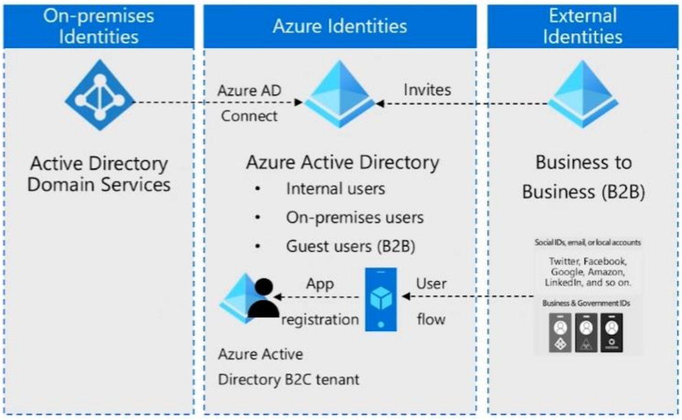
- Integrate with identity providers
- Use conditional access policies to intelligently grant or deny access
- Require MFA for guest users



When to use Azure AD Business to Customer (B2C)?

Azure AD B2C is a type of Azure AD tenant that you use to manage customer identities and their access to your applications.

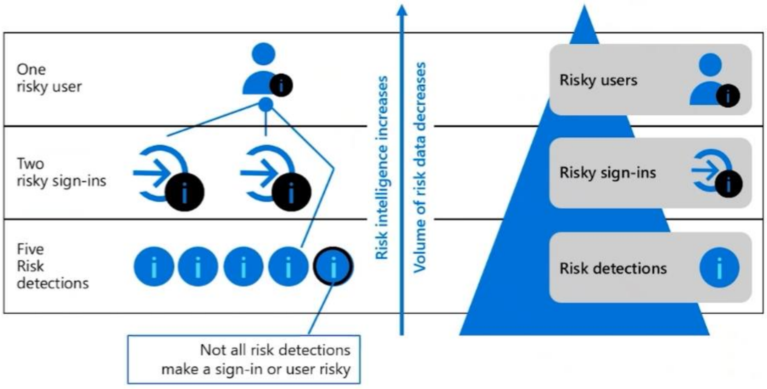
- Integrate with external user stores
- Provide single sign-on access with a user-provided identity
- Create a custom-branded identity solution
- Use policies to configure user journeys
- Use progressive profiling to gradually collect user information
- Pass user data to a 3rd party for validation



When to use Azure AD Identity Protection?

Identity protection is an Azure AD tool that automates the detection and remediation of identity-based risks.

- Configure the policies and actively review the results
- Set the sign-in risk policy to Medium and above, and allow self-remediation options
- Set the user risk policy threshold to High
- Allow for excluding users - emergency access or break-glass administrator accounts
- Send data to Conditional Access or other security information and event management (SIEM) tool



One
risky user
Two
risky sign-ins
Five
Risk
detections
Not all risk detections make a sign-in or user risky
Risk intelligence increases
Volume of risk data decreases
Risky users
Risky sign-ins
Risk detections

User risk represents the probability that a given identity or account is compromised. An example is when a user's valid credentials are leaked. User risks are calculated offline by using Microsoft's internal and external threat intelligence sources. Here are some user risks that can be identified:

- **Leaked credentials**: Microsoft checks for leaked credentials from the dark web, paste sites, or other sources. These leaked credentials are checked against Azure AD users' current valid credentials for valid matches.
- **Azure AD threat intelligence**: This risk detection type indicates user activity that's unusual for the given user or is consistent with known attack patterns.

Sign-in risk represents the probability that a given sign-in (authentication request) isn't authorized by the identity owner. Sign-in risk can be calculated in real time or offline. Here are some sign-in risks that can be identified:

- **Anonymous IP address**: A sign-in attempt from an anonymous IP address like a Tor browser or an anonymized VPN.
- **Atypical travel**: Two sign-ins from the same user that originate from a geographically distant location. Given past behavior, at least one of the locations might also be atypical for the user.
- **Malware-linked IP address**: A sign-in from an IP address that's infected with malware and the malware is known to actively communicate with a bot server.
- **Password spray**: A password spray attack where a bad actor tries to defeat lockout and detection by attempting sign-in with different user names and the same password.

Things to consider when using Identity Protection

- **Consider "High" threshold for user risk policy**. (Microsoft recommended) Set the risk policy level for your users to "High." A high setting can detect for leaked credentials and unusual activity for a user, and check for known attack patterns. By setting the policy threshold to "high," you can spread a wide net to prevent attacks that target user credentials.
- **Consider "Medium and above" threshold for sign-in risk policy**. (Microsoft recommended) Configure the risk policy level for sign-in attempts to apps to "Medium and above." This setting supports the Identity Protection self-remediation options. Self-remediation, like password changes and MFA, have less impact than blocking users.
- **Consider investigating risks in the Azure portal**. Investigate risk events in the Azure portal and identify any weak areas in your security implementation. Download the risk events in .CSV format and view the output in the Security section of Azure AD. Use the Microsoft Graph API integrations to aggregate your data with other sources.
- **Consider exporting your risk detection data**. Export the risk detection data by using the Microsoft Sentinel data connector for Identity Protection.

Design for access reviews

An employee of a company might work in several different roles during their tenure.

Each position they hold can require access to different resources or have varying levels of permissions requirements.

When an employee is first hired, they need initial access to corporate resources and apps.

For each position they hold, they can have specific access requirements and privileges.

When the employee leaves the company, their access is removed.

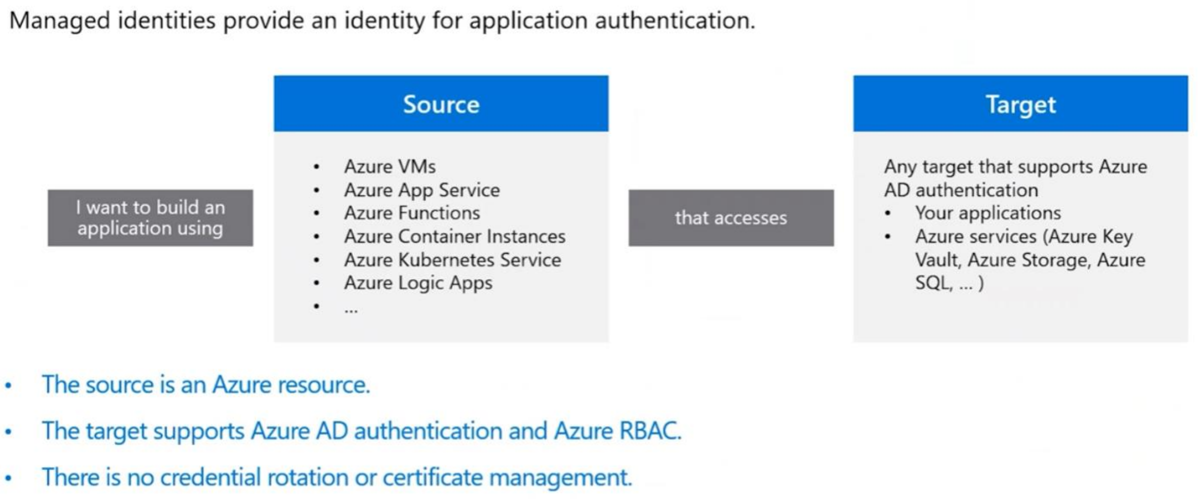
To ensure employees and users always have the correct access, you can perform an access review.

An Azure Active Directory access review is a planned review of the access needs, rights, and history of user access.

Things to know to determine the purpose of the Azure AD access review

- While you consider how to use Azure AD access reviews for Tailwind Traders, think about the following characteristics of an access review.
- Access reviews mitigate risk by protecting, monitoring, and auditing access to critical assets.
- You use access reviews to help ensure the correct users have the correct access to the correct resources.
- Confirm correct user access to apps integrated with Azure AD for single sign-on, including SaaS apps and line-of-business apps.
- Verify group memberships that are synchronized to Azure AD, or created in Azure AD or Microsoft 365, including Microsoft Teams.
- Check access packages that group resources (groups, apps, and sites) into a single package to manage access.
- Access reviews can also be used for Azure AD roles and Azure Resource roles as defined in Privileged Identity Management (PIM).

Design managed identities



Managed identities provide an identity for application authentication.

I want to build an application using

Source

- Azure VMs
- Azure App Service
- Azure Functions
- Azure Container Instances
- Azure Kubernetes Service
- Azure Logic Apps
- …

that accesses

Target

Any target that supports Azure AD authentication

- Your applications
- Azure services (Azure Key Vault, Azure Storage, Azure SQL, …)

- The source is an Azure resource.
- The target supports Azure AD authentication and Azure RBAC.
- There is no credential rotation or certificate management.

There are two types of managed identities:

- **System-assigned**: Some Azure services allow you to enable a managed identity directly on a service instance. When you enable a system-assigned managed identity, an identity is created in Azure AD that's tied to the lifecycle of that service instance. When the resource is deleted, Azure automatically deletes the identity. By design, only that Azure resource can use that identity to request tokens from Azure AD.
- **User-assigned**: You can create a managed identity as a standalone Azure resource. Create a user-assigned managed identity and assign it to one or more instances of an Azure service. A user-assigned identity is managed separately from the resources that use it.

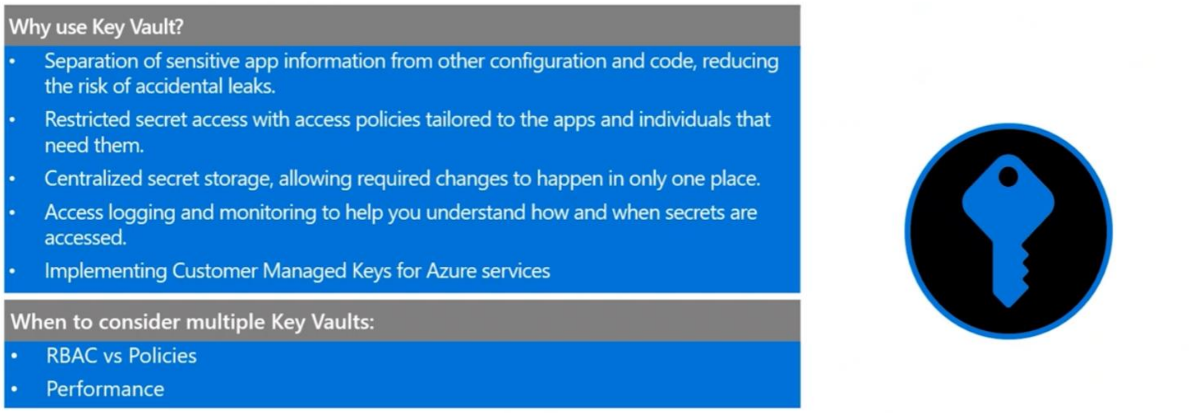
Things to consider when using managed identities

- **Consider your Azure services and your targets**: Build your apps with Azure App Service and access Azure Storage, and by using managed identities, you won't have to manage any credentials.
- **Consider using system-assigned managed identities**. Implement system-assigned managed identities for workloads that are contained within a single Azure resource, or for workloads that need independent identities.
- **Consider choosing user-assigned managed identities**. Choose user-assigned managed identities for workloads that run on multiple resources that can share a single identity. This type of identity is also good for workloads that need pre-authorization to a secure resource as part of a provisioning flow. User-assigned identities are suited for workloads with resources that are recycled frequently, but where permissions should stay consistent.

- **Consider the benefits of managed identities for VMs in Azure**. Review these scenarios that highlight the benefits to being able to use managed identities for VMs that are hosted in Azure.
  - You decide to run the stock-tracking apps inside an Azure-hosted VM that has an assigned managed identity. This setup allows the app to use an Azure key vault to authenticate without having to store a username and password in code.
  - Now that your company has migrated your VM from on-premises to Azure, you can remove the hard-coded authentication details from the application code. You want to use the more secure managed identity token for access to Azure resources.
- **Consider Azure Key Vault authentication for Azure resources**. Authenticate managed identities for Azure resources by integrating with Azure Key Vault.

Design for Azure Key Vault

Azure Key Vault provides a secure storage area so you can manage all your app secrets and properly encrypt your data in transit or while it's being stored.



Why use Key Vault?

- Separation of sensitive app information from other configuration and code, reducing the risk of accidental leaks.
- Restricted secret access with access policies tailored to the apps and individuals that need them.
- Centralized secret storage, allowing required changes to happen in only one place.
- Access logging and monitoring to help you understand how and when secrets are accessed.
- Implementing Customer Managed Keys for Azure services

When to consider multiple Key Vaults:

- RBAC vs Policies
- Performance

Azure Key Vault can help you solve security problems:

- **Manage secrets**. You can securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets.
- **Manage keys**. Key Vault is a key management solution that lets you easily create and control encryption keys to encrypt corporate data.
- **Manage certificates**. Key Vault is also a service that makes it easy to enroll, manage, and deploy public and private Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates for use with Azure and internal connected resources.

Things to know about Azure Key Vault

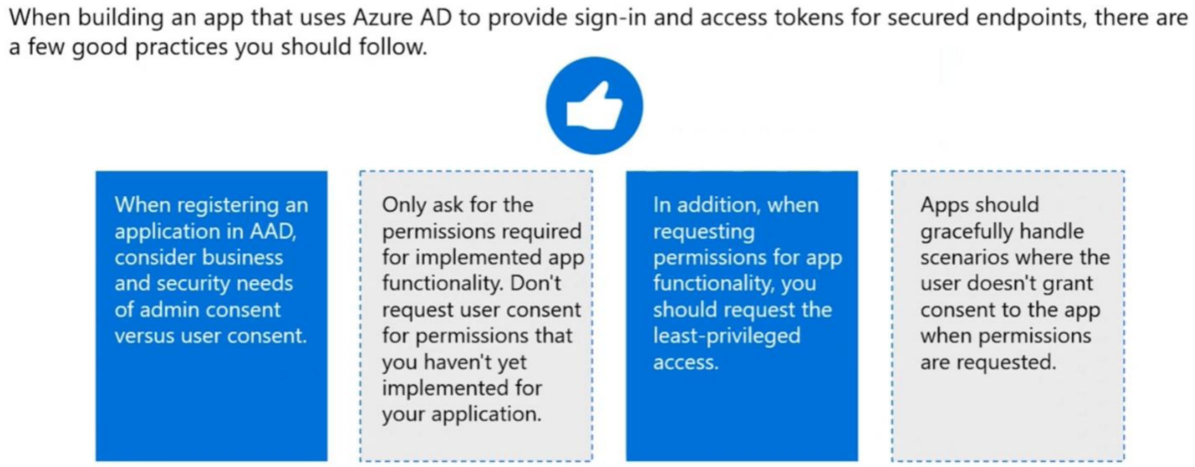
- Key Vault is available in two service tiers:
  - Standard tier lets you encrypt your data with a software key.
  - Premium tier offers hardware security module (HSM)-protected keys.
- You can build access policies with restricted secret access that are tailored to the apps and individuals that need them.
- Sensitive app information can be separated from other configuration and code, which reduces the risk of accidental leaks.
- Centralized secret storage allows required changes to happen in only one place.
- Logging and monitoring in Key Vault helps you understand how and when secrets are accessed.

- Key Vault provides secure access to sensitive information from within your apps:
    - Keys, secrets, and certificates are protected without writing extra code, and you can use these assets from your apps.
    - Customers can own and manage their own keys, secrets, and certificates. Your apps don't own the responsibility or potential liability for customer assets. You can concentrate on providing the core software features for Tailwind Traders apps.
    - Your app can use keys for signing and encryption while keeping key management external from the app.
    - You can manage credentials like passwords, access keys, and shared access signature tokens by storing them in Key Vault as secrets.

Things to consider when using Azure Key Vault

- **Consider using separate key vaults**. Key vaults define security boundaries for stored secrets. Grouping secrets into the same vault increases the blast radius of a security event. Consider what secrets a specific application should have access to, and then separate your key vaults based on this delineation. Separating key vaults by application is the most common boundary.
- **Consider access to the key vault**. Secure access to your key vaults by allowing only authorized applications and users. Here are some suggestions.
    - Create access policies for every vault.
    - Use the principle of least privilege access to grant access.
    - Turn on firewall and virtual network service endpoints.
- **Consider data protection for your key vault**. Turn on soft delete and purge protection to protect your key vault data.
    - Soft delete is designed to prevent accidental deletion of your key vault and keys, secrets, and certificates stored inside key vault. Think of soft-delete like a recycle bin.
    - Purge protection Purge protection is designed to prevent the deletion of your key vault, keys, secrets, and certificates by a malicious insider. Think of this as a recycle bin with a time based lock. You can recover items at any point during the configurable retention period.

Best practices for requesting permissions



When building an app that uses Azure AD to provide sign-in and access tokens for secured endpoints, there are a few good practices you should follow.

When registering an application in AAD, consider business and security needs of admin consent versus user consent.

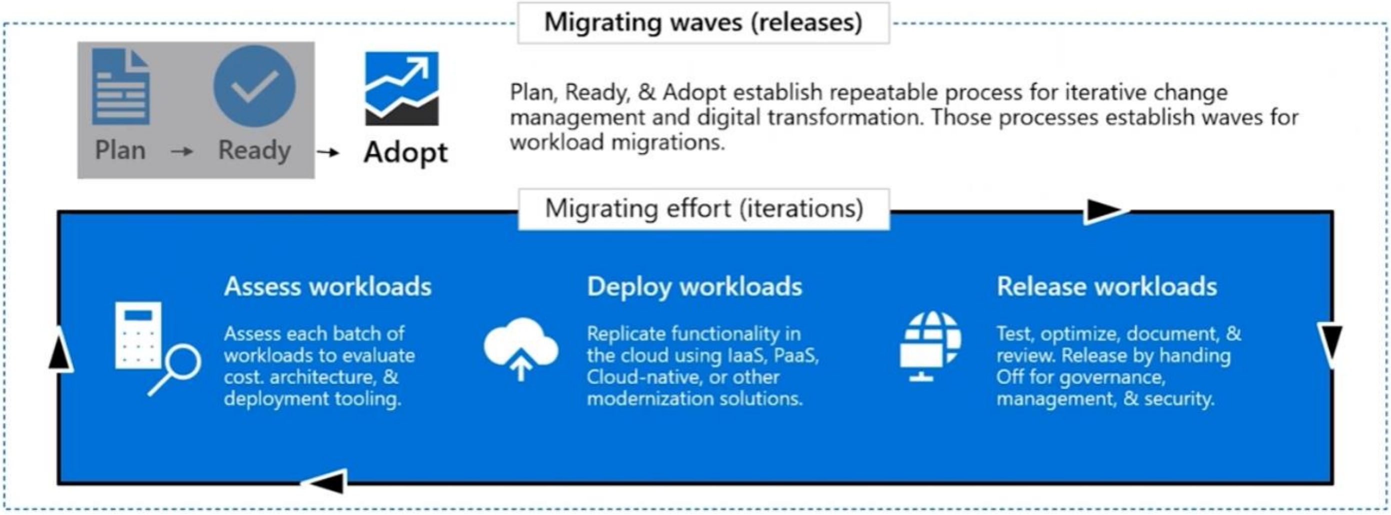
Only ask for the permissions required for implemented app functionality. Don't request user consent for permissions that you haven't yet implemented for your application.

In addition, when requesting permissions for app functionality, you should request the least-privileged access.

Apps should gracefully handle scenarios where the user doesn't grant consent to the app when permissions are requested.

Migrate to Azure

Focus on migration efforts



Migrating waves (releases)

Plan, Ready, & Adopt establish repeatable process for iterative change management and digital transformation. Those processes establish waves for workload migrations.

Migrating effort (iterations)

Assess workloads

Assess each batch of workloads to evaluate cost. architecture, & deployment tooling.

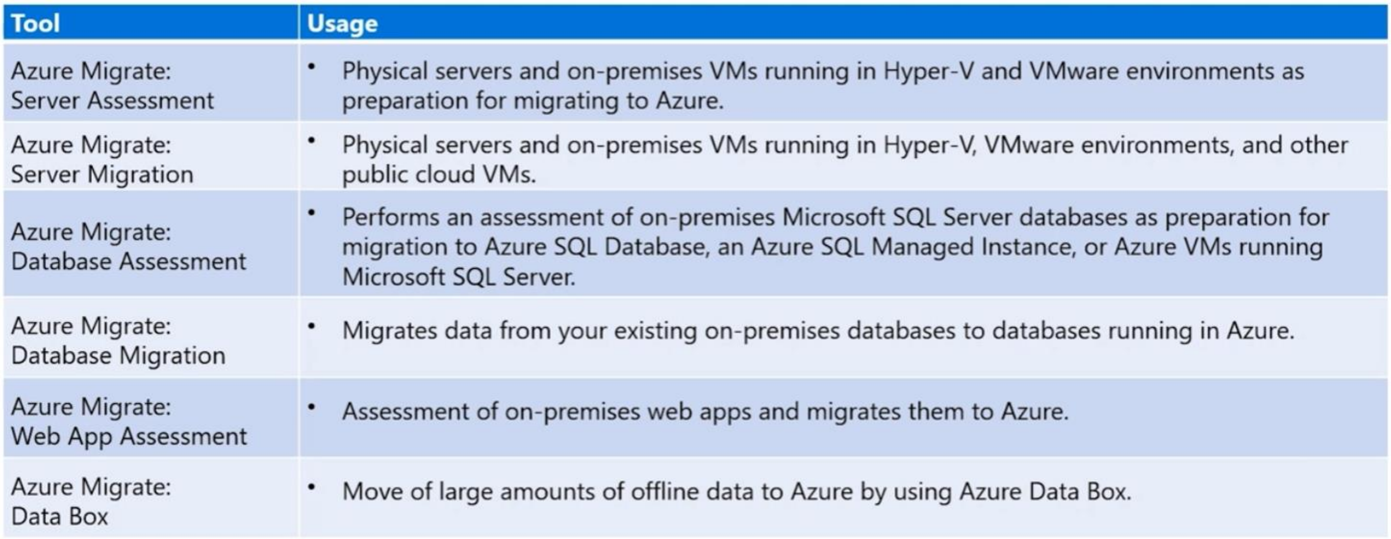
Deploy workloads

Replicate functionality in the cloud using IaaS, PaaS, Cloud-native, or other modernization solutions.

Release workloads

Test, optimize, document, & review. Release by handing Off for governance, management, & security.

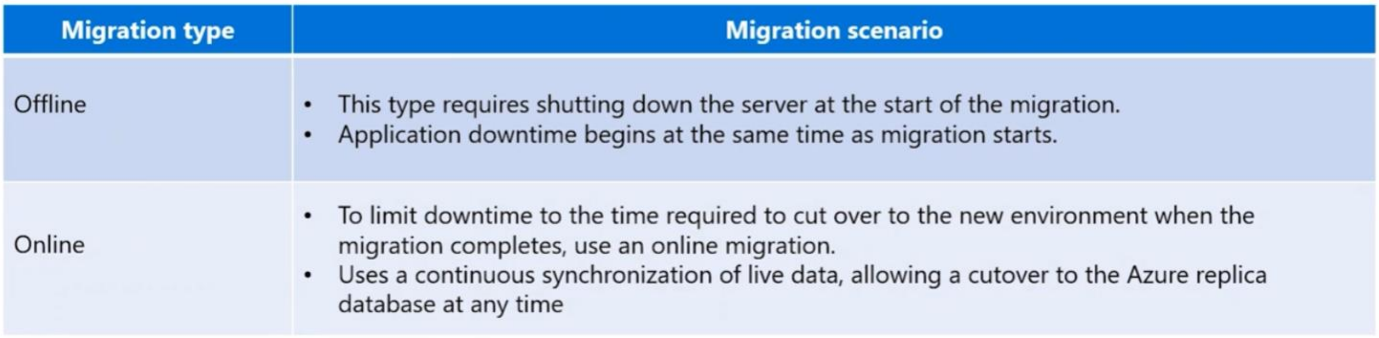
Identify Migration Tools



Tool	Usage
Azure Migrate: Server Assessment	• Physical servers and on-premises VMs running in Hyper-V and VMware environments as preparation for migrating to Azure.
Azure Migrate: Server Migration	• Physical servers and on-premises VMs running in Hyper-V, VMware environments, and other public cloud VMs.
Azure Migrate: Database Assessment	• Performs an assessment of on-premises Microsoft SQL Server databases as preparation for migration to Azure SQL Database, an Azure SQL Managed Instance, or Azure VMs running Microsoft SQL Server.
Azure Migrate: Database Migration	• Migrates data from your existing on-premises databases to databases running in Azure.
Azure Migrate: Web App Assessment	• Assessment of on-premises web apps and migrates them to Azure.
Azure Migrate: Data Box	• Move of large amounts of offline data to Azure by using Azure Data Box.

Select a database migration type

Database migrations can be performed offline and online:



Migration type	Migration scenario
Offline	• This type requires shutting down the server at the start of the migration. • Application downtime begins at the same time as migration starts.
Online	• To limit downtime to the time required to cut over to the new environment when the migration completes, use an online migration. • Uses a continuous synchronization of live data, allowing a cutover to the Azure replica database at any time

Each migration type supports different source and target database pairs:

- Check for support of your migration scenario as migration tools are updated frequently.

Azure Data Box

The Azure Data Box allows a quick, inexpensive, and secure transfer of terabytes of data into Azure. You order the Data Box device via the Azure portal. Microsoft ships you a storage device through a regional carrier.

Data Box is designed to move large amounts of data to Azure with little to no impact to network.

Each storage device has a maximum usable storage capacity of 80 TB.

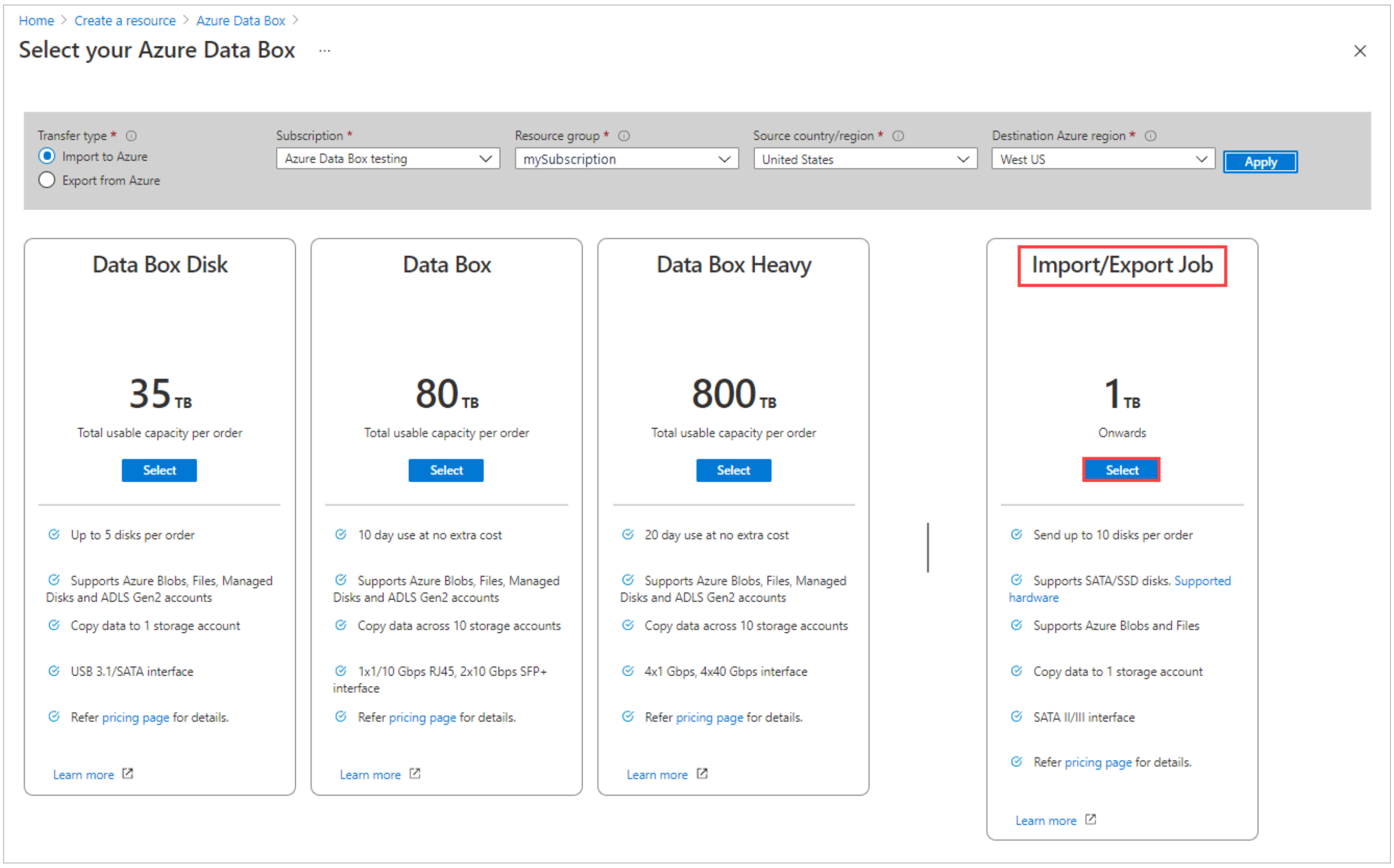
If you have 40 - 500 TB of data that you want to transfer to or from Azure, you would benefit from using Data Box. For data sizes < 40 TB, use Data Box Disk, and for data sizes > 500 TB, sign up for Data Box Heavy.

You can order the Data Box device via the Azure portal to import or export data from Azure. Once the device is received, you can quickly set it up using the local web UI.

Depending on whether you will import or export data, copy the data from your servers to the device or from the device to your servers, and ship the device back to Azure.

If importing data to Azure, in the Azure datacenter, your data is automatically uploaded from the device to Azure. The entire process is tracked end-to-end by the Data Box service in the Azure portal.

Compare Import/Export and Data Box



Home > Create a resource > Azure Data Box >

Select your Azure Data Box

Transfer type *
● Import to Azure
○ Export from Azure

Subscription *: Azure Data Box testing

Resource group *: mySubscription

Source country/region *: United States

Destination Azure region *: West US

Apply

Data Box Disk

35 TB

Total usable capacity per order

Select

- Up to 5 disks per order
- Supports Azure Blobs, Files, Managed Disks and ADLS Gen2 accounts
- Copy data to 1 storage account
- USB 3.1/SATA interface
- Refer pricing page for details.

Learn more

Data Box

80 TB

Total usable capacity per order

Select

- 10 day use at no extra cost
- Supports Azure Blobs, Files, Managed Disks and ADLS Gen2 accounts
- Copy data across 10 storage accounts
- 1x1/10 Gbps RJ45, 2x10 Gbps SFP+ interface
- Refer pricing page for details.

Learn more

Data Box Heavy

800 TB

Total usable capacity per order

Select

- 20 day use at no extra cost
- Supports Azure Blobs, Files, Managed Disks and ADLS Gen2 accounts
- Copy data across 10 storage accounts
- 4x1 Gbps, 4x40 Gbps interface
- Refer pricing page for details.

Learn more

Import/Export Job

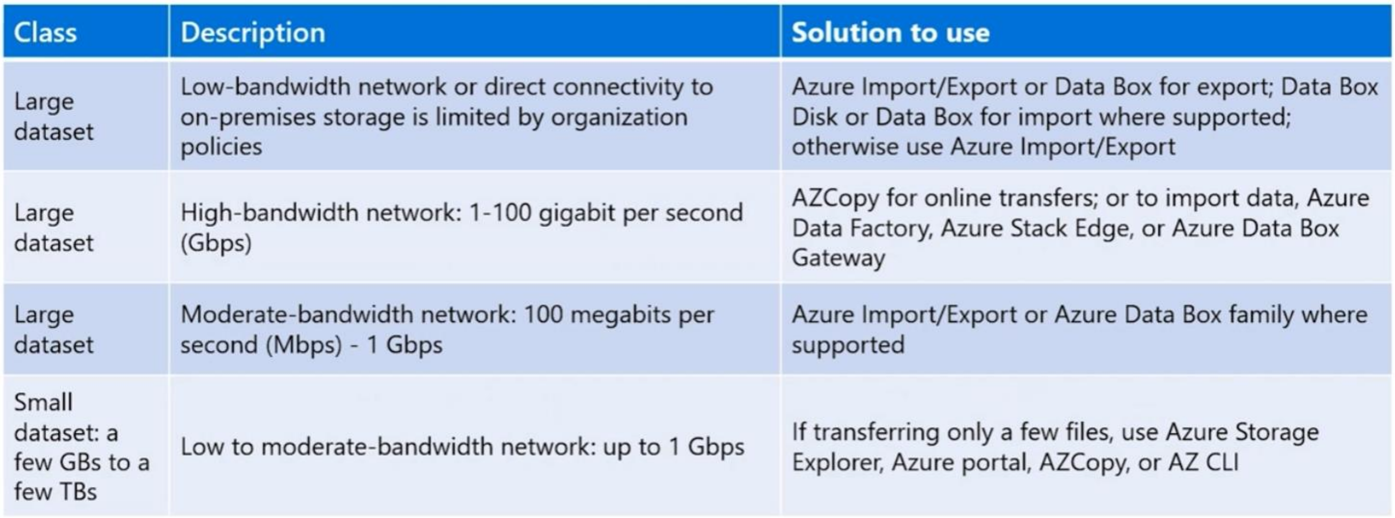
1 TB

Onwards

Select

- Send up to 10 disks per order
- Supports SATA/SSD disks. Supported hardware
- Supports Azure Blobs and Files
- Copy data to 1 storage account
- SATA II/III interface
- Refer pricing page for details.

Learn more



Class	Description	Solution to use
Large dataset	Low-bandwidth network or direct connectivity to on-premises storage is limited by organization policies	Azure Import/Export or Data Box for export; Data Box Disk or Data Box for import where supported; otherwise use Azure Import/Export
Large dataset	High-bandwidth network: 1-100 gigabit per second (Gbps)	AZCopy for online transfers; or to import data, Azure Data Factory, Azure Stack Edge, or Azure Data Box Gateway
Large dataset	Moderate-bandwidth network: 100 megabits per second (Mbps) - 1 Gbps	Azure Import/Export or Azure Data Box family where supported
Small dataset: a few GBs to a few TBs	Low to moderate-bandwidth network: up to 1 Gbps	If transferring only a few files, use Azure Storage Explorer, Azure portal, AZCopy, or AZ CLI

Azure Storage redundancy

Overview

- Azure Storage always stores multiple copies of your data so that it's protected from planned and unplanned events, including transient hardware failures, network or power outages, and massive natural disasters. Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures.
- When deciding which redundancy option is best for your scenario, consider the tradeoffs between lower costs and higher availability.
- The factors that help determine which redundancy option you should choose include:
    - How your data is replicated in the primary region.
    - Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters (geo-replication).
    - Whether your application requires read access to the replicated data in the secondary region if the primary region becomes unavailable for any reason (geo-replication with read access).

Redundancy in the primary region

Data in an Azure Storage account is **always replicated three times in the primary region**. Azure Storage offers two options for how your data is replicated in the primary region:

- **Locally redundant storage (LRS)** copies your data synchronously three times within a single physical location in the primary region. LRS is the least expensive replication option, but isn't recommended for applications requiring high availability or durability.
    - LRS provides at least 99.999999999% (11 nines) durability of objects over a given year.
    - LRS protects your data against server rack and drive failures. However, if a disaster such as fire or flooding occurs within the data center, all replicas of a storage account using LRS may be lost or unrecoverable.
    - To mitigate this risk, Microsoft recommends using zone-redundant storage (ZRS), geo-redundant storage (GRS), or geo-zone-redundant storage (GZRS).
    - If your application stores data that can be easily reconstructed if data loss occurs, you may opt for LRS.
    - If your application is restricted to replicating data only within a country or region due to data governance requirements, you may opt for LRS.
    - If your scenario is using Azure unmanaged disks, you may opt for LRS. While it's possible to create a storage account for Azure unmanaged disks that uses GRS, it isn't recommended due to potential issues with consistency over asynchronous geo-replication.

- **Zone-redundant storage (ZRS)** copies your data synchronously across three Azure availability zones in the primary region. For applications requiring high availability, Microsoft recommends using ZRS in the primary region, and also replicating to a secondary region.
    - ZRS offers durability for storage resources of at least 99.9999999999% (12 9's) over a given year.
    - With ZRS, your data is still accessible for both read and write operations even if a zone becomes unavailable.
    - If a zone becomes unavailable, Azure undertakes networking updates, such as DNS repointing.
    - A write request to a storage account that is using ZRS happens synchronously. The write operation returns successfully only after the data is written to all replicas across the three availability zones. If an availability zone is temporarily unavailable, the operation returns successfully after the data is written to all available zones.
    - Microsoft recommends using ZRS in the primary region for scenarios that require high availability. ZRS is also recommended for restricting replication of data to a particular country or region to meet data governance requirements.
    - ZRS provides excellent performance, low latency, and resiliency for your data if it becomes temporarily unavailable.
    - ZRS by itself **may not protect your data against a regional disaster** where multiple zones are permanently affected.
    - For protection against regional disasters, Microsoft recommends using geo-zone-redundant storage (GZRS), which uses ZRS in the primary region and also geo-replicates your data to a secondary region.

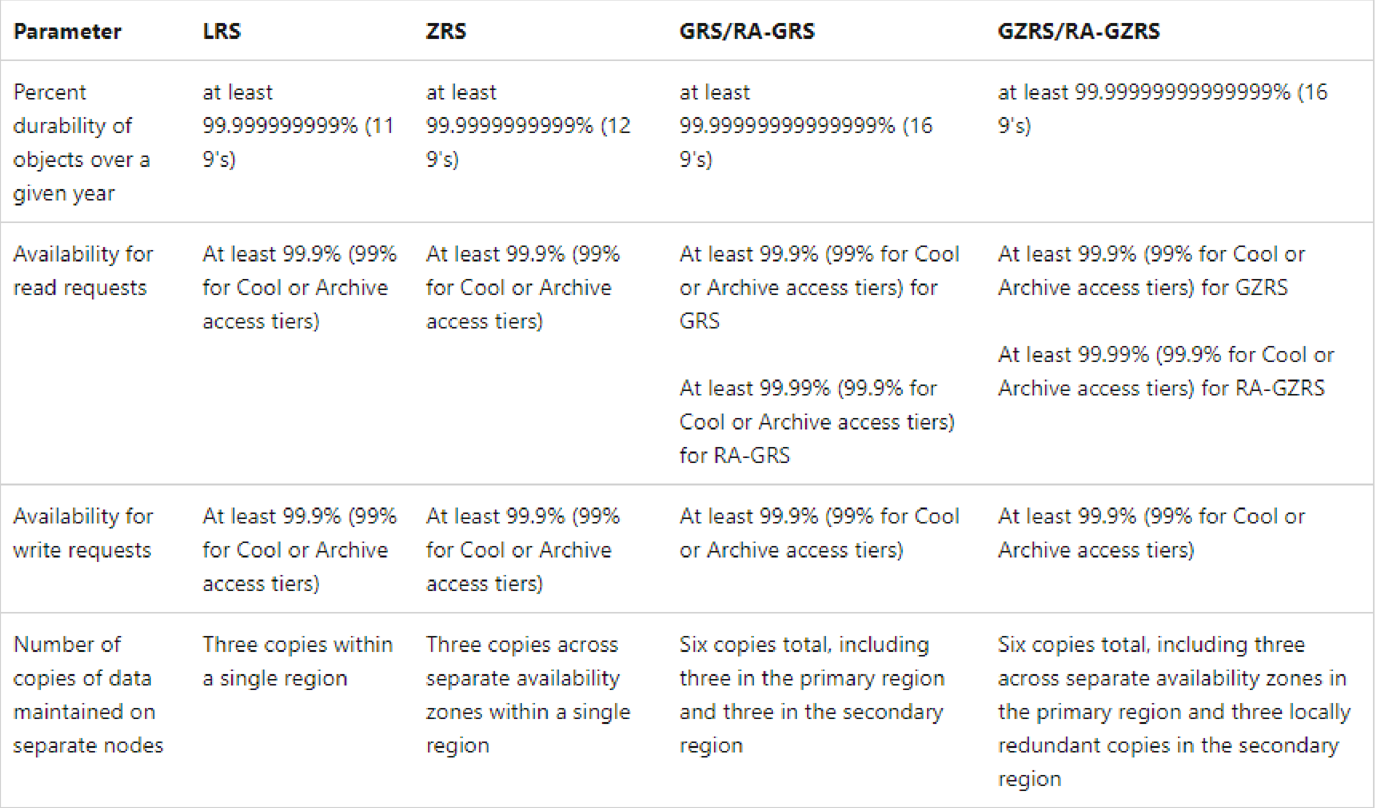
Redundancy in a secondary region

- For applications requiring high durability, you can choose to additionally copy the data in your storage account to a secondary region that is hundreds of miles away from the primary region.
- If your storage account is copied to a secondary region, then **your data is durable even in the case of a complete regional outage or a disaster in which the primary region isn't recoverable**.
- A write operation is first committed to the primary location and replicated using Locally redundant storage (GRS) or Zone-redundant storage (GZRS). The update is then replicated asynchronously to the secondary region. When data is written to the secondary location, it's also replicated within that location using LRS.
- GRS and GZRS are designed to provide at least 99.99999999999999% (16 9's) durability of objects over a given year.
- **Geo-redundant storage (GRS)** copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.
- **Geo-zone-redundant storage (GZRS)** copies your data synchronously across three Azure availability zones in the primary region using ZRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.
    - With a GZRS storage account, you can continue to read and write data if an availability zone becomes unavailable or is unrecoverable.
    - Additionally, your data is also durable in the case of a complete regional outage or a disaster in which the primary region isn't recoverable.
    - This option is recommended by Microsoft for applications requiring maximum consistency, durability, and availability, excellent performance, and resilience for disaster recovery.
- With GRS or GZRS, the data in the secondary region isn't available for read or write access unless there's a failover to the primary region.
- For read access to the secondary region, configure your storage account to use read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS).

Read access to data in the secondary region

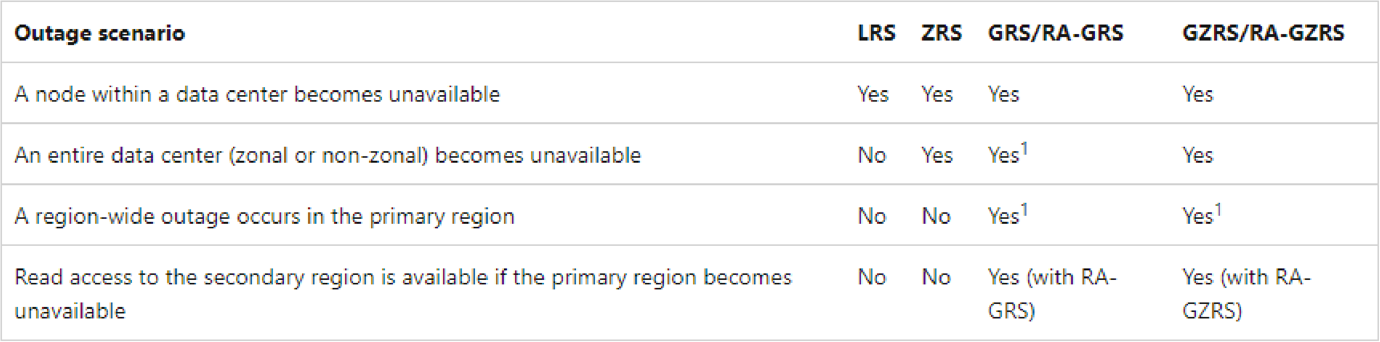
- Geo-redundant storage (with GRS or GZRS) replicates your data to another physical location in the secondary region to protect against regional outages.
- With an account configured for GRS or GZRS, data in the secondary region is not directly accessible to users or applications, unless a failover occurs.
- The failover process updates the DNS entry provided by Azure Storage so that the secondary endpoint becomes the new primary endpoint for your storage account.
- During the failover process, your data is inaccessible.
- After the failover is complete, you can read and write data to the new primary region.
- If your applications require high availability, then you can configure your storage account for read access to the secondary region.
- When you enable read access to the secondary region, then your data is always available to be read from the secondary, including in a situation where the primary region becomes unavailable.
- Read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS) configurations permit read access to the secondary region.
- When read access to the secondary is enabled, your application can be read from the secondary endpoint as well as from the primary endpoint. The secondary endpoint appends the suffix –secondary to the account name. For example, if your primary endpoint for Blob storage is myaccount.blob.core.windows.net, then the secondary endpoint is myaccount-secondary.blob.core.windows.net. The account access keys for your storage account are the same for both the primary and secondary endpoints.
- Because data is replicated asynchronously from the primary to the secondary region, the secondary region is typically behind the primary region in terms of write operations. If a disaster were to strike the primary region, it's likely that some data would be lost.
- Azure Files does not support read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS).

Durability and availability by outage scenario



Parameter	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
Percent durability of objects over a given year	at least 99.999999999% (11 9's)	at least 99.9999999999% (12 9's)	at least 99.99999999999999% (16 9's)	at least 99.99999999999999% (16 9's)
Availability for read requests	At least 99.9% (99% for Cool or Archive access tiers)	At least 99.9% (99% for Cool or Archive access tiers)	At least 99.9% (99% for Cool or Archive access tiers) for GRS At least 99.99% (99.9% for Cool or Archive access tiers) for RA-GRS	At least 99.9% (99% for Cool or Archive access tiers) for GZRS At least 99.99% (99.9% for Cool or Archive access tiers) for RA-GZRS
Availability for write requests	At least 99.9% (99% for Cool or Archive access tiers)	At least 99.9% (99% for Cool or Archive access tiers)	At least 99.9% (99% for Cool or Archive access tiers)	At least 99.9% (99% for Cool or Archive access tiers)
Number of copies of data maintained on separate nodes	Three copies within a single region	Three copies across separate availability zones within a single region	Six copies total, including three in the primary region and three in the secondary region	Six copies total, including three across separate availability zones in the primary region and three locally redundant copies in the secondary region

Durability and availability by outage scenario



Outage scenario	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
A node within a data center becomes unavailable	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes[1]	Yes
A region-wide outage occurs in the primary region	No	No	Yes[1]	Yes[1]
Read access to the secondary region is available if the primary region becomes unavailable	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)

Azure Monitor

Azure Monitor Logs lets you collect and organize data from resources that you monitor. You configure what data is gathered and how it's organized in the platform. Other features in Azure Monitor automatically store their data in Logs. You can use the stored data with your collected data to help monitor the performance of your environment.

Azure Monitor Metrics captures numerical data from your monitored resources and stores the results in a time-organized database. Metrics are collected at intervals you specify. You can use metrics to check how your system is performing at a particular time or under certain circumstances.

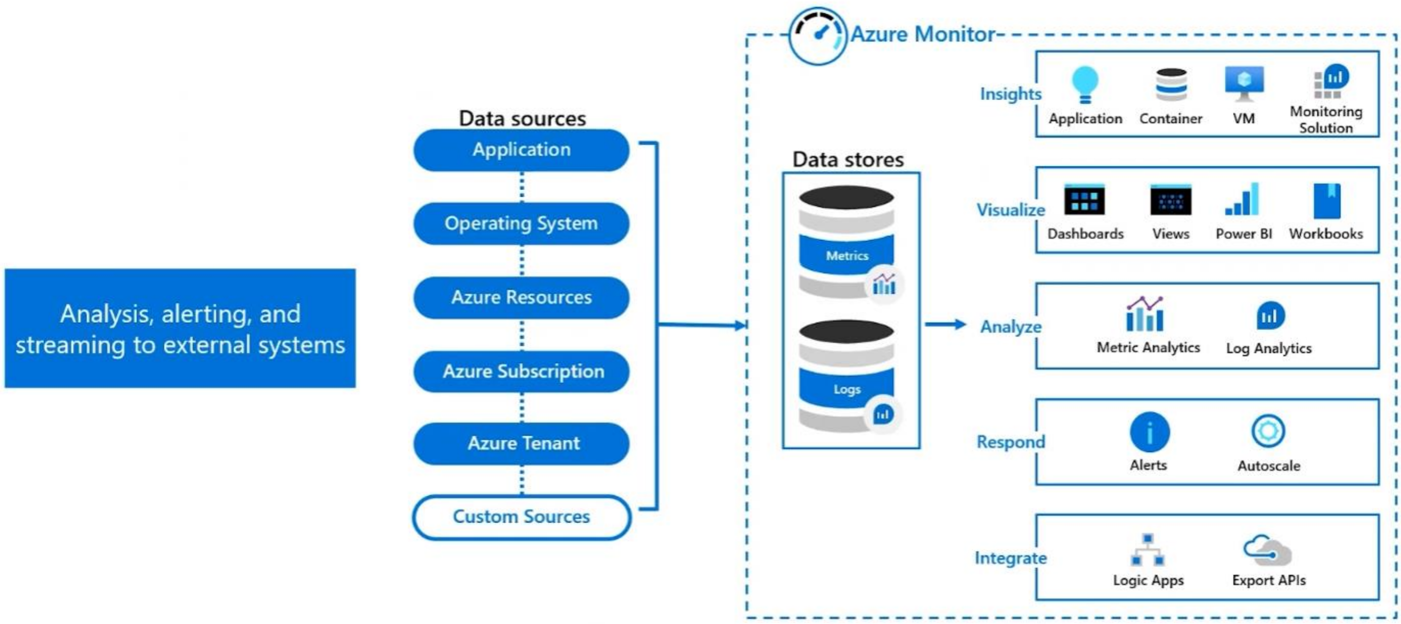
Things to know about Azure Monitor

- Data from multiple resources can be collected into Azure Monitor and analyzed together by using a common set of tools.
- Logs enable complex analysis by using log queries.
- Metrics support near-real-time scenarios like priority alerts and responding to critical issues.
- Monitoring data can be sent to other locations to support certain scenarios, such as tracking and reporting.
- Sources of monitoring data from Azure applications can be organized into tiers, and each tier can be accessed in different ways.
  - The highest tiers are for your application itself.
  - The lower tiers are components of the Azure platform.

Things to consider when using Azure Monitor

- **Consider data sources and data access**. Identify what Tailwind Traders resources to monitor. Consider how data from these resources is accessed by other resources or applications. Azure Monitor collects data automatically from a range of components, and the data is accessed in various ways.
- **Consider queries on Logs data**. Write log queries to analyze your collected data for Tailwind Traders. For more information about log queries, see Log queries in Azure Monitor.
- **Consider alerts based on Logs and Metrics data**. Set up alert rules based on Logs data to be proactively notified about system issues. Use Metrics data to identify when critical Tailwind Traders issues occur, such as values that exceed defined limits.
- **Consider Metrics Explorer to analyze metrics interactively**. Define metrics to monitor about your Tailwind Traders resources, such as peak usage rates, access information, workloads, or incident scenarios. Use the Metrics Explorer to investigate the collected data.

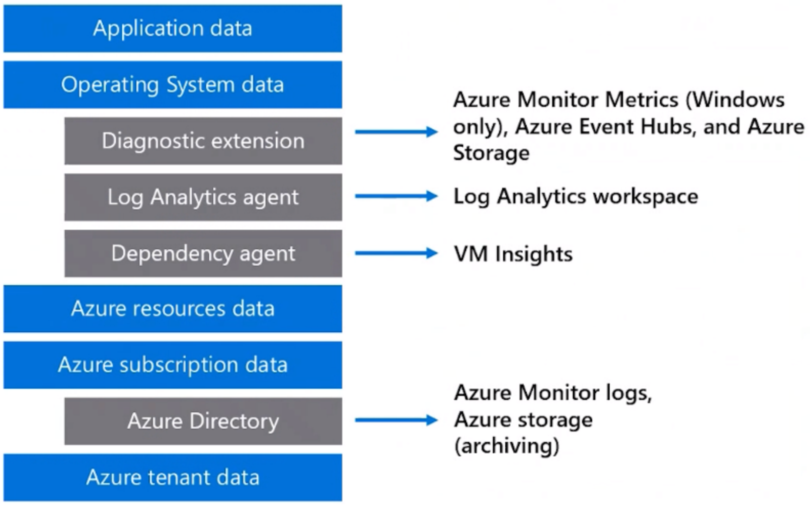
Review Azure Monitor Capabilities



Identify data sources and access method

Azure Monitor collects data automatically from a range of components.

- Data tiers go from Azure applications (highest tier) to Azure platform components (lowest tier).
- The method of accessing data from each tier varies – for example, installing an agent.
- Each data tier can stream to different external systems.
- Prioritize and be deliberate on what data sources you need.



Application data

Operating System data

Diagnostic extension → Azure Monitor Metrics (Windows only), Azure Event Hubs, and Azure Storage

Log Analytics agent → Log Analytics workspace

Dependency agent → VM Insights

Azure resources data

Azure subscription data

Azure Directory → Azure Monitor logs, Azure storage (archiving)

Azure tenant data

Azure Monitor Logs (Log Analytics)

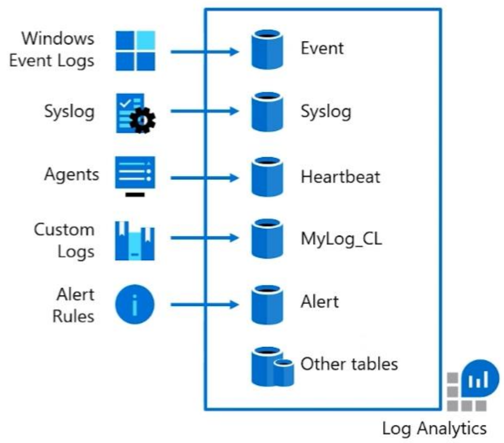
What is Azure Monitor Logs (Log Analytics)?

Log Analytics is a service that helps you collect and analyze data.

- Azure Monitor stores log data in the workspace
- Data in a workspace is organized into tables with properties you can query.

A Log Analytics workspace provides:

- A geographic location for data storage.
- Data isolation by granting different users access rights following one of our recommended design strategies.
- Scope for configuration of settings like pricing tier, retention, and data capping.



Workspace deployment models

Centralized:

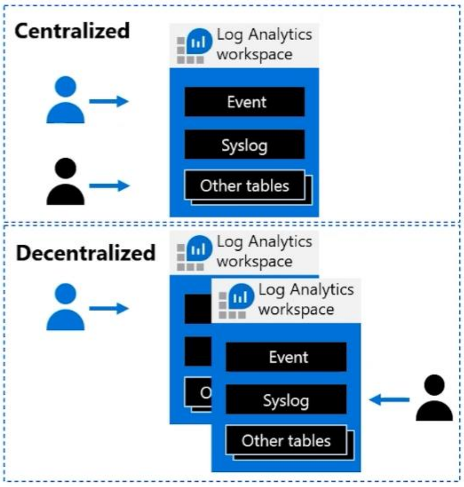
All logs are stored in a central workspace and administered by a single team, with Azure Monitor providing differentiated access per-team.

Decentralized:

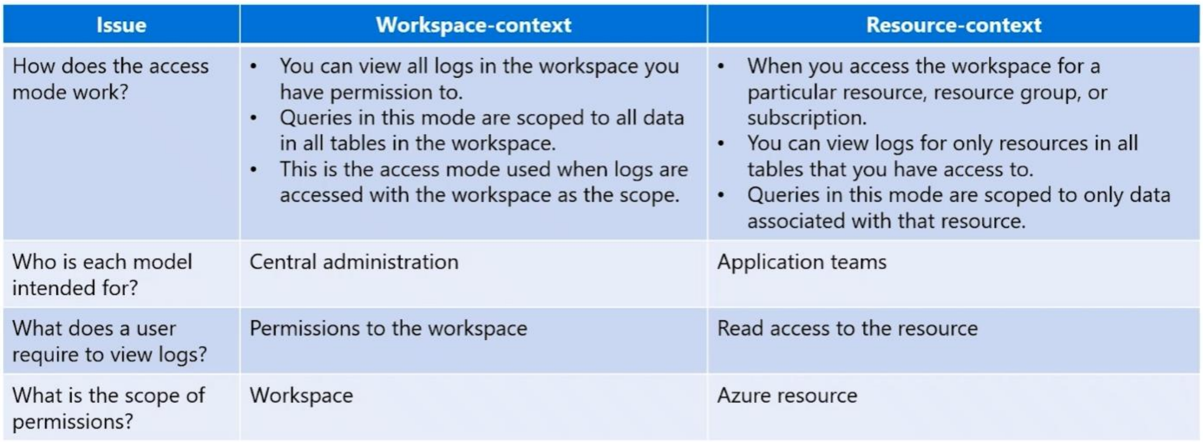
Each team has their own workspace created in a resource group they own and manage, and log data is segregated per resource.

Hybrid:

Security audit compliance requirements further complicate this scenario because many organizations implement both deployment models in parallel.



Azure Monitor Logs workspace access mode



Issue	Workspace-context	Resource-context
How does the access mode work?	• You can view all logs in the workspace you have permission to. • Queries in this mode are scoped to all data in all tables in the workspace. • This is the access mode used when logs are accessed with the workspace as the scope.	• When you access the workspace for a particular resource, resource group, or subscription. • You can view logs for only resources in all tables that you have access to. • Queries in this mode are scoped to only data associated with that resource.
Who is each model intended for?	Central administration	Application teams
What does a user require to view logs?	Permissions to the workspace	Read access to the resource
What is the scope of permissions?	Workspace	Azure resource

Use Case Scenarios

Azure Monitor

- You need to solution for collecting, analyzing, and acting on telemetry from your cloud and on premises environments.

Application Insights

- You need to measure user experience and analyze users' behavior for all external facing applications.
- You need transaction diagnostics and performance statistics (client and server).
- You need usage information on request rates, response times, and failure rates of an application.

Azure Workbooks

- You need data analysis and the creation of rich visual reports.

Azure Monitor Logs (Log Analytics)

- You need to edit and run log queries.

Azure Workbooks

Azure Workbooks is a feature of Azure Monitor. Workbooks provide a flexible canvas for data analysis and the creation of rich visual reports within the Azure portal.

The real power of Workbooks is the ability to combine data from disparate sources within a single report. You can create composite resource views or joins across resources enabling richer data and insights that would otherwise be impossible.

Things to know about Azure Workbooks

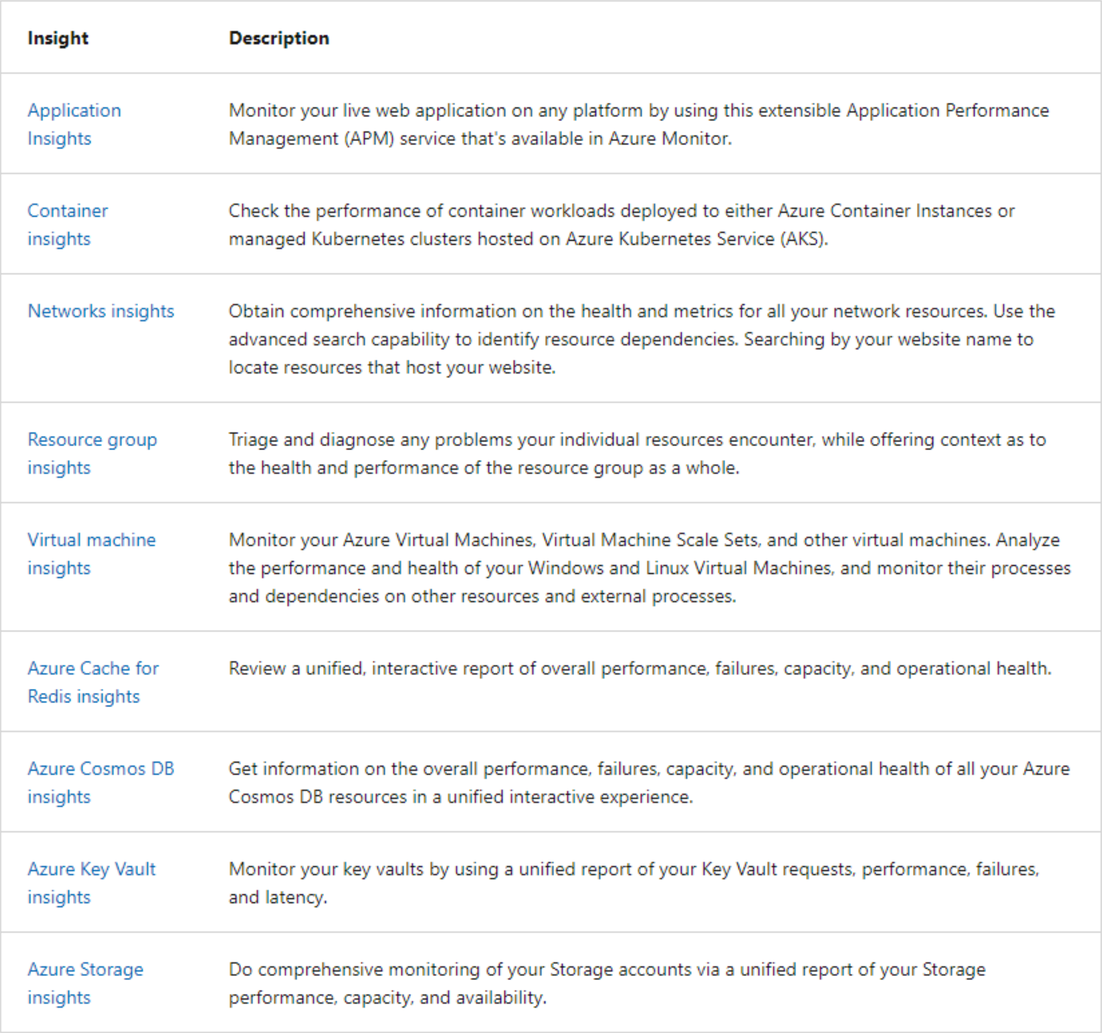
- Azure Workbooks lets you tap into multiple data sources from across Azure and combine them into unified interactive experiences.
- Authors of workbooks can transform ingested data to provide insights into the availability, performance, usage, and overall health of the underlying components.
- You can analyze performance logs from virtual machines to identify high CPU or low memory instances and display the results as a grid in an interactive report.
- Workbooks are currently compatible with the following data sources:
  - Logs
  - Metrics
  - Azure Resource Graph

- Alerts
- Workload Health
- Azure Resource Health
- Azure Data Explorer

Azure insights

Azure insights can help you identify performance issues in the Tailwind Traders architecture. Consider these characteristics about insights:

- Azure insights provide a customized monitoring experience for particular applications and services.
- Azure insights collect and analyze both logs and metrics.
- Many insights are provided as features of Azure Monitor.



Insight	Description
Application Insights	Monitor your live web application on any platform by using this extensible Application Performance Management (APM) service that's available in Azure Monitor.
Container insights	Check the performance of container workloads deployed to either Azure Container Instances or managed Kubernetes clusters hosted on Azure Kubernetes Service (AKS).
Networks insights	Obtain comprehensive information on the health and metrics for all your network resources. Use the advanced search capability to identify resource dependencies. Searching by your website name to locate resources that host your website.
Resource group insights	Triage and diagnose any problems your individual resources encounter, while offering context as to the health and performance of the resource group as a whole.
Virtual machine insights	Monitor your Azure Virtual Machines, Virtual Machine Scale Sets, and other virtual machines. Analyze the performance and health of your Windows and Linux Virtual Machines, and monitor their processes and dependencies on other resources and external processes.
Azure Cache for Redis insights	Review a unified, interactive report of overall performance, failures, capacity, and operational health.
Azure Cosmos DB insights	Get information on the overall performance, failures, capacity, and operational health of all your Azure Cosmos DB resources in a unified interactive experience.
Azure Key Vault insights	Monitor your key vaults by using a unified report of your Key Vault requests, performance, failures, and latency.
Azure Storage insights	Do comprehensive monitoring of your Storage accounts via a unified report of your Storage performance, capacity, and availability.

Things to consider when using Azure insights and Workbooks

- **Consider Azure Workbooks**. Explore how Tailwind Traders apps can be used with Azure Workbooks. Investigate the root cause analysis of incidents, and put together an operational playbook for your team.
- **Consider Azure insights and data analysis**. Include Azure insights for a custom monitoring experience for Tailwind Traders apps and services. Review insights about your network, VMs, and other Azure resources. Collect Logs and Metrics data from Workbooks and analyze the data.
- **Consider combined data sources and visual reporting**. Combine data from Tailwind Traders sources in a single report. Create composite resource views for more robust data and greater insights. Prepare rich visual reports within the Azure portal.

Azure Data Explorer

Azure Data Explorer is a platform for big data that helps you analyze high volumes of data in near real time. Azure Data Explorer comes equipped with features to help you configure an end-to-end solution for ingesting and managing your data, running queries, and generating visualizations.

Things to know about Azure Data Explorer

- Azure Data Explorer is a fast and highly scalable data exploration service for log and telemetry data.
- Azure Data Explorer helps you handle multiple data streams, so you can collect, store, and analyze your data from all resources.
- Analyze large volumes of diverse data from any data source, such as websites, applications, IoT devices, and more.
- Use Azure Data Explorer for diagnostics, monitoring, reporting, machine learning, and other analytics tasks.

Things to consider when using Azure Data Explorer

- Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution.
- Microsoft Sentinel and Azure Monitor SaaS solutions don't offer out-of-the-box support for certain scenarios like application trace logs. You can use Azure Data Explorer to provide monitoring support for all aspects and for all types of logs for Tailwind Traders.

Azure high availability and disaster recovery features for VMs

Availability sets

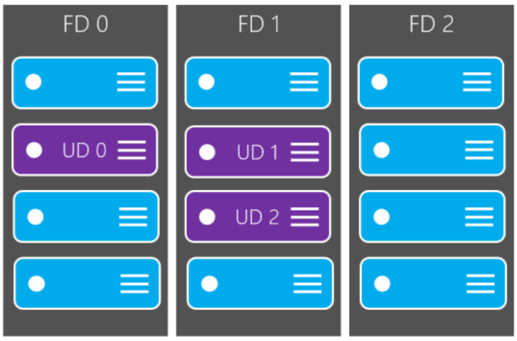
Availability sets provide uptime against Azure-related maintenance and single points of failure in a single data center.

This was one of the first availability features introduced into the Azure platform, and effectively it can be thought of as anti-affinity rules for your VMs. This means if you had two SQL Server VMs in an availability set or log shipping pair, they would be guaranteed to never run on the same physical server.

Availability sets are separated into both fault domains and update domains to support both updates to the underlying Azure Infrastructure.

Fault domains are sets of servers within a data center, which use the same power source and network There can be up to three fault domains in a data center as depicted in the image below by FD 0, 1, and 2.

Update domains, denoted by UD in the image below, indicate groups of virtual machines and underlying physical hardware that can be rebooted at the same time. Different update domains ensure separation.



Availability sets and zones don't protect against in-guest failures, such as an OS or RDBMS crash; which is why you need to implement additional solutions such as AGs or FCIs to ensure you meet RTOs and RPOs.

Both availability sets and zones are designed to limit the impact of environmental problems at the Azure level such as datacenter failure, physical hardware failure, network outages, and power interruptions.

For a multi-tier application, you should put each tier of the application into its own availability set. For example, if you were building a web application that has a SQL Server backend along with Active Directory Domain Services (AD DS), you would create an availability set for each tier (web, database, and AD DS).

Availability sets are not the only way to separate IaaS VMs. Azure also provides Availability Zones, but the two can't be combined. You can pick one or the other.

Availability zones

Availability zones account for data center-level failure in Azure. Each Azure region consists of many data centers with low latency network connections between them.

When you deploy VM resources in a region that supports Availability Zones, you have the option to deploy those resources into Zone 1, 2, or 3. A zone is a unique physical location, that is, a data center, within an Azure region.

Zone numbers are logical representations. For example, if two Azure subscribers both deploy a VM into Zone 1 in their own subscriptions, that doesn't mean those VMs exist in the same physical Azure data center.

Additionally, because of the distance there can be some additional latency introduced into zonal deployments. You should test the latency between your VMs to ensure that the latency meets performance targets. In most cases round-trip latency will be less than 1 millisecond, which supports synchronous data movement in features like availability groups.

You can also deploy Azure SQL Database into Availability Zones.

Azure Site Recovery

Azure Site Recovery provides enhanced availability for VMs at the Azure level and can work with VMs hosting SQL Server.

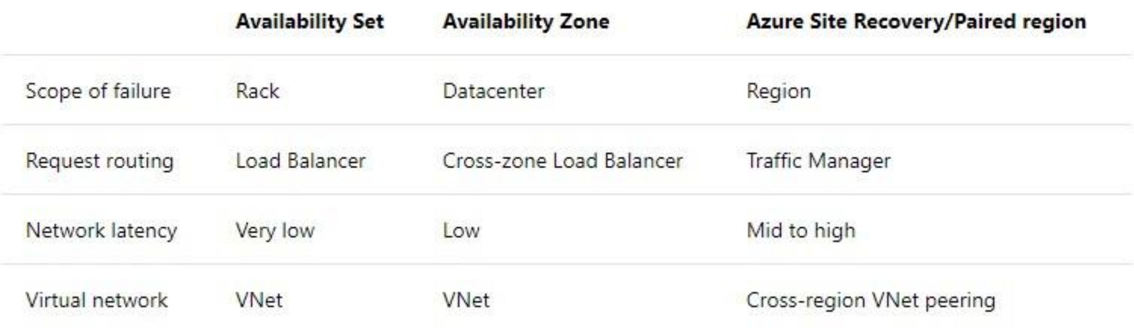
Azure Site Recovery replicates a VM from one Azure region to another to create a disaster recovery solution for that VM.

As noted earlier, this feature does not know that SQL Server is running in the VM and knows nothing about transactions.

While Azure Site Recovery may meet RTO, it may not meet RPO since it isn't accounting for where data is inside SQL Server.

Azure Site Recovery has a stated monthly RTO of two hours.

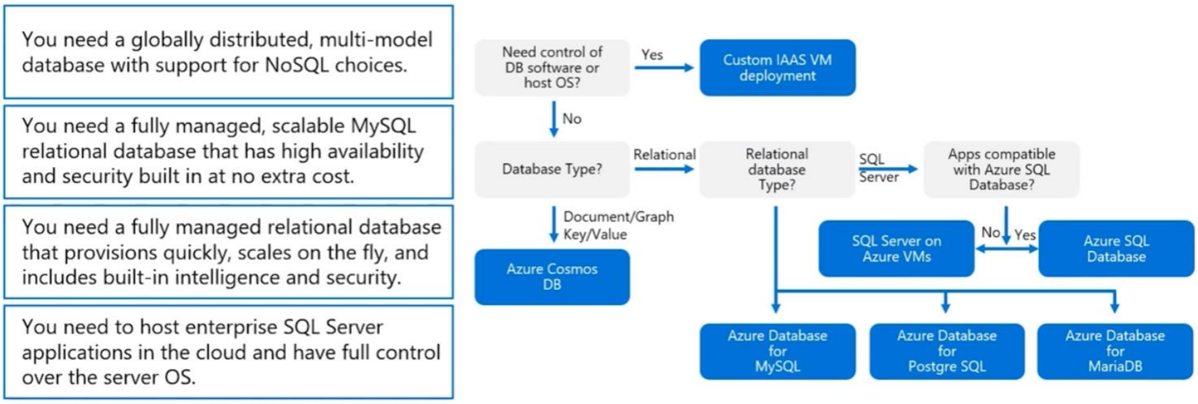
While most database professionals may prefer to use a database-based method for disaster recovery, Azure Site Recovery works well if it meets your RTO and RPO needs.



	Availability Set	Availability Zone	Azure Site Recovery/Paired region
Scope of failure	Rack	Datacenter	Region
Request routing	Load Balancer	Cross-zone Load Balancer	Traffic Manager
Network latency	Very low	Low	Mid to high
Virtual network	VNet	VNet	Cross-region VNet peering

Design a data storage solution for relational data

Select a structures data product (matching)



You need a globally distributed, multi-model database with support for NoSQL choices.

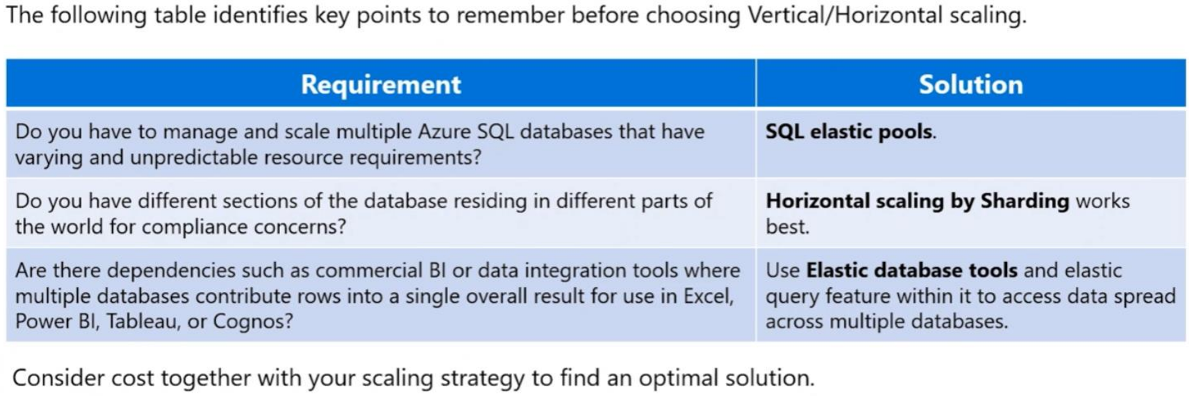
You need a fully managed, scalable MySQL relational database that has high availability and security built in at no extra cost.

You need a fully managed relational database that provisions quickly, scales on the fly, and includes built-in intelligence and security.

You need to host enterprise SQL Server applications in the cloud and have full control over the server OS.

Need control of DB software or host OS?
Yes
Custom IAAS VM deployment
No
Database Type?
Relational
Relational database Type?
SQL Server
Apps compatible with Azure SQL Database?
Document/Graph Key/Value
Azure Cosmos DB
SQL Server on Azure VMs
No
Yes
Azure SQL Database
Azure Database for MySQL
Azure Database for Postgre SQL
Azure Database for MariaDB

Database scaling strategy

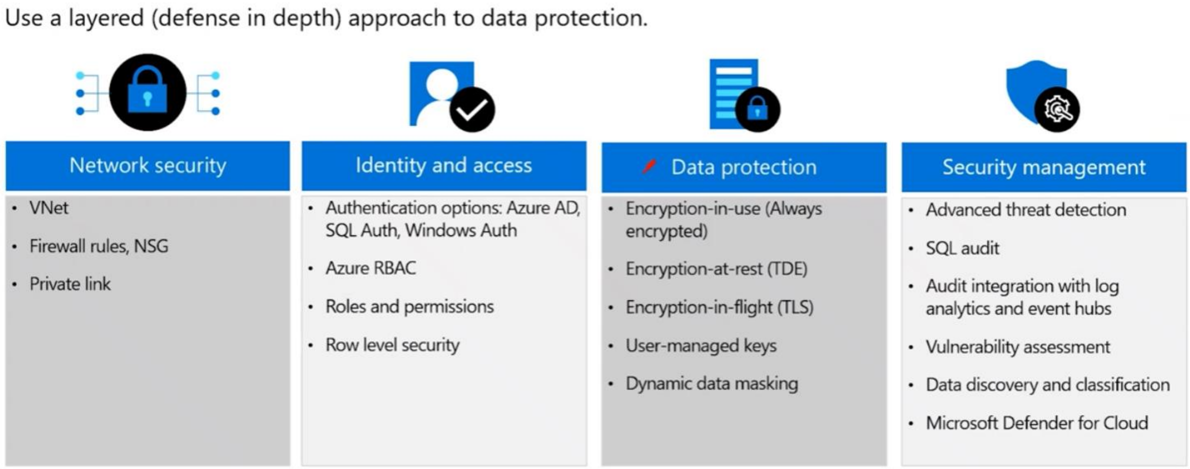


The following table identifies key points to remember before choosing Vertical/Horizontal scaling.

Requirement	Solution
Do you have to manage and scale multiple Azure SQL databases that have varying and unpredictable resource requirements?	**SQL elastic pools**.
Do you have different sections of the database residing in different parts of the world for compliance concerns?	**Horizontal scaling by Sharding** works best.
Are there dependencies such as commercial BI or data integration tools where multiple databases contribute rows into a single overall result for use in Excel, Power BI, Tableau, or Cognos?	Use **Elastic database tools** and elastic query feature within it to access data spread across multiple databases.

Consider cost together with your scaling strategy to find an optimal solution.

Protect your database



Use a layered (defense in depth) approach to data protection.

Network security

- VNet
- Firewall rules, NSG
- Private link

Identity and access

- Authentication options: Azure AD, SQL Auth, Windows Auth
- Azure RBAC
- Roles and permissions
- Row level security

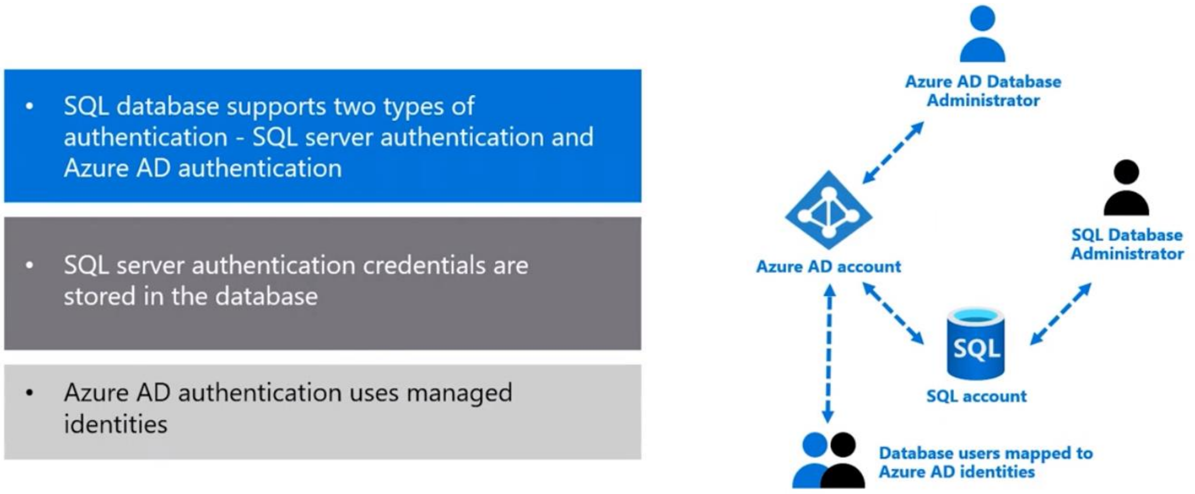
Data protection

- Encryption-in-use (Always encrypted)
- Encryption-at-rest (TDE)
- Encryption-in-flight (TLS)
- User-managed keys
- Dynamic data masking

Security management

- Advanced threat detection
- SQL audit
- Audit integration with log analytics and event hubs
- Vulnerability assessment
- Data discovery and classification
- Microsoft Defender for Cloud

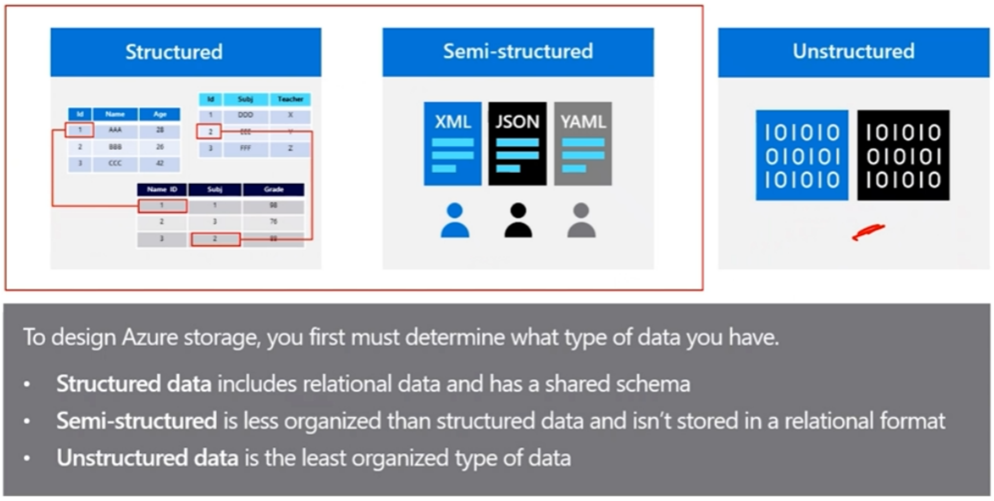
Authenticate to an Azure SQL Database



- SQL database supports two types of authentication - SQL server authentication and Azure AD authentication
- SQL server authentication credentials are stored in the database
- Azure AD authentication uses managed identities

Azure AD Database Administrator
Azure AD account
SQL Database Administrator
SQL
SQL account
Database users mapped to Azure AD identities

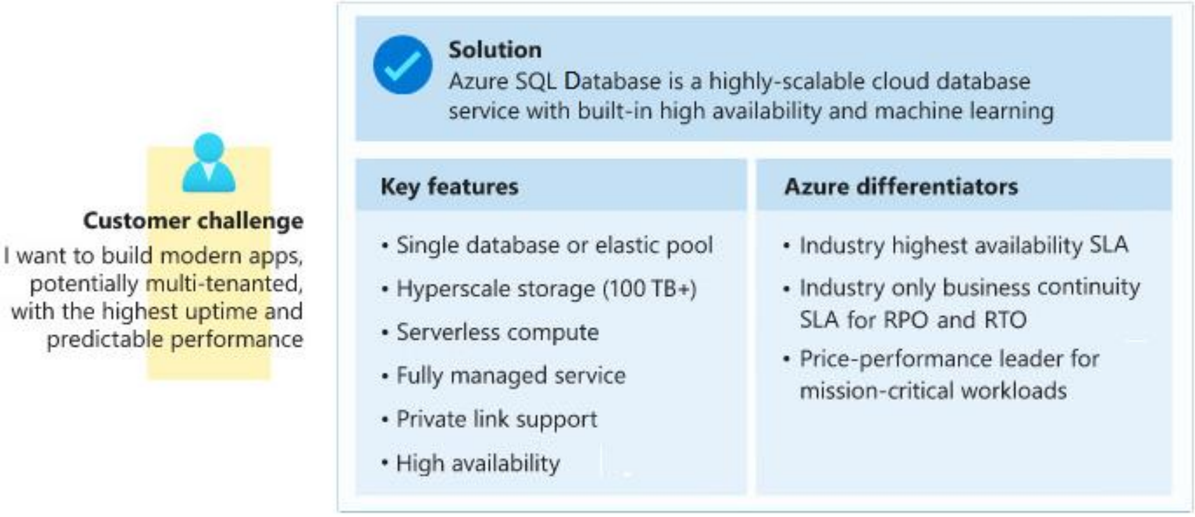
Design for structured and semi-structured data



To design Azure storage, you first must determine what type of data you have.

- **Structured data** includes relational data and has a shared schema
- **Semi-structured** is less organized than structured data and isn't stored in a relational format
- **Unstructured data** is the least organized type of data

Azure SQL Database



Customer challenge
I want to build modern apps, potentially multi-tenanted, with the highest uptime and predictable performance

Solution
Azure SQL Database is a highly-scalable cloud database service with built-in high availability and machine learning

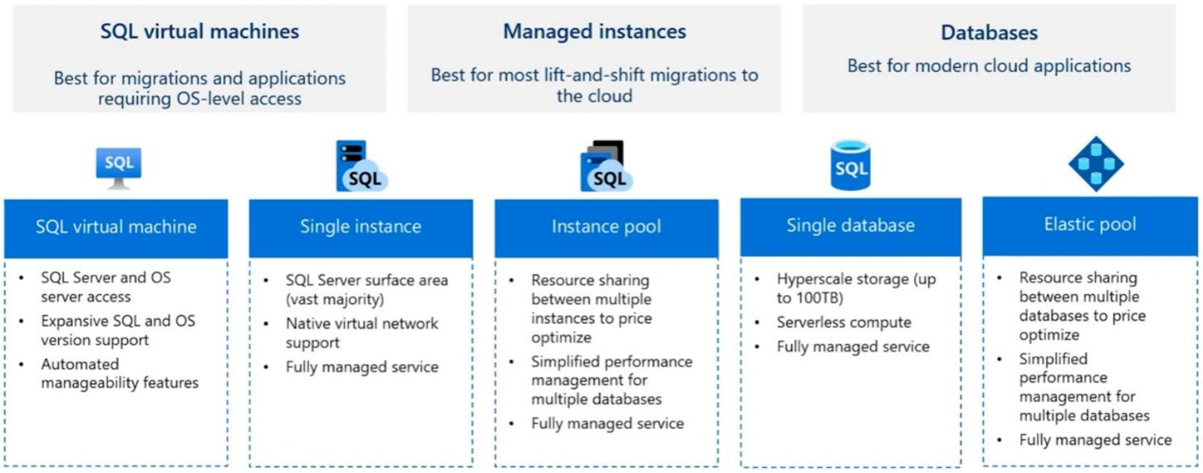
Key features

- Single database or elastic pool
- Hyperscale storage (100 TB+)
- Serverless compute
- Fully managed service
- Private link support
- High availability

Azure differentiators

- Industry highest availability SLA
- Industry only business continuity SLA for RPO and RTO
- Price-performance leader for mission-critical workloads

When to use Azure SQL Databases



SQL virtual machines

Best for migrations and applications requiring OS-level access

Managed instances

Best for most lift-and-shift migrations to the cloud

Databases

Best for modern cloud applications

SQL virtual machine

- SQL Server and OS server access
- Expansive SQL and OS version support
- Automated manageability features

Single instance

- SQL Server surface area (vast majority)
- Native virtual network support
- Fully managed service

Instance pool

- Resource sharing between multiple instances to price optimize
- Simplified performance management for multiple databases
- Fully managed service

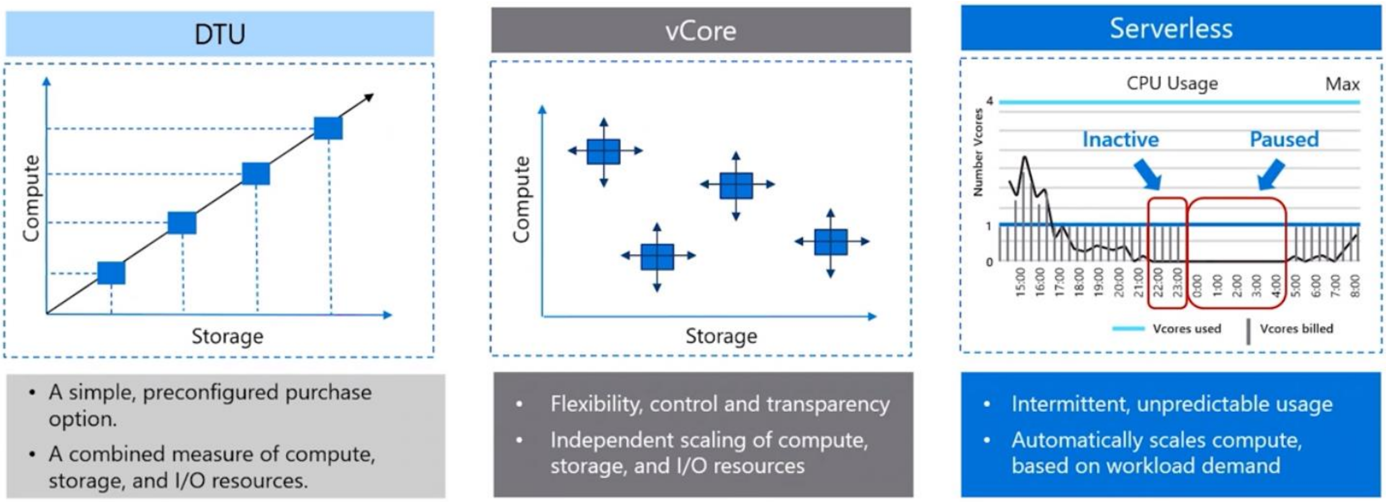
Single database

- Hyperscale storage (up to 100TB)
- Serverless compute
- Fully managed service

Elastic pool

- Resource sharing between multiple databases to price optimize
- Simplified performance management for multiple databases
- Fully managed service

Select and Azure SQL Database pricing model



DTU

- A simple, preconfigured purchase option.
- A combined measure of compute, storage, and I/O resources.

vCore

- Flexibility, control and transparency
- Independent scaling of compute, storage, and I/O resources

Serverless

- Intermittent, unpredictable usage
- Automatically scales compute, based on workload demand

Elastic Database Pools

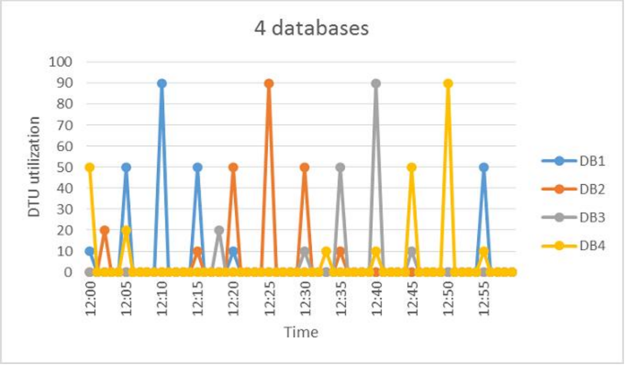
Set of compute and storage resources to share among all SQL databases in an elastic pool.

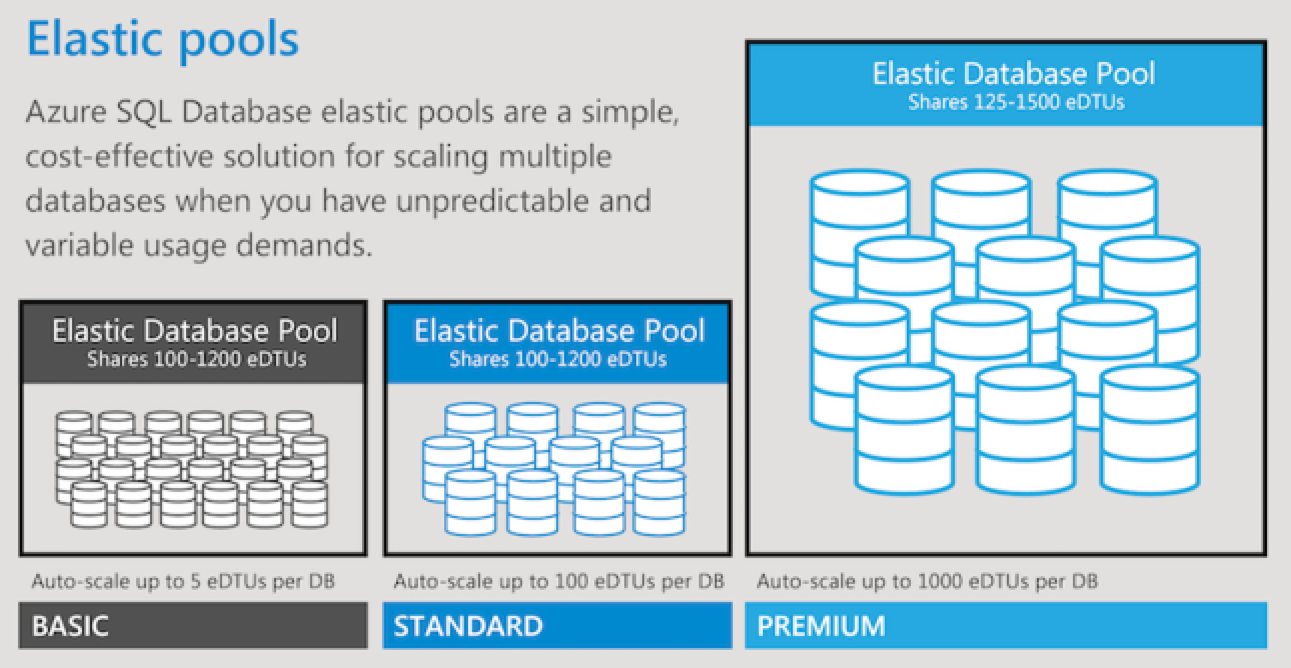
Pools are well suited for a large number of databases with specific utilization patterns. For a given database, this pattern is characterized by low average utilization with infrequent utilization spikes.

Conversely, multiple databases with persistent medium-high utilization shouldn't be placed in the same elastic pool.

The more databases you can add to a pool, the greater your savings become, because you pay by the pool’s capacity, not number of databases like single instance.

The peak utilization for each database occurs at different points in time.

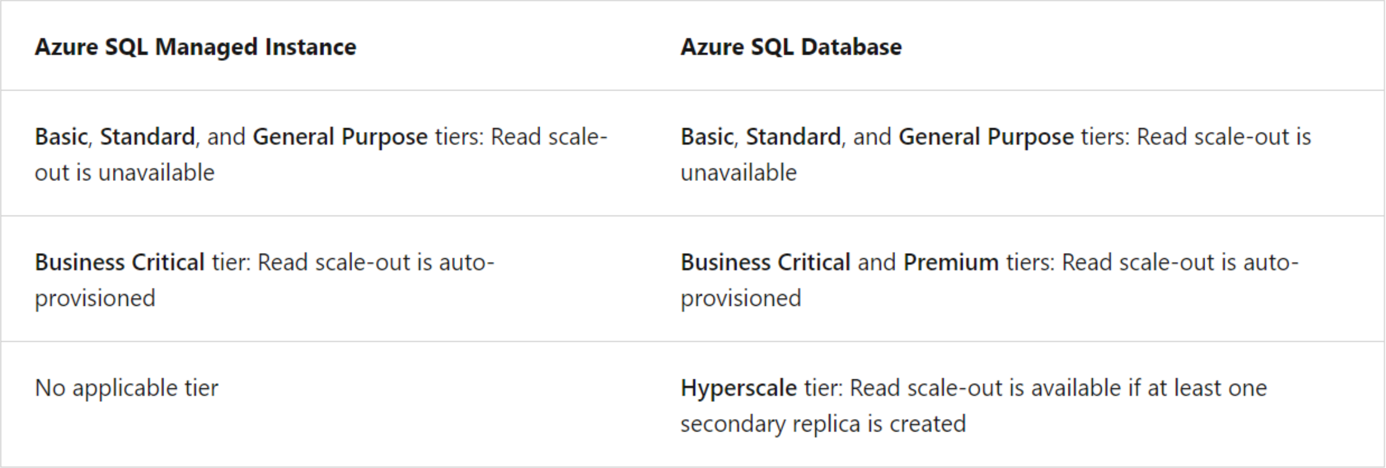




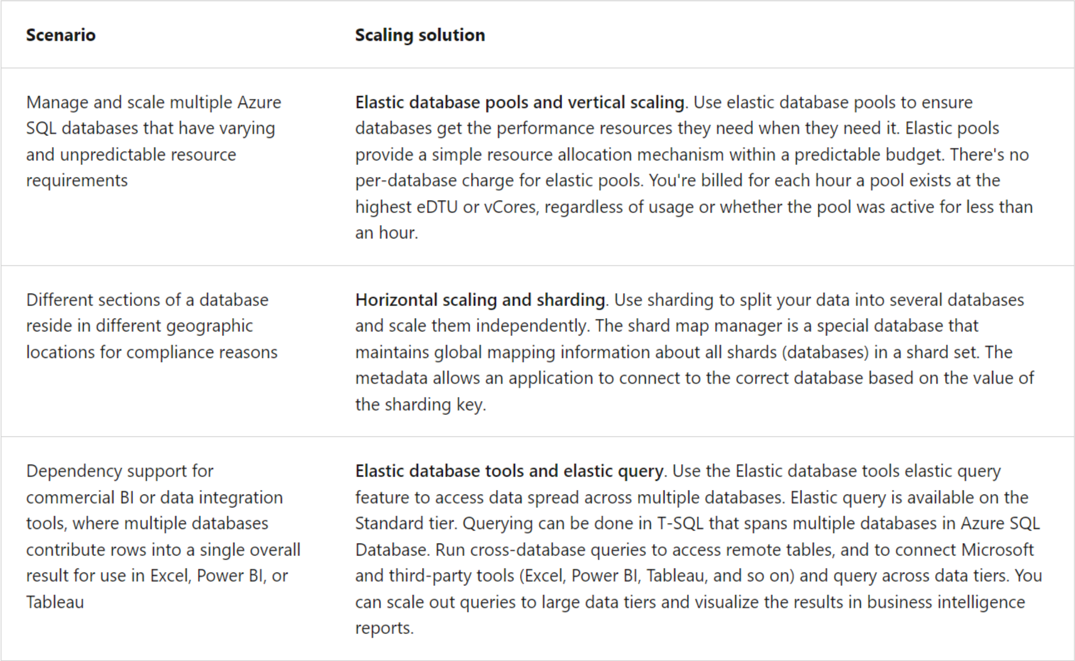
Elastic pools

Azure SQL Database elastic pools are a simple, cost-effective solution for scaling multiple databases when you have unpredictable and variable usage demands.

Horizontal Scaling



Azure SQL Managed Instance	Azure SQL Database
Basic, **Standard**, and **General Purpose** tiers: Read scale-out is unavailable	**Basic**, **Standard**, and **General Purpose** tiers: Read scale-out is unavailable
Business Critical tier: Read scale-out is auto-provisioned	**Business Critical** and **Premium** tiers: Read scale-out is auto-provisioned
No applicable tier	**Hyperscale** tier: Read scale-out is available if at least one secondary replica is created



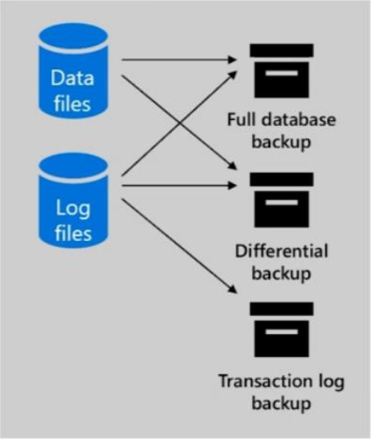
Scenario	Scaling solution
Manage and scale multiple Azure SQL databases that have varying and unpredictable resource requirements	**Elastic database pools and vertical scaling**. Use elastic database pools to ensure databases get the performance resources they need when they need it. Elastic pools provide a simple resource allocation mechanism within a predictable budget. There's no per-database charge for elastic pools. You're billed for each hour a pool exists at the highest eDTU or vCores, regardless of usage or whether the pool was active for less than an hour.
Different sections of a database reside in different geographic locations for compliance reasons	**Horizontal scaling and sharding**. Use sharding to split your data into several databases and scale them independently. The shard map manager is a special database that maintains global mapping information about all shards (databases) in a shard set. The metadata allows an application to connect to the correct database based on the value of the sharding key.
Dependency support for commercial BI or data integration tools, where multiple databases contribute rows into a single overall result for use in Excel, Power BI, or Tableau	**Elastic database tools and elastic query**. Use the Elastic database tools elastic query feature to access data spread across multiple databases. Elastic query is available on the Standard tier. Querying can be done in T-SQL that spans multiple databases in Azure SQL Database. Run cross-database queries to access remote tables, and to connect Microsoft and third-party tools (Excel, Power BI, Tableau, and so on) and query across data tiers. You can scale out queries to large data tiers and visualize the results in business intelligence reports.

How Azure SQL backup works

SQL Database and SQL Managed Instances automatically backup.

Full backups once a week

- Full backups once a week
- Differential backups every 12-24 hours
- Transactional log backups every 5-10 minutes



Considerations for Azure SQL Backup

Restore in the retention period or use a long-term retention policy

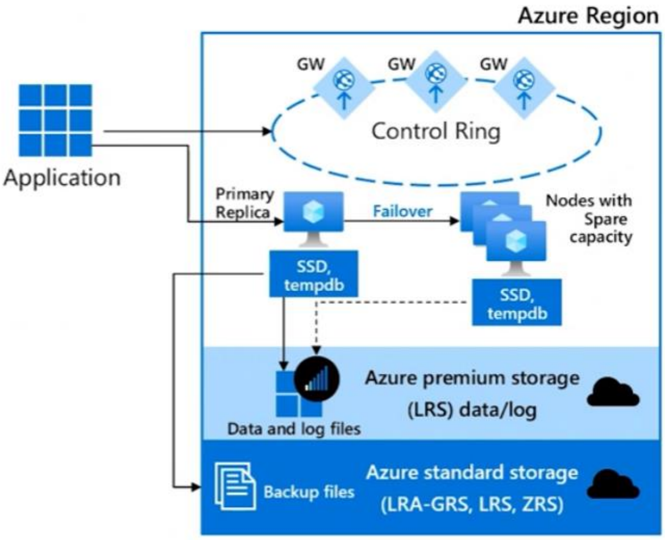
- Restore an existing database to a point in time in the past within the retention period
- Restore a deleted database to the time of deletion or to any point in time within the retention period
- Restore a database to another geographic region
- Restore a database from a specific long-term backup of a single database or pooled database
- Long term retention uses read-access geo-redundant storage (RA-GRS)

Retention period	Long term retention
35 days	Up to 10 years

High availability with the General Purpose/Standard tier

Azure SQL Database offers three service tiers that are designed for different types of applications:

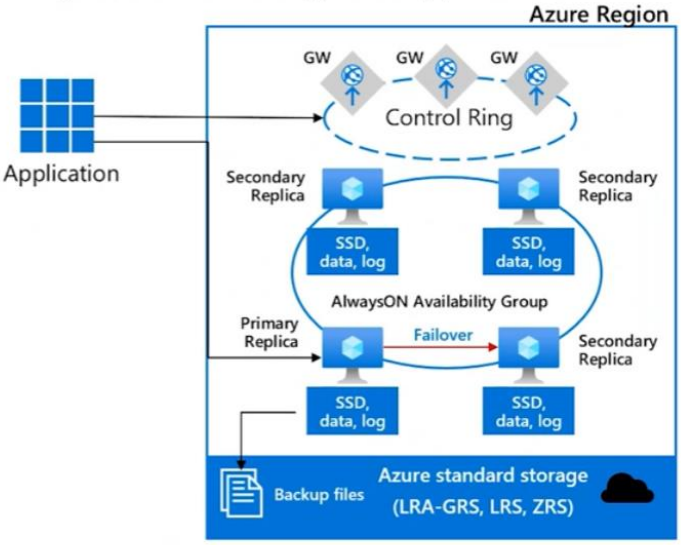
- Designed for common workloads
- Budget oriented balanced compute and storage
- Uses nodes with spare capacity to spin up a new SQL Server instances
- Uses LRS and RA-GRS (backup files)



High availability with the Business Critical/Premium tier

Azure SQL Database offers three service tiers that are designed for different types of applications:

- Designed for OLTP applications
- High transaction rate and low I/O latency
- Offers the highest resilience to failures by using several isolated replicas
- Deploys an Always On availability group using multiple synchronously updated replicas
- Uses local SSD storage and RA-GRS (backup files)



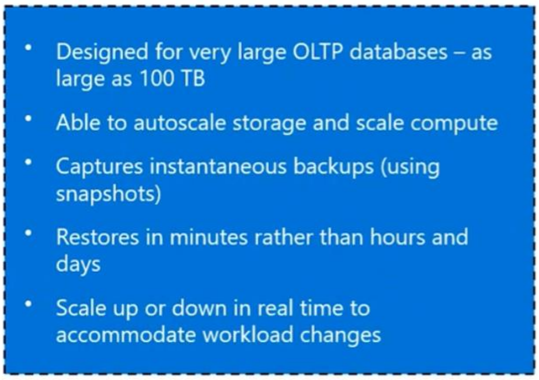
Encryption methods

Data state	Encryption method	Encryption level
Data at rest	Transparent data encryption (TDE)	Always encrypted
Data in motion	Secure Socket Layers and Transport Layer Security (SSL/TLS)	Always encrypted
Data in process	Dynamic data masking	Specific data is unencrypted, Remaining data is encrypted

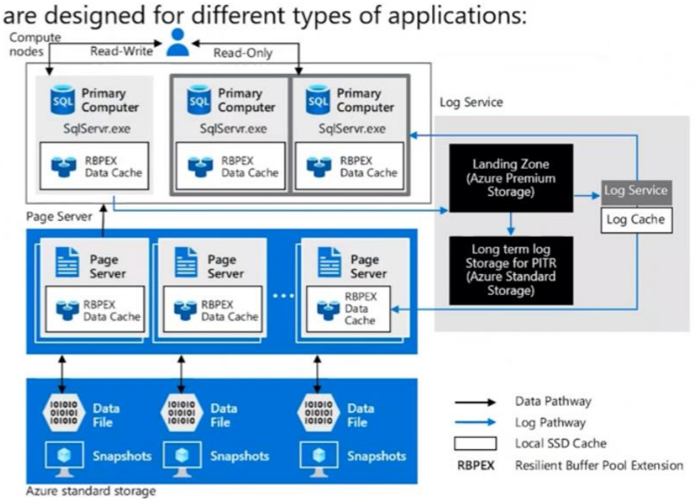
Scenario	Possible security solution
Secure access from multiple workstations located on-premises to an Azure virtual network	Use site-to-site VPN
Secure access from an individual workstation located on-premises to an Azure virtual network	Use point-to-site VPN
Move large data sets over a dedicated high-speed wide-area network (WAN) link	Use Azure ExpressRoute
Interact with Azure Storage through the Azure portal	All transactions are done by using HTTPS. You can also use the Azure Storage REST API over HTTPS to interact with Azure Storage and Azure SQL Database.

High availability with the Hyperscale tier

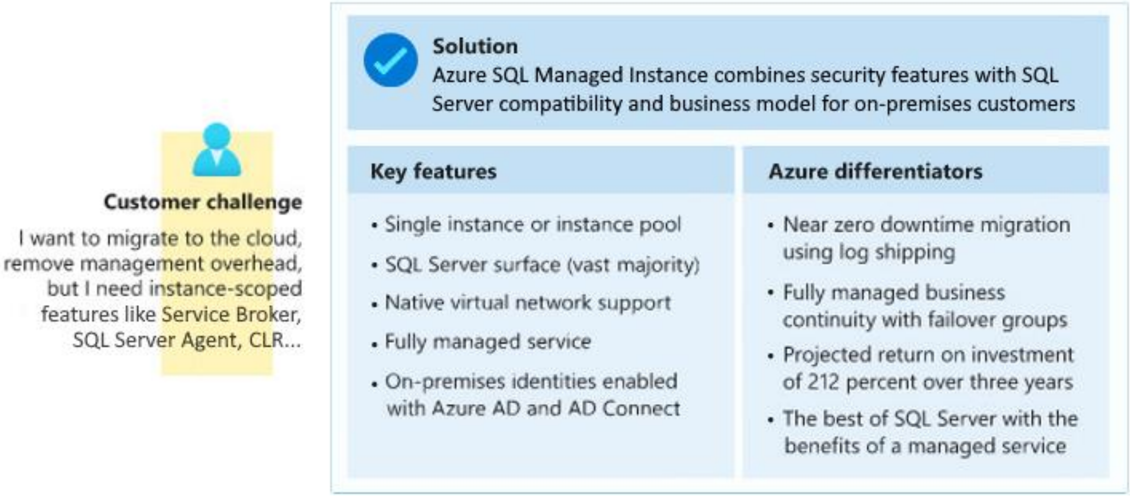
Azure SQL Database offers three service tiers that are designed for different types of applications:



- Designed for very large OLTP databases – as large as 100 TB
- Able to autoscale storage and scale compute
- Captures instantaneous backups (using snapshots)
- Restores in minutes rather than hours and days
- Scale up or down in real time to accommodate workload changes



Azure SQL Managed Instance



Customer challenge

I want to migrate to the cloud, remove management overhead, but I need instance-scoped features like Service Broker, SQL Server Agent, CLR...

Solution

Azure SQL Managed Instance combines security features with SQL Server compatibility and business model for on-premises customers

Key features

- Single instance or instance pool
- SQL Server surface (vast majority)
- Native virtual network support
- Fully managed service
- On-premises identities enabled with Azure AD and AD Connect

Azure differentiators

- Near zero downtime migration using log shipping
- Fully managed business continuity with failover groups
- Projected return on investment of 212 percent over three years
- The best of SQL Server with the benefits of a managed service

Azure SQL Managed Instance is a scalable cloud data service that provides the broadest SQL Server database engine compatibility with all the benefits of a fully managed platform as a service.

Depending on your scenario, Azure SQL Managed Instance might offer more options for your database needs.

Like Azure SQL Database, Azure SQL Managed Instance is a platform as a service (PaaS) database engine, which means that your company will be able to take advantage of the best features of moving your data to the cloud in a fully managed environment.

Azure SQL Managed Instance provides **several options that might not be available to Azure SQL Database**, such as SQL Server Agent, Common language runtime (CLR), Database Mail, Distributed transactions, and Machine Learning Services.

SQL Managed Instance uses vCores mode. You can define the maximum CPU cores and maximum storage allocated to your managed instance.

All databases within the managed instance share the resources allocated to the instance.

Azure Database for MySQL

Azure Database for MySQL is a relational database service in the cloud, and it's based on the MySQL Community Edition database engine, versions 5.6, 5.7, and 8.0.

Supports several websites on-premises that use the LAMP stack (Linux, Apache, MySQL, PHP).

Azure Database for MySQL delivers:

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

Azure Database for PostgreSQL

Azure Database for PostgreSQL is a relational database service in the cloud. The server software is based on the community version of the open-source PostgreSQL database engine.

Your familiarity with tools and expertise with PostgreSQL is applicable when you're using Azure Database for PostgreSQL.

Moreover, Azure Database for PostgreSQL delivers the following benefits:

- **Built-in high availability (99.99 percent SLA)** compared to on-premises resources. There's no additional configuration, replication, or cost required to make sure your applications are always available.
- **Simple and flexible pricing**. You have predictable performance based on a selected pricing tier choice that includes software patching, automatic backups, monitoring, and security.
- **Scale up or down as needed, within seconds**. You can scale compute or storage independently as needed to make sure you adapt your service to match usage.
- Monitoring and alerting to assess your server.
- **Adjustable automatic backups and point-in-time-restore for up to 35 days**.
- **Enterprise-grade security and compliance to protect sensitive data at rest and in motion**. This security covers data encryption on disk and SSL encryption between client and server communication.

Hyperscale

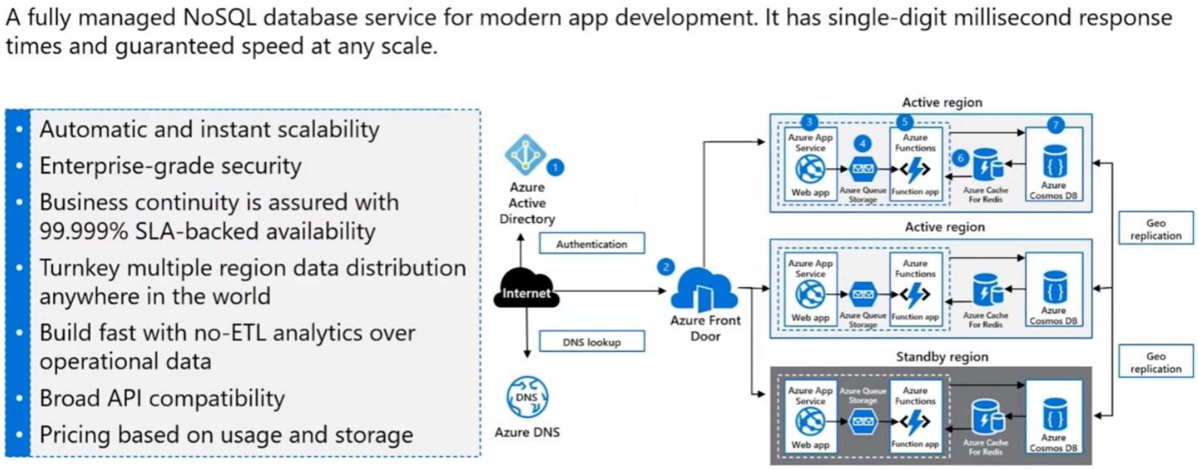
- The Hyperscale (Citus) option horizontally scales queries across multiple machines by using sharding. Its query engine parallelizes incoming SQL queries across these servers for faster responses on large datasets. It serves applications that require greater scale and performance, generally workloads that are approaching, or already exceed, 100 GB of data.
- The Hyperscale (Citus) deployment option supports multi-tenant applications, real-time operational analytics, and high-throughput transactional workloads. Applications built for PostgreSQL can run distributed queries on Hyperscale (Citus) with standard connection libraries and minimal changes.

Azure CosmosDB

Azure Cosmos DB is a fully managed NoSQL database service for modern app development.

As a fully managed service, Azure Cosmos DB takes database administration off your hands with automatic management, updates, and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

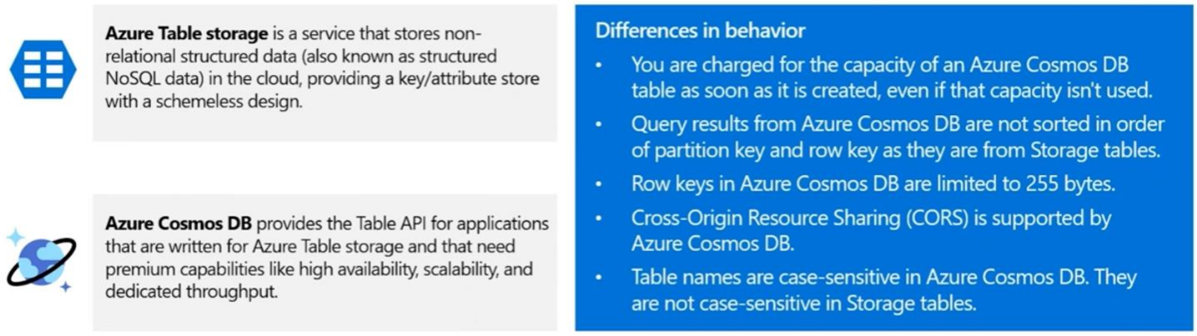
When to use Azure Cosmos DB



A fully managed NoSQL database service for modern app development. It has single-digit millisecond response times and guaranteed speed at any scale.

- Automatic and instant scalability
- Enterprise-grade security
- Business continuity is assured with 99.999% SLA-backed availability
- Turnkey multiple region data distribution anywhere in the world
- Build fast with no-ETL analytics over operational data
- Broad API compatibility
- Pricing based on usage and storage

Azure Storage tables and Azure Cosmos DB tables

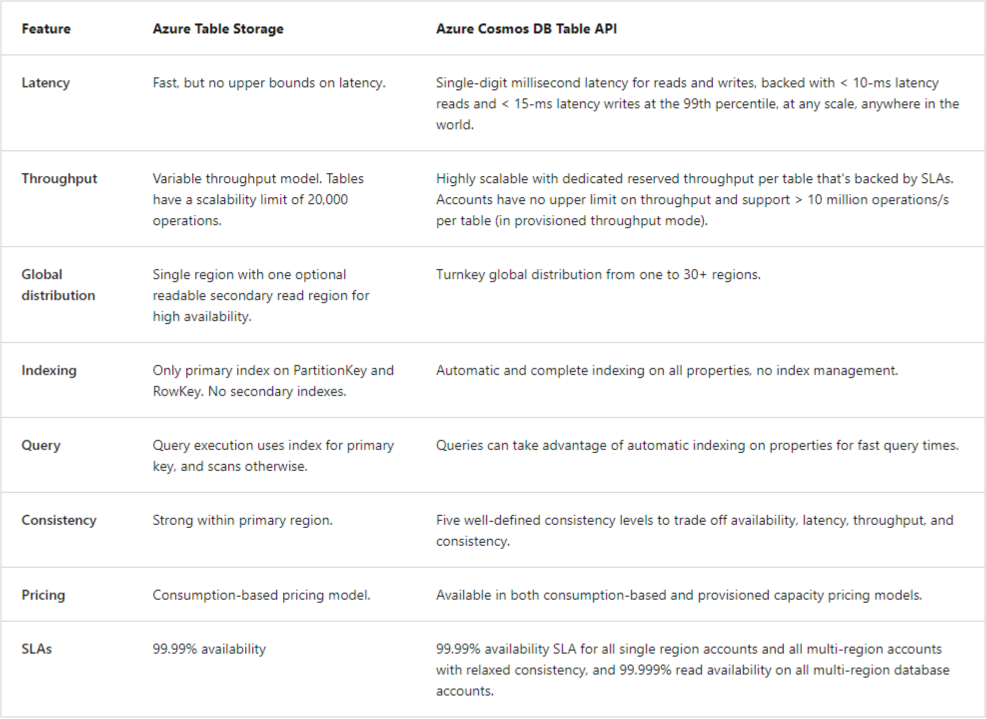


Azure Table storage is a service that stores non-relational structured data (also known as structured NoSQL data) in the cloud, providing a key/attribute store with a schemeless design.

Azure Cosmos DB provides the Table API for applications that are written for Azure Table storage and that need premium capabilities like high availability, scalability, and dedicated throughput.

Differences in behavior

- You are charged for the capacity of an Azure Cosmos DB table as soon as it is created, even if that capacity isn't used.
- Query results from Azure Cosmos DB are not sorted in order of partition key and row key as they are from Storage tables.
- Row keys in Azure Cosmos DB are limited to 255 bytes.
- Cross-Origin Resource Sharing (CORS) is supported by Azure Cosmos DB.
- Table names are case-sensitive in Azure Cosmos DB. They are not case-sensitive in Storage tables.



Feature	Azure Table Storage	Azure Cosmos DB Table API
Latency	Fast, but no upper bounds on latency.	Single-digit millisecond latency for reads and writes, backed with < 10-ms latency reads and < 15-ms latency writes at the 99th percentile, at any scale, anywhere in the world.
Throughput	Variable throughput model. Tables have a scalability limit of 20,000 operations.	Highly scalable with dedicated reserved throughput per table that's backed by SLAs. Accounts have no upper limit on throughput and support > 10 million operations/s per table (in provisioned throughput mode).
Global distribution	Single region with one optional readable secondary read region for high availability.	Turnkey global distribution from one to 30+ regions.
Indexing	Only primary index on PartitionKey and RowKey. No secondary indexes.	Automatic and complete indexing on all properties, no index management.
Query	Query execution uses index for primary key, and scans otherwise.	Queries can take advantage of automatic indexing on properties for fast query times.
Consistency	Strong within primary region.	Five well-defined consistency levels to trade off availability, latency, throughput, and consistency.
Pricing	Consumption-based pricing model.	Available in both consumption-based and provisioned capacity pricing models.
SLAs	99.99% availability	99.99% availability SLA for all single region accounts and all multi-region accounts with relaxed consistency, and 99.999% read availability on all multi-region database accounts.

High availability and disaster recovery options for PaaS

For the SQL Server-based options of Azure SQL Database and Azure SQL Database Managed Instance, the options are active geo-replication (Azure SQL Database only) and autofailover groups (Azure SQL Database or Azure SQL Database Managed Instance).

Azure Database for MySQL has a service level agreement, which guarantees availability of 99.99, meaning nearly no downtime should be encountered. For Azure Database for MySQL, if a node-level problem happens such as hardware failure, a built-in failover mechanism will kick in. All transactional changes to the MySQL database are written synchronously to storage upon commit. If a node-level interruption occurs, the database server automatically creates a new node and attaches the data storage.

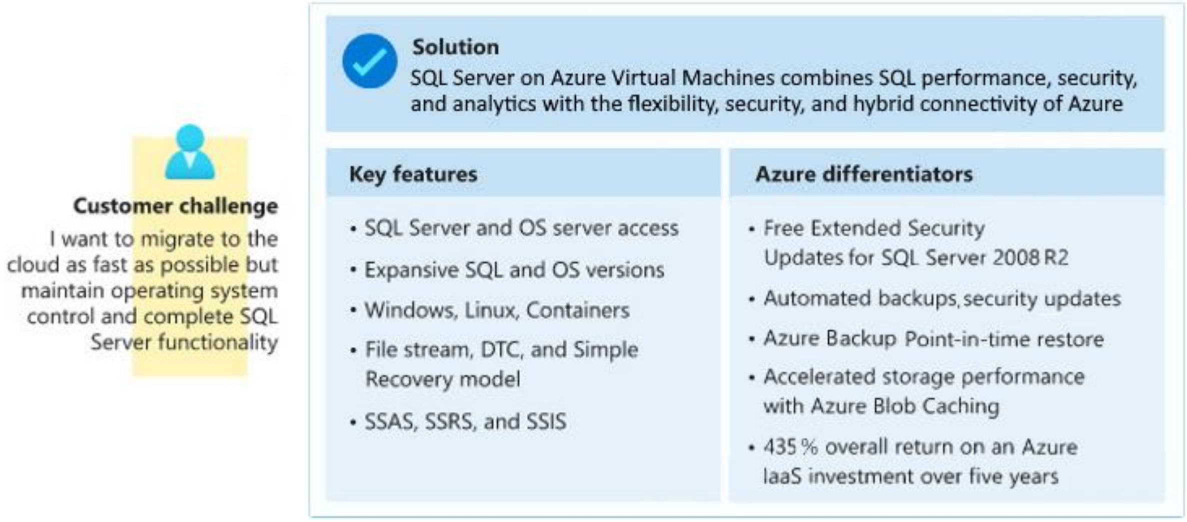
From an application standpoint, you will need to code the necessary retry logic because all connections are dropped as part of spinning up the new node and any in flight transactions are lost. This process is considered a best practice for any cloud application, as they should be designed to handle transient failures.

Azure Database for PostgreSQL uses a similar model to MySQL in its standard deployment model; however, Azure PostgreSQL also offers a scale-out hyperscale solution called Citus. Citus provides both scale-out and additional high availability for a server group. If enabled, a standby replica is configured for every node of a server group, which would also increase cost since it would double the number of servers in the group. In the event, the original node has a problem such as becoming unresponsive or failing completely, the standby takes its place. The data is kept in sync via PostgreSQL synchronous streaming replication.

As with Azure Database for MySQL, solutions that use Azure Database for PostgreSQL must also include retry logic in the application because of dropped connections and loss of in-flight transactions.

Both Azure Database for MySQL and PostgreSQL supports the option for a read replica. This means a replica can be used for activities like reporting to offload work from the primary database. A read replica also enhances availability because it exists in another region.

SQL Server running on Azure Virtual Machine



Customer challenge

I want to migrate to the cloud as fast as possible but maintain operating system control and complete SQL Server functionality

Solution

SQL Server on Azure Virtual Machines combines SQL performance, security, and analytics with the flexibility, security, and hybrid connectivity of Azure

Key features

- SQL Server and OS server access
- Expansive SQL and OS versions
- Windows, Linux, Containers
- File stream, DTC, and Simple Recovery model
- SSAS, SSRS, and SSIS

Azure differentiators

- Free Extended Security Updates for SQL Server 2008 R2
- Automated backups, security updates
- Azure Backup Point-in-time restore
- Accelerated storage performance with Azure Blob Caching
- 435 % overall return on an Azure IaaS investment over five years

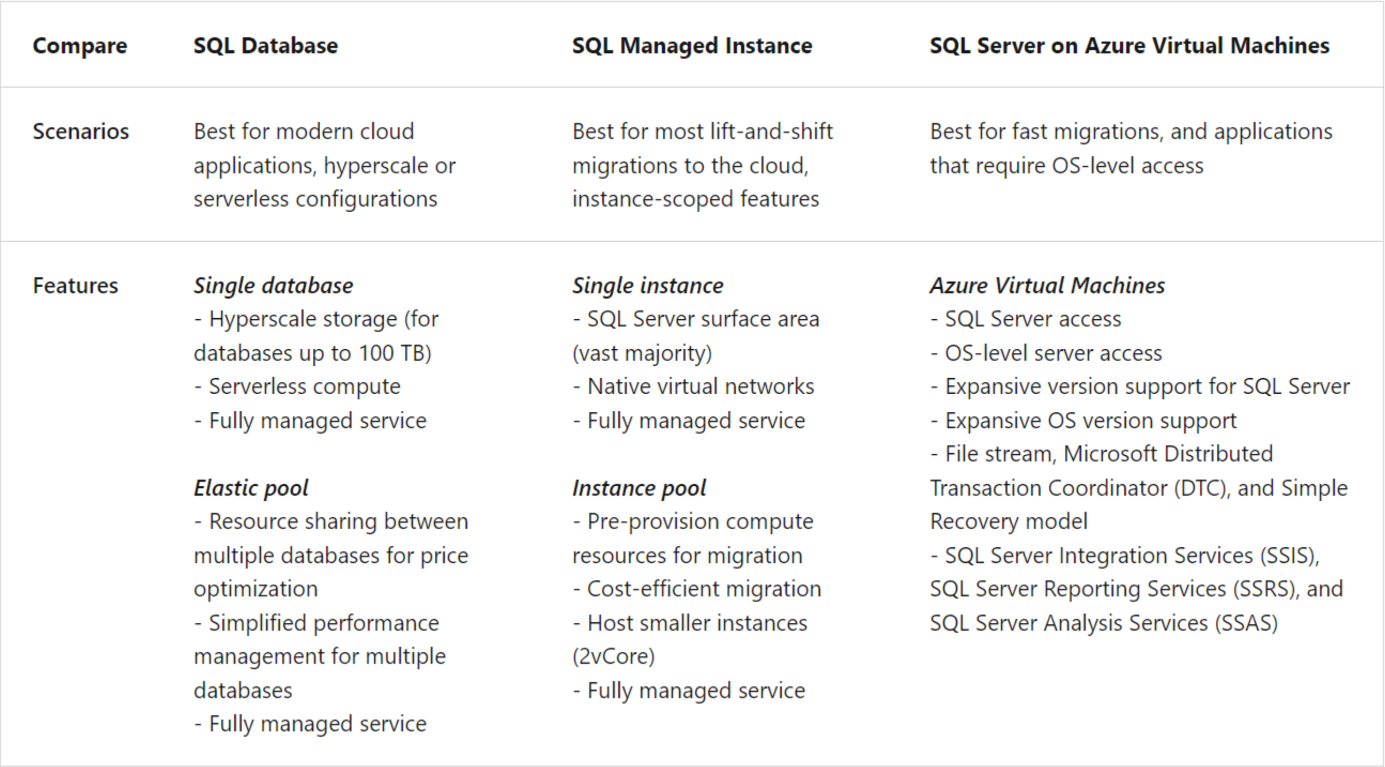
Things to know about SQL Server on Azure Virtual Machines

- When you run SQL Server on Azure Virtual Machines, you have access to the full capabilities of SQL Server.

- All of your SQL Server skills should directly transfer during the migration, and Azure can help automate backups and security patches.
- Unlike the Azure SQL Database and Azure SQL Managed Instance deployment options, you're responsible for version update operations for the OS and SQL Server.

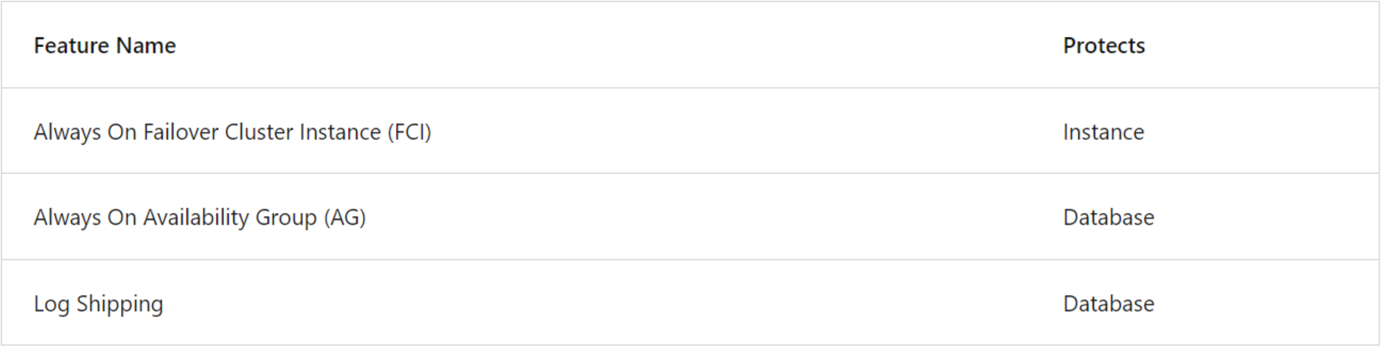
Things to consider when using SQL Server on Azure Virtual Machines

- **Consider server access**. Access your SQL Server and operating system server by implementing SQL Server on your virtual machines. Expansive support is provided for SQL Server and operating system versions.
- **Consider automated management**. Use the automated management features of SQL Server for your virtual machines.
- **Consider Azure Hybrid Benefit**. Exercise the Azure Hybrid Benefit for existing on-premises Windows Server and SQL Server licenses.



Compare	SQL Database	SQL Managed Instance	SQL Server on Azure Virtual Machines
Scenarios	Best for modern cloud applications, hyperscale or serverless configurations	Best for most lift-and-shift migrations to the cloud, instance-scoped features	Best for fast migrations, and applications that require OS-level access
Features	*Single database* - Hyperscale storage (for databases up to 100 TB) - Serverless compute - Fully managed service *Elastic pool* - Resource sharing between multiple databases for price optimization - Simplified performance management for multiple databases - Fully managed service	*Single instance* - SQL Server surface area (vast majority) - Native virtual networks - Fully managed service *Instance pool* - Pre-provision compute resources for migration - Cost-efficient migration - Host smaller instances (2vCore) - Fully managed service	*Azure Virtual Machines* - SQL Server access - OS-level server access - Expansive version support for SQL Server - Expansive OS version support - File stream, Microsoft Distributed Transaction Coordinator (DTC), and Simple Recovery model - SQL Server Integration Services (SSIS), SQL Server Reporting Services (SSRS), and SQL Server Analysis Services (SSAS)

SQL Server HADR Features for Azure Virtual Machine



Feature Name	Protects
Always On Failover Cluster Instance (FCI)	Instance
Always On Availability Group (AG)	Database
Log Shipping	Database

Always On Failover Cluster Instances

An FCI is configured when SQL Server is installed. A standalone instance of SQL Server cannot be converted to an FCI.

Applications and end users would use the unique name of the FCI for access. This abstraction enables applications to not have to know where the instance is running.

One major difference between Azure-based FCIs versus on-premises FCIs, is that for Azure, an internal load balancer (ILB) is required. The ILB is used to help ensure applications and end users can connect to the FCI's unique name.

When an FCI fails over to another node of a cluster, whether it is initiated manually or happens due to a problem, the entire instance restarts on another node. That means the failover process is a full stop and start of SQL Server.

Any applications or end users connected to the FCI will be disconnected during failover and only applications that can handle and recover from this interruption can reconnect automatically.

The FCI will be consistent to the point of failure, so technically there will be no data loss but any transactions that need to be rolled back will do so as part of recovery.

As noted above, because this is instance-level protection, everything necessary (logins, SQL Server Agent jobs, etc.) is already there so business can continue as usual once the databases are ready.

Why is this architecture worth considering?

- FCIs are still a popular availability solution.
- The shared storage story is improving with feature like Azure Shared Disk.
- This architecture meets most RTO and RPO for HA (although DR is not handled).
- This architecture provides an easy, standardized method for applications to access the clustered instance of SQL Server.
- This architecture provides enhanced availability during patching scenarios.
- This architecture separates out the WSFC as a single point of failure if all nodes lose communication
- In this architecture, one primary is not synchronizing all secondary replicas.
- This architecture can provide failing back from one location to another.

Always On availability groups

AGs were introduced in SQL Server 2012 Enterprise Edition and as of SQL Server 2016, are also in Standard Edition. In Standard Edition, an AG can contain one database whereas in Enterprise Edition, an AG can have more than one database. While AGs share some similarities with FCIs, in most ways they are different.

The biggest difference between an FCI and an AG is that AGs provide database-level protection.

The primary replica is the instance participating in an AG that contains the read/write databases.

A secondary replica is where the primary sends transactions over the log transport to keep it synchronized. Data movement between a primary replica can be synchronous or asynchronous.

The databases on any secondary replica are in a loading state, which means they can receive transactions but cannot be a fully writeable copy until that replica becomes the primary.

An AG in Standard Edition can have at most two replicas (one primary, one secondary) whereas Enterprise Edition supports up to nine (one primary, eight secondary).

A secondary replica is initialized either from a backup of the database, or as of SQL Server 2016, you can use a feature called 'automatic seeding'.

Automatic seeding uses the log stream transport to stream the backup to the secondary replica for each database of the availability group using the configured endpoints.

An AG provides abstraction with the listener. The listener functions like the unique name assigned to an FCI and has its own name and IP address that is different from anything else (WSFC, node, etc.). The listener also requires an ILB and goes through a stop and start.

Applications and end users can use the listener to connect, but unlike an FCI, if desired, the listener does not have to be used. Connections directly to the instance can occur.

With Enterprise Edition, secondary replicas in Enterprise Edition can also be configured for read-only access if desired and can be used for other functionality such as database consistency checks (DBCCs) and backups.

AGs can have a quicker failover time compared to an FCI, which is one reason they are attractive. While AGs do not require shared storage, each replica has a copy of the data, which increases the total number of copies of the database and overall storage costs.

Remember that any object that exists outside of the database or is not captured in the database's transaction log must manually be created and accounted for on any other SQL Server instance should that instance need to become the new primary replica.

Examples of objects you would be responsible for include SQL Server Agent jobs, instance-level logins, and linked servers. If you can use Windows authentication or use contained databases with AGs, it will simplify access.

If you are using AGs, one option is **to configure the AG across multiple Azure regions or potentially as a hybrid architecture**. This means that all nodes which contain the replicas participate in the same WSFC. This assumes good network connectivity, especially if this is a hybrid configuration.

One of the biggest considerations would be the witness resource for the WSFC. This architecture would require AD DS and DNS to be available in every region and potentially on premises as well if this is a hybrid solution.

Why is this architecture worth considering?

- This architecture protects data by having more than one copy on different virtual machines (VMs).
- This architecture allows you to meet recovery time objective (RTO) and recovery point objective (RPO) with minimal-to-no data loss if implemented properly.
- This architecture provides an easy, standardized method for applications to access both primary and secondary replicas (if things like read-only replicas will be used).
- This architecture provides enhanced availability during patching scenarios.
- This architecture needs no shared storage, so there is less complication than when using a failover cluster instance (FCI).
- This architecture works with Standard and Enterprise editions of SQL Server.

Log shipping

Log shipping has been around since the early days of SQL Server. The feature is based on backup, copy, and restore and is one of the simplest methods of achieving HADR for SQL Server.

Log shipping is primarily used for disaster recovery, but it could also be used to enhance local availability.

Log shipping, like AGs, provides database-level protection, which means you still need to account for SQL Server Agent jobs, linked servers, instance-level logins, etc.

There is no abstraction provided natively by log shipping, so a switch to another server participating in log shipping must be able to tolerate a name change. If that is not possible, there are methods such as a DNS alias, which can be configured at the network layer to try to mitigate the name change issues.

The log shipping mechanism is simple: first, take a full backup of the source database on the primary server, restore it in a loading state (STANDBY or NORECOVERY) on another instance known as a secondary server or warm standby.

This new copy of the database is known as a secondary database. An automated process built into SQL Server will then automatically backup the primary database's transaction log, copy the backup to the standby server, and finally, restore the backup onto the standby.

Why is this architecture worth considering?

- Log shipping is a tried-and-true feature that has been around for over 20 years
- Log shipping is easy to deploy and administer since it is based on backup and restore.
- Log shipping is tolerant of networks that are not robust.
- Log shipping meets most RTO and RPO goals for DR.
- Log shipping is a good way to protect FCIs.

Azure Site Recovery

For those who do not want to implement a SQL Server-based disaster solution, Azure Site Recovery is a potential option. However, most data professionals prefer a database-centric approach as it will generally have a lower RPO.

Why is this architecture worth considering?

- Azure Site Recovery supports deploying to another region.
- Azure Site Recovery will work with more than just SQL Server
- Azure Site Recovery may meet RTO and possibly RPO.
- Azure Site Recovery is provided as part of the Azure platform.

Azure Data Factory

Azure Data Factory is a cloud-based data integration service that can help you create and schedule data-driven workflows.

You can use Azure Data Factory to orchestrate data movement and transform data at scale.

The data-driven workflows, or pipelines, ingest data from disparate data stores.

Azure Data Factory is an ETL data integration process, which stands for extract, transform, and load. This integration process combines data from multiple data sources into a single data store.

Components of Azure Data Factory

- Pipelines and activities: Pipelines provide a logical grouping of activities that perform a task. An activity is a single processing step in a pipeline. Azure Data Factory supports data movement, data transformation, and control activities.
- Datasets: Datasets are data structures within your data stores.
- Linked services: Linked services define the required connection information needed for Azure Data Factory to connect to external resources.
- Data flows: Data flows allow data engineers to develop data transformation logic without writing code. Data flow activities can be operationalized by using existing Azure Data Factory scheduling, control, flow, and monitoring capabilities.
- Integration runtimes: Integration runtimes are the bridge between the activity and linked Services objects. There are three types of integration runtime: Azure, self-hosted, and Azure-SSIS.

Things to consider when using Azure Data Factory

- **Consider requirements for data integration**. Azure Data Factory serves two communities: the big data community and the relational data warehousing community that uses SQL Server Integration Services (SSIS). Depending on your organization's data needs, you can set up pipelines in the cloud by using Azure Data Factory. You can access data from both cloud and on-premises data services.
- **Consider coding resources**. If you prefer a graphical interface to set up pipelines, then Azure Data Factory authoring and monitoring tool is the right fit for your needs. Azure Data Factory provides a low code/no code process for working with data sources.
- **Consider support for multiple data sources**. Azure Data Factory supports 90+ connectors to integrate with disparate data sources.
- **Consider serverless infrastructure**. There are advantages to using a fully managed, serverless solution for data integration. There's no need to maintain, configure or deploy servers, and you gain the ability to scale with fluctuating workloads.

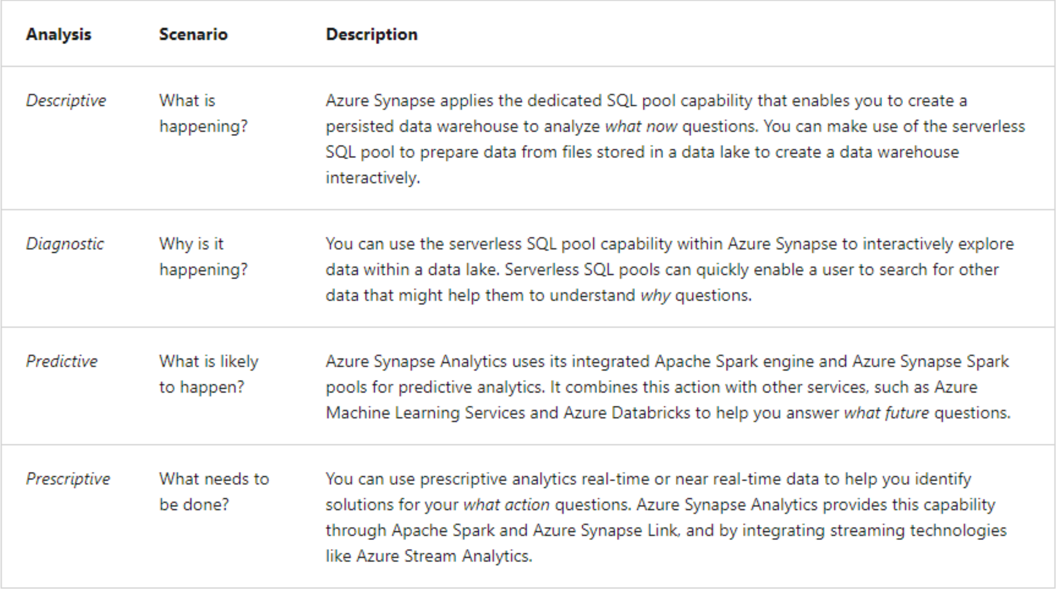
Azure Synapse Analytics

Azure Synapse Analytics (formerly Azure SQL Data Warehouse) is a limitless analytics service that brings together enterprise-data warehousing and big-data analytics.

You can query data on your terms by using either serverless or provisioned resources at scale.

You have a unified experience to ingest, prepare, manage, and serve data for immediate business intelligence and machine learning needs.

Analytical options



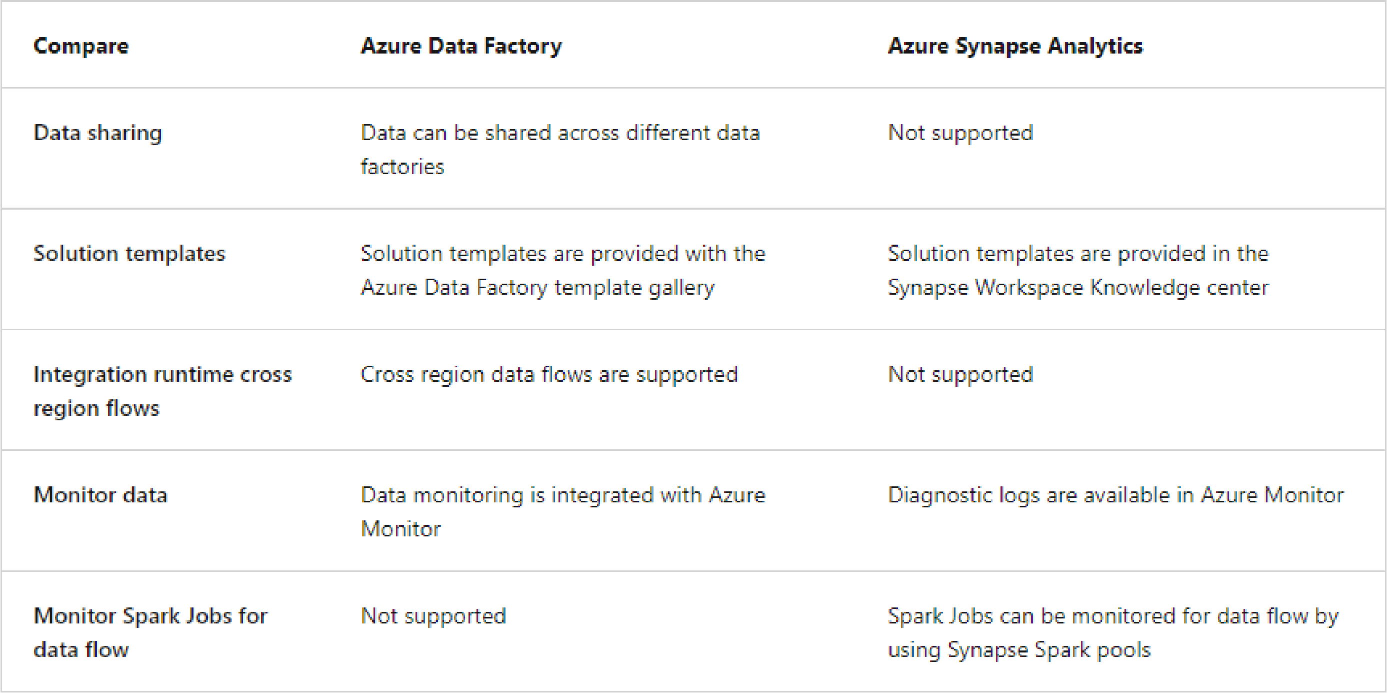
Analysis	Scenario	Description
Descriptive	What is happening?	Azure Synapse applies the dedicated SQL pool capability that enables you to create a persisted data warehouse to analyze *what now* questions. You can make use of the serverless SQL pool to prepare data from files stored in a data lake to create a data warehouse interactively.
Diagnostic	Why is it happening?	You can use the serverless SQL pool capability within Azure Synapse to interactively explore data within a data lake. Serverless SQL pools can quickly enable a user to search for other data that might help them to understand *why* questions.
Predictive	What is likely to happen?	Azure Synapse Analytics uses its integrated Apache Spark engine and Azure Synapse Spark pools for predictive analytics. It combines this action with other services, such as Azure Machine Learning Services and Azure Databricks to help you answer *what future* questions.
Prescriptive	What needs to be done?	You can use prescriptive analytics real-time or near real-time data to help you identify solutions for your *what action* questions. Azure Synapse Analytics provides this capability through Apache Spark and Azure Synapse Link, and by integrating streaming technologies like Azure Stream Analytics.

Things to consider when choosing Azure Data Factory or Azure Synapse Analytics

- **Consider variety of data sources**. When you have various data sources that use Azure Synapse Analytics for code-free ETL and data flow activities.
- **Consider Machine Learning**. When you need to implement Machine Learning solutions by using Apache Spark, you can use Azure Synapse Analytics for built-in support for AzureML.

- **Consider data lake integration**. When you have existing data stored on a data lake and need integration with Azure Data Lake and other input sources, Azure Synapse Analytics provides seamless integration between the two components.
- **Consider real-time analytics**. When you require real-time analytics, you can use features like Azure Synapse Link to analyze data in real-time and offer insights.

Azure Data Factory vs Azure Synapse Analytics



Compare	Azure Data Factory	Azure Synapse Analytics
Data sharing	Data can be shared across different data factories	Not supported
Solution templates	Solution templates are provided with the Azure Data Factory template gallery	Solution templates are provided in the Synapse Workspace Knowledge center
Integration runtime cross region flows	Cross region data flows are supported	Not supported
Monitor data	Data monitoring is integrated with Azure Monitor	Diagnostic logs are available in Azure Monitor
Monitor Spark Jobs for data flow	Not supported	Spark Jobs can be monitored for data flow by using Synapse Spark pools

Azure HDInsight

Azure HDInsight is a fully managed, open-source analytics service for enterprises.

It's a cloud service that makes it easier, faster, and more cost-effective to process massive amounts of data.

You can run popular open-source frameworks and create cluster types such as Apache Spark, Apache Hadoop, Apache Kafka, Apache HBase.

HDInsight also supports a broad range of scenarios such as extraction, transformation, and loading (ETL), data warehousing, machine learning, and IoT.

Azure Databricks

Azure Databricks helps you unlock insights from all your data and build artificial intelligence solutions.

You can set up your Apache Spark environment in minutes, then autoscale and collaborate on shared projects in an interactive workspace.

Azure Databricks supports Python, Scala, R, Java, and SQL, as well as data science frameworks and libraries including TensorFlow, PyTorch, and scikit-learn.

Things to consider when using Azure Databricks

- **Consider data science preparation of data**. Create, clone, and edit clusters of complex, unstructured data. Turn the data clusters into specific jobs. Deliver the results to data scientists and data analysts for review.
- **Consider insights in the data**. Implement Azure Databricks to build recommendation engines, churn analysis, and intrusion detection.
- **Consider productivity across data and analytics teams**. Create a collaborative environment and shared workspaces for data engineers, analysts, and scientists. Teams can work together across the data science lifecycle with shared workspaces, which helps to save valuable time and resources.
- **Consider big data workloads**. Exercise Azure Data Lake and the engine to get the best performance and reliability for your big data workloads. Create no-fuss multi-step data pipelines.
- **Consider machine learning programs**. Take advantage of the integrated end-to-end machine learning environment. It incorporates managed services for experiment tracking, model training, feature development and management, and feature and model serving.

Azure Data Lake Analytics

Azure Data Lake Analytics is an on-demand analytics job service that simplifies big data.

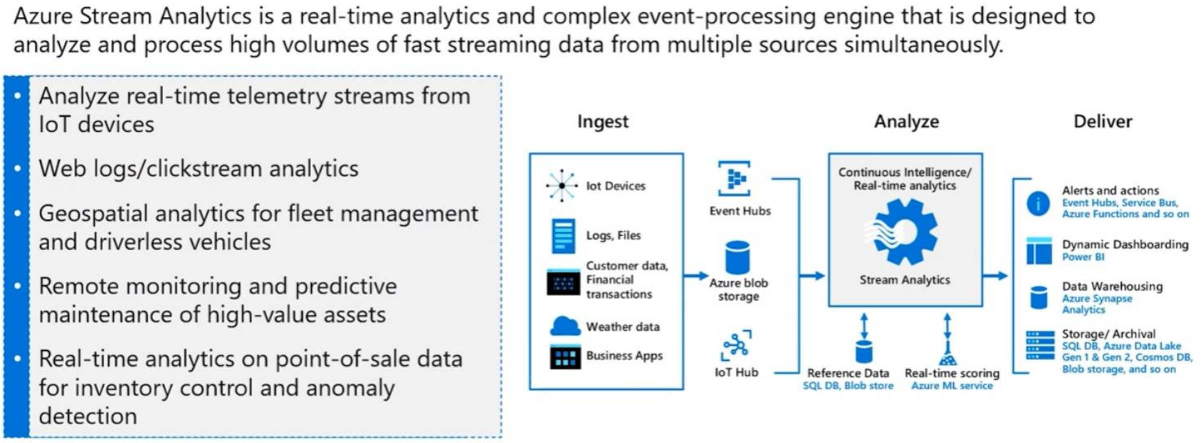
Instead of deploying, configuring, and tuning hardware, you can write queries to transform your data and extract valuable insights.

The analytics service can handle jobs of any scale instantly by setting the dial for how much power you need.

You only pay for your job when it's running, making it more cost-effective.

Azure Stream Analytics

Overview



Azure Stream Analytics is a real-time analytics and complex event-processing engine that is designed to analyze and process high volumes of fast streaming data from multiple sources simultaneously.

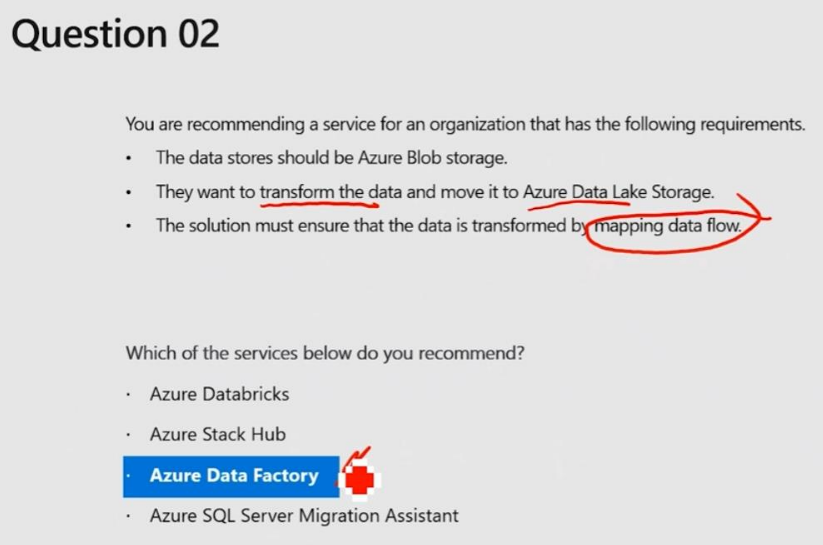
- Analyze real-time telemetry streams from IoT devices
- Web logs/clickstream analytics
- Geospatial analytics for fleet management and driverless vehicles
- Remote monitoring and predictive maintenance of high-value assets
- Real-time analytics on point-of-sale data for inventory control and anomaly detection

Things to consider when using Azure Stream Analytics

- **Consider provisioning requirements**. Azure Stream Analytics is a fully managed service. It's offered as a PaaS (Platform as a Service) offering, so there's no overhead of provisioning any hardware or infrastructure. Azure

Stream Analytics fully manages your job, so you can focus on your business logic and not on the infrastructure.

- **Consider costs**. Stream Analytics is low cost. Billing is done by Streaming Units (SUs) consumed that represents the amount of CPU and memory resources allocated. Scaling up and down are based on business needs, which can also lower costs. No maintenance charges are involved.
- **Consider implementation**. You can run Azure Stream Analytics in the cloud for large-scale analytics. For ultra-low latency analytics, run Stream Analytics on IoT Edge or Azure Stack.
- **Consider performance**. Stream Analytics offers reliable performance guarantees. It supports higher performance by partitioning, which allows complex queries to be parallelized and executed on multiple streaming nodes. Stream Analytics can process millions of events every second. It can deliver results with ultra-low latencies.
- **Consider security**. Stream Analytics encrypts all incoming and outgoing communications and supports TLS 1.2. Built-in checkpoints are also encrypted. Stream Analytics doesn't store the incoming data because all processing is done in-memory.



Question 02

You are recommending a service for an organization that has the following requirements.

- The data stores should be Azure Blob storage.
- They want to transform the data and move it to Azure Data Lake Storage.
- The solution must ensure that the data is transformed by mapping data flow.

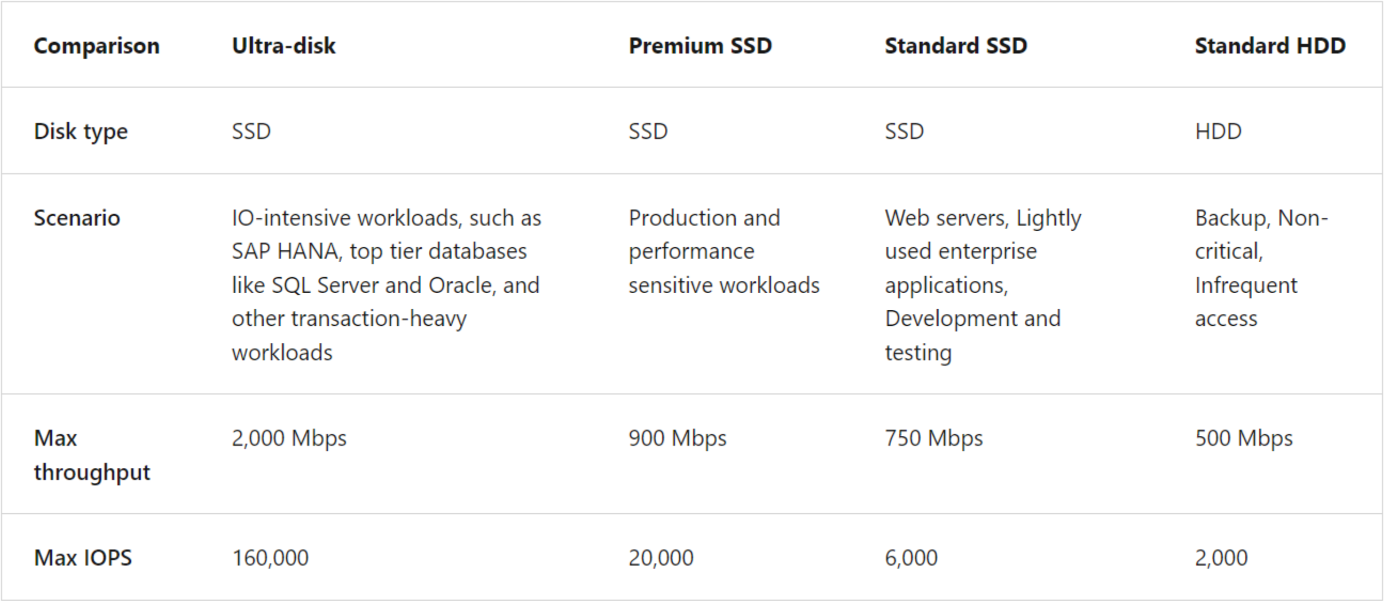
Which of the services below do you recommend?

- Azure Databricks
- Azure Stack Hub
- [✓ Azure Data Factory]
- Azure SQL Server Migration Assistant

Azure Managed Disks

Data disks are used by virtual machines to store data like database files, website static content, or custom application code. The number of data disks you can add depends on the virtual machine size. Each data disk has a maximum capacity of 32,767 GB.

Microsoft recommends always using Azure managed disks. You specify the disk size, the disk type, and provision the disk. Azure handles the remaining operations.



Comparison	Ultra-disk	Premium SSD	Standard SSD	Standard HDD
Disk type	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads, such as SAP HANA, top tier databases like SQL Server and Oracle, and other transaction-heavy workloads	Production and performance sensitive workloads	Web servers, Lightly used enterprise applications, Development and testing	Backup, Non-critical, Infrequent access
Max throughput	2,000 Mbps	900 Mbps	750 Mbps	500 Mbps
Max IOPS	160,000	20,000	6,000	2,000

Encryption options

- **Azure Disk Encryption (ADE)** encrypts the VM's virtual hard disks (VHDs). If VHD is protected with ADE, the disk image is accessible only by the VM that owns the disk.
- **Server-Side Encryption (SSE)** is performed on the physical disks in the data center. If someone directly accesses the physical disk, the data will be encrypted. When the data is accessed from the disk, it's decrypted and loaded into memory. This form of encryption is also referred to as encryption at rest or Azure Storage encryption.
- **Encryption at host** ensures that data stored on the VM host is encrypted at rest and flows encrypted to the Storage service. Disks with encryption at host enabled aren't encrypted with SSE. Instead, the server hosting your VM provides the encryption for your data, and that encrypted data flows into Azure Storage.

Data caching: Improve performance with disk caching (up to 4 TB). Azure Virtual Machines disk caching optimizes read and write access to the virtual hard disk (VHD) files. The VHDs are attached to Azure Virtual Machines. For OS disks, the default cache setting is ReadWrite, and for data disks, the default is ReadOnly.

Azure Blob Storage

Azure Blob Storage is an object storage solution for the cloud.

It can store massive amounts of data, such as text or binary data.

Azure Blob Storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold.

Blob Storage can manage thousands of simultaneous uploads, massive amounts of video data, constantly growing log files, and can be reached from anywhere with an internet connection.

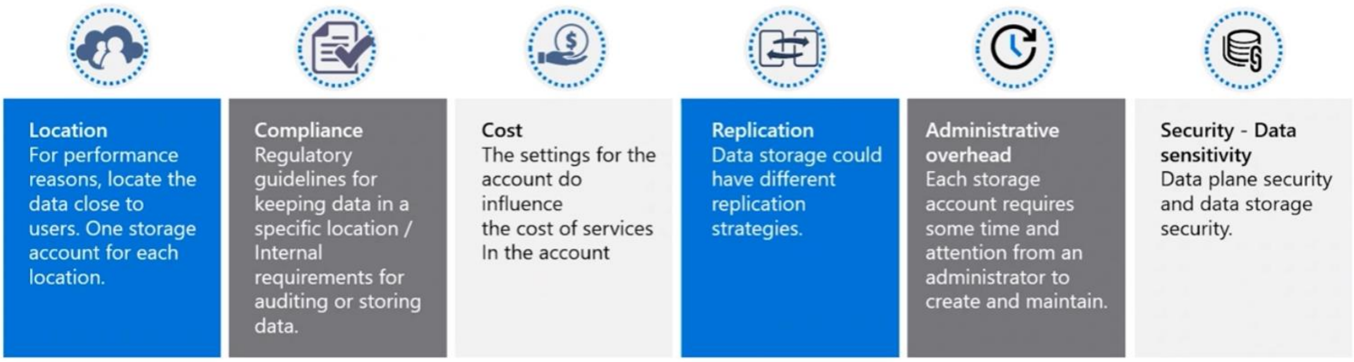
Blob Storage is ideal for:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Storing data for backup and restore, disaster recovery, and archiving.

- Storing data for analysis by an on-premises or Azure-hosted service.
- Storing up to 8 TB of data for virtual machines.

Considerations for storage accounts

It is important to plan your storage accounts.



Location
For performance reasons, locate the data close to users. One storage account for each location.

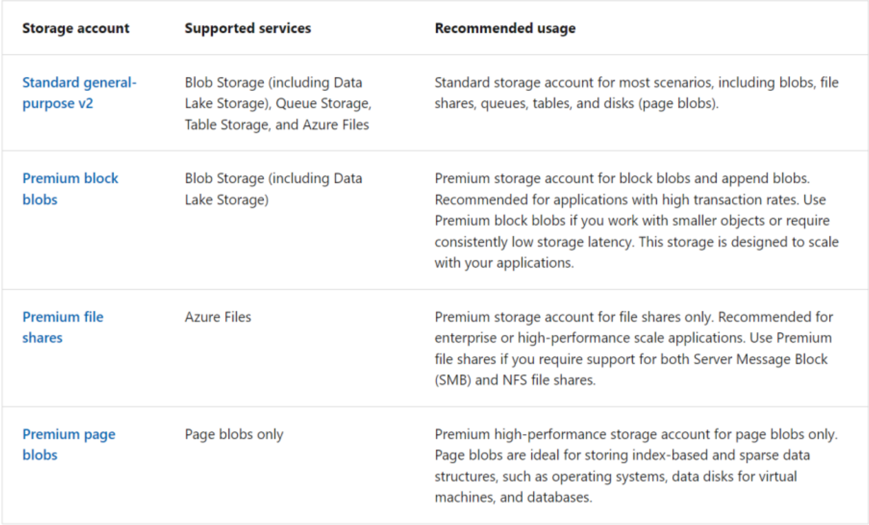
Compliance
Regulatory guidelines for keeping data in a specific location / Internal requirements for auditing or storing data.

Cost
The settings for the account do influence the cost of services In the account

Replication
Data storage could have different replication strategies.

Administrative overhead
Each storage account requires some time and attention from an administrator to create and maintain.

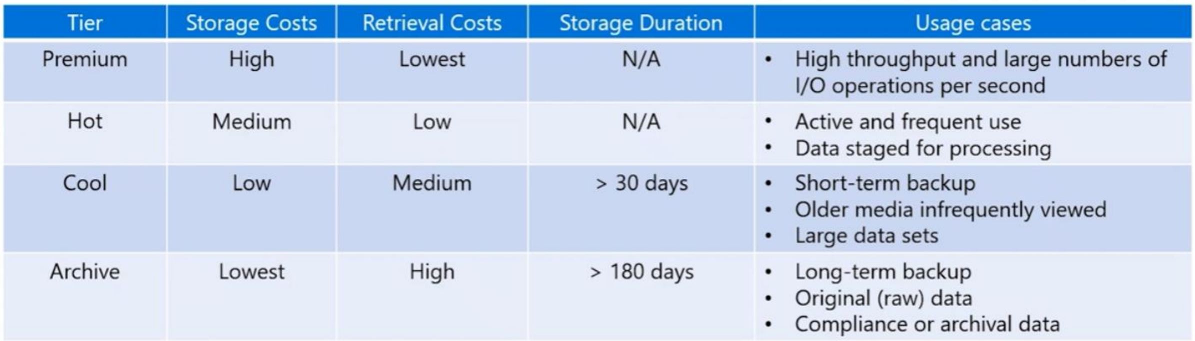
Security - Data sensitivity
Data plane security and data storage security.



Storage account	Supported services	Recommended usage
Standard general-purpose v2	Blob Storage (including Data Lake Storage), Queue Storage, Table Storage, and Azure Files	Standard storage account for most scenarios, including blobs, file shares, queues, tables, and disks (page blobs).
Premium block blobs	Blob Storage (including Data Lake Storage)	Premium storage account for block blobs and append blobs. Recommended for applications with high transaction rates. Use Premium block blobs if you work with smaller objects or require consistently low storage latency. This storage is designed to scale with your applications.
Premium file shares	Azure Files	Premium storage account for file shares only. Recommended for enterprise or high-performance scale applications. Use Premium file shares if you require support for both Server Message Block (SMB) and NFS file shares.
Premium page blobs	Page blobs only	Premium high-performance storage account for page blobs only. Page blobs are ideal for storing index-based and sparse data structures, such as operating systems, data disks for virtual machines, and databases.

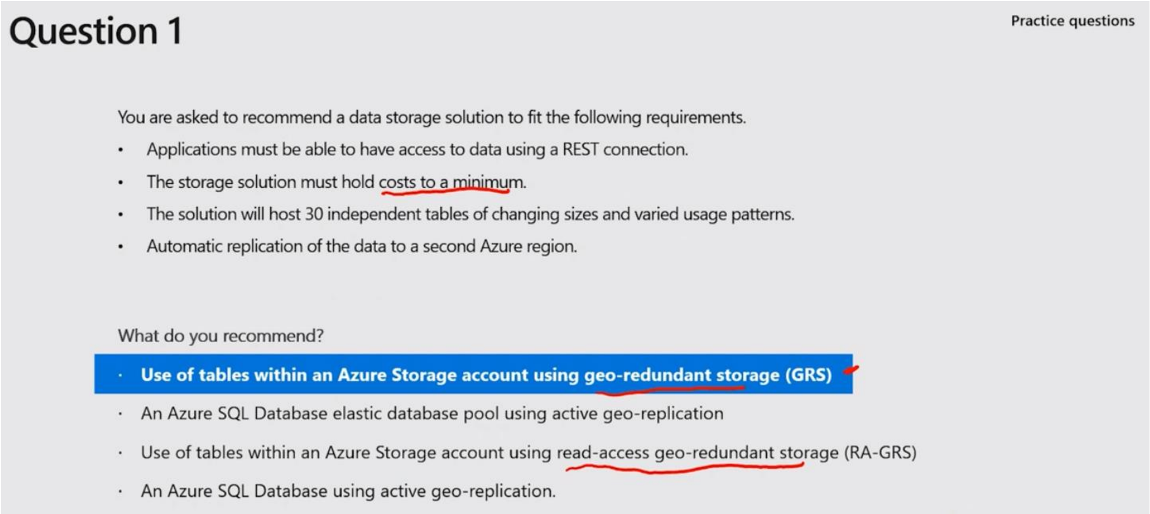
Determine the storage tier

Blob storage is an object store used for storing vast amounts of unstructured data.



Tier	Storage Costs	Retrieval Costs	Storage Duration	Usage cases
Premium	High	Lowest	N/A	• High throughput and large numbers of I/O operations per second
Hot	Medium	Low	N/A	• Active and frequent use • Data staged for processing
Cool	Low	Medium	> 30 days	• Short-term backup • Older media infrequently viewed • Large data sets
Archive	Lowest	High	> 180 days	• Long-term backup • Original (raw) data • Compliance or archival data

- Use lifecycle rules to transition blob data to the appropriate access tiers.
- Consider a data lifecycle rule to expire or delete data.



Question 1

Practice questions

You are asked to recommend a data storage solution to fit the following requirements.

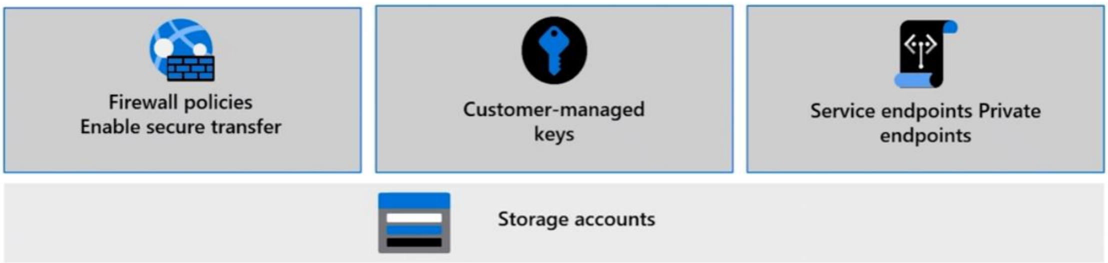
- Applications must be able to have access to data using a REST connection.
- The storage solution must hold costs to a minimum.
- The solution will host 30 independent tables of changing sizes and varied usage patterns.
- Automatic replication of the data to a second Azure region.

What do you recommend?

- **Use of tables within an Azure Storage account using geo-redundant storage (GRS)**
- An Azure SQL Database elastic database pool using active geo-replication
- Use of tables within an Azure Storage account using read-access geo-redundant storage (RA-GRS)
- An Azure SQL Database using active geo-replication.

Considerations for storage security

Use a layered security model to secure and control access.

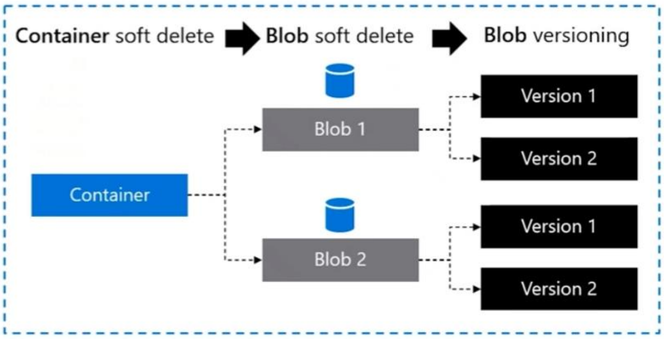


- Grant limited access to Azure Storage resources
- Enable firewall rules to limit access to access - IP addresses or subnets
- Use private endpoints and private links for clients
- Use virtual network service endpoints to provide direct connection
- Use customer managed encryption keys

Considerations for soft delete

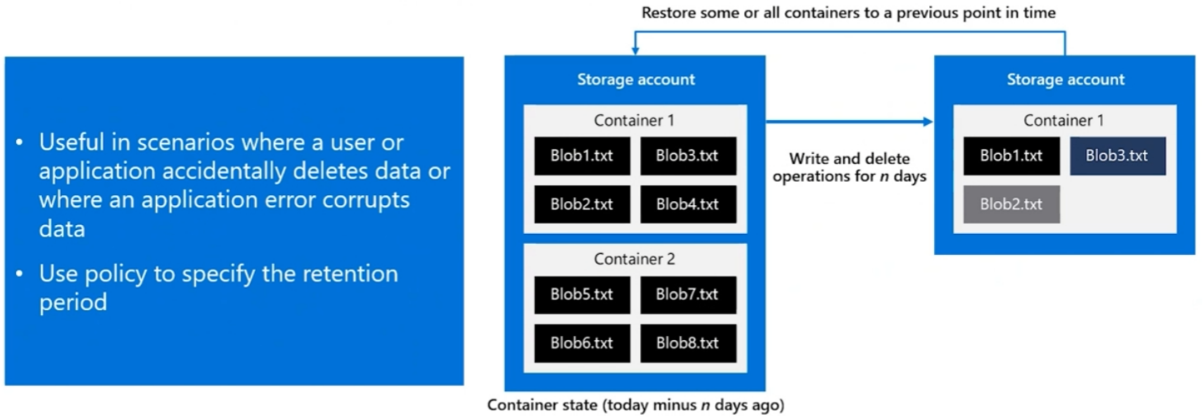
Consider soft delete with recovery times from 1 to 365 days

- Maintains the deleted data in the system for a specified period of time
- Soft delete protects blobs, snapshot, containers, or versions from accidental deletes or overwrites
- Soft delete maintains the deleted data in the system for a specified retention period



Considerations for point-in-time restore

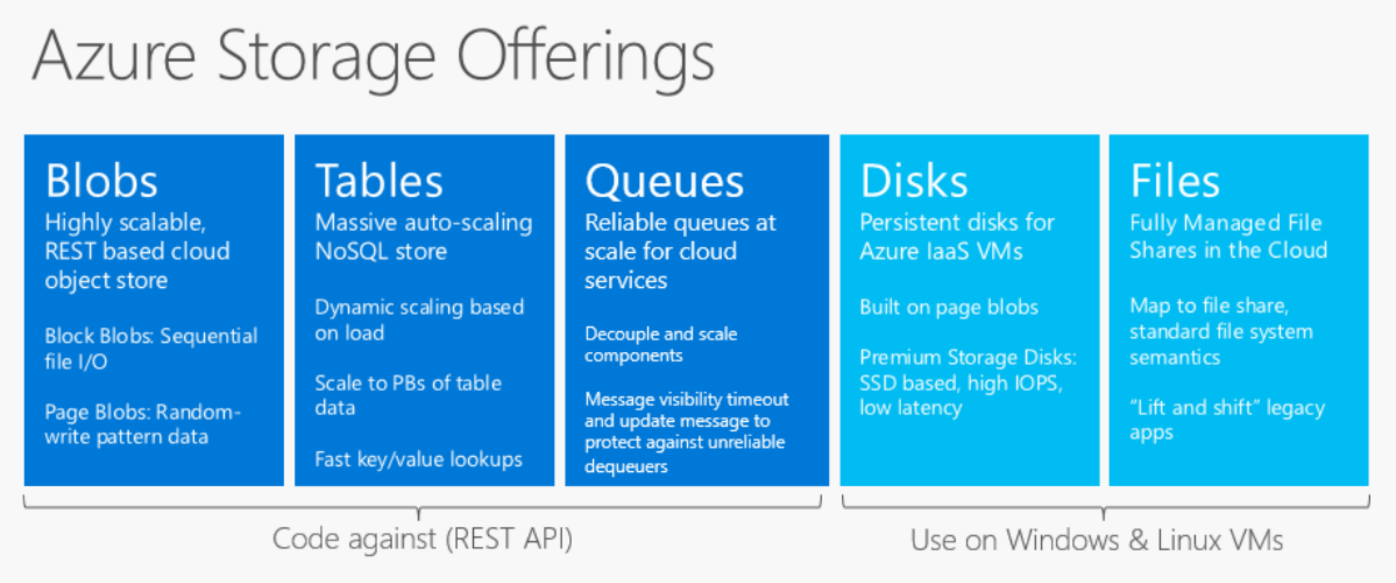
Consider point-in-time restore for block blobs



- Useful in scenarios where a user or application accidentally deletes data or where an application error corrupts data
- Use policy to specify the retention period

Comparing storage accounts

Storage Type	Description	Pricing
Azure Block Blobs	Scalable object storage for documents, videos, images, and unstructured text or binary data. There are 3 tiers to choose from Hot, Cool or Archive.	Prices for LRS Archive Block Blob with 3 years of reserved capacity start at $0.00081 / GB per month.
Azure Data Lake Storage Gen2	Combines the power of a Hadoop-compatible file system (which uses an integrated hierarchical namespace) with the massive scale and economy of Azure Blob Storage.	Prices for LRS archive storage with 3 years of reserved capacity start at $0.00081 / GB per month.
Azure Managed Disks	Persistent, secure disks that support easy and scalable virtual machine deployment; designed to achieve 99.999% availability.	Prices for standard managed disks start at $1.54 per month.
Azure Files	Fully managed file shares in the cloud (accessible via standard Server Message Block (SMB) protocol) for applications using Windows APIs or REST API.	Prices for LRS file storage start at $0.058 / GB per month.
Azure Page Blobs	Optimized for random read / write options that are ideal for overwriting small segments at a known address. Page blobs can be accessed via the REST protocol or attached to a VM to support disk traffic as unmanaged disks.	Prices for LRS file storage start at $0.14 / GB per month.
Azure Table Storage	Offers NoSQL storage for unstructured and semi-structured data using a schema-less design that relies on key/attribute storage which is ideal for web applications, address books and other user data.	Prices for LRS file storage start at $0.045 / GB per month.
Azure Queues Storage	Provides a reliable messaging solution for your apps and is generally used to store messages that are processed asynchronously; messages can be up to 64 KB in size. It can handle large numbers of messages (up to millions) into queues, which are accessible from anywhere in the world via authenticated HTTP/HTTPS calls	Prices for LRS file storage start at $0.045 / GB per month.



Azure Storage Offerings

Blobs
Highly scalable, REST based cloud object store

Block Blobs: Sequential file I/O

Page Blobs: Random-write pattern data

Tables
Massive auto-scaling NoSQL store

Dynamic scaling based on load

Scale to PBs of table data

Fast key/value lookups

Queues
Reliable queues at scale for cloud services

Decouple and scale components

Message visibility timeout and update message to protect against unreliable dequeuers

Disks
Persistent disks for Azure IaaS VMs

Built on page blobs

Premium Storage Disks: SSD based, high IOPS, low latency

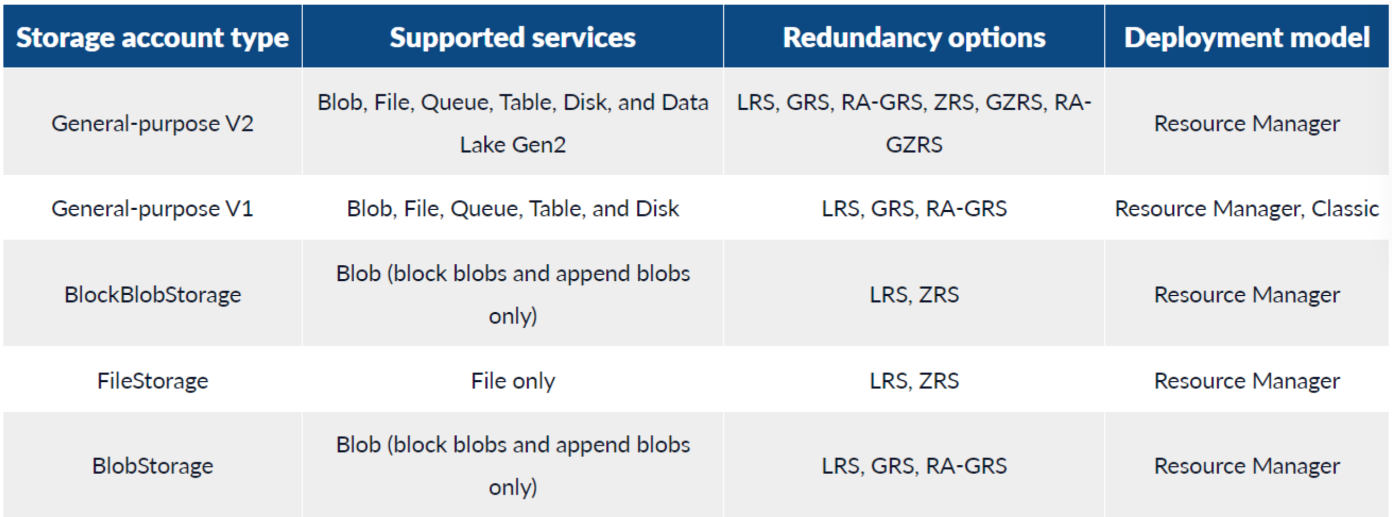
Files
Fully Managed File Shares in the Cloud

Map to file share, standard file system semantics

"Lift and shift" legacy apps

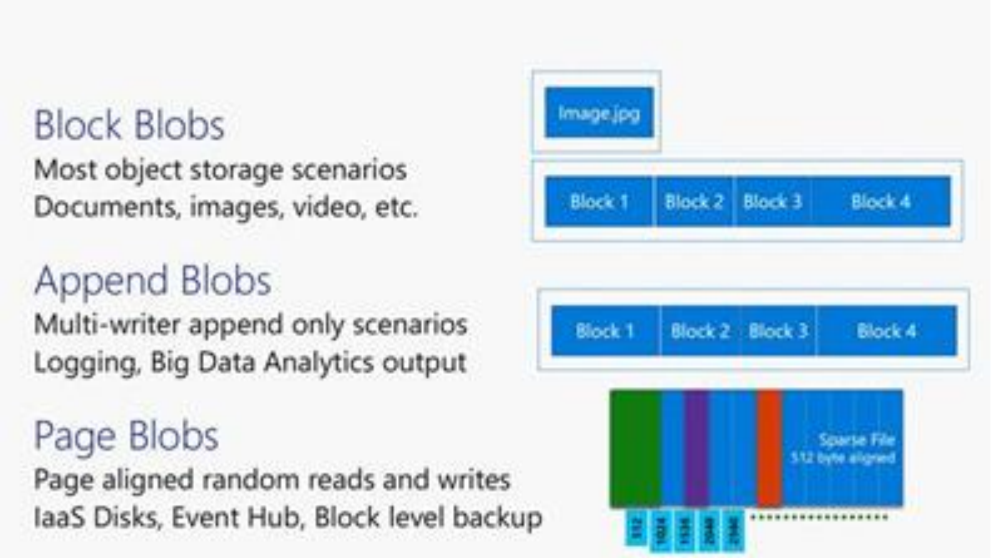
Code against (REST API)

Use on Windows & Linux VMs



Storage account type	Supported services	Redundancy options	Deployment model
General-purpose V2	Blob, File, Queue, Table, Disk, and Data Lake Gen2	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS	Resource Manager
General-purpose V1	Blob, File, Queue, Table, and Disk	LRS, GRS, RA-GRS	Resource Manager, Classic
BlockBlobStorage	Blob (block blobs and append blobs only)	LRS, ZRS	Resource Manager
FileStorage	File only	LRS, ZRS	Resource Manager
BlobStorage	Blob (block blobs and append blobs only)	LRS, GRS, RA-GRS	Resource Manager

Types of Blobs



Block Blobs
Most object storage scenarios
Documents, images, video, etc.

Append Blobs
Multi-writer append only scenarios
Logging, Big Data Analytics output

Page Blobs
Page aligned random reads and writes
IaaS Disks, Event Hub, Block level backup

Azure Files

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) and Network File System protocols (NFS).

Azure file shares can be mounted concurrently by cloud or on-premises deployments of Windows, Linux, and macOS.

Applications running in Azure virtual machines or cloud services can mount a file storage share to access file data, just as a desktop application would mount a typical SMB share.

Use Azure Files for the following situations:

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

Azure Files provides the capability to take share snapshots of file shares. Share snapshots give you an extra level of security, and help reduce the risk of data corruption or accidental deletion. You can also use share snapshots as a general backup for disaster recovery.

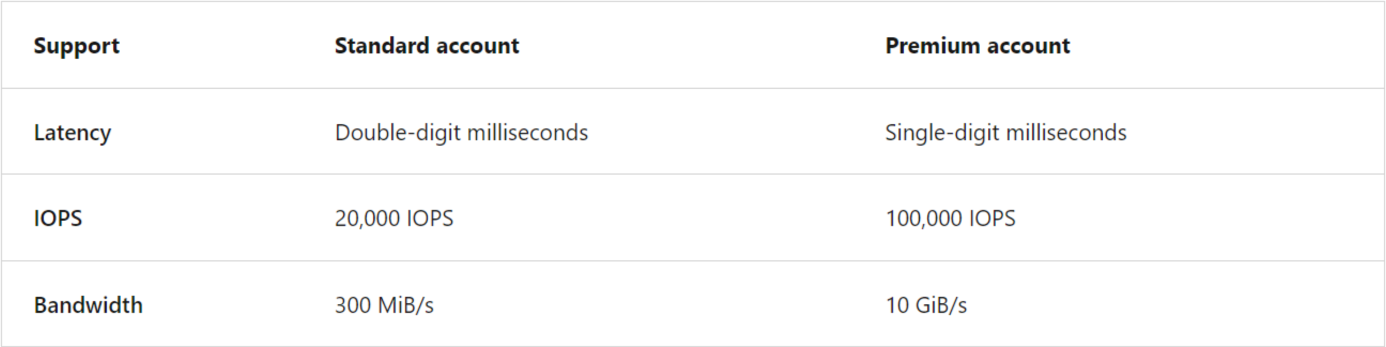
You can automate and manage your Azure file shares snapshots. Automating snapshot backups with Azure Backup is the recommended approach.

When Azure Backup is enabled on the file share, the soft delete feature is also enabled.

You can configure snapshot backups for daily, weekly, monthly or yearly retention, according to your requirements.

Azure file shares can be used in two ways. You can directly mount serverless Azure file shares (SMB) or cache Azure file shares on-premises by using Azure File Sync.

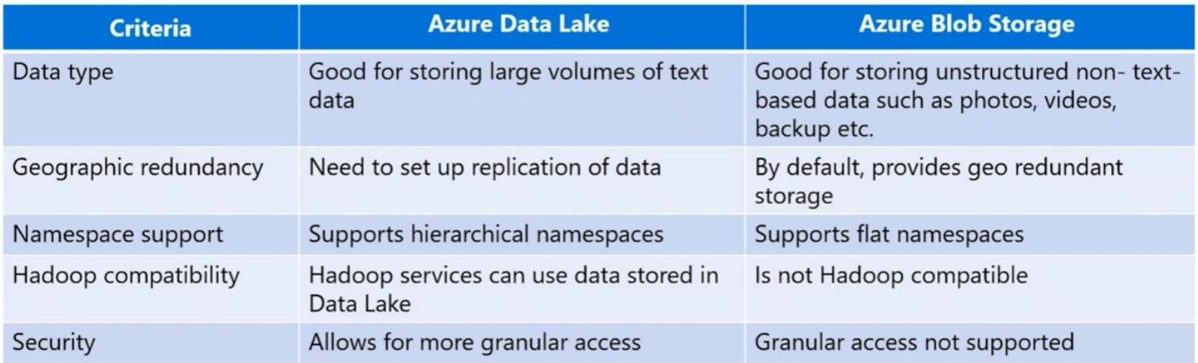
- **Direct mount of Azure file shares**: Because Azure Files provides SMB access, you can mount Azure file shares on-premises or in the cloud. Mounting uses the standard SMB client available in Windows, macOS, and Linux. Because Azure file shares are serverless, deploying for production scenarios doesn't require managing a file server or NAS device. Direct mounting means you don't have to apply software patches or swap out physical disks.
- **Cache Azure file shares on-premises with Azure File Sync**: Azure File Sync lets you centralize your organization's file shares. Azure Files provides the flexibility, performance, and compatibility of an on-premises file server. Azure File Sync transforms an on-premises (or cloud) Windows Server into a quick cache of your Azure file share.



Support	Standard account	Premium account
Latency	Double-digit milliseconds	Single-digit milliseconds
IOPS	20,000 IOPS	100,000 IOPS
Bandwidth	300 MiB/s	10 GiB/s

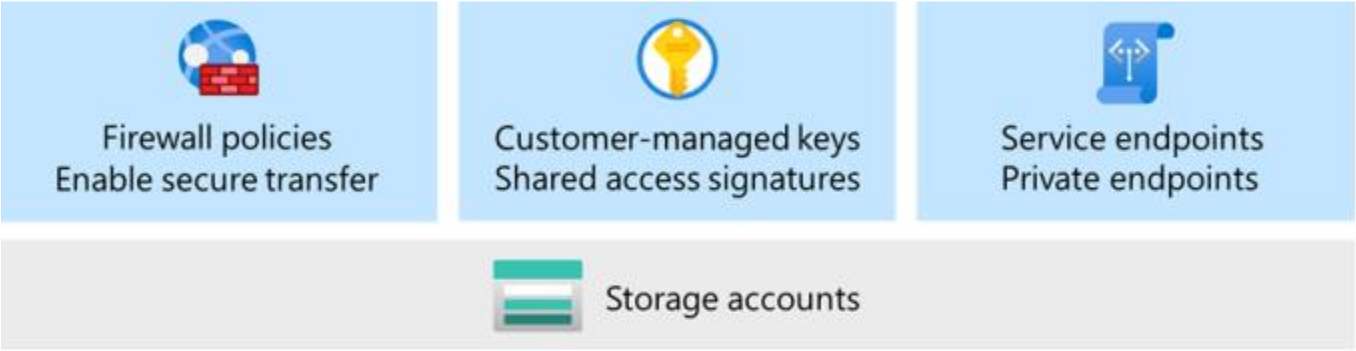
Azure Data Lake

Compare Azure Data Lake to Azure Blob Storage



Criteria	Azure Data Lake	Azure Blob Storage
Data type	Good for storing large volumes of text data	Good for storing unstructured non- text-based data such as photos, videos, backup etc.
Geographic redundancy	Need to set up replication of data	By default, provides geo redundant storage
Namespace support	Supports hierarchical namespaces	Supports flat namespaces
Hadoop compatibility	Hadoop services can use data stored in Data Lake	Is not Hadoop compatible
Security	Allows for more granular access	Granular access not supported

Storage Security



Firewall policies
Enable secure transfer

Customer-managed keys
Shared access signatures

Service endpoints
Private endpoints

Storage accounts

Azure security baseline for Azure Storage grants limited access to Azure Storage resources. Azure security baseline provides a comprehensive list of ways to secure your Azure storage.

Shared access signatures (SAS) provide secure delegated access to resources in your storage account. With a SAS, you have granular control over how a client can access your data.

Firewall policies and rules limit access to your storage account. Requests can be limited to specific IP addresses or ranges, or to a list of subnets in an Azure virtual network. The Azure Storage firewall provides access control for the public endpoint of your storage account.

Virtual network service endpoints restrict network access and provide direct connection to your Azure storage. You can secure storage accounts to your virtual network, and enable private IP addresses in the virtual network to reach the service endpoint. With private endpoints, you can create a special network interface for an Azure service in your virtual network.

Secure transfer enables an Azure storage account to accept requests from secure connections. When you require secure transfer, any requests originating from non-secure connections are rejected. Microsoft recommends that you always require secure transfer for all your storage accounts.

Data in your storage account **is automatically encrypted**. Azure Storage encryption offers two ways to manage encryption keys at the storage account level:

- Microsoft-managed keys: By default, Microsoft manages the keys used to encrypt your storage account.

- Customer-managed keys: You can optionally choose to manage encryption keys for your storage account. Customer-managed keys must be stored in Azure Key Vault.

Azure Virtual Machines (VM)

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

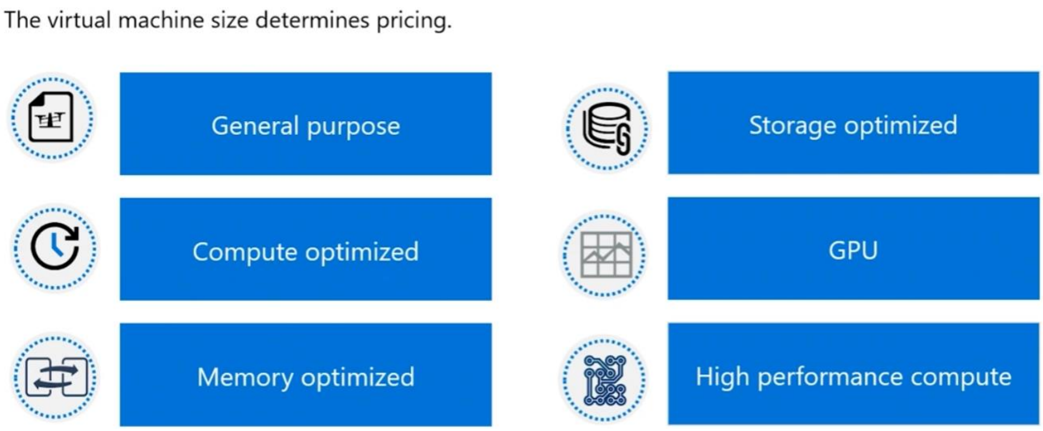
- Total control over the operating system (OS).
- The ability to run custom software.
- To use custom hosting configurations.

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

Examples of when to use VMs

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

Determine the virtual machine family



The virtual machine size determines pricing.

General purpose

Storage optimized

Compute optimized

GPU

Memory optimized

High performance compute

Azure Virtual Machines scale sets

Azure Virtual Machine Scale Sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule.

Scale sets provide the following key benefits:

- **Easy to create and manage multiple VMs**
    - When you have many VMs that run your application, it's important to maintain a consistent configuration across your environment. For reliable performance of your application, the VM size, disk configuration, and application installs should match across all VMs.
    - With scale sets, all VM instances are created from the same base OS image and configuration. This approach lets you easily manage hundreds of VMs without extra configuration tasks or network management.
    - Scale sets support the use of the Azure load balancer for basic layer-4 traffic distribution, and Azure Application Gateway for more advanced layer-7 traffic distribution and TLS termination.

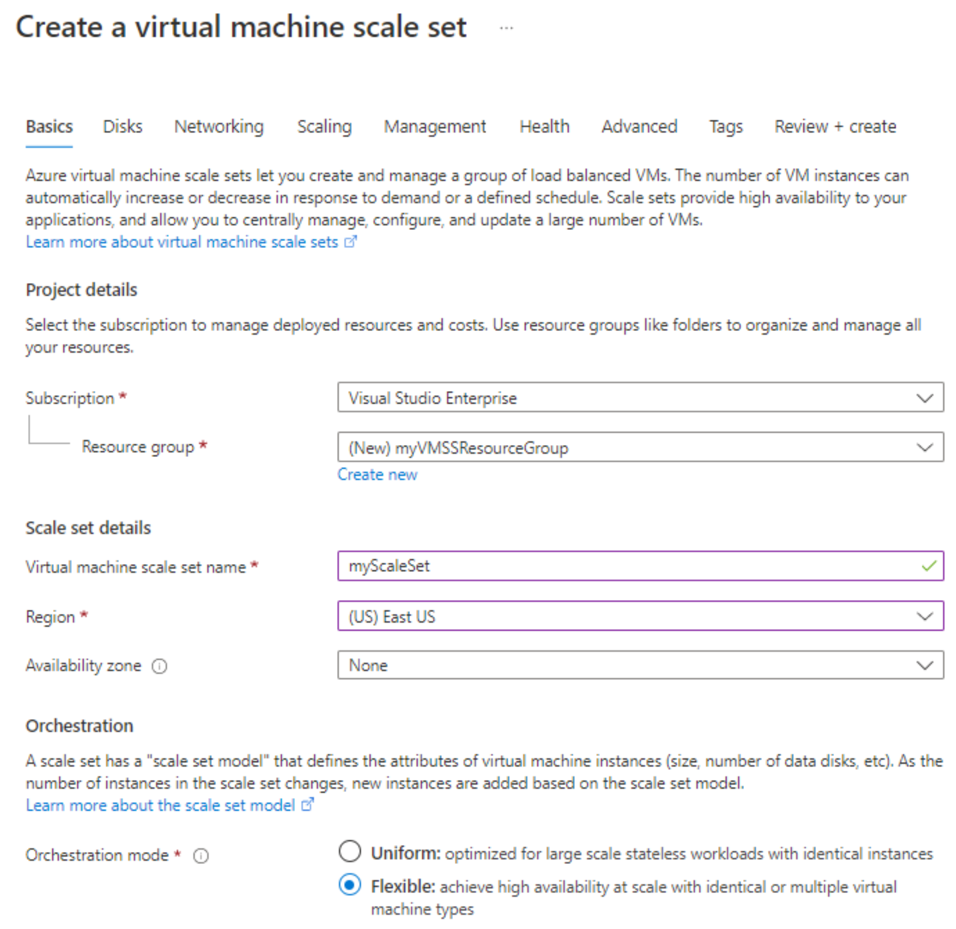
- **Provides high availability and application resiliency** by distributing VMs across availability zones or fault domains
    - Scale sets are used to run multiple instances of your application. If one of these VM instances has a problem, customers continue to access your application through one of the other VM instances with minimal interruption.
    - For more availability, you can use Availability Zones to automatically distribute VM instances in a scale set within a single datacenter or across multiple datacenters.

- Allows your application to **automatically scale** as resource demand changes
    - To match customer demand, scale sets can automatically increase the number of VM instances as application demand increases, then reduce the number of VM instances as demand decreases.
    - Autoscale also minimizes the number of unnecessary VM instances that run your application when demand is low, while customers continue to receive an acceptable level of performance as demand grows and additional VM instances are automatically added.
    - This ability helps reduce costs and efficiently create Azure resources as required.

- **Works at large-scale**
    - Scale sets support up to 1,000 VM instances for standard marketplace images and custom images through the Azure Compute Gallery (formerly known as Shared Image Gallery). If you create a scale set using a managed image, the limit is 600 VM instances.
    - For the best performance with production workloads, use Azure Managed Disks.

With Flexible orchestration, Azure provides a unified experience across the Azure VM ecosystem. Flexible orchestration offers high availability guarantees (up to 1000 VMs) by spreading VMs across fault domains in a region or within an Availability Zone.

In the Azure portal search bar, search for and select **Virtual Machine Scale Sets**.



Create a virtual machine scale set ...

Azure virtual machine scale sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update a large number of VMs.
Learn more about virtual machine scale sets

Project details

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

Subscription *	Visual Studio Enterprise
Resource group *	(New) myVMSSResourceGroup Create new

Scale set details

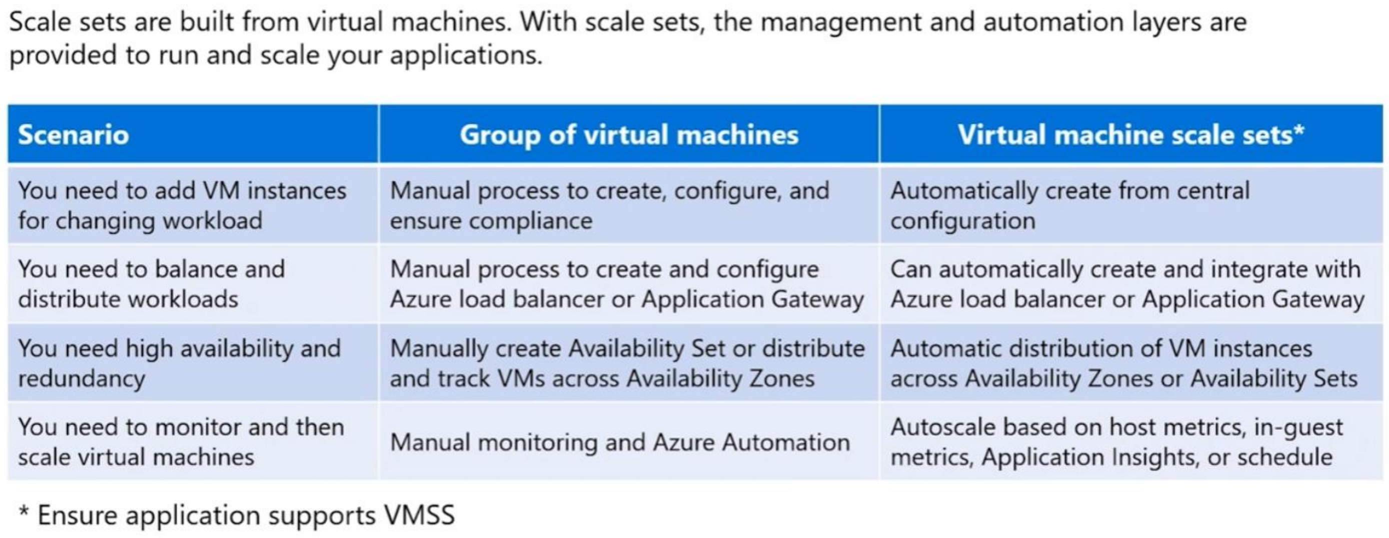
Virtual machine scale set name *	myScaleSet ✓
Region *	(US) East US
Availability zone ⓘ	None

Orchestration

A scale set has a "scale set model" that defines the attributes of virtual machine instances (size, number of data disks, etc). As the number of instances in the scale set changes, new instances are added based on the scale set model.
Learn more about the scale set model

Orchestration mode * ⓘ
○ **Uniform:** optimized for large scale stateless workloads with identical instances
● **Flexible:** achieve high availability at scale with identical or multiple virtual machine types

When to select virtual machines scale sets



Scale sets are built from virtual machines. With scale sets, the management and automation layers are provided to run and scale your applications.

Scenario	Group of virtual machines	Virtual machine scale sets*
You need to add VM instances for changing workload	Manual process to create, configure, and ensure compliance	Automatically create from central configuration
You need to balance and distribute workloads	Manual process to create and configure Azure load balancer or Application Gateway	Can automatically create and integrate with Azure load balancer or Application Gateway
You need high availability and redundancy	Manually create Availability Set or distribute and track VMs across Availability Zones	Automatic distribution of VM instances across Availability Zones or Availability Sets
You need to monitor and then scale virtual machines	Manual monitoring and Azure Automation	Autoscale based on host metrics, in-guest metrics, Application Insights, or schedule

* Ensure application supports VMSS

- To provide redundancy and improved performance, applications are typically distributed across multiple instances. Customers may access your application through a load balancer that distributes requests to one of the application instances.
- If you need to perform maintenance or update an application instance, your customers must be distributed to another available application instance.
- Azure Virtual Machine Scale Sets provide the management capabilities for applications that run across many VMs, automatic scaling of resources, and load balancing of traffic.

Azure Batch

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

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

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

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

Azure Virtual Desktop (AVD)

Azure Virtual Desktop is a desktop and application virtualization service that runs on the cloud.

It enables your users to use a cloud-hosted version of Windows from any location.

Azure Virtual Desktop works across devices like Windows, Mac, iOS, Android, and Linux. It works with apps that you can use to access remote desktops and apps.

You can also use most modern browsers to access Azure Virtual Desktop-hosted experiences.

Users have the freedom to connect to Azure Virtual Desktop with any device over the internet.

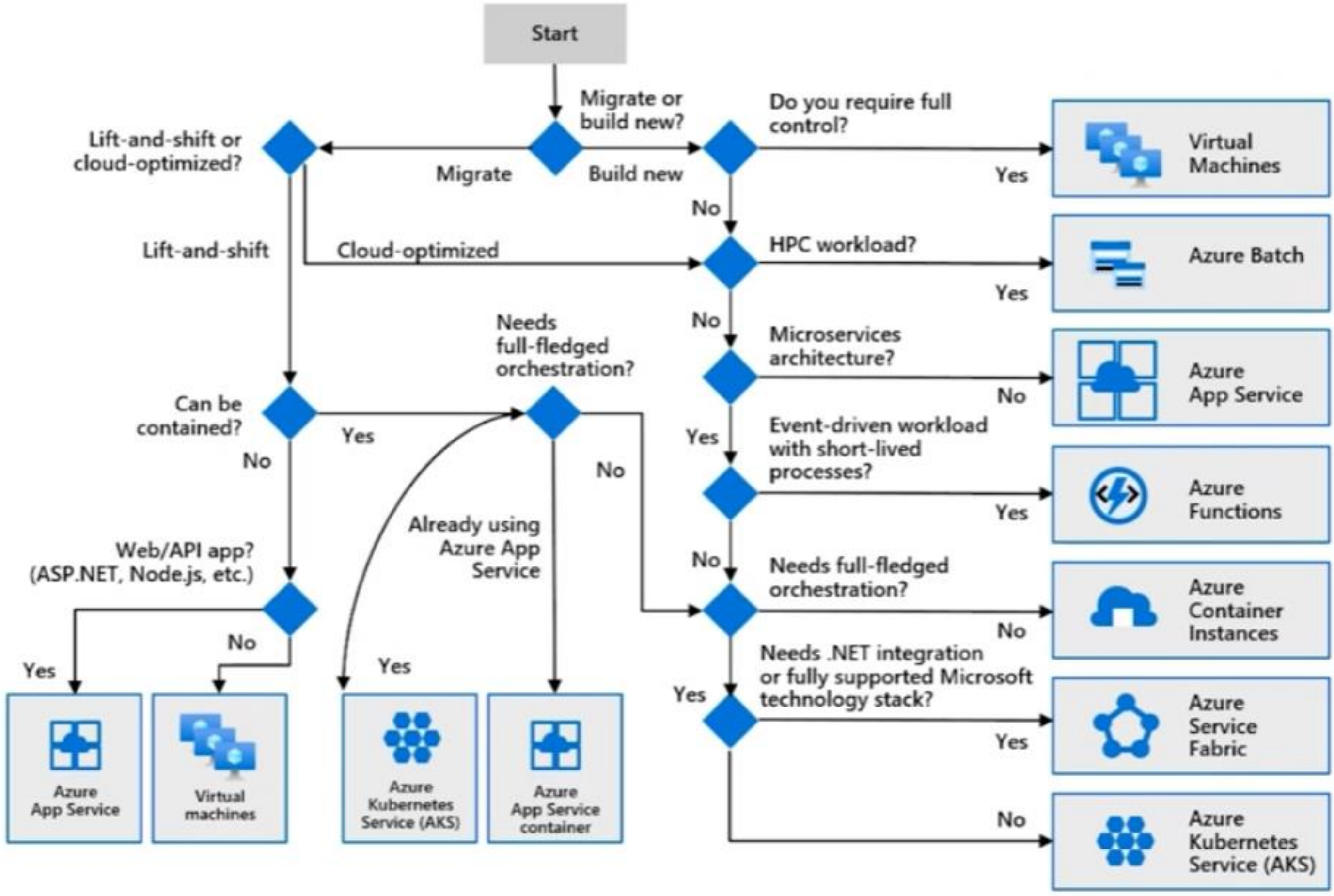
They use a Azure Virtual Desktop client to connect to their published Windows desktop and applications. This client could either be a native application on the device or the Azure Virtual Desktop HTML5 web client.

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

You can enable multifactor authentication to secure user sign-ins. You can also secure access to data by assigning granular role-based access controls (RBACs) to users.

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

Choose a compute service for your application


Start
Migrate or build new?
Migrate
Build new
Lift-and-shift or cloud-optimized?
Lift-and-shift
Cloud-optimized
Do you require full control?
Yes
Virtual Machines
No
HPC workload?
Yes
Azure Batch
No
Microservices architecture?
No
Azure App Service
Yes
Event-driven workload with short-lived processes?
Yes
Azure Functions
No
Needs full-fledged orchestration?
No
Azure Container Instances
Yes
Needs .NET integration or fully supported Microsoft technology stack?
Yes
Azure Service Fabric
No
Azure Kubernetes Service (AKS)
Can be contained?
Yes
No
Needs full-fledged orchestration?
No
Already using Azure App Service
Yes
Web/API app? (ASP.NET, Node.js, etc.)
Yes
No
Azure App Service
Virtual machines
Azure Kubernetes Service (AKS)
Azure App Service container

Comparing Container Apps with other Azure container options

Azure Container Apps

Azure Container Apps enables you to build serverless microservices based on containers.

Distinctive features of Container Apps include:

- Optimized for running general purpose containers, especially for applications that span many microservices deployed in containers.
- Powered by Kubernetes and open-source technologies like Dapr, KEDA, and envoy.
- Supports Kubernetes-style apps and microservices with features like service discovery and traffic splitting.
- Enables event-driven application architectures by supporting scale based on traffic and pulling from event sources like queues, including scale to zero.
- Support of long running processes and can run background tasks.

Azure Container Apps doesn't provide direct access to the underlying Kubernetes APIs.

If you require access to the Kubernetes APIs and control plane, you should use Azure Kubernetes Service.

However, if you would like to build Kubernetes-style applications and don't require direct access to all the native Kubernetes APIs and cluster management, Container Apps provides a fully managed experience based on best-practices.

For these reasons, many teams may prefer to start building container microservices with Azure Container Apps.

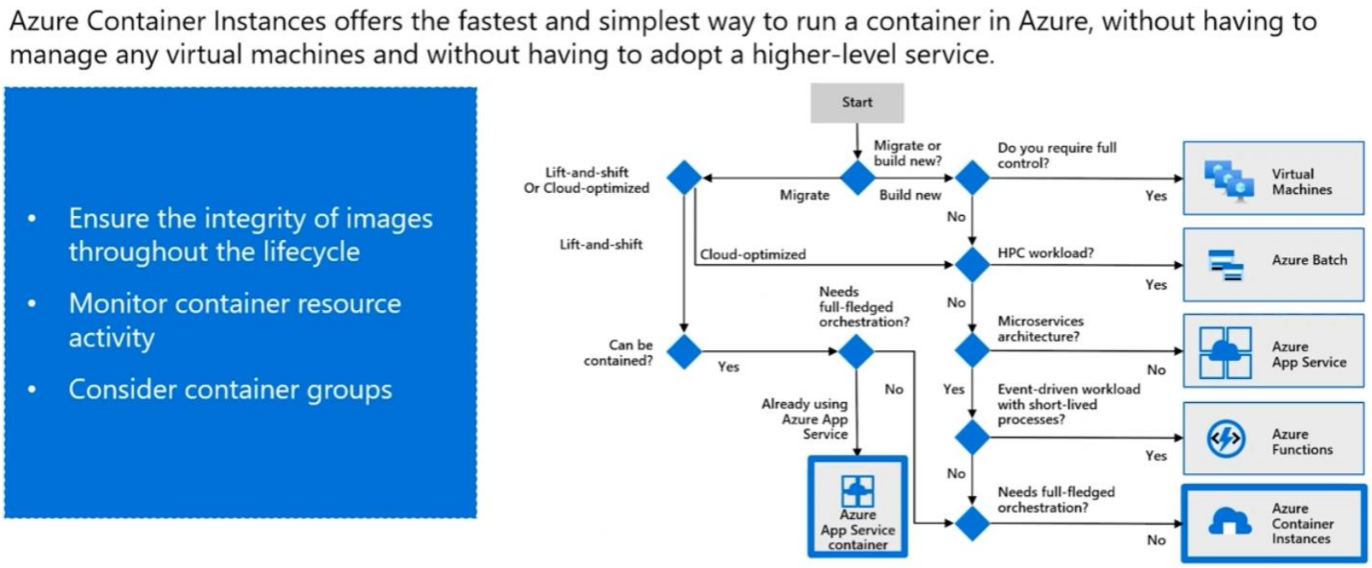
Azure App Service

- Azure App Service provides fully managed hosting for web applications including websites and web APIs.

- These web applications may be deployed using code or containers.
- Azure App Service is optimized for web applications.
- Azure App Service is integrated with other Azure services including Azure Container Apps or Azure Functions.
- With App Service, you can host most common app service styles like:
  - Web apps
  - API apps
  - WebJobs
  - Mobile apps
- App Service handles most of the infrastructure decisions you deal with in hosting web-accessible apps:
  - Deployment and management are integrated into the platform.
  - Endpoints can be secured.
  - Sites can be scaled quickly to handle high traffic loads.
  - The built-in load balancing and traffic manager provide high availability.

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

Azure Container Instances

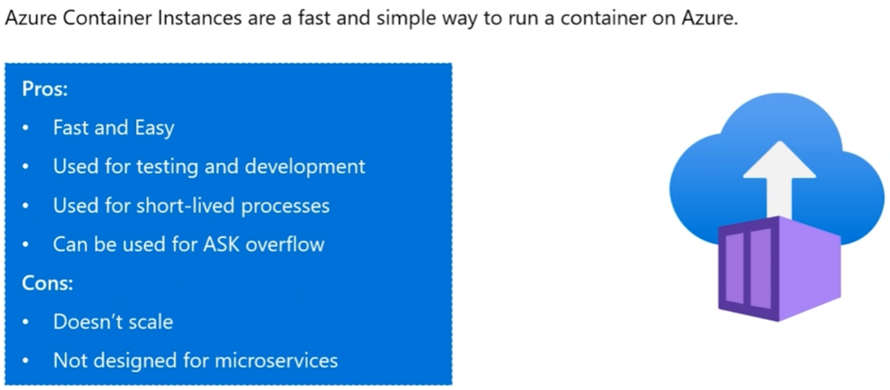


Azure Container Instances offers the fastest and simplest way to run a container in Azure, without having to manage any virtual machines and without having to adopt a higher-level service.

- Ensure the integrity of images throughout the lifecycle
- Monitor container resource activity
- Consider container groups

Start
Migrate or build new?
Lift-and-shift Or Cloud-optimized
Migrate
Build new
Do you require full control?
Yes
Virtual Machines
No
Lift-and-shift
Cloud-optimized
HPC workload?
Yes
Azure Batch
No
Needs full-fledged orchestration?
Microservices architecture?
Can be contained?
Yes
No
Azure App Service
Yes
Event-driven workload with short-lived processes?
Already using Azure App Service
Yes
Azure Functions
No
Azure App Service container
Needs full-fledged orchestration?
No
Azure Container Instances

- Azure Container Instances (ACI) provides a single pod of Hyper-V isolated containers on demand.
- It can be thought of as a lower-level "building block" option compared to Container Apps.
- Concepts like scale, load balancing, and certificates are not provided with ACI containers. For example, to scale to five container instances, you create five distinct container instances.
- Azure Container Apps provide many application-specific concepts on top of containers, including certificates, revisions, scale, and environments.
- Users often interact with Azure Container Instances through other services. For example, Azure Kubernetes Service can layer orchestration and scale on top of ACI through virtual nodes.
- If you need a less "opinionated" building block that doesn't align with the scenarios Azure Container Apps is optimizing for, Azure Container Instances is an ideal option.



Azure Container Instances are a fast and simple way to run a container on Azure.

Pros:

- Fast and Easy
- Used for testing and development
- Used for short-lived processes
- Can be used for ASK overflow

Cons:

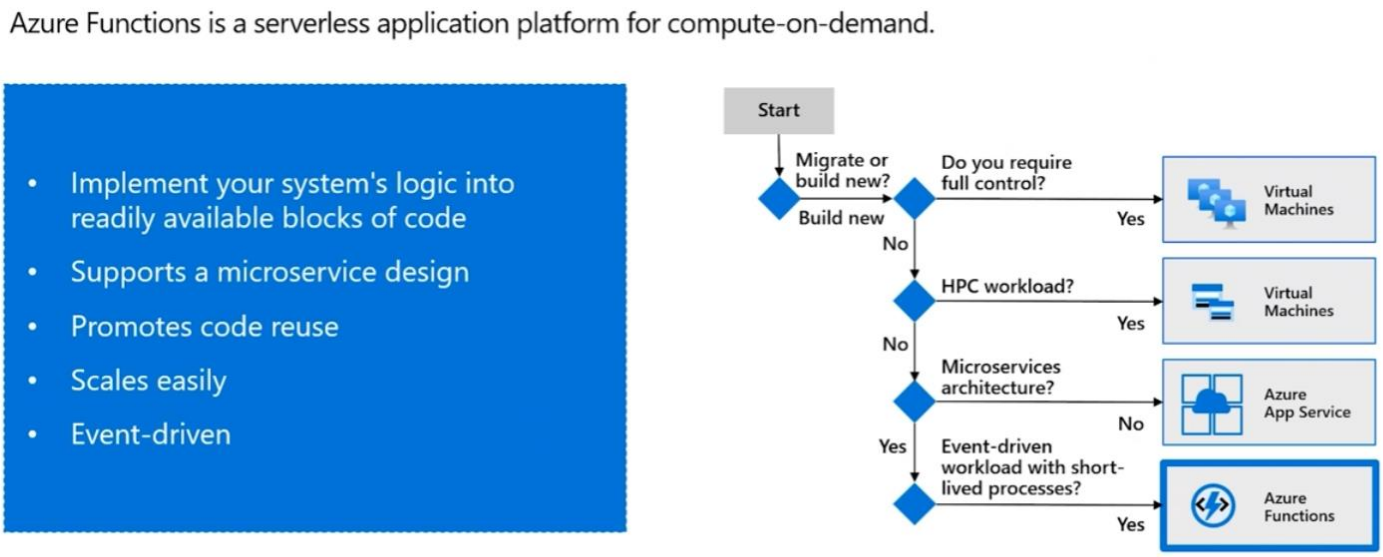
- Doesn't scale
- Not designed for microservices

Azure Kubernetes Service

- Azure Kubernetes Service (AKS) provides a fully managed Kubernetes option in Azure.
- It supports direct access to the Kubernetes API and runs any Kubernetes workload.
- The full cluster resides in your subscription, with the cluster configurations and operations within your control and responsibility.
- Teams looking for a fully managed version of Kubernetes in Azure, Azure Kubernetes Service is an ideal option.

Azure Functions

- Azure Functions is a serverless Functions-as-a-Service (FaaS) solution.
- It's optimized for running event-driven applications using the functions programming model.
- It shares many characteristics with Azure Container Apps around scale and integration with events, but optimized for ephemeral functions deployed as either code or containers.
- The Azure Functions programming model provides productivity benefits for teams looking to trigger the execution of your functions on events and bind to other data sources.
- When building FaaS-style functions, Azure Functions is the ideal option.
- The Azure Functions programming model is available as a base container image, making it portable to other container based compute platforms allowing teams to reuse code as environment requirements change.
- Functions scale automatically based on demand, so they're a solid choice when demand is variable.
- With functions, Azure runs your code when it's triggered and automatically deallocates resources when the function is finished. In this model, you're only charged for the CPU time used while your function runs.
- When you're concerned only about the code running your service, and not the underlying platform or infrastructure, using Azure Functions is ideal.



Azure Functions is a serverless application platform for compute-on-demand.

- Implement your system's logic into readily available blocks of code
- Supports a microservice design
- Promotes code reuse
- Scales easily
- Event-driven

Azure Logic Apps

Logic apps are similar to functions. Both enable you to trigger logic based on an event.

Where functions execute code, logic apps execute workflows that are designed to automate business scenarios and are built from predefined logic blocks.

Every Azure logic app workflow starts with a trigger, which fires when a specific event happens or when newly available data meets specific criteria.

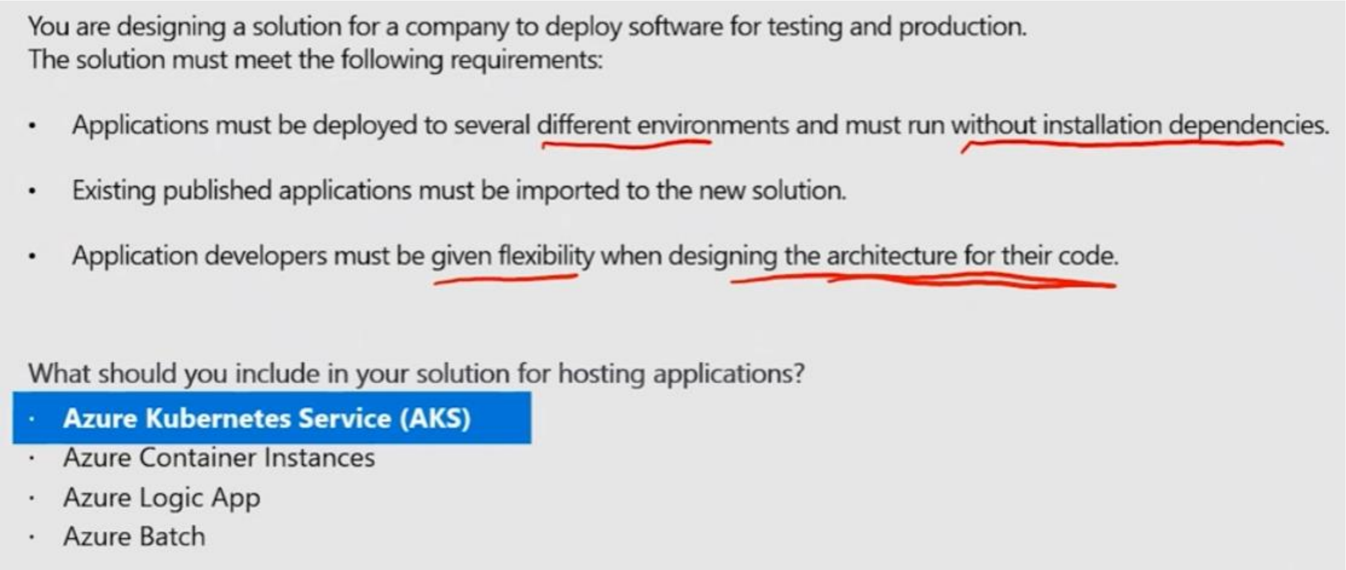
Many triggers include basic scheduling capabilities, so developers can specify how regularly their workloads will run.

Each time the trigger fires, the Logic Apps engine creates a logic app instance that runs the actions in the workflow. These actions can also include data conversions and flow controls, such as conditional statements, switch statements, loops, and branching.

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

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

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



You are designing a solution for a company to deploy software for testing and production.
The solution must meet the following requirements:

- Applications must be deployed to several different environments and must run without installation dependencies.
- Existing published applications must be imported to the new solution.
- Application developers must be given flexibility when designing the architecture for their code.

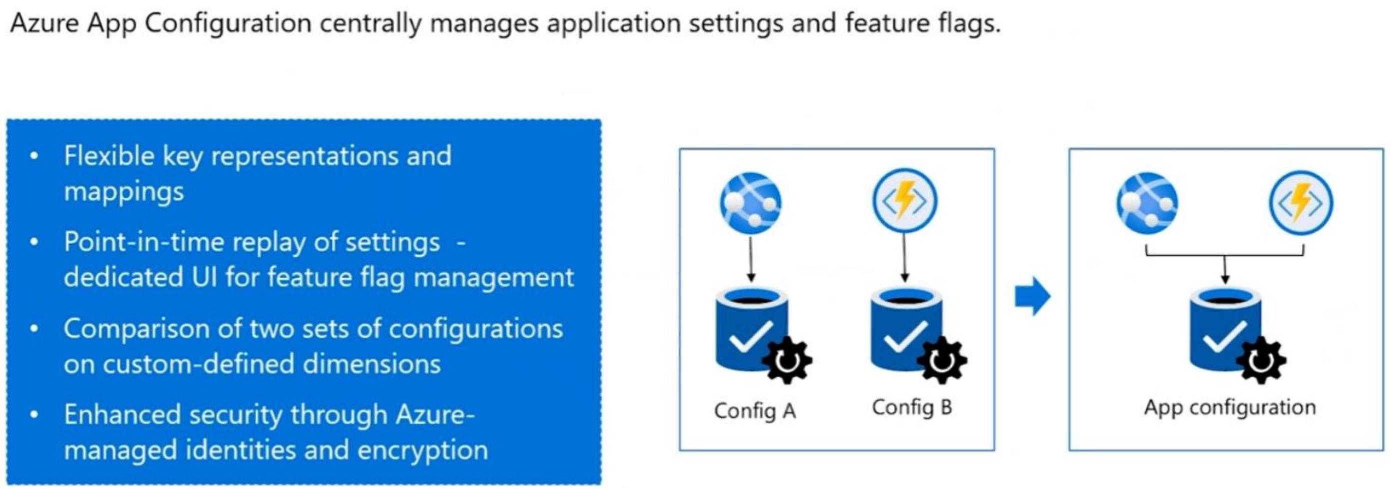
What should you include in your solution for hosting applications?

- [✓ **Azure Kubernetes Service (AKS)**]
- Azure Container Instances
- Azure Logic App
- Azure Batch

Azure Functions vs Azure Logic Apps

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

Azure App Configuration



Azure App Configuration centrally manages application settings and feature flags.

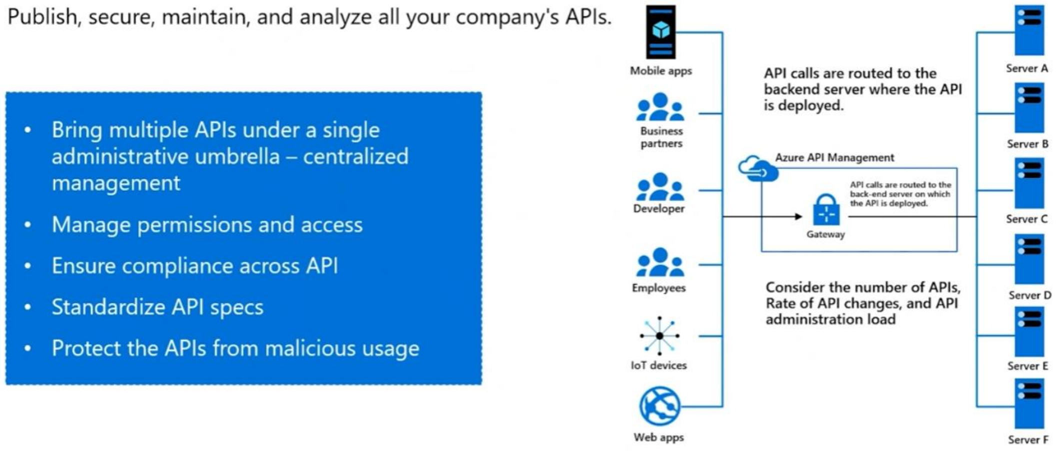
- Flexible key representations and mappings
- Point-in-time replay of settings - dedicated UI for feature flag management
- Comparison of two sets of configurations on custom-defined dimensions
- Enhanced security through Azure-managed identities and encryption

API Management

Overview

Azure API Management is a hybrid, multi-cloud management platform for APIs across all environments.

As a platform-as-a-service, API Management supports the complete API lifecycle.



Publish, secure, maintain, and analyze all your company's APIs.

- Bring multiple APIs under a single administrative umbrella – centralized management
- Manage permissions and access
- Ensure compliance across API
- Standardize API specs
- Protect the APIs from malicious usage

API calls are routed to the backend server where the API is deployed.

API calls are routed to the back-end server on which the API is deployed.

Consider the number of APIs, Rate of API changes, and API administration load

Azure API Management helps customers meet these challenges:

- Abstract backend architecture diversity and complexity from API consumers
- Securely expose services hosted on and outside of Azure as APIs
- Protect, accelerate, and observe APIs
- Enable API discovery and consumption by internal and external users

Common scenarios include:

- **Unlocking legacy assets** - APIs are used to abstract and modernize legacy backends and make them accessible from new cloud services and modern applications. APIs allow innovation without the risk, cost, and delays of migration.
- **API-centric app integration** - APIs are easily consumable, standards-based, and self-describing mechanisms for exposing and accessing data, applications, and processes. They simplify and reduce the cost of app integration.
- **Multi-channel user experiences** - APIs are frequently used to enable user experiences such as web, mobile, wearable, or Internet of Things applications. Reuse APIs to accelerate development and ROI.
- **B2B integration** - APIs exposed to partners and customers lower the barrier to integrate business processes and exchange data between business entities. APIs eliminate the overhead inherent in point-to-point integration. Especially with self-service discovery and onboarding enabled, APIs are the primary tools for scaling B2B integration.

Key concepts

- **APIs**: Each API represents a set of operations available to app developers. Each API contains a reference to the backend service that implements the API, and its operations map to backend operations. Operations in API Management are highly configurable, with control over URL mapping, query and path parameters, request and response content, and operation response caching.

- **Products:** Products are how APIs are surfaced to developers. Products in API Management have one or more APIs, and can be open or protected.
    - Protected products require a subscription key, while open products can be consumed freely.

    - When a product is ready for use by developers, it can be published. Once published, it can be viewed or subscribed to by developers. Subscription approval is configured at the product level and can either require an administrator's approval or be automatic.

- **Groups:** Groups are used to manage the visibility of products to developers. API Management has the following built-in groups:
    - Administrators - Manage API Management service instances and create the APIs, operations, and products that are used by developers. Azure subscription administrators are members of this group.
    - Developers - Authenticated developer portal users that build applications using your APIs. Developers are granted access to the developer portal and build applications that call the operations of an API.
    - Guests - Unauthenticated developer portal users, such as prospective customers visiting the developer portal. They can be granted certain read-only access, such as the ability to view APIs but not call them.

- **Policies:** With policies, an API publisher can change the behavior of an API through configuration.
    - Policies are a collection of statements that are executed sequentially on the request or response of an API.
    - Popular statements include format conversion from XML to JSON and call-rate limiting to restrict the number of incoming calls from a developer.
    - Policies can be applied at different scopes, depending on your needs: global (all APIs), a product, a specific API, or an API operation.

Components

API gateway

All requests from client applications first reach the API gateway, which then forwards them to respective backend services. The API gateway acts as a facade to the backend services, allowing API providers to abstract API implementations and evolve backend architecture without impacting API consumers. The gateway enables consistent configuration of routing, security, throttling, caching, and observability.

Specifically, the gateway:

- Acts as a facade to backend services by accepting API calls and routing them to appropriate backends
- Verifies API keys and other credentials such as JWT tokens and certificates presented with requests
- Enforces usage quotas and rate limits
- Optionally transforms requests and responses as specified in policy statements
- If configured, caches responses to improve response latency and minimize the load on backend services
- Emits logs, metrics, and traces for monitoring, reporting, and troubleshooting

Management plane

API providers interact with the service through the management plane, which provides full access to the API Management service capabilities.

Customers interact with the management plane through Azure tools including the Azure portal, Azure PowerShell, Azure CLI, a Visual Studio Code extension, or client SDKs in several popular programming languages.

Use the management plane to:

- Provision and configure API Management service settings
- Define or import API schemas from a wide range of sources, including OpenAPI specifications, Azure compute services, or WebSocket or GraphQL backends

- Package APIs into products
- Set up policies like quotas or transformations on the APIs
- Get insights from analytics
- Manage users

Developer portal

The open-source developer portal is an automatically generated, fully customizable website with the documentation of your APIs.

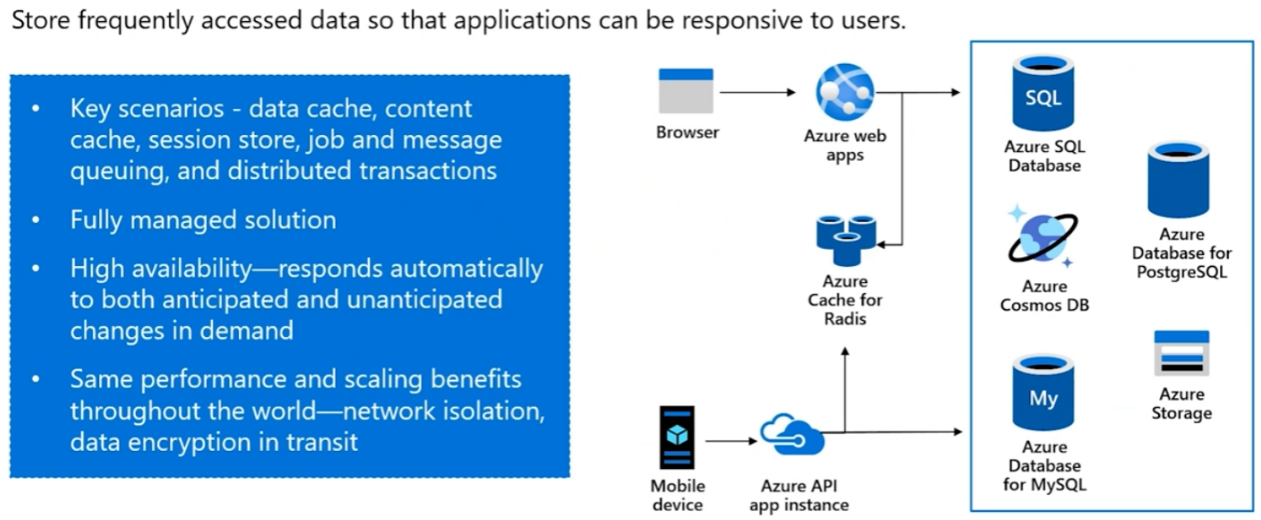
API providers can customize the look and feel of the developer portal by adding custom content, customizing styles, and adding their branding. Extend the developer portal further by self-hosting.

App developers use the open-source developer portal to discover the APIs, onboard to use them, and learn how to consume them in applications. (APIs can also be exported to the Power Platform for discovery and use by citizen developers.)

Using the developer portal, developers can:

- Read API documentation
- Call an API via the interactive console
- Create an account and subscribe to get API keys
- Access analytics on their own usage
- Download API definitions
- Manage API keys

Azure Cache for Redis



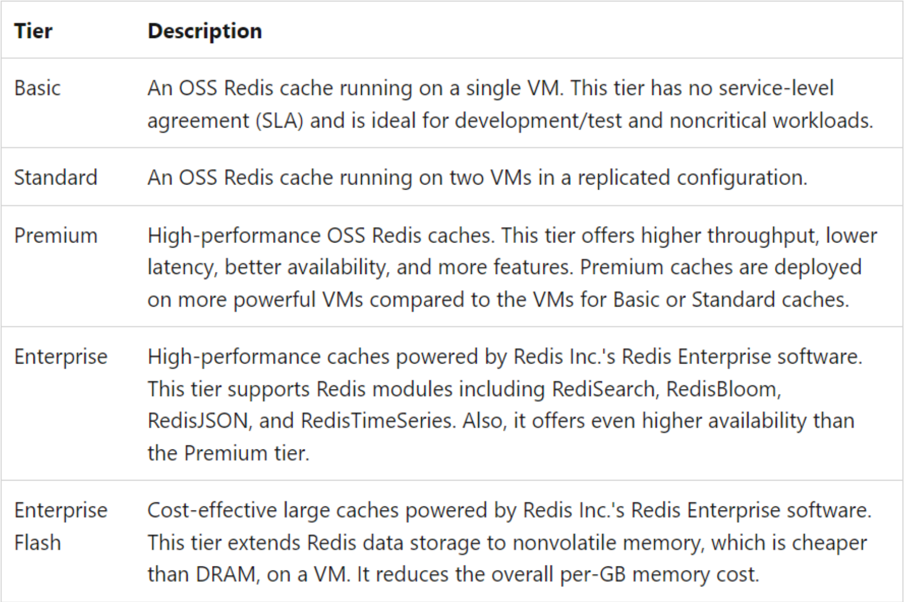
Store frequently accessed data so that applications can be responsive to users.

- Key scenarios - data cache, content cache, session store, job and message queuing, and distributed transactions
- Fully managed solution
- High availability—responds automatically to both anticipated and unanticipated changes in demand
- Same performance and scaling benefits throughout the world—network isolation, data encryption in transit

- 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.
- Key scenarios:
  - **Data cache**: Cache data from databases using cache-aside pattern. When the system makes changes to the data or cached data is too old, Redis updates the data into the cache.
  - **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 and could be cached.

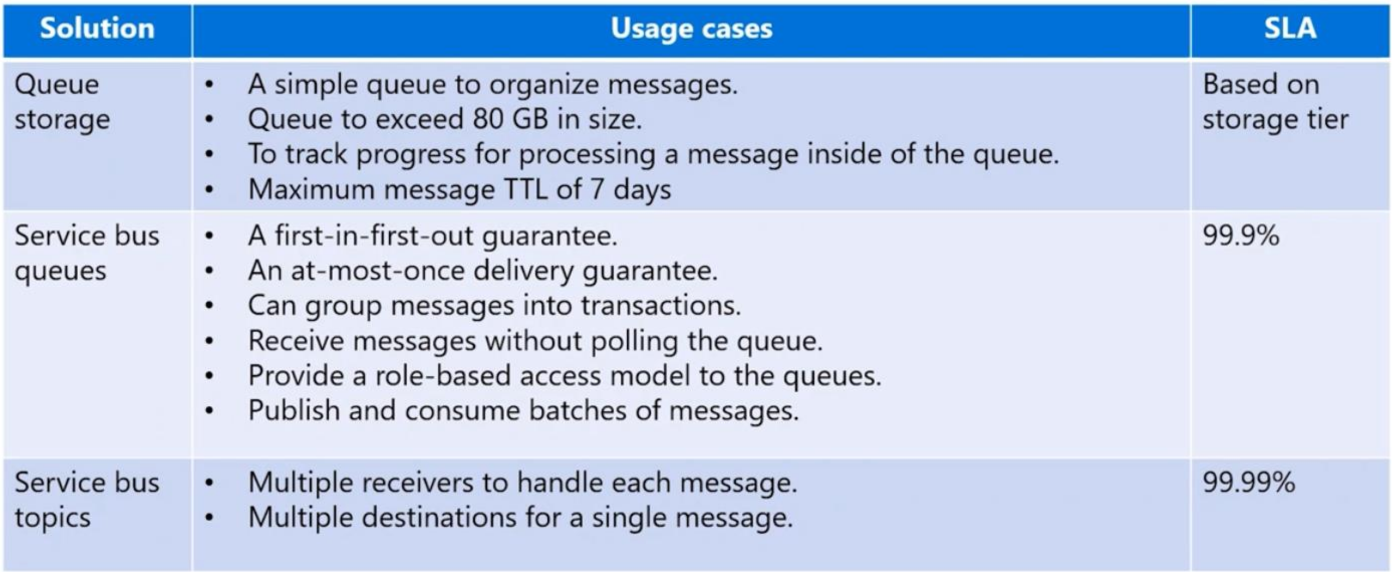
- **Session store:** Commonly used with shopping carts and other user history data that a web application might associate with user cookies.

Service tiers



Tier	Description
Basic	An OSS Redis cache running on a single VM. This tier has no service-level agreement (SLA) and is ideal for development/test and noncritical workloads.
Standard	An OSS Redis cache running on two VMs in a replicated configuration.
Premium	High-performance OSS Redis caches. This tier offers higher throughput, lower latency, better availability, and more features. Premium caches are deployed on more powerful VMs compared to the VMs for Basic or Standard caches.
Enterprise	High-performance caches powered by Redis Inc.'s Redis Enterprise software. This tier supports Redis modules including RediSearch, RedisBloom, RedisJSON, and RedisTimeSeries. Also, it offers even higher availability than the Premium tier.
Enterprise Flash	Cost-effective large caches powered by Redis Inc.'s Redis Enterprise software. This tier extends Redis data storage to nonvolatile memory, which is cheaper than DRAM, on a VM. It reduces the overall per-GB memory cost.

Comparing messaging solutions



Solution	Usage cases	SLA
Queue storage	• A simple queue to organize messages. • Queue to exceed 80 GB in size. • To track progress for processing a message inside of the queue. • Maximum message TTL of 7 days	Based on storage tier
Service bus queues	• A first-in-first-out guarantee. • An at-most-once delivery guarantee. • Can group messages into transactions. • Receive messages without polling the queue. • Provide a role-based access model to the queues. • Publish and consume batches of messages.	99.9%
Service bus topics	• Multiple receivers to handle each message. • Multiple destinations for a single message.	99.99%

Azure Service Bus

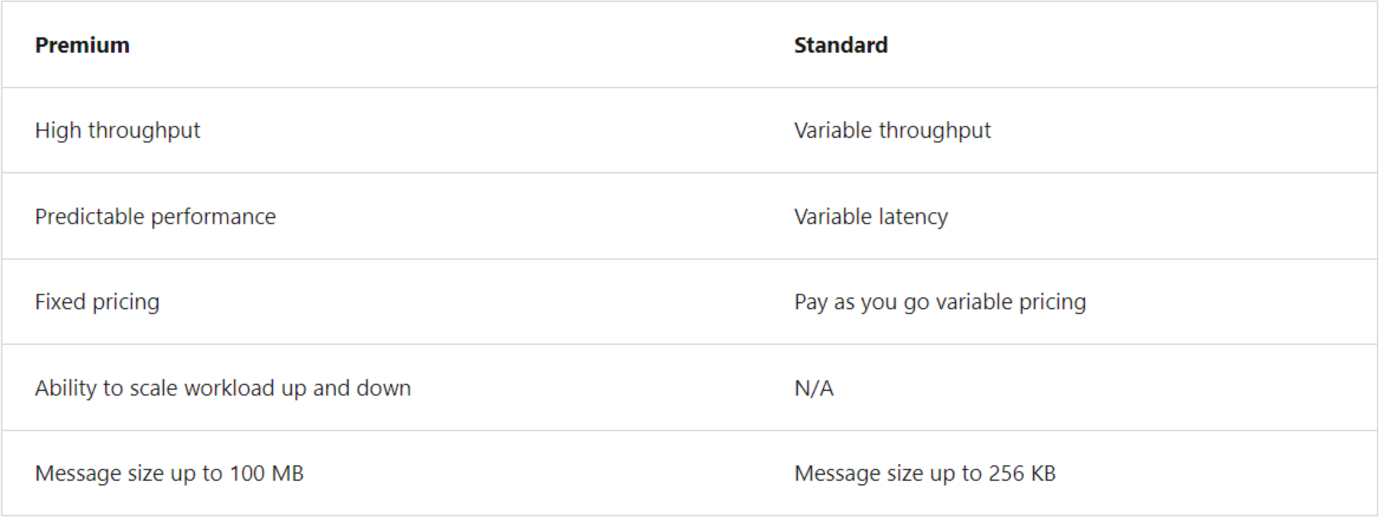
Microsoft Azure Service Bus is a fully managed enterprise integration message broker. Service Bus can decouple applications and services. Data is transferred between different applications and services using messages. A message is a container decorated with metadata, and contains data.

The data can be any kind of information, including structured data encoded with the common formats such as the following ones: JSON, XML, Apache Avro, Plain Text.

Some common messaging scenarios are:

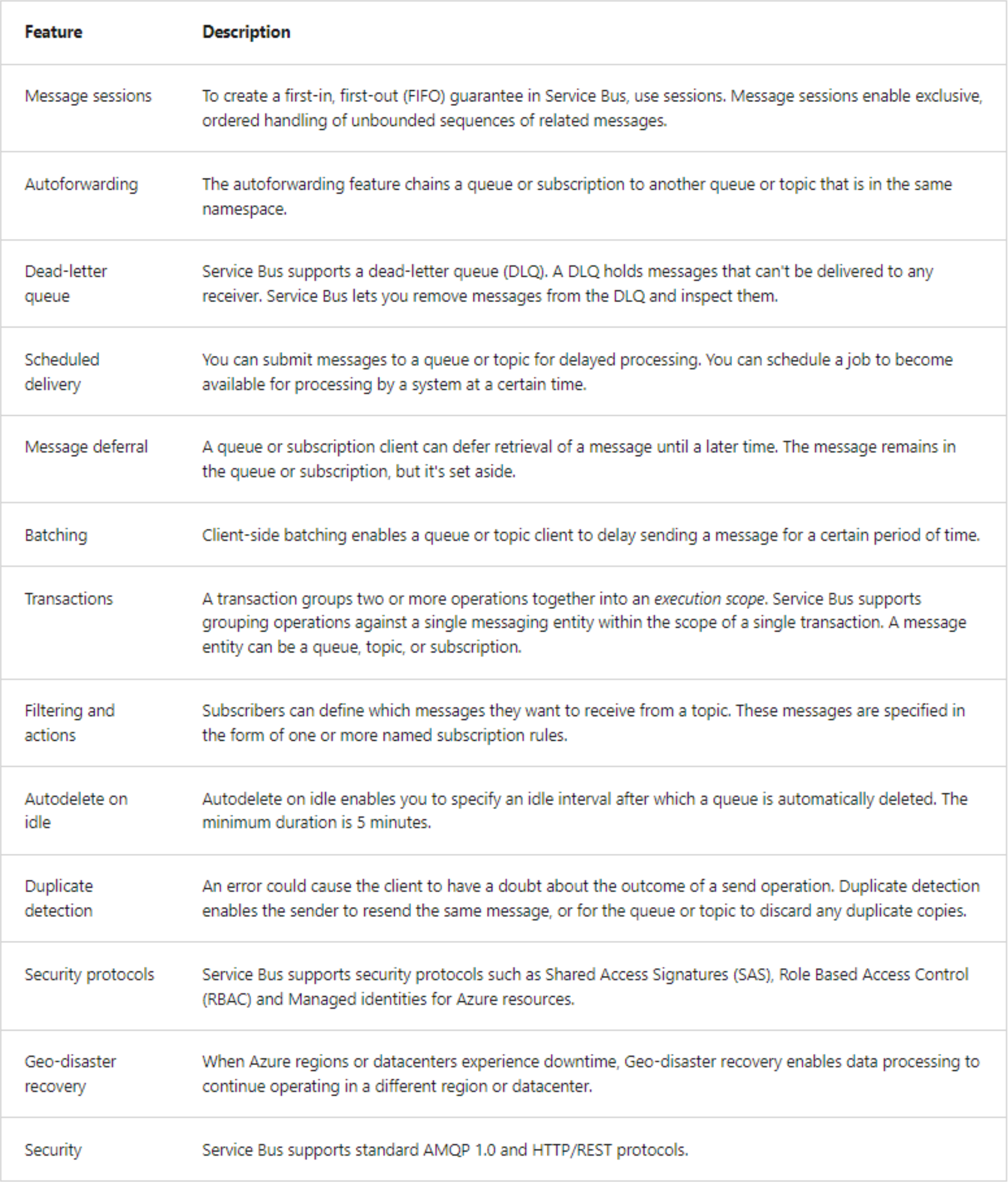
- **Messaging**: Transfer business data, such as sales or purchase orders, journals, or inventory movements.
- **Decouple applications**: Improve reliability and scalability of applications and services. Client and service don't have to be online at the same time.

- **Topics and subscriptions**: Enable 1:n relationships between publishers and subscribers.
- **Message sessions:** Implement workflows that require message ordering or message deferral.



Premium	Standard
High throughput	Variable throughput
Predictable performance	Variable latency
Fixed pricing	Pay as you go variable pricing
Ability to scale workload up and down	N/A
Message size up to 100 MB	Message size up to 256 KB

Advanced Features



Feature	Description
Message sessions	To create a first-in, first-out (FIFO) guarantee in Service Bus, use sessions. Message sessions enable exclusive, ordered handling of unbounded sequences of related messages.
Autoforwarding	The autoforwarding feature chains a queue or subscription to another queue or topic that is in the same namespace.
Dead-letter queue	Service Bus supports a dead-letter queue (DLQ). A DLQ holds messages that can't be delivered to any receiver. Service Bus lets you remove messages from the DLQ and inspect them.
Scheduled delivery	You can submit messages to a queue or topic for delayed processing. You can schedule a job to become available for processing by a system at a certain time.
Message deferral	A queue or subscription client can defer retrieval of a message until a later time. The message remains in the queue or subscription, but it's set aside.
Batching	Client-side batching enables a queue or topic client to delay sending a message for a certain period of time.
Transactions	A transaction groups two or more operations together into an *execution scope*. Service Bus supports grouping operations against a single messaging entity within the scope of a single transaction. A message entity can be a queue, topic, or subscription.
Filtering and actions	Subscribers can define which messages they want to receive from a topic. These messages are specified in the form of one or more named subscription rules.
Autodelete on idle	Autodelete on idle enables you to specify an idle interval after which a queue is automatically deleted. The minimum duration is 5 minutes.
Duplicate detection	An error could cause the client to have a doubt about the outcome of a send operation. Duplicate detection enables the sender to resend the same message, or for the queue or topic to discard any duplicate copies.
Security protocols	Service Bus supports security protocols such as Shared Access Signatures (SAS), Role Based Access Control (RBAC) and Managed identities for Azure resources.
Geo-disaster recovery	When Azure regions or datacenters experience downtime, Geo-disaster recovery enables data processing to continue operating in a different region or datacenter.
Security	Service Bus supports standard AMQP 1.0 and HTTP/REST protocols.

Consider using Service Bus queues

As a solution architect/developer, you should consider using Service Bus queues when:

- Your solution needs to receive messages without having to poll the queue. With Service Bus, you can achieve it by using a long-polling receive operation using the TCP-based protocols that Service Bus supports.
- Your solution requires the queue to provide a guaranteed first-in-first-out (FIFO) ordered delivery.
- Your solution needs to support automatic duplicate detection.
- You want your application to process messages as parallel long-running streams (messages are associated with a stream using the session ID property on the message). In this model, each node in the consuming application competes for streams, as opposed to messages. When a stream is given to a consuming node, the node can examine the state of the application stream state using transactions.
- Your solution requires transactional behavior and atomicity when sending or receiving multiple messages from a queue.
- Your application handles messages that can exceed 64 KB but won't likely approach the 256-KB limit.

Azure Queue Storage

Azure Queue Storage is a service for storing large numbers of messages.

You access messages from anywhere in the world via authenticated calls using HTTP or HTTPS.

A queue message can be up to 64 KB in size.

A queue may contain millions of messages, up to the total capacity limit of a storage account.

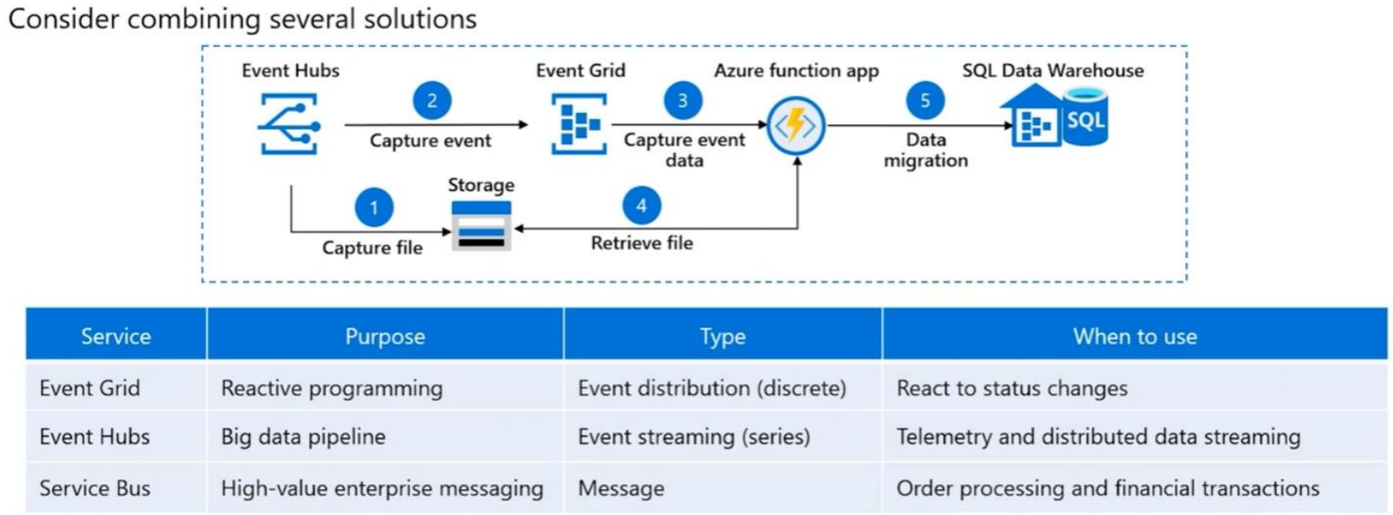
Queues are commonly used to create a backlog of work to process asynchronously.

Consider using Storage queues

As a solution architect/developer, you should consider using Storage queues when:

- Your application must store over 80 gigabytes of messages in a queue.
- Your application wants to track progress for processing a message in the queue. It's useful if the worker processing a message crashes. Another worker can then use that information to continue from where the prior worker left off.
- You require server side logs of all of the transactions executed against your queues.

Comparing message and event solutions



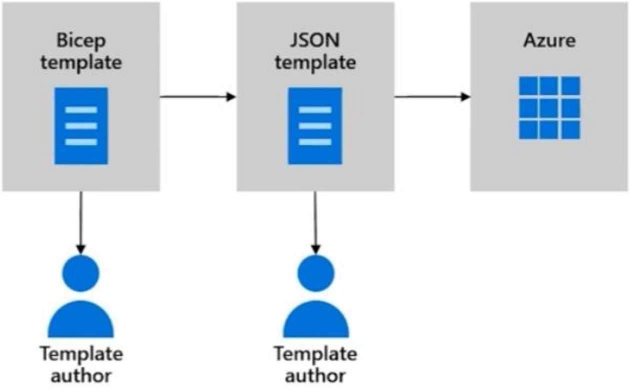
Consider combining several solutions

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

Provision resources with Infrastructure As Code (IaC)

Infrastructure as Code (IaC) is the process of automating your infrastructure provisioning.

- Azure Resource Manager templates – Bicep, JSON
- Azure Automation
- Azure DevOps services
- GitHub actions
- Terraform
- Jenkins



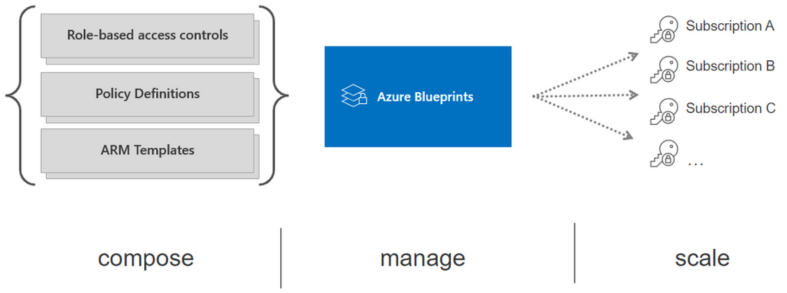
Azure Blueprints

Overview

Azure Blueprints enables cloud architects and central information technology groups to define a repeatable set of Azure resources that implements and adheres to an organization's standards, patterns, and requirements.

Azure Blueprints makes it possible for development teams to rapidly build and start up new environments with trust they're building within organizational compliance with a set of built-in components, such as networking, to speed up development and delivery.

deploy and update cloud environments in a repeatable manner using composable artifacts



Blueprints are a declarative way to orchestrate the deployment of various resource templates and other artifacts such as:

- Role Assignments
- Policy Assignments
- Azure Resource Manager templates (ARM templates)
- Resource Groups

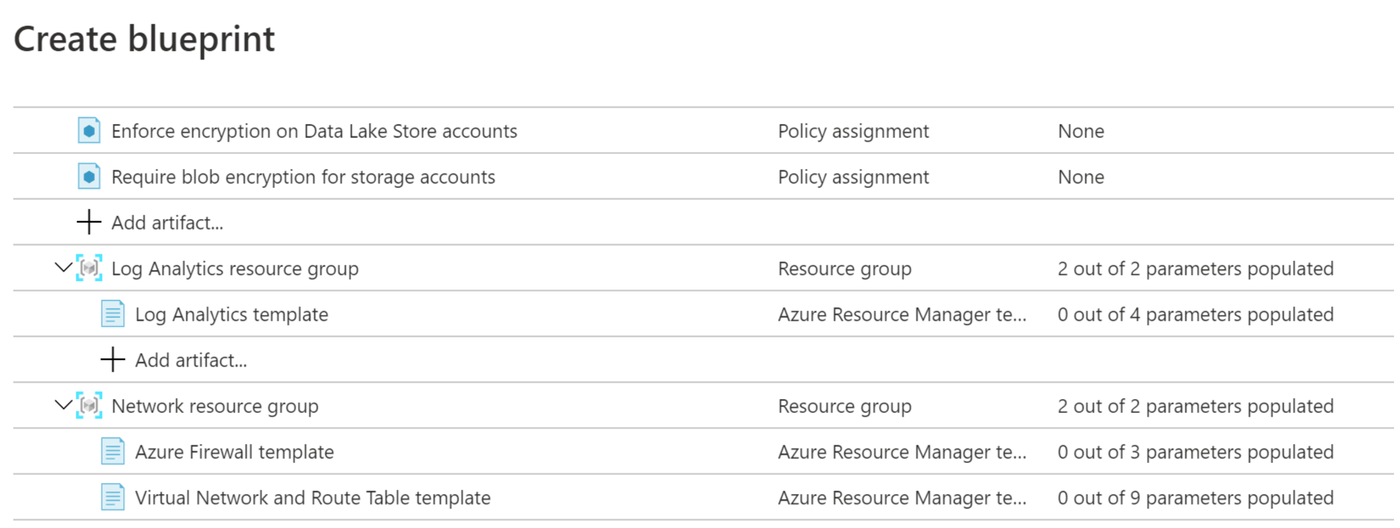
Azure Blueprints service is designed to help with environment setup. This setup often consists of a set of resource groups, policies, role assignments, and Resource Manager (ARM) template deployments. A blueprint is a package to bring each of these artifact types together and allow you to compose and version that package -- including through a CI/CD pipeline. Ultimately, each is assigned to a subscription in a single operation that can be audited and tracked.

Nearly everything that you want to include for deployment in Azure Blueprints can be accomplished with an ARM template. However, an ARM template is a document that doesn't exist natively in Azure - each is stored either locally or in source control or in Templates (preview).

The template gets used for deployments of one or more Azure resources, but once those resources deploy there's no active connection or relationship to the template.

With Azure Blueprints, the relationship between the blueprint definition (what should be deployed) and the blueprint assignment (what was deployed) is preserved. This connection supports improved tracking and auditing of deployments. Azure Blueprints can also upgrade several subscriptions at once that are governed by the same blueprint.

There's no need to choose between an ARM template and a blueprint. Each blueprint can consist of zero or more ARM template artifacts. This support means that previous efforts to develop and maintain a library of ARM templates are reusable in Azure Blueprints.



Create blueprint

Enforce encryption on Data Lake Store accounts	Policy assignment	None
Require blob encryption for storage accounts	Policy assignment	None
+ Add artifact...		
Log Analytics resource group	Resource group	2 out of 2 parameters populated
Log Analytics template	Azure Resource Manager te...	0 out of 4 parameters populated
+ Add artifact...		
Network resource group	Resource group	2 out of 2 parameters populated
Azure Firewall template	Azure Resource Manager te...	0 out of 3 parameters populated
Virtual Network and Route Table template	Azure Resource Manager te...	0 out of 9 parameters populated

Azure Blueprints vs ARM Templates

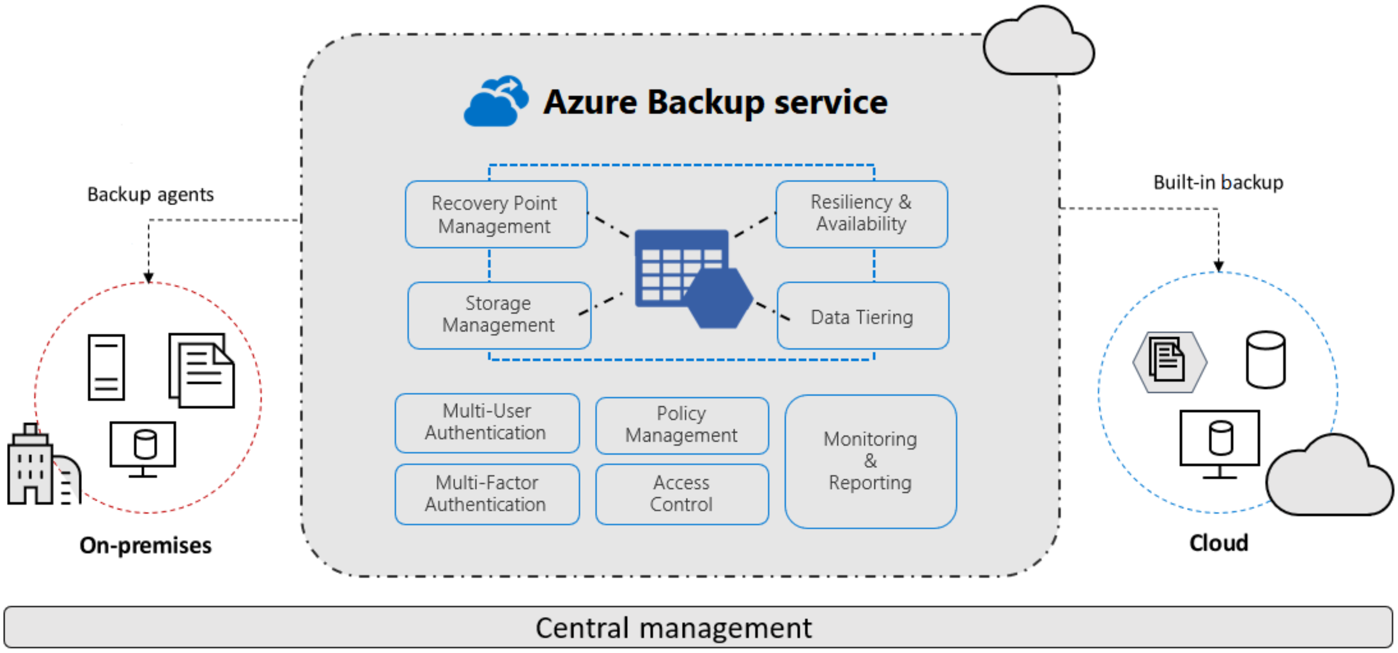
	Azure Blueprints	Azure ARM Template
Definition	Provide a way to package and deploy a repeatable set of managed resources such as Azure Policy, role assignments, and ARM templates as a single unit. You can combine Azure Policies, ARM templates, and resource groups together into a Blueprint. Essentially, this Blueprint is an allotment of cloud resources, components, or services that adhere to your organization's standards, patterns, and requirements.	Used for defining, deploying and managing infrastructure as code in Azure. Essentially, this means that you can describe all the necessary resources for your Azure application (such as virtual machines, storage accounts, networks, web applications, and your entire Azure infrastructure) in one declarative JSON template. This way, you can automate the entire deployment process while keeping your infrastructure configuration tracked in source control.
Scope	Provide you with greater control over the deployment of resources. They are scoped at the subscription level, which means that you can deploy and manage multiple resources groups at once across different regions and subscriptions.	Scoped at the resource group level, which provides granularity and control over the deployed resources. You can create, update, or delete multiple Azure resources within a resource group using ARM templates. However, they don't let you define the whole environment or infrastructure and include more than one resource group.

Customization	Are designed to be less flexible. They are intended to provide a standardized and auditable approach to resource governance for those with enterprise-level concerns. The Blueprints are pre-defined and locked for consistency to ensure the Azure services and resources your organization employs are secure and compliant.	Azure ARM templates provide more flexibility. They can be created and modified using various editors, such as Visual Studio or the Azure Cloud Shell, or even by hand using a text editor. Since ARM templates are code, you can modify them as necessary to meet your specific deployment needs.

Disaster Recovery Solutions

Azure Backup

- The Azure Backup service keeps your data safe and recoverable.
- You need to back up on-premises machines and workloads.
- Back up files, folders, and system state with the Microsoft Azure Recovery Services (MARS) agent.
- You can also use System Center Data Protection Manager (DPM) or the Microsoft Azure Backup Server (MABS) agent to protect on-premises virtual machines (both Hyper-V and VMware) and other on-premises workloads.
- You have Azure VMs running production workloads.
- Back up Azure file shares to a storage account.
- You need application-consistent backups for Linux virtual machines.
- Azure Backup storage vaults
  - **Azure Backup vault**: Azure Backup vaults are used with Azure Backup only. Supported data sources include Azure Database for PostgreSQL servers, Azure blobs, and Azure disks.
  - **Azure Recovery Services vault**: Azure Recovery Services vaults can be used with Azure Backup or Azure Site Recovery. Supported data sources include Azure virtual machines, SQL or SAP HANA in an Azure virtual machine, and Azure file shares. You can back up data to a Recovery Services vault from Azure Backup Server, Azure Backup Agent, and System Center Data Protection Manager.
  - Supports both locally redundant storage (LRS) to protect against failure in a datacenter and geo-redundant storage (GRS) to protect against region-wide outages.



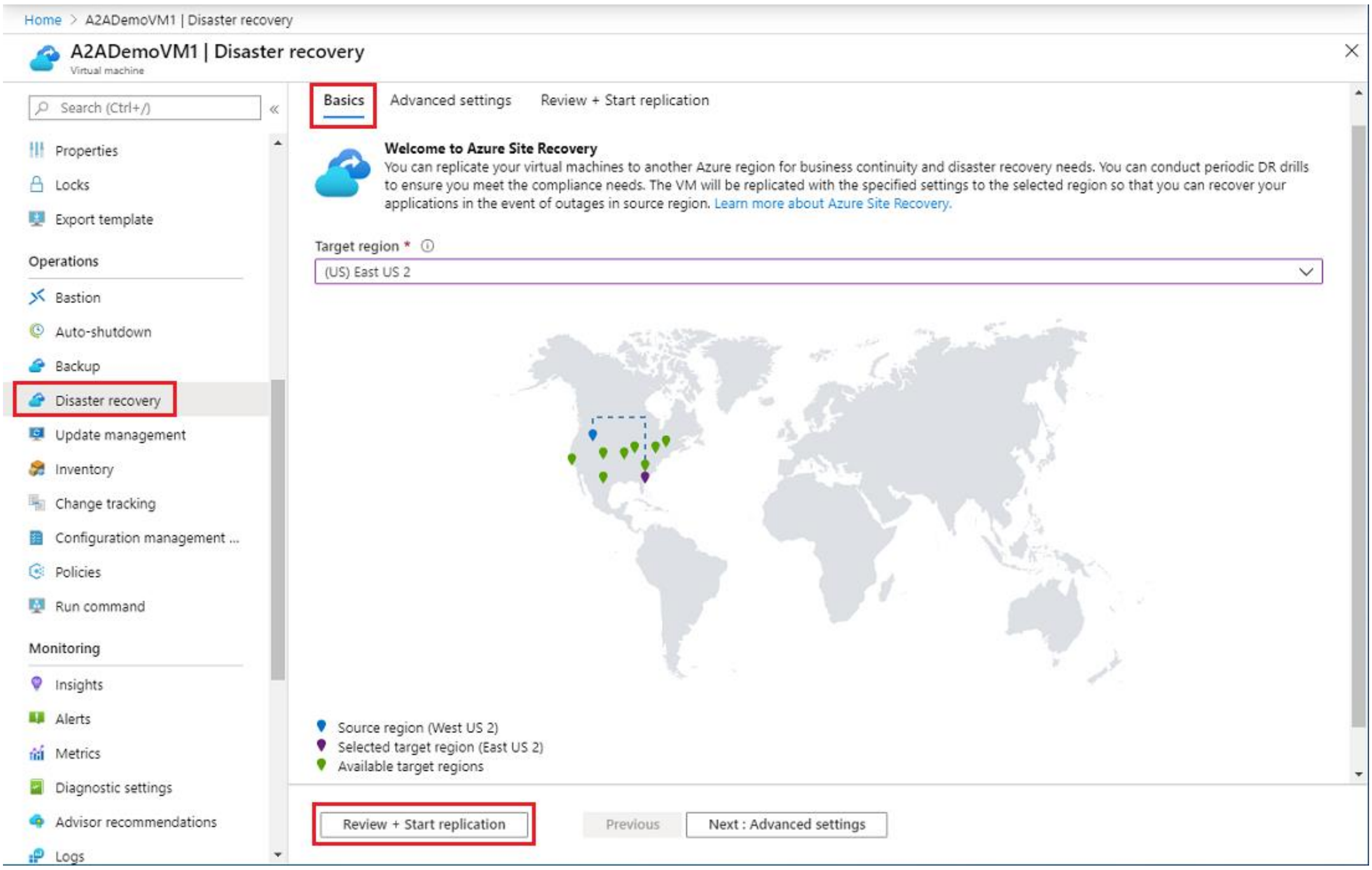
Azure Site Recovery

- You need to cover disaster scenarios like an entire regional outage.

- Site Recovery replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to a secondary location, and access apps from there. After the primary location is running again, you can fail back to it.
- Site Recovery can manage replication for:
    - Azure VMs replicating between Azure regions
    - Replication from Azure Public Multi-Access Edge Compute (MEC) to the region
    - Replication between two Azure Public MECs
    - On-premises VMs, Azure Stack VMs, and physical servers

What does Site Recovery provide?

- **Simple BCDR solution**: Using Site Recovery, you can set up and manage replication, failover, and failback from a single location in the Azure portal.
- **Azure VM replication**: You can set up disaster recovery of Azure VMs from a primary region to a secondary region or from Azure Public MEC to the Azure region or from one Azure Public MEC to another Azure Public MEC connected to the same Azure region.
- **VMware VM replication**: You can replicate VMware VMs to Azure using the improved Azure Site Recovery replication appliance that offers better security and resilience than the configuration server. For more information, see Disaster recovery of VMware VMs.
- **On-premises VM replication**: You can replicate on-premises VMs and physical servers to Azure, or to a secondary on-premises datacenter. Replication to Azure eliminates the cost and complexity of maintaining a secondary datacenter.
- **Workload replication**: Replicate any workload running on supported Azure VMs, on-premises Hyper-V and VMware VMs, and Windows/Linux physical servers.
- **Data resilience:** Site Recovery orchestrates replication without intercepting application data. When you replicate to Azure, data is stored in Azure storage, with the resilience that provides. When failover occurs, Azure VMs are created based on the replicated data. This also applies to Public MEC to Azure region Azure Site Recovery scenario. In case of Azure Public MEC to Public MEC Azure Site Recovery scenario (the ASR functionality for Public MEC is in preview state), data is stored in the Public MEC.
- **RTO and RPO targets**: Keep recovery time objectives (RTO) and recovery point objectives (RPO) within organizational limits. Site Recovery provides continuous replication for Azure VMs and VMware VMs, and replication frequency as low as 30 seconds for Hyper-V. You can reduce RTO further by integrating with Azure Traffic Manager.
- **Keep apps consistent over failover**: You can replicate using recovery points with application-consistent snapshots. These snapshots capture disk data, all data in memory, and all transactions in process.
- **Testing without disruption**: You can easily run disaster recovery drills, without affecting ongoing replication.
- **Flexible failovers**: You can run planned failovers for expected outages with zero-data loss. Or, unplanned failovers with minimal data loss, depending on replication frequency, for unexpected disasters. You can easily fail back to your primary site when it's available again.
- **Customized recovery plans**: Using recovery plans, you can customize and sequence the failover and recovery of multi-tier applications running on multiple VMs. You group machines together in a recovery plan, and optionally add scripts and manual actions. Recovery plans can be integrated with Azure Automation runbooks.
- **BCDR integration**: Site Recovery integrates with other BCDR technologies. For example, you can use Site Recovery to protect the SQL Server backend of corporate workloads, with native support for SQL Server Always On, to manage the failover of availability groups.
- **Azure automation integration**: A rich Azure Automation library provides production-ready, application-specific scripts that can be downloaded and integrated with Site Recovery.
- **Network integration**: Site Recovery integrates with Azure for application network management. For example, to reserve IP addresses, configure load-balancers, and use Azure Traffic Manager for efficient network switchovers.

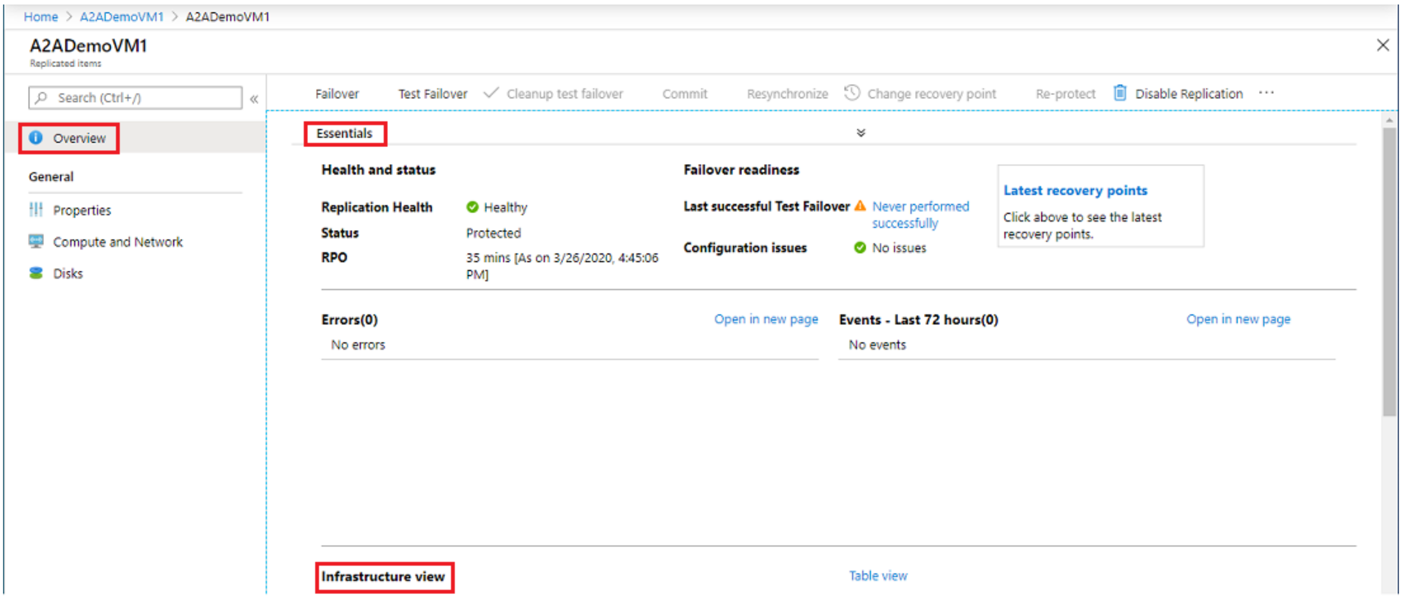


Welcome to Azure Site Recovery

You can replicate your virtual machines to another Azure region for business continuity and disaster recovery needs. You can conduct periodic DR drills to ensure you meet the compliance needs. The VM will be replicated with the specified settings to the selected region so that you can recover your applications in the event of outages in source region. Learn more about Azure Site Recovery.

Target region * : (US) East US 2

● Source region (West US 2)
● Selected target region (East US 2)
● Available target regions



Health and status

Replication Health: Healthy
Status: Protected
RPO: 35 mins [As on 3/26/2020, 4:45:06 PM]

Failover readiness

Last successful Test Failover: Never performed successfully
Configuration issues: No issues

Latest recovery points

Click above to see the latest recovery points.

Errors(0) Open in new page

No errors

Events - Last 72 hours(0) Open in new page

No events

VM Snapshot

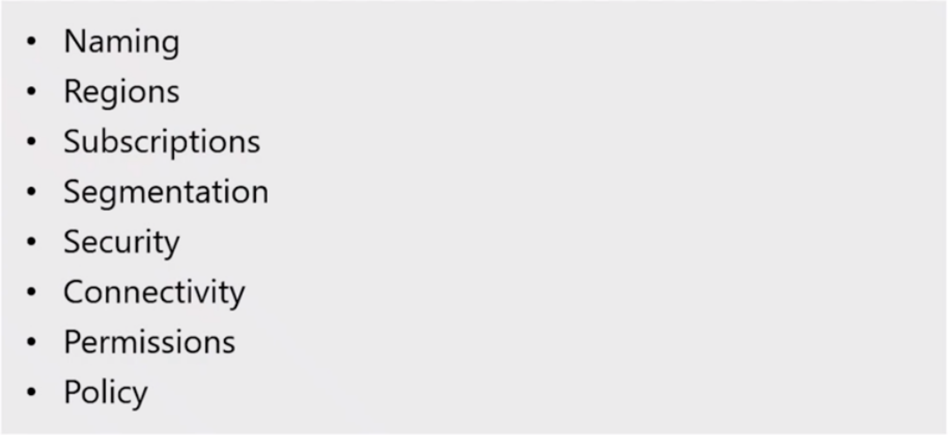
- You need to back up your managed disks at any point in time.
- You need a read-only full copy of a managed disk.

Microsoft Azure Backup Services (MABS)

- You need to back up on-premises machines and workloads (depending on the workload).

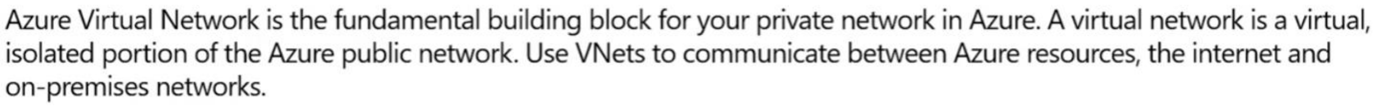
Networking

Plan Virtual Networks (VNETs) and subnets – design considerations

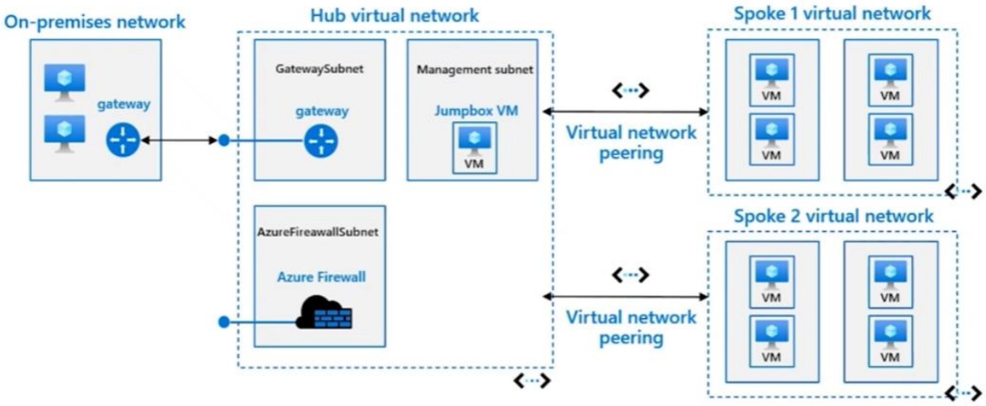


- Naming
- Regions
- Subscriptions
- Segmentation
- Security
- Connectivity
- Permissions
- Policy

Design Azure Virtual Networks



Azure Virtual Network is the fundamental building block for your private network in Azure. A virtual network is a virtual, isolated portion of the Azure public network. Use VNets to communicate between Azure resources, the internet and on-premises networks.



Communicate between Azure resources

You can enable Azure resources to communicate securely with each other, in one of two ways:

- **Virtual networks**: Virtual networks can connect not only VMs but other Azure resources, such as the App Service Environment for Power Apps, Azure Kubernetes Service, and Azure Virtual Machine Scale Sets.
- **Service endpoints**: You can use service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts. This approach enables you to link multiple Azure resources to virtual networks to improve security and provide optimal routing between resources.

Communicate with on-premises resources

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

- **Point-to-site virtual private networks**: The typical approach to a virtual private network (VPN) connection is from a computer outside your organization, back into your corporate network. In this case, the client computer initiates an encrypted VPN connection to connect that computer to the Azure virtual network.

- **Site-to-site virtual private networks**: A site-to-site VPN links your on-premises VPN device or gateway to the Azure VPN gateway in a virtual network. In effect, the devices in Azure can appear as being on the local network. The connection is encrypted and works over the internet.
- **Azure ExpressRoute**: For environments where you need greater bandwidth and even higher levels of security, Azure ExpressRoute is the best approach. ExpressRoute provides a dedicated private connectivity to Azure that doesn't travel over the internet.

Virtual Network Address space

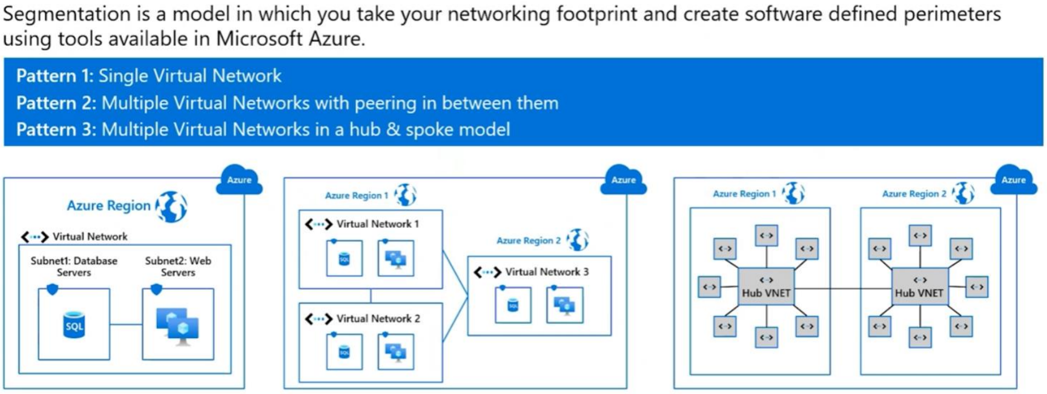
When you set up a virtual network, you define the internal address space in Classless Interdomain Routing (CIDR) format. This address space needs to be unique within your subscription and any other networks that you connect to.

Let's assume you choose an address space of 10.0.0.0/24 for your first virtual network. The addresses defined in this address space range from 10.0.0.1 to 10.0.0.254.

You then create a second virtual network and choose an address space of 10.0.0.0/8. The addresses in this address space range from 10.0.0.1 to 10.255.255.254. Some of the addresses overlap and can't be used for the two virtual networks.

But you can use 10.0.0.0/16, with addresses that range from 10.0.0.1 to 10.0.255.254, and 10.1.0.0/16, with addresses that range from 10.1.0.1 to 10.1.255.254. You can assign these address spaces to your virtual networks because there's no address overlap.

Design network topology



Segmentation is a model in which you take your networking footprint and create software defined perimeters using tools available in Microsoft Azure.

Pattern 1: Single Virtual Network
Pattern 2: Multiple Virtual Networks with peering in between them
Pattern 3: Multiple Virtual Networks in a hub & spoke model

Design Outbound Connectivity

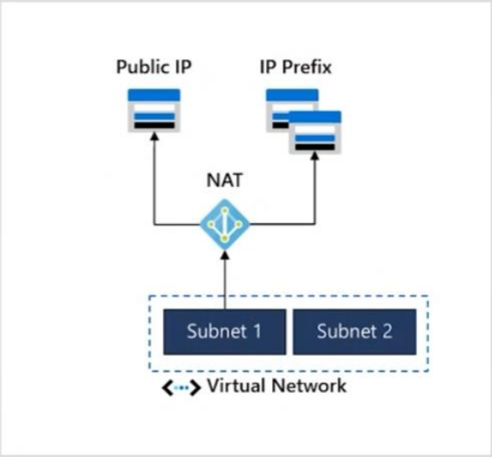
Virtual Network NAT (network address translation) simplifies outbound-only Internet connectivity for virtual networks. When configured on a subnet, all outbound connectivity uses your specified static public IP addresses. NAT is fully managed and highly resilient.

Options include:

- Azure Firewall
- Load balancer
- Virtual Network NAT gateway

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



VPN Gateway

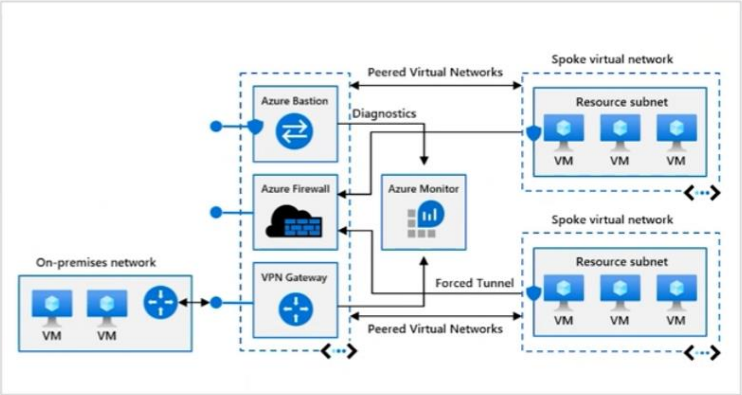
A VPN gateway is a type of virtual network gateway that sends encrypted traffic between an Azure virtual network and an on-premises location.

Benefits

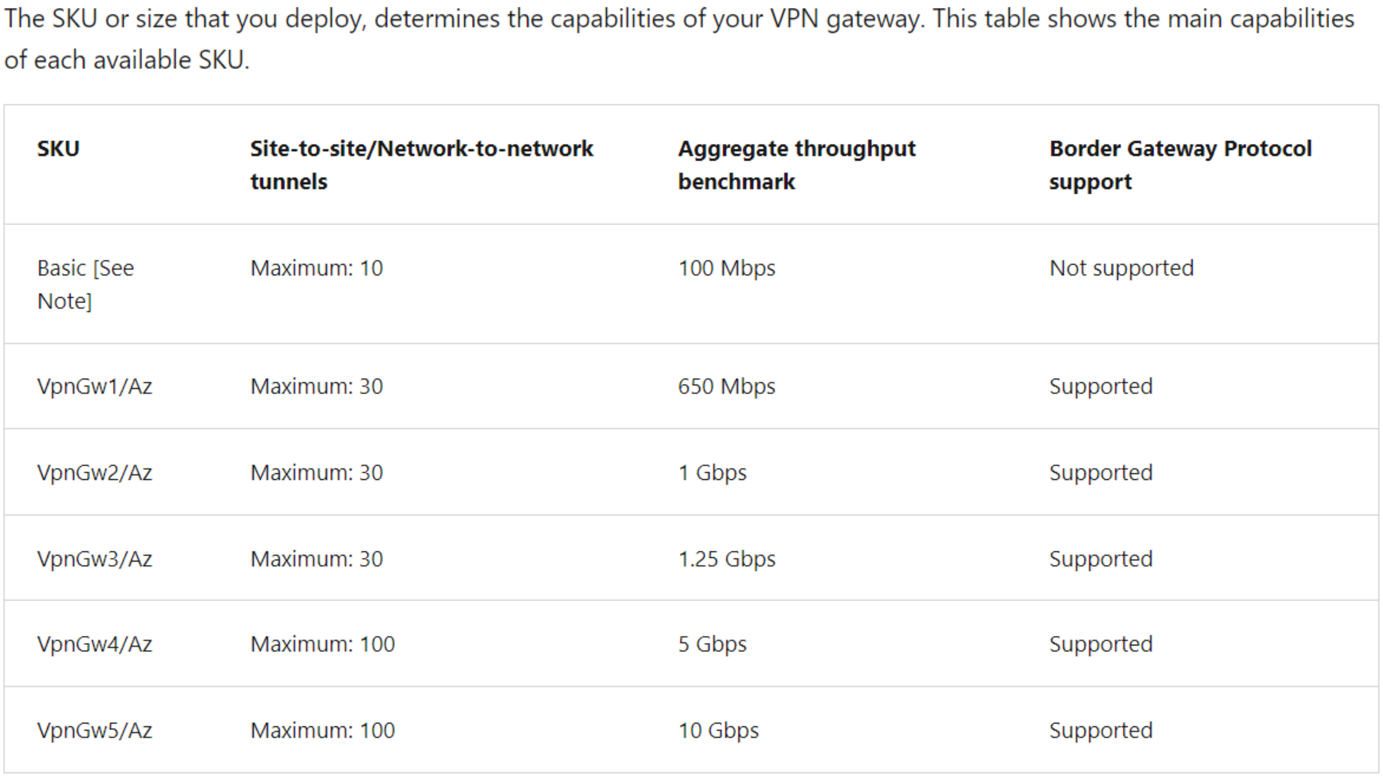
- Simple to configure.
- Up to 10 Gbps depending on the VPN Gateway SKU.

Challenges

- Requires an on-premises VPN device.
- Although Microsoft guarantees 99.9% availability for each VPN Gateway, this SLA only covers the VPN gateway, and not your network connection to the gateway or throughput



VPN Gateway sizes



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

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

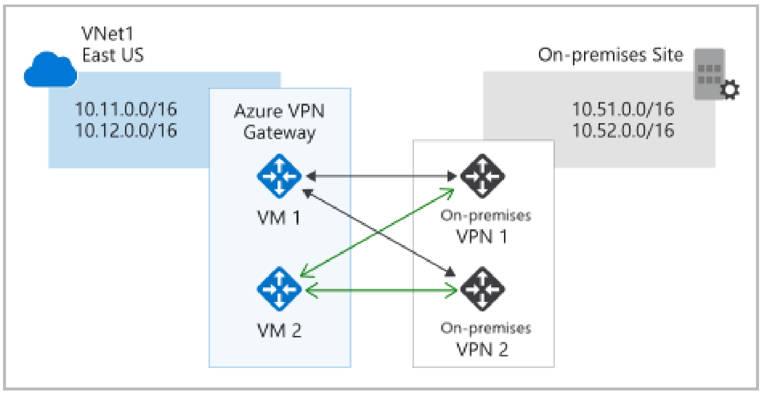
VPN Gateway High-availability scenarios

Active/standby: By default, VPN gateways are deployed as two instances in an active/standby configuration, even if you only see one VPN gateway resource in Azure.

- When planned maintenance or unplanned disruption affects the active instance, the standby instance automatically assumes responsibility for connections without any user intervention.
- Connections are interrupted during this failover, but they're typically restored within a few seconds for planned maintenance and within 90 seconds for unplanned disruptions.

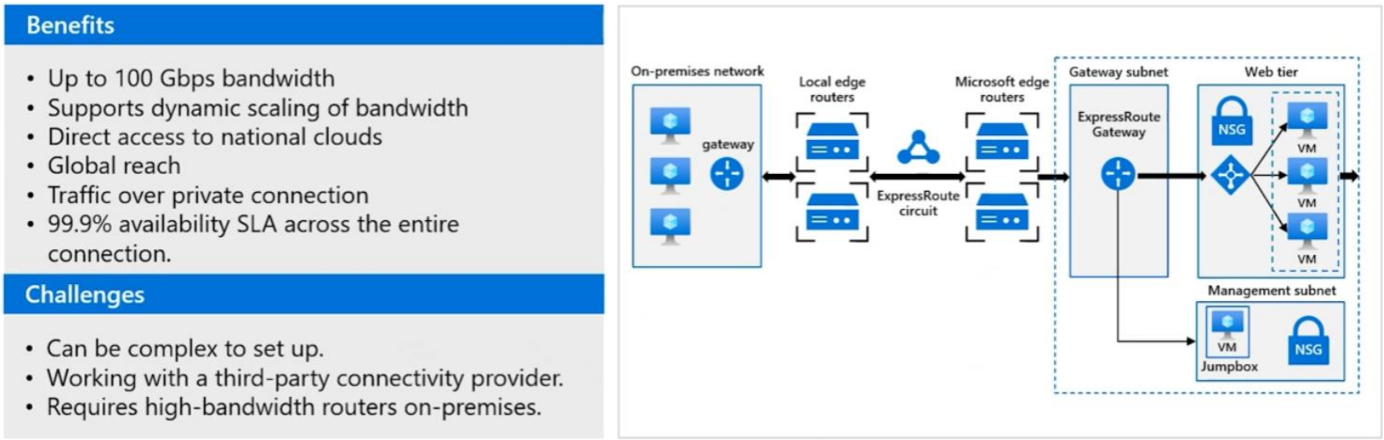
Active/active: With the introduction of support for the BGP routing protocol, you can also deploy VPN gateways in an active/active configuration.

- In this configuration, you assign a unique public IP address to each instance.
- You then create separate tunnels from the on-premises device to each IP address.
- You can extend the high availability by deploying another VPN device on-premises.



Azure ExpressRoute and ExpressRoute Direct connection

ExpressRoute connections use a private, dedicated connection through a third-party connectivity provider. This connection is private.



Benefits

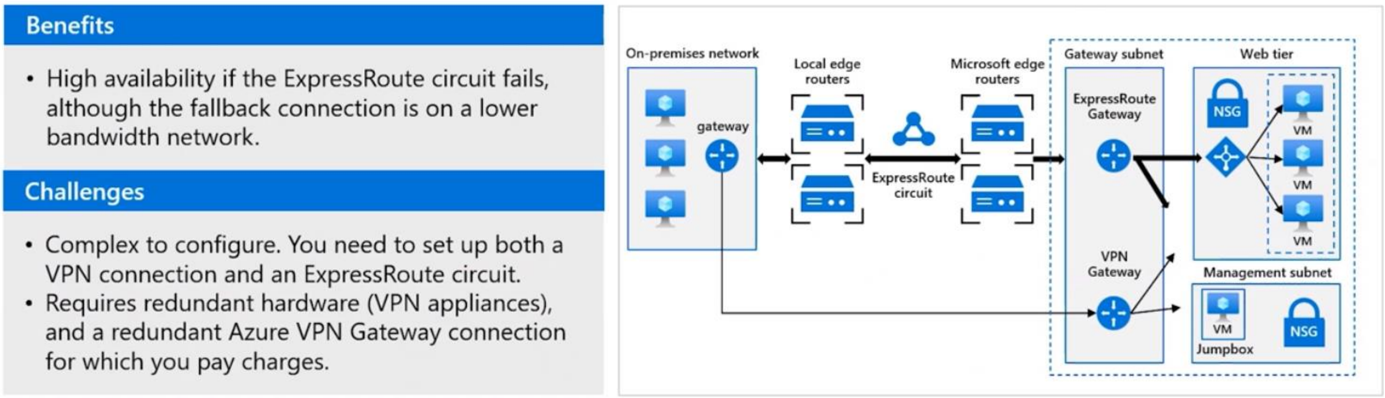
- Up to 100 Gbps bandwidth
- Supports dynamic scaling of bandwidth
- Direct access to national clouds
- Global reach
- Traffic over private connection
- 99.9% availability SLA across the entire connection.

Challenges

- Can be complex to set up.
- Working with a third-party connectivity provider.
- Requires high-bandwidth routers on-premises.

Azure 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.



Benefits

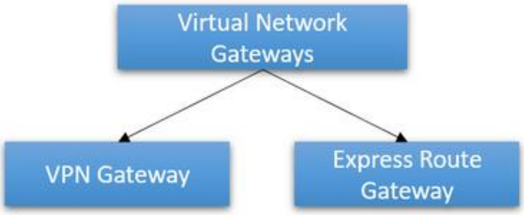
- High availability if the ExpressRoute circuit fails, although the fallback connection is on a lower bandwidth network.

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.

Azure VPN Gateway vs ExpressRoute

For environments where you need greater bandwidth and even higher levels of security, Azure ExpressRoute is the best approach. ExpressRoute provides a dedicated private connectivity to Azure that doesn't travel over the internet.



	Point-to-Site	Site-to-Site	ExpressRoute
Azure Supported Services	Cloud Services and Virtual Machines	Cloud Services and Virtual Machines	Services list
Typical Bandwidths	Based on the gateway SKU	Typically < 10 Gbps aggregate	50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps, 100 Gbps

Protocols Supported	Secure Sockets Tunneling Protocol (SSTP), OpenVPN, and IPsec	IPsec/ IKE	Direct connection over VLANs, NSP's VPN technologies (MPLS, VPLS,...)
Routing	RouteBased (dynamic)	We support PolicyBased (static routing) and RouteBased (dynamic routing VPN)	BGP
Connection resiliency	active-passive	active-passive or active-active	active-active
High Availability	-	Highly Available cross-premises and VNet-to-VNet connectivity Multiple on-premises VPN devices Active-active VPN gateways Dual-redundancy: active-active VPN gateways for both Azure and on-premises networks Highly Available VNet-to-Vnet	Designing for high availability with ExpressRoute First-mile physical layer design considerations Active-active connections NAT for Microsoft peering Fine-tuning features for private peering Availability Zone aware ExpressRoute virtual network gateways Improving failure detection time
Typical use case	Secure access to Azure virtual networks for remote users	Dev/test / lab scenarios and small to medium-scale production workloads for cloud services and virtual machines	Access to all Azure services (validated list), Enterprise-class and mission-critical workloads, Backup, Big Data, Azure as a DR site

VPN Gateway

A virtual network gateway is composed of two or more Azure-managed VMs automatically configured and deployed to a specific subnet you create called the GatewaySubnet.

When you create a VPN gateway, gateway VMs are deployed to the gateway subnet and configured with your specified settings. This process can take 45 minutes or more to complete, depending on your selected gateway SKU.

- **Point-to-site virtual private networks**: The typical approach to a virtual private network (VPN) connection is from a computer outside your organization, back into your corporate network. In this case, the client computer initiates an encrypted VPN connection to connect that computer to the Azure virtual network.
- **Site-to-site virtual private networks**: A site-to-site VPN links your on-premises VPN device or gateway to the Azure VPN gateway in a virtual network. In effect, the devices in Azure can appear as being on the local network. The connection is encrypted and works over the internet.

ExpressRoute Gateway

Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider.

With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure and Microsoft 365.

You must first create a virtual network gateway to connect your Azure virtual network and your on-premises network using ExpressRoute.

A virtual network gateway serves two purposes: exchanging IP routes between the networks and routing traffic.

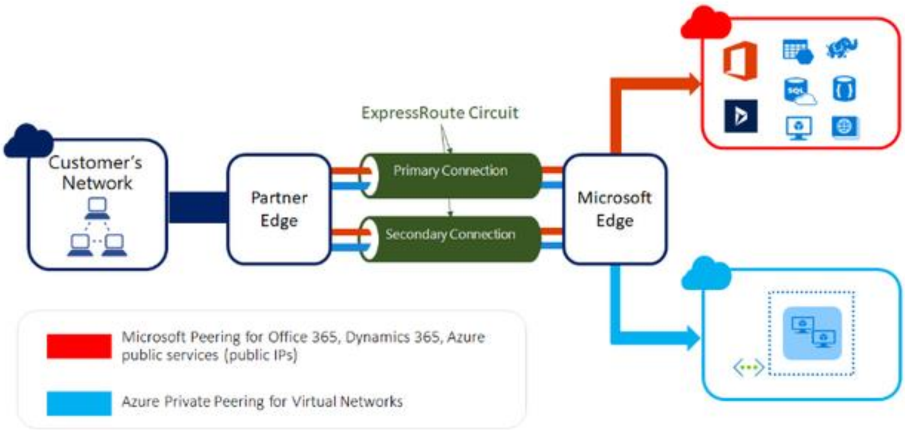
The dedicated version of an Azure ExpressRoute is called ExpressRoute Direct. You create and manage these separately from other ExpressRoute circuits.

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

ExpressRoute Direct gives you the ability to connect directly into the Microsoft global network at peering locations strategically distributed around the world. ExpressRoute Direct provides dual 100-Gbps or 10-Gbps connectivity, that supports Active/Active connectivity at scale.

Benefits of using Azure ExpressRoute:

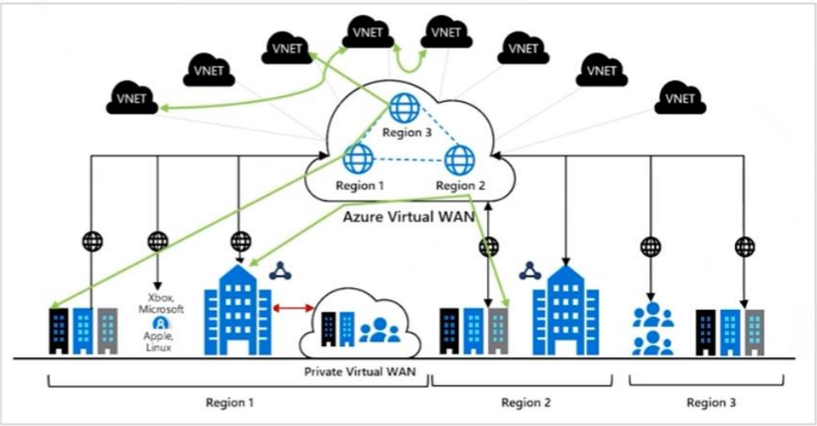
- Layer 3 connectivity between your on-premises network and the Microsoft Cloud through a connectivity provider. Connectivity can be from an any-to-any (IPVPN) network, a point-to-point Ethernet connection, or through a virtual cross-connection via an Ethernet exchange.
- Connectivity to Microsoft cloud services across all regions in the geopolitical region.
- Global connectivity to Microsoft services across all regions with the ExpressRoute premium add-on.
- Dynamic routing between your network and Microsoft via BGP.
- Built-in redundancy in every peering location for higher reliability.
- Connection uptime SLA.
- QoS support for Skype for Business.
- ExpressRoute connections don't go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, consistent latencies, and higher security than typical connections over the Internet.
- ExpressRoute does provide private connectivity, but it is NOT encrypted.


ExpressRoute Circuit
Customer's Network
Partner Edge
Primary Connection
Secondary Connection
Microsoft Edge
Microsoft Peering for Office 365, Dynamics 365, Azure public services (public IPs)
Azure Private Peering for Virtual Networks

Azure Virtual WAN

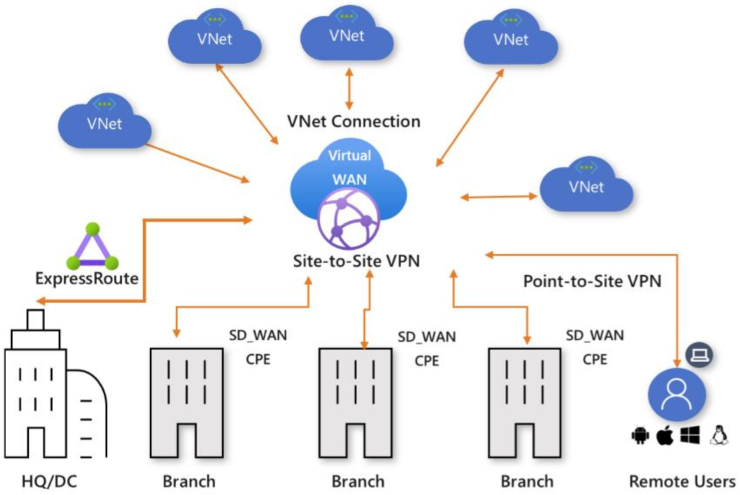
Azure Virtual WAN is a networking service that brings many networking, security, and routing functionalities together to provide a single operational interface.

- 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
- Separation of concerns between central IT (SecOps, InfraOps) and workloads (DevOps).



Some of the main features include:

- Branch connectivity (via connectivity automation from Virtual WAN Partner devices such as SD-WAN or VPN CPE).
- Site-to-site VPN connectivity.
- Remote user VPN connectivity (point-to-site).
- Private connectivity (ExpressRoute).
- Intra-cloud connectivity (transitive connectivity for virtual networks).
- VPN ExpressRoute inter-connectivity.
- Routing, Azure Firewall, and encryption for private connectivity.



Virtual WAN offers the following advantages:

- **Integrated connectivity solutions in hub and spoke**: Automate site-to-site configuration and connectivity between on-premises sites and an Azure hub.
- **Automated spoke setup and configuration**: Connect your virtual networks and workloads to the Azure hub seamlessly.
- **Intuitive troubleshooting**: You can see the end-to-end flow within Azure, and then use this information to take required actions.

Virtual WAN Types:

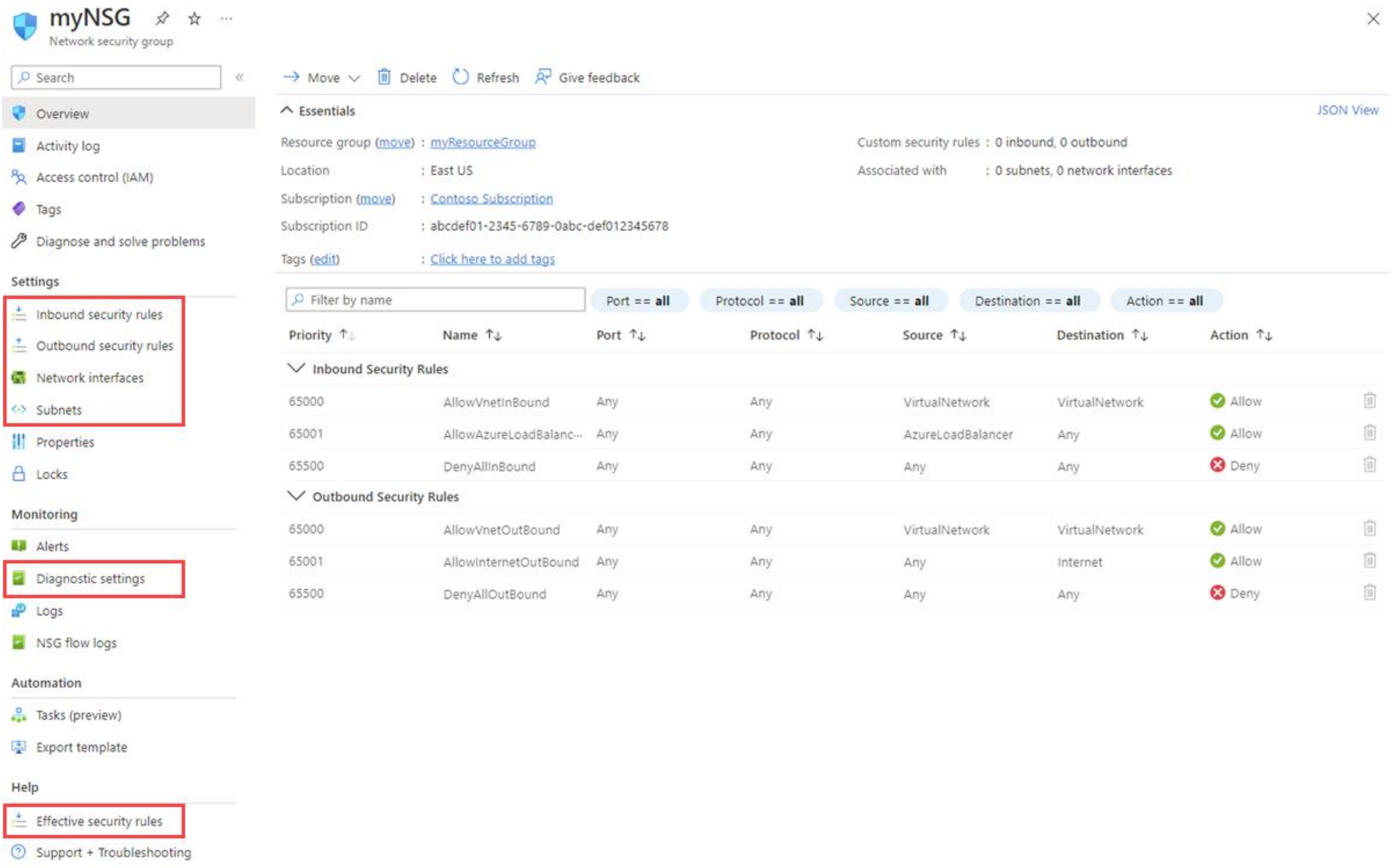
- Basic:
  - Site-to-site VPN Only
- Standard:
  - ExpressRoute
  - User VPN (P2S)
  - VPN (site-to-site)
  - Inter-hub and VNet-to-VNet transiting through the virtual hub
  - Azure Firewall
  - NVA in a virtual WAN

Network Security Group (NSG)

You can filter network traffic to and from Azure resources in an Azure virtual network with a network security group.

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 contain two sets of rules: inbound and outbound. The priority for a rule must be unique within each set.



Resource group (move) : myResourceGroup

Location : East US

Subscription (move) : Contoso Subscription

Subscription ID : abcdef01-2345-6789-0abc-def012345678

Tags (edit) : Click here to add tags

Custom security rules : 0 inbound, 0 outbound

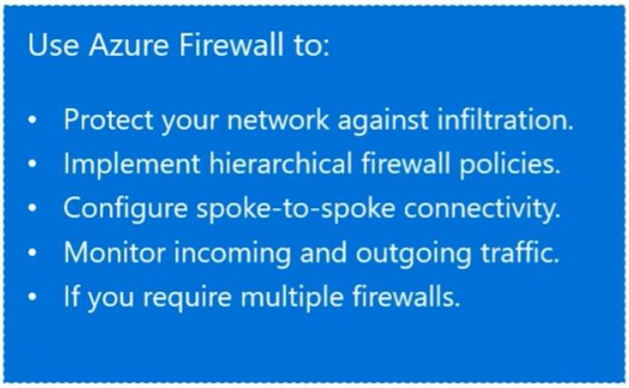
Associated with : 0 subnets, 0 network interfaces

Priority	Name	Port	Protocol	Source	Destination	Action
Inbound Security Rules						
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalanc...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
Outbound Security Rules						
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

Azure Firewall

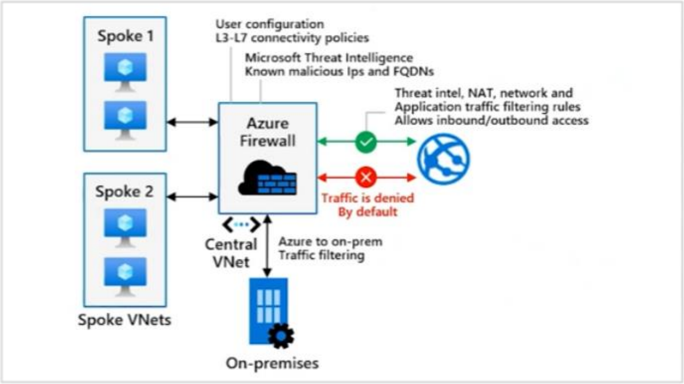
Azure firewall is a cloud-native network security service offering high-availability and scalability.

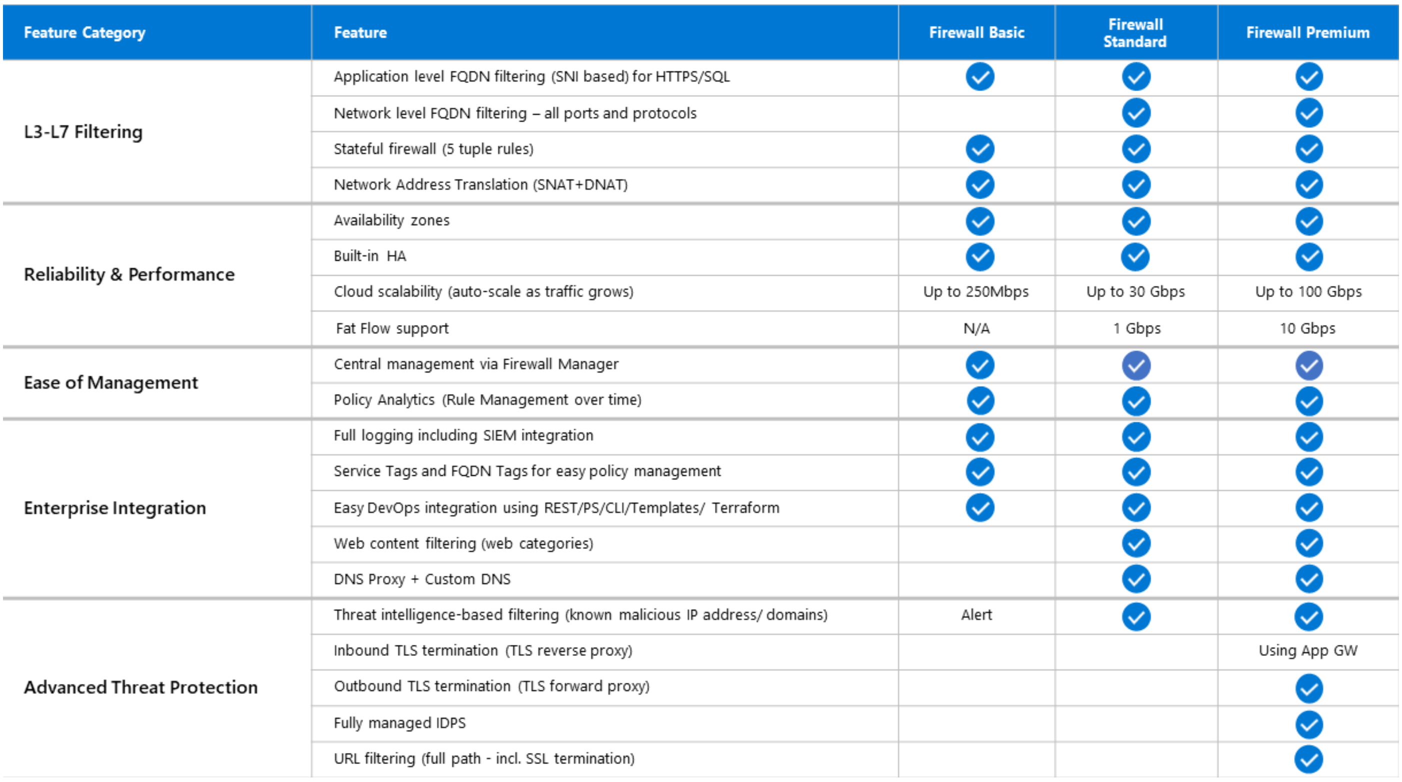
Azure Firewall provides inbound protection for non-HTTP/S protocols (for example, RDP, SSH, FTP), outbound network-level protection for all ports and protocols, and application-level protection for outbound HTTP/S.



Use Azure Firewall to:

- Protect your network against infiltration.
- Implement hierarchical firewall policies.
- Configure spoke-to-spoke connectivity.
- Monitor incoming and outgoing traffic.
- If you require multiple firewalls.


Spoke 1
Spoke 2
Spoke VNets
User configuration
L3-L7 connectivity policies
Microsoft Threat Intelligence
Known malicious Ips and FQDNs
Threat intel, NAT, network and
Application traffic filtering rules
Allows inbound/outbound access
Azure Firewall
Traffic is denied
By default
Central VNet
Azure to on-prem
Traffic filtering
On-premises

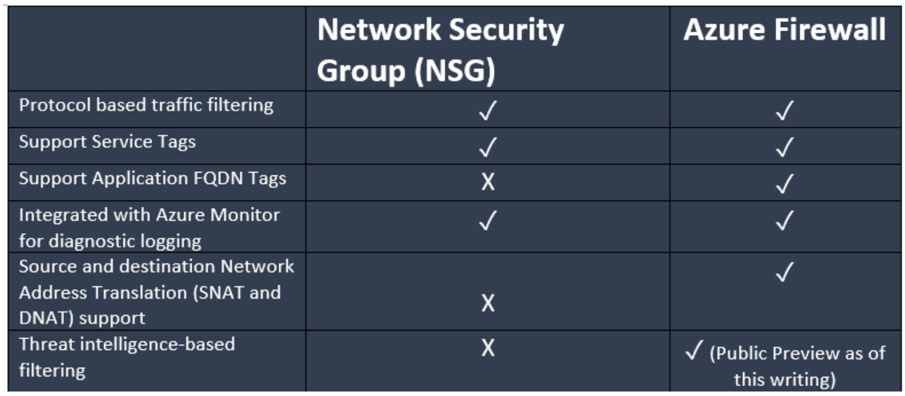


Feature Category	Feature	Firewall Basic	Firewall Standard	Firewall Premium
L3-L7 Filtering	Application level FQDN filtering (SNI based) for HTTPS/SQL	✓	✓	✓
	Network level FQDN filtering – all ports and protocols		✓	✓
	Stateful firewall (5 tuple rules)	✓	✓	✓
	Network Address Translation (SNAT+DNAT)	✓	✓	✓
Reliability & Performance	Availability zones	✓	✓	✓
	Built-in HA	✓	✓	✓
	Cloud scalability (auto-scale as traffic grows)	Up to 250Mbps	Up to 30 Gbps	Up to 100 Gbps
	Fat Flow support	N/A	1 Gbps	10 Gbps
Ease of Management	Central management via Firewall Manager	✓	✓	✓
	Policy Analytics (Rule Management over time)	✓	✓	✓
Enterprise Integration	Full logging including SIEM integration	✓	✓	✓
	Service Tags and FQDN Tags for easy policy management	✓	✓	✓
	Easy DevOps integration using REST/PS/CLI/Templates/ Terraform	✓	✓	✓
	Web content filtering (web categories)		✓	✓
	DNS Proxy + Custom DNS		✓	✓
Advanced Threat Protection	Threat intelligence-based filtering (known malicious IP address/ domains)	Alert	✓	✓
	Inbound TLS termination (TLS reverse proxy)			Using App GW
	Outbound TLS termination (TLS forward proxy)			✓
	Fully managed IDPS			✓
	URL filtering (full path - incl. SSL termination)			✓

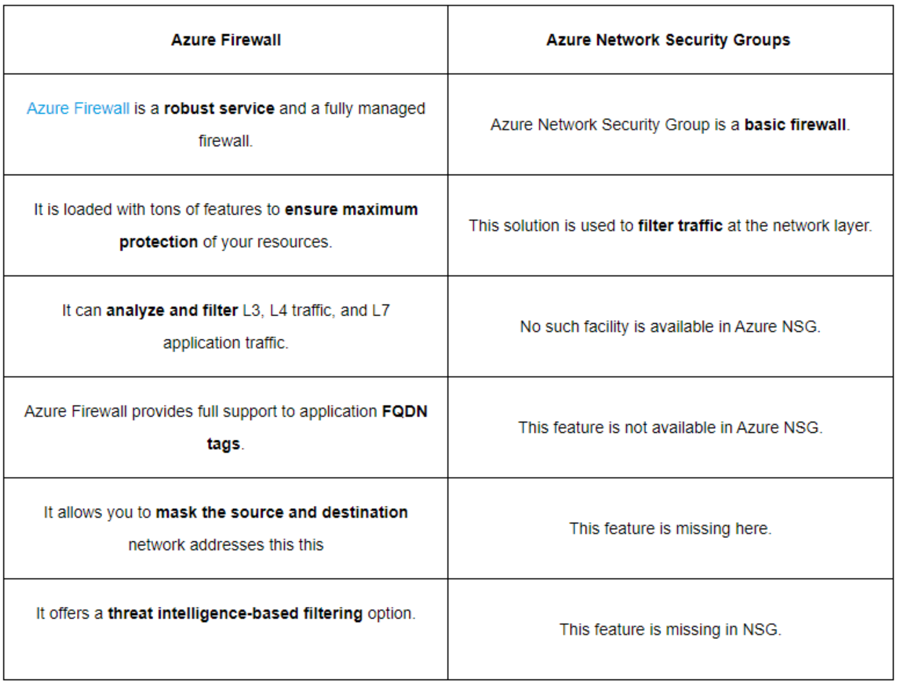
Azure Firewall now supports three different SKUs to cater to a wide range of customer use cases and preferences.

- **Azure Firewall Premium** is recommended to secure highly sensitive applications (such as payment processing). It supports advanced threat protection capabilities like malware and TLS inspection.
- **Azure Firewall Standard** is recommended for customers looking for Layer 3–Layer 7 firewall and needs autoscaling to handle peak traffic periods of up to 30 Gbps. It supports enterprise features like threat intelligence, DNS proxy, custom DNS, and web categories.
- **Azure Firewall Basic** is recommended for SMB customers with throughput needs of 250 Mbps.

Azure Network Security Group vs Azure Firewall

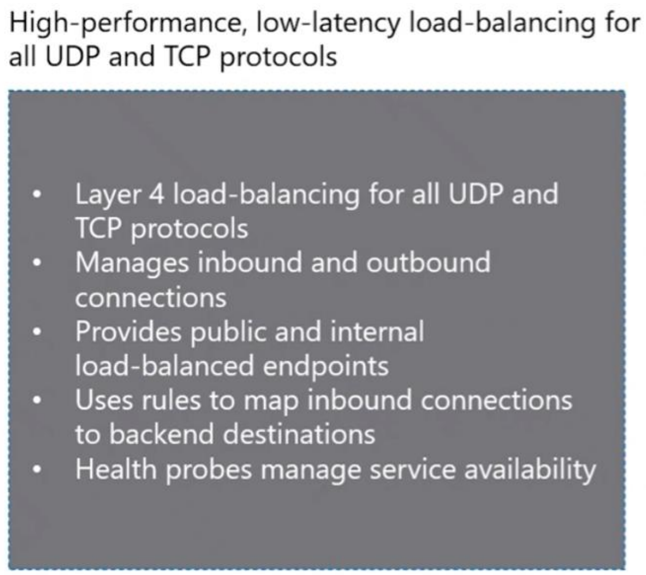


	Network Security Group (NSG)	Azure Firewall
Protocol based traffic filtering	✓	✓
Support Service Tags	✓	✓
Support Application FQDN Tags	X	✓
Integrated with Azure Monitor for diagnostic logging	✓	✓
Source and destination Network Address Translation (SNAT and DNAT) support	X	✓
Threat intelligence-based filtering	X	✓ (Public Preview as of this writing)



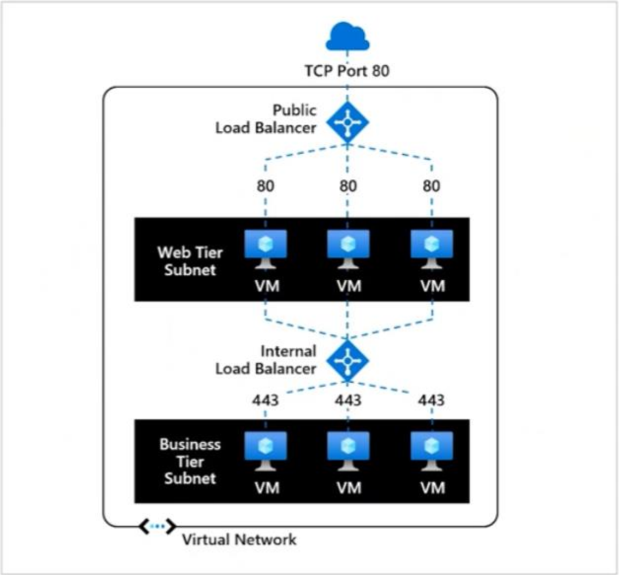
Azure Firewall	Azure Network Security Groups
Azure Firewall is a **robust service** and a fully managed firewall.	Azure Network Security Group is a **basic firewall**.
It is loaded with tons of features to **ensure maximum protection** of your resources.	This solution is used to **filter traffic** at the network layer.
It can **analyze and filter** L3, L4 traffic, and L7 application traffic.	No such facility is available in Azure NSG.
Azure Firewall provides full support to application **FQDN tags**.	This feature is not available in Azure NSG.
It allows you to **mask the source and destination** network addresses this this	This feature is missing here.
It offers a **threat intelligence-based filtering** option.	This feature is missing in NSG.

Azure Load Balancer



High-performance, low-latency load-balancing for all UDP and TCP protocols

- Layer 4 load-balancing for all UDP and TCP protocols
- Manages inbound and outbound connections
- Provides public and internal load-balanced endpoints
- Uses rules to map inbound connections to backend destinations
- Health probes manage service availability



A **public load balancer** can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. Public Load Balancers are used to load balance internet traffic to your VMs.

An **internal (or private) load balancer** is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic inside a virtual network. A load balancer frontend can be accessed from an on-premises network in a hybrid scenario.

Key scenarios that you can accomplish using Azure Standard Load Balancer include:

- Load balance internal and external traffic to Azure virtual machines.
- Increase availability by distributing resources within and across zones.
- Configure outbound connectivity for Azure virtual machines.
- Use health probes to monitor load-balanced resources.
- Employ port forwarding to access virtual machines in a virtual network by public IP address and port.
- Enable support for load-balancing of IPv6.
- Standard load balancer provides multi-dimensional metrics through Azure Monitor. These metrics can be filtered, grouped, and broken out for a given dimension. They provide current and historic insights into performance and health of your service. Insights for Azure Load Balancer offers a preconfigured dashboard with useful visualizations for these metrics. Resource Health is also supported. Review Standard load balancer diagnostics for more details.
- Load balance services on multiple ports, multiple IP addresses, or both.
- Move internal and external load balancer resources across Azure regions.
- Load balance TCP and UDP flow on all ports simultaneously using HA ports.
- Chain Standard Load Balancer and Gateway Load Balancer.

Secure by default

- Standard load balancer is built on the zero trust network security model.
- Standard Load Balancer is secure by default and part of your virtual network. The virtual network is a private and isolated network.
- Standard load balancers and standard public IP addresses are closed to inbound connections unless opened by Network Security Groups. NSGs are used to explicitly permit allowed traffic. If you don't have an NSG on a subnet or NIC of your virtual machine resource, traffic isn't allowed to reach this resource. To learn about NSGs and how to apply them to your scenario, see Network Security Groups.
- Basic load balancer is open to the internet by default.
- Load balancer doesn't store customer data.

Other Load Balancers

- If you are looking to do DNS based global routing and do not have requirements for Transport Layer Security (TLS) protocol termination ("SSL offload"), per-HTTP/HTTPS request or application-layer processing, review **Traffic Manager**.
- If you want to load balance between your servers in a region at the application layer, review **Application Gateway**.
- If you need to optimize global routing of your web traffic and optimize top-tier end-user performance and reliability through quick global failover, see **Front Door**.

Azure Front Door

Overview

- Whether you're delivering content and files or building global apps and APIs, Azure Front Door can help you deliver higher availability, lower latency, greater scale, and more secure experiences to your users wherever they are.
- Azure Front Door is Microsoft's modern cloud Content Delivery Network (CDN) that provides fast, reliable, and secure access between your users and your applications' static and dynamic web content across the globe.
- Azure Front Door delivers your content using Microsoft's global edge network with hundreds of global and local points of presence (PoPs) distributed around the world close to both your enterprise and consumer end users.
- For web workloads, we highly recommend utilizing Azure DDoS protection and a web application firewall (WAF) to safeguard against emerging DDoS attacks. Another option is to employ Azure Front Door along with a web application firewall. Azure Front Door offers platform-level protection against network-level DDoS attacks.

Why use Azure Front Door?

Build and operate modern internet-first architectures that have dynamic, high-quality digital experiences with highly automated, secure, and reliable platforms.

Accelerate and deliver your app and content globally at scale to your users wherever they're creating opportunities for you to compete, weather change, and quickly adapt to new demand and markets.

Intelligently secure your digital estate against known and new threats with intelligent security that embrace a Zero Trust framework.

Global delivery scale using Microsoft's network

- Leverage over 118 edge locations across 100 metro cities connected to Azure using a private enterprise-grade WAN and improve latency for apps by up to 3 times.
- Accelerate application performance by using Front Door's anycast network and split TCP connections.
- Terminate SSL offload at the edge and use integrated certificate management.
- Natively support end-to-end IPv6 connectivity and the HTTP/2 protocol.

Deliver modern apps and architectures

- Integrate with DevOps friendly command line tools across SDKs of different languages, Bicep, ARM templates, CLI and PowerShell.
- Define your own custom domain with flexible domain validation.
- Load balance and route traffic across origins and use intelligent health probe monitoring across apps or content hosted in Azure or anywhere.
- Integrate with other Azure services such as DNS, Web Apps, Storage and many more for domain and origin management.
- Move your routing business logic to the edge with enhanced rules engine capabilities including regular expressions and server variables.
- Analyze built-in reports with an all-in-one dashboard for both Front Door and security patterns.
- Monitoring your Front Door traffic in real time and configure alerts that integrate with Azure Monitor.
- Log each Front Door request and failed health probes.

Simple and cost-effective

- Unified static and dynamic delivery offered in a single tier to accelerate and scale your application through caching, SSL offload, and layer 3-4 DDoS protection.
- Free, autorotation managed SSL certificates that save time and quickly secure apps and content.
- Low entry fee and a simplified cost model that reduces billing complexity by having fewer meters needed to plan for.
- Azure to Front Door integrated egress pricing that removes the separate egress charge from Azure regions to Azure Front Door.

Intelligent secure internet perimeter

- Secure applications with built-in layer 3-4 DDoS protection, seamlessly attached Web Application Firewall (WAF), and Azure DNS to protect your domains.
- Protect your applications against layer 7 DDoS attacks using WAF.
- Protect your applications from malicious actors with Bot manager rules based on Microsoft's own Threat Intelligence.
- Privately connect to your backend behind Azure Front Door with Private Link and embrace a zero-trust access model.
- Provide a centralized security experience for your application via Azure Policy and Azure Advisor that ensures consistent security features across apps.

DDoS protection on Front Door

- Front Door is protected by the default Azure infrastructure DDoS protection. The full scale and capacity of Front Door's globally deployed network provides defense against common network layer attacks through always-on traffic monitoring and real-time mitigation.
- Front Door only accepts traffic on the HTTP and HTTPS protocols, and will only process valid requests with a known Host header. This behavior helps to mitigate some common DDoS attack types including volumetric attacks that are spread across a range of protocols and ports, DNS amplification attacks, and TCP poisoning attacks.
- Front Door is located at the edge of Azure's network, absorbing and geographically isolating large volume attacks.
- Front Door's caching capabilities can be used to protect backends from large traffic volumes generated by an attack. Cached resources will be returned from the Front Door edge nodes so they don't get forwarded to your backend. Even short cache expiry times (seconds or minutes) on dynamic responses can greatly reduce load on backend services.
- Enable Azure DDoS Protection on the origin VNet to protect your public IPs against DDoS attacks. DDoS Protection customers receive extra benefits including cost protection, SLA guarantee, and access to experts from the DDoS Rapid Response Team for immediate help during an attack.

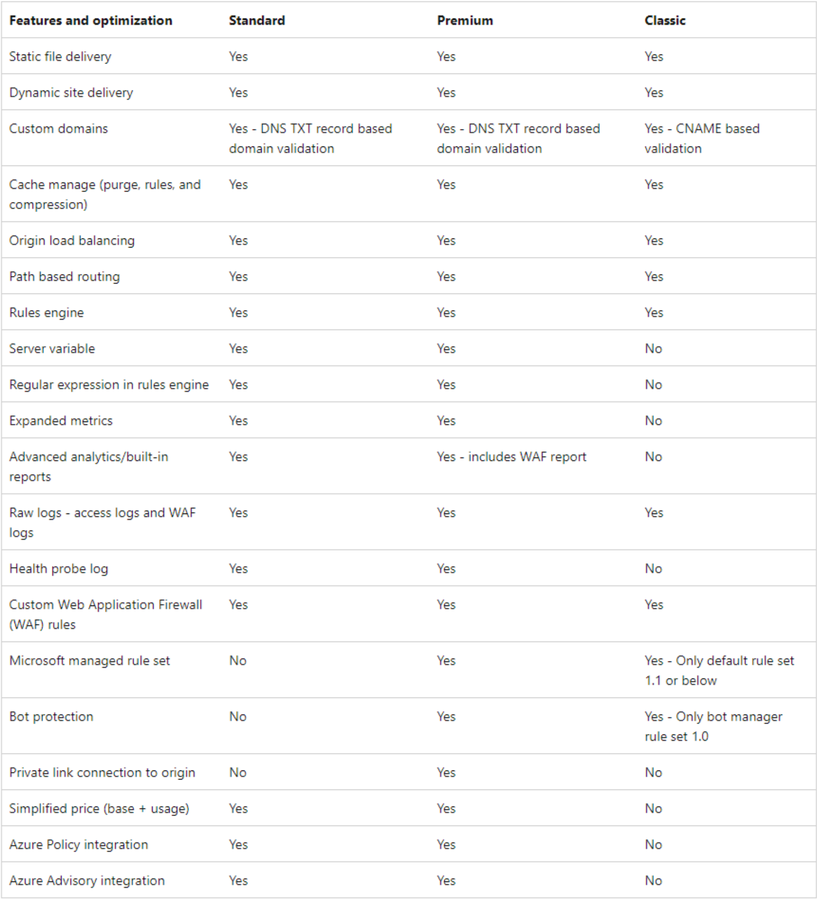
Web Application Firewall (WAF)

Front Door's Web Application Firewall (WAF) can be used to mitigate many different types of attacks:

- Using the managed rule set provides protection against many common attacks.
- Traffic from outside a defined geographic region, or within a defined region, can be blocked or redirected to a static webpage.
- IP addresses and ranges that you identify as malicious can be blocked.
- Rate limiting can be applied to prevent IP addresses from calling your service too frequently.

- You can create custom WAF rules to automatically block and rate limit HTTP or HTTPS attacks that have known signatures.
- Using the bot protection managed rule set provides protection against known bad bots.

Service Tiers



Features and optimization	Standard	Premium	Classic
Static file delivery	Yes	Yes	Yes
Dynamic site delivery	Yes	Yes	Yes
Custom domains	Yes - DNS TXT record based domain validation	Yes - DNS TXT record based domain validation	Yes - CNAME based validation
Cache manage (purge, rules, and compression)	Yes	Yes	Yes
Origin load balancing	Yes	Yes	Yes
Path based routing	Yes	Yes	Yes
Rules engine	Yes	Yes	Yes
Server variable	Yes	Yes	No
Regular expression in rules engine	Yes	Yes	No
Expanded metrics	Yes	Yes	No
Advanced analytics/built-in reports	Yes	Yes - includes WAF report	No
Raw logs - access logs and WAF logs	Yes	Yes	Yes
Health probe log	Yes	Yes	No
Custom Web Application Firewall (WAF) rules	Yes	Yes	Yes
Microsoft managed rule set	No	Yes	Yes - Only default rule set 1.1 or below
Bot protection	No	Yes	Yes - Only bot manager rule set 1.0
Private link connection to origin	No	Yes	No
Simplified price (base + usage)	Yes	Yes	No
Azure Policy integration	Yes	Yes	No
Azure Advisory integration	Yes	Yes	No

Azure DDoS protection

Overview

- Azure DDoS Protection, combined with application design best practices, provides enhanced DDoS mitigation features to defend against DDoS attacks.
- Azure DDoS Protection protects at layer 3 and layer 4 network layers. For web applications protection at layer 7, you need to add protection at the application layer using a WAF offering.

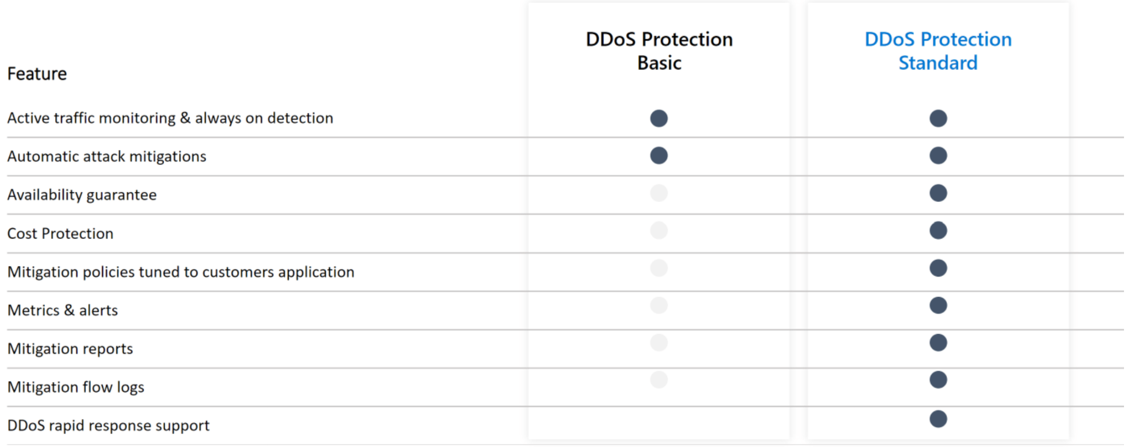
Key benefits

- **Always-on traffic monitoring:** Your application traffic patterns are monitored 24 hours a day, 7 days a week, looking for indicators of DDoS attacks. Azure DDoS Protection instantly and automatically mitigates the attack, once it's detected.
- **Adaptive real time tuning**: Intelligent traffic profiling learns your application's traffic over time, and selects and updates the profile that is the most suitable for your service. The profile adjusts as traffic changes over time.
- **DDoS Protection telemetry, monitoring, and alerting**: Azure DDoS Protection applies three auto-tuned mitigation policies (TCP SYN, TCP, and UDP) for each public IP of the protected resource, in the virtual network that has DDoS enabled. The policy thresholds are auto-configured via machine learning-based

network traffic profiling. DDoS mitigation occurs for an IP address under attack only when the policy threshold is exceeded.

- **Azure DDoS Rapid Response**: During an active attack, Azure DDoS Protection customers have access to the DDoS Rapid Response (DRR) team, who can help with attack investigation during an attack and post-attack analysis. For more information, see Azure DDoS Rapid Response.
- **SKU**: Azure DDoS Protection is offered in two available SKUs, DDoS IP Protection and DDoS Network Protection.
- **Native platform integration**: Natively integrated into Azure. Includes configuration through the Azure portal. Azure DDoS Protection understands your resources and resource configuration.
- **Turnkey protection**: Simplified configuration immediately protects all resources on a virtual network as soon as DDoS Network Protection is enabled. No intervention or user definition is required. Similarly, simplified configuration immediately protects a public IP resource when DDoS IP Protection is enabled for it.
- **Multi-Layered protection**: When deployed with a web application firewall (WAF), Azure DDoS Protection protects both at the network layer (Layer 3 and 4, offered by Azure DDoS Protection) and at the application layer (Layer 7, offered by a WAF). WAF offerings include Azure Application Gateway WAF SKU and third-party web application firewall offerings available in the Azure Marketplace.
- **Extensive mitigation scale**: All L3/L4 attack vectors can be mitigated, with global capacity, to protect against the largest known DDoS attacks.
- **Attack analytics**: Get detailed reports in five-minute increments during an attack, and a complete summary after the attack ends. Stream mitigation flow logs to Microsoft Sentinel or an offline security information and event management (SIEM) system for near real-time monitoring during an attack. See View and configure DDoS diagnostic logging to learn more.
- **Attack metrics**: Summarized metrics from each attack are accessible through Azure Monitor. See View and configure DDoS protection telemetry to learn more.
- **Attack alerting**: Alerts can be configured at the start and stop of an attack, and over the attack's duration, using built-in attack metrics. Alerts integrate into your operational software like Microsoft Azure Monitor logs, Splunk, Azure Storage, Email, and the Azure portal. See View and configure DDoS protection alerts to learn more.
- **Cost guarantee**: Receive data-transfer and application scale-out service credit for resource costs incurred as a result of documented DDoS attacks.

Service Tiers



Feature	DDoS Protection Basic	DDoS Protection Standard
Active traffic monitoring & always on detection	●	●
Automatic attack mitigations	●	●
Availability guarantee	○	●
Cost Protection	○	●
Mitigation policies tuned to customers application	○	●
Metrics & alerts	○	●
Mitigation reports	○	●
Mitigation flow logs	○	●
DDoS rapid response support		●

Choose a load balancing solution using Azure

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**. 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?

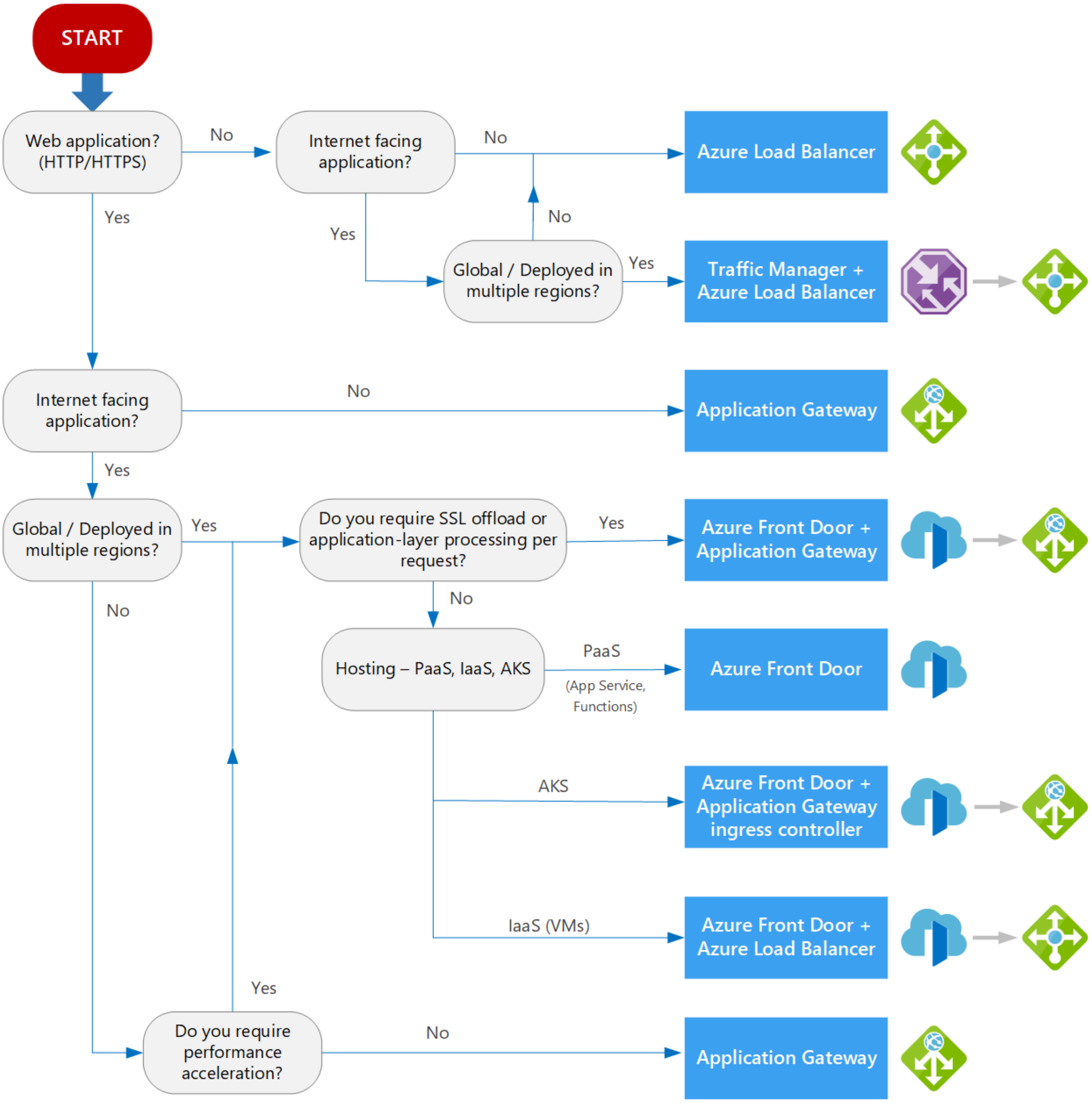
Overall features

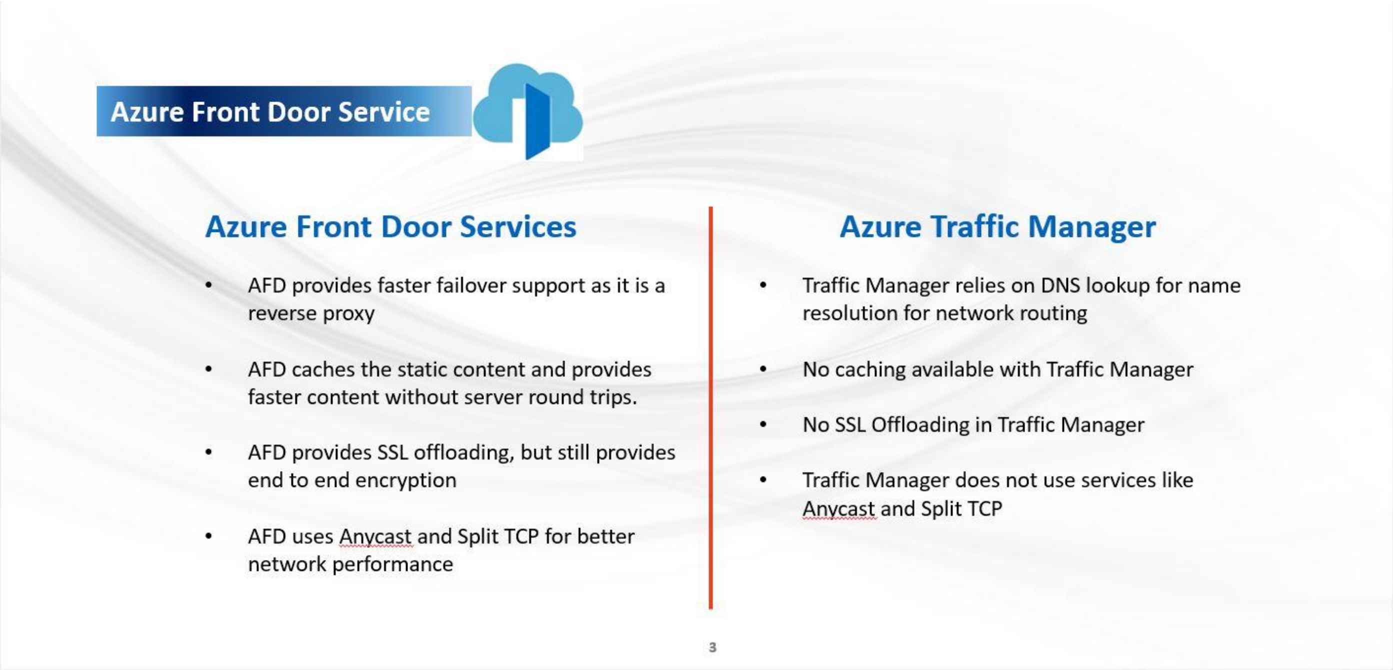
- **Front Door** is an application delivery network that provides global load balancing and site acceleration service for web applications. It offers Layer 7 (Application) capabilities for your application like SSL offload, path-based routing, fast failover, caching, etc, to improve performance and high-availability of your applications.
- **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. Because Traffic Manager is a DNS-based load-balancing service, it load balances only at the domain level. For that reason, it can't fail over as quickly as Front Door, because of common challenges around DNS caching and systems not honoring DNS TTLs.
- **Application Gateway** provides application delivery controller (ADC) as a service, offering various Layer 7 (Application) load-balancing capabilities. Use it to optimize web farm productivity by offloading CPU-intensive SSL termination to the gateway.
- **Azure Load Balancer** is a high-performance, ultra low-latency Layer 4 (Transport) load-balancing service (inbound and outbound) for all UDP and TCP protocols. It is built to handle millions of requests per second while ensuring your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across Availability Zones.

Summary

- **Azure Load Balancer**: Non-HTTP/HTTPS traffic. Could be internet facing (Public Load Balancer) or not (Private Load Balancer), but is deployed on a single region.
- **Traffic Manager + Azure Load Balancer**: Non-HTTP/HTTPS traffic. It is publicly accessible from the internet and is Global (deployed in multiple regions).
- **Application Gateway**: Traffic from Web applications (HTTP/HTTPS). Could be publicly accessible from the internet or not, could be Global or not and do not accelerate applications.
- **Azure Front Door**: Traffic from Web applications (HTTP/HTTPS), it is publicly accessible from internet, Global (deployed in multiple regions) and hosting is PaaS (App Service, Azure Functions).
- **Azure Front Door + Application Gateway**: Traffic from Web applications (HTTP/HTTPS), it is publicly accessible from internet, Global (deployed in multiple regions) and require SSL offload of application-layer processing per request.
- **Azure Front Door + Application Gateway ingress controller**: Traffic from Web applications (HTTP/HTTPS), it is publicly accessible from internet, Global (deployed in multiple regions) and hosting is AKS.
- **Azure Front Door + Azure Load Balancer**: Traffic from Web applications (HTTP/HTTPS), it is publicly accessible from internet, Global (deployed in multiple regions) and hosting is IaaS (VMs).

Service	Global/regional	Recommended traffic
Azure Front Door	Global	HTTP(S)
Traffic Manager	Global	non-HTTP(S)
Application Gateway	Regional	HTTP(S)
Azure Load Balancer	Regional	non-HTTP(S)


START
Web application? (HTTP/HTTPS)
No
Internet facing application?
No
Azure Load Balancer
Yes
No
Yes
Global / Deployed in multiple regions?
Yes
Traffic Manager + Azure Load Balancer
Internet facing application?
No
Application Gateway
Yes
Global / Deployed in multiple regions?
Yes
Do you require SSL offload or application-layer processing per request?
Yes
Azure Front Door + Application Gateway
No
No
Hosting – PaaS, IaaS, AKS
PaaS
(App Service, Functions)
Azure Front Door
AKS
Azure Front Door + Application Gateway ingress controller
IaaS (VMs)
Azure Front Door + Azure Load Balancer
Yes
Do you require performance acceleration?
No
Application Gateway



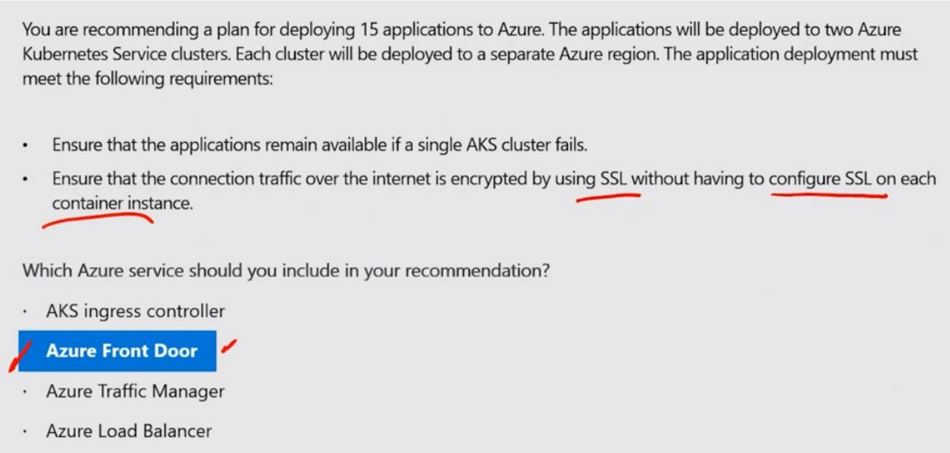
Azure Front Door Service

Azure Front Door Services

- AFD provides faster failover support as it is a reverse proxy
- AFD caches the static content and provides faster content without server round trips.
- AFD provides SSL offloading, but still provides end to end encryption
- AFD uses Anycast and Split TCP for better network performance

Azure Traffic Manager

- Traffic Manager relies on DNS lookup for name resolution for network routing
- No caching available with Traffic Manager
- No SSL Offloading in Traffic Manager
- Traffic Manager does not use services like Anycast and Split TCP

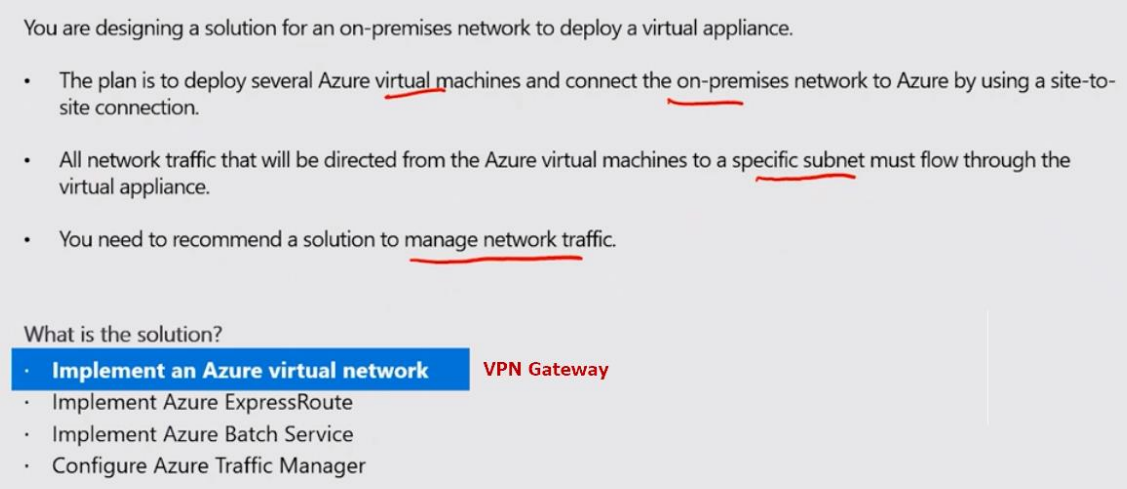


You are recommending a plan for deploying 15 applications to Azure. The applications will be deployed to two Azure Kubernetes Service clusters. Each cluster will be deployed to a separate Azure region. The application deployment must meet the following requirements:

- Ensure that the applications remain available if a single AKS cluster fails.
- Ensure that the connection traffic over the internet is encrypted by using SSL without having to configure SSL on each container instance.

Which Azure service should you include in your recommendation?

- AKS ingress controller
- [✓ Azure Front Door] ✓
- Azure Traffic Manager
- Azure Load Balancer



You are designing a solution for an on-premises network to deploy a virtual appliance.

- The plan is to deploy several Azure virtual machines and connect the on-premises network to Azure by using a site-to-site connection.
- All network traffic that will be directed from the Azure virtual machines to a specific subnet must flow through the virtual appliance.
- You need to recommend a solution to manage network traffic.

What is the solution?

- [✓ Implement an Azure virtual network] VPN Gateway
- Implement Azure ExpressRoute
- Implement Azure Batch Service
- Configure Azure Traffic Manager