Secure your cloud data

Security, responsibility, and trust in Azure

Security is a shared responsibility

The first shift you'll make is from on-premises data centers to infrastructure as a service (IaaS). With IaaS, you are leveraging the lowest-level service and asking Azure to create virtual machines (VMs) and virtual networks. At this level, it's still your responsibility to patch and secure your operating systems and software, as well as configure your network to be secure. At Contoso Shipping, you are taking advantage of IaaS when you start using Azure VMs instead of your on-premises physical servers. In addition to the operational advantages, you receive the security advantage of having outsourced concern over protecting the physical parts of the network.

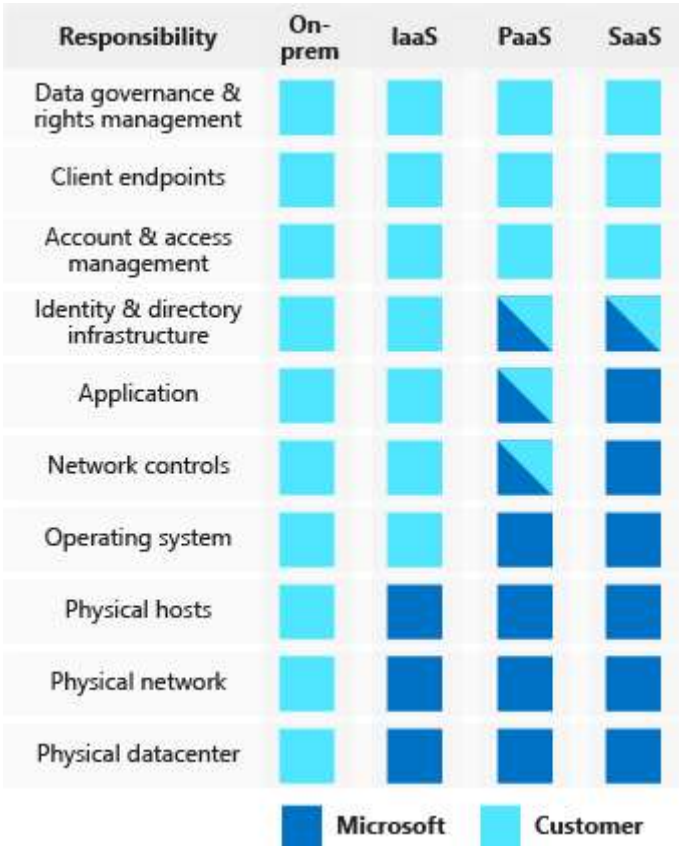
Moving to platform as a service (PaaS) outsources several security concerns. At this level, Azure is taking care of the operating system and of most foundational software like database management systems. Everything is updated with the latest security patches and can be integrated with Azure Active Directory for access controls. PaaS also comes with many operational advantages. Rather than building whole infrastructures and subnets for your environments by hand, you can "point and click" within the Azure portal or run automated scripts to bring complex, secured systems up and down, and scale them as needed. Contoso Shipping uses Azure Event Hubs for ingesting telemetry data from drones and trucks — as well as a web app with an Azure Cosmos DB back end with its mobile apps — which are all examples of PaaS.

With software as a service (SaaS), you outsource almost everything. SaaS is software that runs with an internet infrastructure. The code is controlled by the vendor but configured to be used by the customer. Like so many companies, Contoso Shipping uses Microsoft 365 (formerly Office 365), which is a great example of SaaS!

For all cloud deployment types, you own your data and identities. You are responsible for helping secure your data and identities, your on-premises resources, and the cloud components you control (which vary by service type).

Regardless of the deployment type, you always retain responsibility for the following items:

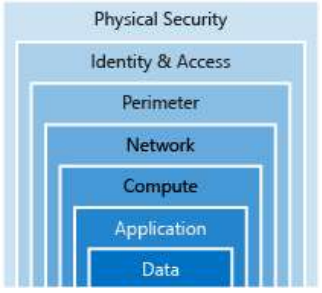
- Data
- Endpoints
- Accounts
- Access management



A layered approach to security

Defense in depth is a strategy that employs a series of mechanisms to slow the advance of an attack aimed at acquiring unauthorized access to information. Each layer provides protection so that if one layer is breached, a subsequent layer is already in place to prevent further exposure. Microsoft applies a layered approach to security, both in physical data centers and across Azure services. The objective of defense in depth is to protect and prevent information from being stolen by individuals who are not authorized to access it.

Defense in depth can be visualized as a set of concentric rings, with the data to be secured at the center. Each ring adds an additional layer of security around the data. This approach removes reliance on any single layer of protection and acts to slow down an attack and provide alert telemetry that can be acted upon, either automatically or manually. Let's take a look at each of the layers.



Data

In almost all cases, attackers are after data:

- Stored in a database
- Stored on disk inside virtual machines
- Stored on a SaaS application such as Microsoft 365
- Stored in cloud storage

It's the responsibility of those storing and controlling access to data to ensure that it's properly secured. Often, there are regulatory requirements that dictate the controls and processes that must be in place to ensure the confidentiality, integrity, and availability of the data.

Application

- Ensure applications are secure and free of vulnerabilities.
- Store sensitive application secrets in a secure storage medium.
- Make security a design requirement for all application development.

Integrating security into the application development life cycle will help reduce the number of vulnerabilities introduced in code. We encourage all development teams to ensure their applications are secure by default, and that they're making security requirements non-negotiable.

Compute

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

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

Networking

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

At this layer, the focus is on limiting the network connectivity across all your resources to allow only what is required. By limiting this communication, you reduce the risk of lateral movement throughout your network.

Perimeter

- Use distributed denial of service (DDoS) protection to filter large-scale attacks before they can cause a denial of service for end users.
- Use perimeter firewalls to identify and alert on malicious attacks against your network.

At the network perimeter, it's about protecting from network-based attacks against your resources. Identifying these attacks, eliminating their impact, and alerting you when they happen are important ways to keep your network secure.

Identity and access

- Control access to infrastructure and change control.
- Use single sign-on and multi-factor authentication.
- Audit events and changes.

The identity and access layer is all about ensuring identities are secure, access granted is only what is needed, and changes are logged.

Physical security

- Physical building security and controlling access to computing hardware within the data center is the first line of defense.

With physical security, the intent is to provide physical safeguards against access to assets. These safeguards ensure that other layers can't be bypassed, and loss or theft is handled appropriately.

Azure helps alleviate your security concerns. But security is still a shared responsibility. How much of that responsibility falls on us depends on which model we use with Azure. We use the *defense in depth* rings as a guideline for considering what protections are adequate for our data and environments.

Azure Security Center

A great place to start when examining the security of your Azure-based solutions is **Azure Security Center.** Security Center is a monitoring service that provides threat protection across all of your services both in Azure, and on-premises. Security Center can:

- Provide security recommendations based on your configurations, resources, and networks.
- Monitor security settings across on-premises and cloud workloads, and automatically apply required security to new services as they come online.
- Continuously monitor all your services, and perform automatic security assessments to identify potential vulnerabilities before they can be exploited.
- Use machine learning to detect and block malware from being installed on your virtual machines and services. You can also define a list of allowed applications to ensure that only the apps you validate are allowed to execute.
- Analyze and identify potential inbound attacks, and help to investigate threats and any post-breach activity that might have occurred.
- Provide just-in-time access control for ports, reducing your attack surface by ensuring the network only allows traffic that you require.

Authentication and authorization

Two fundamental concepts that need to be understood when talking about identity and access control are authentication and authorization. They underpin everything else that happens and occur sequentially in any identity and access process:

- **Authentication** is the process of establishing the identity of a person or service looking 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 use. It establishes if they are 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.

What is Azure Active Directory?

Azure AD is a cloud-based identity service. It has built in support for synchronizing with your existing on-premises Active Directory or can be used stand-alone. This means that all your applications, whether on-premises, in the cloud (including Microsoft 365), or even mobile can share the same credentials. Administrators and developers can control access to internal and external data and applications using centralized rules and policies configured in Azure AD.

Azure AD provides services such as:

- **Authentication.** This includes verifying identity to access applications and resources, and providing functionality such as self-service password reset, multi-factor authentication (MFA), a custom banned password list, and smart lockout services.
- **Single-Sign-On (SSO).** SSO enables users to remember only one ID and one password to access multiple applications. A single identity is tied to a user, simplifying the security model. As users change roles or leave an organization,

access modifications are tied to that identity, greatly reducing the effort needed to change or disable accounts.

- **Application management.** You can manage your cloud and on-premises apps using Azure AD Application Proxy, SSO, the My apps portal (also referred to as Access panel), and SaaS apps.
- **Business to business (B2B) identity services.** Manage your guest users and external partners while maintaining control over your own corporate data
- **Business-to-Customer (B2C) identity services.** Customize and control how users sign up, sign in, and manage their profiles when using your apps with services.
- **Device Management.** Manage how your cloud or on-premises devices access your corporate data.

Single sign-on

The more identities a user has to manage, the greater the risk of a credential-related security incident. More identities mean more passwords to remember and change. Password policies can vary between applications and, as complexity requirements increase, it becomes increasingly difficult for users to remember them.

Now, consider the logistics of managing all those identities. Additional strain is placed on help desks as they deal with account lockouts and password reset requests. If a user leaves an organization, tracking down all those identities and ensuring they are disabled can be challenging. If an identity is overlooked, this could allow access when it should have been eliminated.

With single sign-on (SSO), users need to remember only one ID and one password. Access across applications is granted to a single identity tied to a user, simplifying the security model. As users change roles or leave an organization, access modifications are tied to the single identity, greatly reducing the effort needed to change or disable accounts. Using single sign-on for accounts will make it easier for users to manage their identities and will increase the security capabilities in your environment.

SSO with Azure Active Directory

By leveraging Azure AD for SSO you'll also have the ability to combine multiple data sources into an intelligent security graph. This security graph enables the ability to provide threat analysis and real-time identity protection to all accounts in Azure AD, including accounts that are synchronized from your on-premises AD. By using a centralized identity provider, you'll have centralized the security controls, reporting, alerting, and administration of your identity infrastructure.

Multi-factor authentication

Multi-factor authentication (MFA) provides additional security for your identities by requiring two or more elements for full authentication. These elements fall into three categories:

- Something you know

- Something you possess
- Something you are

Something you know would be a password or the answer to a security question. **Something you possess** could be a mobile app that receives a notification or a token-generating device. **Something you are** is typically some sort of biometric property, such as a fingerprint or face scan used on many mobile devices.

Using MFA increases security of your identity by limiting the impact of credential exposure. An attacker who has a user's password would also need to have possession of their phone or their security token generator in order to fully authenticate. Authentication with only a single factor verified is insufficient, and the attacker would be unable to use only those credentials to authenticate. The benefits this brings to security are huge, and we can't emphasize enough the importance of enabling MFA wherever possible.

Azure AD has MFA capabilities built in and will integrate with other third-party MFA providers. MFA should be used for users in the Global Administrator role in Azure AD, because these are highly sensitive accounts. All other accounts can also have MFA enabled.

Providing identities to services

It's usually valuable for services to have identities. Often, and against best practices, credential information is embedded in configuration files. With no security around these configuration files, anyone with access to the systems or repositories can access these credentials and risk exposure.

Azure AD addresses this problem through two methods: service principals and managed identities for Azure services.

Service principals

To understand service principals, it's useful to first understand the words **identity** and **principal**, because of how they are used in the identity management world.

An **identity** is just a thing that can be authenticated. Obviously, this includes users with a user name and password, but it can also include applications or other servers, which might authenticate with secret keys or certificates.

A **principal** is an identity acting with certain roles or claims. Usually, it is not useful to consider identity and principal separately, but think of using 'sudo' on a Bash prompt in Linux or on Windows using "run as Administrator." In both those cases, you are still logged in as the same identity as before, but you've changed the role under which you are executing. Groups are often also considered principals because they can have rights assigned.

A **service principal** is an identity that is used by a service or application. And like other identities, it can be assigned roles.

Managed identities for Azure services

The creation of service principals can be a tedious process, and there are a lot of touch points that can make maintaining them difficult. Managed identities for Azure services are much easier and will do most of the work for you.

A managed identity can be instantly created for any Azure service that supports it—and the list is constantly growing. When you create a managed identity for a service, you are creating an account on your organization's Active Directory (a specific organization's Active Directory instance is known as an "Active Directory Tenant"). The Azure infrastructure will automatically take care of authenticating the service and managing the account. You can then use that account like any other Azure AD account, including allowing the authenticated service secure access of other Azure resources.

Role-based access control

Roles are sets of permissions, like "Read-only" or "Contributor", that users can be granted to access an Azure service instance.

Identities are mapped to roles directly or through group membership. Separating security principals, access permissions, and resources provides simple access management and fine-grained control. Administrators are able to ensure the minimum necessary permissions are granted.

Roles can be granted at the individual service instance level, but they also flow down the Azure Resource Manager hierarchy.

Privileged Identity Management

In addition to managing Azure resource access with role-based access control (RBAC), a comprehensive approach to infrastructure protection should consider including the ongoing auditing of role members as their organization changes and evolves. Azure AD Privileged Identity Management (PIM) is an additional, paid-for offering that provides oversight of role assignments, self-service, and just-in-time role activation and Azure AD and Azure resource access reviews.

What is encryption?

Encryption is the process of making data unreadable and unusable to unauthorized viewers. To use or read the encrypted data, it must be decrypted, which requires the use of a secret key. There are two top-level types of encryption: **symmetric** and **asymmetric**.

Symmetric encryption uses the same key to encrypt and decrypt the data. Consider a desktop password manager application. You enter your passwords and they are encrypted with your own personal key (your key is often derived from your master password). When the data needs to be retrieved, the same key is used, and the data is decrypted.

Asymmetric encryption uses a public key and private key pair. Either key can encrypt but a single key can't decrypt its own encrypted data. To decrypt, you need the paired key. Asymmetric encryption is used for things like Transport Layer Security (TLS) (used in HTTPS) and data signing.

Encryption at rest

Data at rest is the data that has been stored on a physical medium. This data could be stored on the disk of a server, data stored in a database, or data stored in a storage account. Regardless of the storage mechanism, encryption of data at rest ensures that the stored data is unreadable without the keys and secrets needed to decrypt it. If an attacker was to obtain a hard drive with encrypted data and did not have access to the encryption keys, the attacker would not compromise the data without great difficulty.

Encryption in transit

Data in transit is the data actively moving from one location to another, such as across the internet or through a private network. Secure transfer can be handled by several different layers. It could be done by encrypting the data at the application layer prior to sending it over a network. HTTPS is an example of application layer in transit encryption.

You can also set up a secure channel, like a virtual private network (VPN), at a network layer, to transmit data between two systems.

Encrypting data in transit protects the data from outside observers and provides a mechanism to transmit data while limiting risk of exposure.

Encrypt raw storage

Azure Storage Service Encryption for data at rest helps you protect your data to meet your organizational security and compliance commitments. With this feature, the Azure storage platform automatically encrypts your data before persisting it to Azure Managed Disks, Azure Blob storage, Azure Files, or Azure Queue storage, and decrypts the data before retrieval. The handling of encryption, encryption at rest, decryption, and key management in Storage Service Encryption is transparent to applications using the services.

Encrypt virtual machine disks

Storage Service Encryption provides low-level encryption protection for data written to physical disk, but how do you protect the virtual hard disks (VHDs) of virtual machines? If malicious attackers gained access to your Azure subscription and got the VHDs of your virtual machines, how would you ensure they would be unable to access the stored data?

Azure Disk Encryption is a capability that helps you encrypt your Windows and Linux IaaS virtual machine disks. Azure Disk Encryption leverages the industry-standard BitLocker feature of Windows and the dm-crypt feature of Linux to provide volume encryption for the OS and data disks. The solution is integrated with Azure Key Vault to help you control and

manage the disk encryption keys and secrets (and you can use managed service identities for accessing Key Vault).

Encrypt databases

Transparent data encryption (TDE) helps protect Azure SQL Database and Azure Data Warehouse against the threat of malicious activity. It performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. By default, TDE is enabled for all newly deployed Azure SQL Database instances.

TDE encrypts the storage of an entire database by using a symmetric key called the database encryption key. By default, Azure provides a unique encryption key per logical SQL Server instance and handles all the details. Bring your own key (BYOK) is also supported with keys stored in Azure Key Vault (see below).

Encrypt secrets

We've seen that the encryption services all use keys to encrypt and decrypt data, so how do we ensure that the keys themselves are secure? Corporations may also have passwords, connection strings, or other sensitive pieces of information that they need to securely store. In Azure, we can use **Azure Key Vault** to protect our secrets.

Azure Key Vault is a centralized cloud service for storing your application secrets. Key Vault helps you control your applications' secrets by keeping them in a single, central location and by providing secure access, permissions control, and access logging capabilities. It is useful for a variety of scenarios:

- Secrets management. You can use Key Vault to securely store and tightly control access to tokens, passwords, certificates, Application Programming Interface (API) keys, and other secrets.
- Key management. You also can use Key Vault as a key management solution. Key Vault makes it easier to create and control the encryption keys used to encrypt your data.
- Certificate management. Key Vault lets you provision, manage, and deploy your public and private Secure Sockets Layer/ Transport Layer Security (SSL/ TLS) certificates for your Azure, and internally connected, resources more easily.
- Store secrets backed by hardware security modules (HSMs). The secrets and keys can be protected either by software, or by FIPS 140-2 Level 2 validated HSMs.

The benefits of using Key Vault include:

- Centralized application secrets. Centralizing storage for application secrets allows you to control their distribution, and reduces the chances that secrets may be accidentally leaked.

- Securely stored secrets and keys. Azure uses industry-standard algorithms, key lengths, and HSMs, and access requires proper authentication and authorization.
- Monitor access and use. Using Key Vault, you can monitor and control access to company secrets.
- Simplified administration of application secrets. Key Vault makes it easier to enroll and renew certificates from public Certificate Authorities (CAs). You can also scale up and replicate content within regions, and use standard certificate management tools.
- Integrate with other Azure services. You can integrate Key Vault with storage accounts, container registries, event hubs, and many more Azure services.

Because Azure AD identities can be granted access to use Azure Key Vault secrets, applications with managed service identities enabled can automatically and seamlessly acquire the secrets they need.

Certificates used in Azure are **x.509 v3** and can be signed by a trusted certificate authority, or they can be self-signed. A self-signed certificate is signed by its own creator; therefore, it is not trusted by default. Most browsers can ignore this problem. However, you should only use self-signed certificates when developing and testing your cloud services. These certificates can contain a private or a public key and have a thumbprint that provides a means to identify a certificate in an unambiguous way. This thumbprint is used in the Azure configuration file to identify which certificate a cloud service should use.

Types of certificates

Certificates are used in Azure for two primary purposes and are given a specific designation based on their intended use.

- **Service certificates** are used for cloud services
- **Management certificates** are used for authenticating with the management API

Service certificates

Service certificates are attached to cloud services and enable secure communication to and from the service. For example, if you deploy a web site, you would want to supply a certificate that can authenticate an exposed HTTPS endpoint. Service certificates, which are defined in your service definition, are automatically deployed to the VM that is running an instance of your role.

You can upload service certificates to Azure either using the Azure portal or by using the classic deployment model. Service certificates are associated with a specific cloud service. They are assigned to a deployment in the service definition file.

Management certificates

Management certificates allow you to authenticate with the classic deployment model. Many programs and tools (such as Visual Studio or the Azure SDK) use these certificates to

automate configuration and deployment of various Azure services. However, these types of certificates are not related to cloud services.

Using Azure Key Vault with certificates

You can store your certificates in Azure Key Vault - much like any other secret. However, Key Vault provides additional features above and beyond the typical certificate management.

- You can create certificates in Key Vault, or import existing certificates
- You can securely store and manage certificates without interaction with private key material.
- You can create a policy that directs Key Vault to manage the life cycle of a certificate.
- You can provide contact information for notification about life-cycle events of expiration and renewal of certificate.
- You can automatically renew certificates with selected issuers - Key Vault partner x509 certificate providers / certificate authorities.

A layered approach to network security

You've probably noticed that a common theme throughout this module is the emphasis of a layered approach to security. This approach is also recommended at the network layer. It's not enough to just focus on securing the network perimeter, or focusing on the network security between services inside a network. A layered approach provides multiple levels of protection, so that if an attacker gets through one layer, there are further protections in place to limit further attack.

Let's take a look at how Azure can provide the tools for a layered approach to securing your network footprint.

Internet protection

If we start on the perimeter of the network, we're focused on limiting and eliminating attacks from the internet. We suggest first assessing the resources that are internet-facing, and to only allow inbound and outbound communication where necessary. Make sure you identify all resources that are allowing inbound network traffic of any type, and then ensure they are restricted to only the ports and protocols required. Azure Security Center is a great place to look for this information, because it will identify internet-facing resources that don't have network security groups associated with them, as well as resources that are not secured behind a firewall.

What is a Firewall?

A firewall is a service that grants server access based on the originating IP address of each request. You create firewall rules that specify ranges of IP addresses. Only clients from these granted IP addresses will be allowed to access the server. Firewall rules, generally speaking, also include specific network protocol and port information.

To provide inbound protection at the perimeter, you have several choices.

- **Azure Firewall** is a managed, cloud-based, network security service that protects your Azure Virtual Network resources. It is a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability. Azure Firewall provides inbound protection for non-HTTP/S protocols. Examples of non-HTTP/S protocols include: Remote Desktop Protocol (RDP), Secure Shell (SSH), and File Transfer Protocol (FTP). It also provides outbound, network-level protection for all ports and protocols, and application-level protection for outbound HTTP/S.
- **Azure Application Gateway** is a load balancer that includes a Web Application Firewall (WAF) that provides protection from common, known vulnerabilities in websites. It is designed to protect HTTP traffic.
- **Network virtual appliances (NVAs)** are ideal options for non-HTTP services or advanced configurations, and are similar to hardware firewall appliances.

Stopping Distributed Denial of Service (DDoS) attacks

Any resource exposed on the internet is at risk of being attacked by a denial of service attack. These types of attacks attempt to overwhelm a network resource by sending so many requests that the resource becomes slow or unresponsive.

When you combine **Azure DDoS Protection** with application design best practices, you help provide defense against DDoS attacks. DDoS Protection leverages the scale and elasticity of Microsoft's global network to bring DDoS mitigation capacity to every Azure region. The Azure DDoS Protection service protects your Azure applications by monitoring traffic at the Azure network edge before it can impact your service's availability. Within a few minutes of attack detection, you are notified using Azure Monitor metrics.

Azure DDoS Protection provides the following service tiers:

- **Basic** - The Basic service tier is automatically enabled as part of the Azure platform. Always-on traffic monitoring and real-time mitigation of common network-level attacks provide the same defenses that Microsoft's online services use. Azure's global network is used to distribute and mitigate attack traffic across regions.
- **Standard** - The Standard service tier provides additional mitigation capabilities that are tuned specifically to Microsoft Azure Virtual Network resources. DDoS Protection Standard is simple to enable and requires no application changes. Protection policies are tuned through dedicated traffic monitoring and machine learning algorithms. Policies are applied to public IP addresses associated with resources deployed in virtual networks, such as Azure Load Balancer and Application Gateway.

Controlling the traffic inside your virtual network

Virtual network security

Once inside a virtual network (VNet), it's crucial that you limit communication between resources to only what is required.

For communication between virtual machines, Network Security Groups (NSGs) are a critical piece to restrict unnecessary communication.

Network Security Groups allow you to filter network traffic to and from Azure resources in an Azure virtual network. An NSG can contain multiple inbound and outbound security rules that enable you to filter traffic to and from resources by source and destination IP address, port, and protocol. They provide a list of allowed and denied communication to and from network interfaces and subnets, and are fully customizable.

You can completely remove public internet access to your services by restricting access to service endpoints. With service endpoints, Azure service access can be limited to your virtual network.

Network integration

It's common to have existing network infrastructure that needs to be integrated to provide communication from on-premises networks or to provide improved communication between services in Azure. There are a few key ways to handle this integration and improve the security of your network.

Virtual private network (VPN) connections are a common way of establishing secure communication channels between networks. Connections between Azure Virtual Network and an on-premises VPN device are a great way to provide secure communication between your network and your VNet on Azure.

To provide a dedicated, private connection between your network and Azure, you can use Azure ExpressRoute. ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Microsoft 365, and Dynamics 365. ExpressRoute connections improve the security of your on-premises communication by sending this traffic over the private circuit instead of over the public internet. You don't need to allow access to these services for your end users over the public internet, and you can send this traffic through appliances for further traffic inspection.

Protect your shared documents

Microsoft Azure Information Protection (sometimes referred to as AIP) is a cloud-based solution that helps organizations classify and optionally protect documents and emails by applying labels.

Labels can be applied automatically based on rules and conditions. Labels can also be applied manually. You can also guide users to choose recommended labels with a combination of automatic and manual steps.

Azure Advanced Threat Protection

Azure Advanced Threat Protection (Azure ATP) is a cloud-based security solution that identifies, detects, and helps you investigate advanced threats, compromised identities, and malicious insider actions directed at your organization.

Azure ATP is capable of detecting known malicious attacks and techniques, security issues, and risks against your network.

Azure ATP portal

Azure ATP has its own portal, through which you can monitor and respond to suspicious activity. The Azure ATP portal allows you to create your Azure ATP instance, and view the data received from Azure ATP sensors. You can also use the portal to monitor, manage, and investigate threats in your network environment. You can sign in to the Azure ATP portal at https://portal.atp.azure.com . Your user accounts must be assigned to an Azure AD security group that has access to the Azure ATP portal to be able to sign in.

Azure ATP sensor

Azure ATP sensors are installed directly on your domain controllers. The sensor monitors domain controller traffic without requiring a dedicated server or configuring port mirroring.

Azure ATP cloud service

Azure ATP cloud service runs on Azure infrastructure and is currently deployed in the United States, Europe, and Asia. Azure ATP cloud service is connected to Microsoft's intelligent security graph.

Understand Security Considerations for Application Lifecycle Management Solutions

The **Microsoft Security Development Lifecycle** (SDL) introduces security and privacy considerations throughout all phases of the development process. It helps developers build highly secure software, address security compliance requirements, and reduce development costs. The guidance, best practices, tools, and processes in the SDL are practices used internally at Microsoft to build more secure products and services.

Links

- Microsoft Learn: Design for Security
- Azure Security (Trust Center)
- Azure Security Center planning and operations guide
- What is Microsoft Azure Information Protection?
- Azure Advanced Threat Protection
- Configuring SSL for an application in Azure.
- What are service certificates?
- What are management certificates?

- Get started with Key Vault certificates

Summary

1. Cloud security is a shared responsibility between you and your cloud provider. Which category of cloud services requires the greatest security effort on your part?
Infrastructure as a service (IaaS)
At this level, the cloud provider provides physical security to compute resources. However, it's your responsibility to patch and secure your operating systems and software, as well as configure your network to be secure.

2. Which of these options helps you most easily disable an account when an employee leaves your company?
Use single sign-on (SSO)
SSO centralizes user identity, so you can disable an inactive account in a single step.

3. Which of these approaches is the strongest way to protect sensitive customer data?
Encrypt data both as it sits in your database and as it travels over the network
Encrypting your data at all times, both as it sits in your database and as it travels over the network, minimizes the opportunity for an attacker to access your data in plain text.

4. There has been an attack on your public-facing website, and the application's resources have been overwhelmed and exhausted, and are now unavailable to users. What service should you use to prevent this type of attack?
DDoS protection
DDoS protection is the correct answer, because it will help prevent DDoS attacks.

5. You want to store certificates in Azure to centrally manage them for your services. Which Azure service should you use?
Azure Key Vault
Azure Key Vault is the correct answer, because it is a centralized cloud service for storing application secrets, referred to as a secret store.

Top 5 security items to consider before pushing to production

What is Azure Security Center?

Azure Security Center (ASC) is a monitoring service that provides threat protection across all of your services both in Azure, and on-premises. It can:

- Provide security recommendations based on your configurations, resources, and networks.
- Monitor security settings across on-premises and cloud workloads and automatically apply required security to new services as they come online.
- Continuously monitor all your services and perform automatic security assessments to identify potential vulnerabilities before they can be exploited.
- Use machine learning to detect and block malware from being installed in your services and virtual machines. You can also allowlist applications to ensure that only the apps you validate are allowed to execute.
- Analyze and identify potential inbound attacks and help to investigate threats and any post-breach activity which might have occurred.
- Just-In-Time access control for ports, reducing your attack surface by ensuring the network only allows traffic you require.

Always encode your output

Any output you present either visually or within a document should always be encoded and escaped. This can protect you in case something was missed in the sanitization pass, or the code accidentally generates something that can be used maliciously. This design principle will make sure that everything is displayed as output and not inadvertently interpreted as something that should be executed, which is another common attack technique that is referred to as "Cross-Site Scripting" (XSS).

Since XSS prevention is a common application requirement, this security technique is another area where ASP.NET will do the work for you. By default, all output is already encoded. If you are using another web framework, you can verify your options for output encoding on websites with the OWASP XSS Prevention Cheatsheet.

What is Azure Key Vault

Azure Key Vault is a secret store: a centralized cloud service for storing application secrets. Key Vault keeps your confidential data safe by keeping application secrets in a single central location and providing secure access, permissions control, and access logging.

Secrets are stored in individual vaults, each with their own configuration and security policies to control access. You can then get to your data through a REST API, or through a client SDK available for most languages.

Why use a Key Vault for my secrets

Key management and storing secrets can be complicated and error-prone when performed manually. Rotating certificates manually means potentially going without for a few hours, or days. As mentioned above, saving your connections strings in your configuration file or code repository means someone could steal your credentials.

Key Vault allows users to store connection strings, secrets, passwords, certificates, access policies, file locks (making items in Azure read-only), and automation scripts. It also logs access and activity, allows you to monitor access control (IAM) in your subscription, and it also has diagnostic, metrics, alerts and troubleshooting tools, to ensure you have the access you need.

Summary

1. True or false: Azure Security Center works for Azure resources and on-premises resources.
True
Azure Security Center can monitor all your services, both in Azure and on premises.

2. Which of the following data sources need to be validated?
Data from a 3rd party API
Data from the URL parameter
Data collected from the user via an input field
All of the above
All these sources of data need to be validated. Never trust any data that could have been modified.

3. Parameterized queries (stored procedures in SQL) are a secure way to talk to the database because:
Parameterized queries substitute variables before running queries, meaning it avoids the opportunity for code to be submitted in place of a variable.
Parameter fields used in parameterized queries are treated as data, not code, protecting against injection vulnerabilities. For more information on how to implement parameterized queries please see the OWASP Query Parameterization Cheat Sheet.

4. Which of the following data needs to be output encoded?
Data to be output to the screen
Data sent to the screen needs to be output encoded to ensure it is never interpreted as code.

Configure security policies to manage data

Digital data always exists in one of three states: at **rest**, in **process**, and in **transit.**

All three states require unique technical solutions for data classification, but the applied principles of data classification should be the same for each. Data that is classified as confidential needs to stay **confidential** in each state.

Protect data at rest

Data encryption at rest is a mandatory step toward data privacy, compliance, and data sovereignty.

Best practice	Solution
Apply disk encryption to help safeguard your data.	Use Microsoft Azure Disk Encryption, which enables IT administrators to encrypt both Windows infrastructure as a service (IaaS) and Linux IaaS virtual machine (VM) disks. Disk encryption combines the industry-standard BitLocker feature and the Linux DM-Crypt feature to provide volume encryption for the operating system (OS) and the data disks. Azure Storage and Azure SQL Database encrypt data at rest by default, and many services offer encryption as an option. You can use Azure Key Vault to maintain control of keys that access and encrypt your data. See Azure resource providers encryption model support to learn more.
Use encryption to help mitigate risks related to unauthorized data access.	Encrypt your drives before you write sensitive data to them.

Organizations that don’t enforce data encryption risk higher exposure to data-integrity issues. For example, unauthorized users or malicious hackers might steal data in compromised accounts or gain unauthorized access to data coded in Clear Format. To comply with industry regulations, companies also must prove that they are diligent and using correct security controls to enhance their data security.

Protect data in transit

Protecting data in transit should be an essential part of your data protection strategy. Because data is moving back and forth from many locations, we generally recommend that you always use SSL/TLS protocols to exchange data across different locations. In some circumstances, you might want to isolate the entire communication channel between your on-premises and cloud infrastructures by using a VPN.

For data moving between your on-premises infrastructure and Azure, consider appropriate safeguards such as HTTPS or VPN. When sending encrypted traffic between an Azure virtual network and an on-premises location over the public internet, use Azure VPN Gateway.

The following table lists best practices specific to using Azure VPN Gateway, SSL/TLS, and HTTPS.

Best practice	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. If you choose to use ExpressRoute, you can also encrypt the data at the application level by using SSL/TLS or other protocols for added protection.
Interact with Azure Storage through the Azure portal	All transactions occur via HTTPS. You can also use Storage REST API over HTTPS to interact with Azure Storage and Azure SQL Database.

Organizations that fail to protect data in transit are more susceptible to man-in-the-middle attacks, eavesdropping, and session hijacking. These attacks can be the first step in gaining access to confidential data.

Data discovery

Data discovery and classification (currently in preview) provides advanced capabilities built into Azure SQL Database for discovering, classifying, labeling and protecting sensitive data (such as business, personal data (PII), and financial information) in your databases. Finding and classifying this data can play a pivotal role in your organizational information protection stature. It can serve as infrastructure for:

- Helping meet data privacy standards and regulatory compliance requirements.
- Addressing various security scenarios such as monitoring, auditing, and alerting on anomalous access to sensitive data.
- Controlling access to and hardening the security of databases containing highly sensitive data.

Data discovery and classification is part of the Advanced Data Security offering, which is a unified package for advanced Microsoft SQL Server security capabilities. You access and manage data discovery and classification via the central SQL Advanced Data Security portal.

Data discovery and classification introduces a set of advanced services and SQL capabilities, forming a SQL Information Protection paradigm aimed at protecting the data, not just the database:

- **Discovery and recommendations** - The classification engine scans your database and identifies columns containing potentially sensitive data. It then provides you with a more natural way to review and apply the appropriate classification recommendations via the Azure portal.
- **Labeling** - Sensitivity classification labels can be persistently tagged on columns using new classification metadata attributes introduced into the SQL Server Engine. This metadata can then be utilized for advanced sensitivity-based auditing and protection scenarios.
- **Query result set sensitivity** - The sensitivity of the query result set is calculated in real-time for auditing purposes.
- **Visibility** - You can view the database classification state in a detailed dashboard in the Azure portal. Additionally, you can download a report (in Microsoft Excel format) that you can use for compliance and auditing purposes, in addition to other needs.

Classifications have two metadata attributes:

- **Labels** - These are the main classification attributes used to define the sensitivity level of the data stored in the column.
- **Information Types** - These provide additional granularity into the type of data stored in the column.

SQL Information Protection (SQL IP)

SQL IP brings a set of advanced services and SQL capabilities, forming a new information protection paradigm in SQL aimed at protecting the data, not just the database:

- **Azure SQL Auditing –** Azure SQL Auditing tracks database events and writes them to an audit log in your Azure storage account, Log Analytics workspace or Event Hub.
- **Data Discovery & Classifications** – Is built into Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. It provides advanced capabilities for discovering, classifying, labeling, and reporting the sensitive data in your databases.
- **Dynamic data masking –** Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics support dynamic data masking. Dynamic data masking limits sensitive data exposure by masking it to non-privileged users.
- **Security Center –** Scans your database and makes recommendations to improve security. Also allows you to set up and monitor Security Alerts.
- **Transparent data encryption –** Transparent data encryption encrypts your databases, backups, and logs at rest without any changes to your application. To enable encryption, go to each database.

Immutable storage and data retention

Immutable storage for Azure Blob Storage enables users to store business-critical data in a write once, read many (WORM) state. This state makes the data unerasable and unmodifiable for a user-specified interval. Blobs can be created and read, but not modified or deleted, for the duration of the retention interval.

Immutable storage enables:

- **Time-based retention policy support** - Users set policies to store data for a specified interval.
- **Legal hold policy support** - When the retention interval is not known, users can set legal holds to store data immutably until the legal hold is cleared. When a legal hold is set, blobs can be created and read, but not modified or deleted. Each legal hold is associated with a user-defined alphanumeric tag that is used as an identifier string (such as a case ID).
- **Support for all blob tiers** - WORM policies are independent of the Azure Blob Storage tier and apply to all tiers: hot, cool, and archive. Users can transition data to the most cost-optimized tier for their workloads while maintaining data immutability.
- **Container-level configuration** - Users can configure time-based retention policies and legal hold tags at the container level. By using simple container-level settings, users can create and lock time-based retention policies, extend retention intervals, set and clear legal holds, and more. These policies apply to all the blobs in the container, both existing and new.

- **Audit logging support** - Each container includes an audit log, which displays up to five time-based retention commands for locked time-based retention policies. However, the log has a maximum of three logs for retention interval extensions or time-based retention. The log contains the user ID, command type, time stamps, and retention interval. For legal holds, the log contains the user ID, command type, time stamps, and legal hold tags.

The audit log is kept for the lifetime of the container, in accordance with the SEC 17a-4(f) regulatory guidelines. The Azure Activity Log shows a more comprehensive log of all the control plane activities. It is the user's responsibility to store those logs persistently, as might be required for regulatory or other purposes.

Immutable storage for Azure Blob storage supports two types of WORM or immutable policies: time-based retention and legal holds.

Cross-region activities number key

- **Azure Compute (IaaS)** - You must provision additional compute resources in advance to ensure resources are available in another region during a disaster. For more information, see Designing resilient applications for Azure.
- **Azure Storage**- Geo-redundant storage (GRS) is configured by default when an Azure Storage account is created. With GRS, data is automatically replicated three times within the primary region, and three times in a paired region. For more information, see Azure Storage redundancy.
- **Azure SQL Database** - With Azure SQL Database geo-replication, you can configure asynchronous replication of transactions to any region in the world; however, we recommend you deploy these resources in a paired region for most disaster recovery scenarios. For more information, see Configure active geo-replication for Azure SQL Database in the Azure portal.
- **Azure Resource Manager** - Resource Manager inherently provides logical isolation of components across regions. This means that logical failures in one region are less likely to impact other regions.

Benefits of Azure paired regions number key:

- **Physical isolation** - When possible, Azure services prefer at least 300 miles of separation between datacenters in a regional pair (although this isn't practical or possible in all geographies). Physical datacenter separation reduces the likelihood of both regions being affected simultaneously as a result of natural disasters, civil unrest, power outages, or physical network outages. Isolation is subject to the constraints within the geography, such as geography size, power, and network infrastructure availability, and regulations.
- **Platform-provided replication** - Some services such as geo-redundant storage provide automatic replication to the paired region.
- **Region recovery order** - In the event of a widespread outage, recovery of one region is prioritized out of every pair. Applications that are deployed across paired regions are guaranteed to have one of the regions recovered with priority. If an

application is deployed across regions that are not paired, recovery might be delayed. In the worst case, the chosen regions might be the last two to be recovered.

- **Sequential updates** - Planned Azure system updates are rolled out to paired regions sequentially, not at the same time. This helps minimize downtime, the effect of bugs, and logical failures in the rare event of a bad update.
- **Data residency** - To meet data residency requirements for tax and law enforcement jurisdiction purposes, a region resides within the same geography as its pair (with the exception of Brazil South).

Links

- Azure Security Center
- Azure Information Protection
- Azure SQL Database and SQL Data Warehouse data discovery & classification
- Information protection in Microsoft 365

Configure and manage secrets in Azure Key Vault

Azure Key Vault helps safeguard cryptographic keys and secrets that cloud applications and services use. Key Vault streamlines the key management process and enables you to maintain control of keys that access and encrypt your data. Developers can create keys for development and testing in minutes, and then migrate them to production keys. Security administrators can grant (and revoke) permission to keys, as needed.

Azure Key Vault is a centralized cloud service for storing application secrets such as encryption keys, certificates, and server-side tokens. Key Vault helps you control your applications' secrets by keeping them in a single central location and providing secure access, permissions control, and access logging.

There are three primary concepts used in an Azure Key Vault: vaults, keys, and secrets.

Vaults

You use Azure Key Vault to create multiple secure containers, called vaults. Vaults help reduce the chances of accidental loss of security information by centralizing application secrets storage. Organizations will have several key vaults. Each key vault is a collection of cryptographic keys and cryptographically protected data (call them "secrets") managed by one or more responsible individuals within your organization. These key vaults represent the logical groups of keys and secrets for your organization; those that you want to manage together. They are like folders in the file system. Key vaults also control and log the access to anything stored in them.

Keys

Keys are the central actor in the Azure Key Vault service. A given key in a key vault is a cryptographic asset destined for a particular use such as the asymmetric master key of Microsoft Azure RMS, or the asymmetric keys used for SQL Server TDE (Transparent Data Encryption), CLE (Column Level Encryption) and Encrypted backup.

Microsoft and your apps don't have access to the stored keys directly once a key is created or added to a key vault. Applications must use your keys by calling cryptography methods on the Key Vault service. The Key Vault service performs the requested operation within its hardened boundary. The application never has direct access to the keys.

Keys can be single instanced (only one key exists) or be versioned. In the versioned case, a key is an object with a primary (active) key and a collection of one or more secondary (archived) keys created when keys are rolled (renewed). Key Vault supports asymmetric keys (RSA 2048). Your applications may use these for encryption or digital signatures.

There are two variations on keys in Key Vault: hardware-protected, and software-protected.

Hardware protected keys

The Key Vault service supports using HSMs that provide a hardened, tamper-resistant environment for cryptographic processing and key generation. Azure has dedicated HSMs validated to FIPS 140-2 Level 2 that Key Vault uses to generate or store keys. These HSM-backed keys are always locked to the boundary of the HSM. When you ask the Key Vault service to decrypt or sign with a key, the operation is performed inside an HSM.

You can import keys from your own hardware security modules (HSMs) and transfer them to Key Vault without leaving the HSM boundary. This scenario is often referred to as bring your own key, or BYOK. More details on generating your own HSM-protected key and then transferring it to Azure Key Vault is available in the summary of this module. You can also use these Azure HSMs directly through the Microsoft Azure Dedicated Hardware Security Module (HSM) service if you need to migrate HSM-protected apps or maintain a high security compliance requirement.

Software protected keys

Key Vault can also generate and protect keys using software-based RSA and ECC algorithms. In general, software-protected keys offer most of the features as HSM-protected keys except the FIPS 140-2 Level 2 assurance:

- Your key is still isolated from the application (and Microsoft) in a container that you manage
- It's stored at rest encrypted with HSMs
- You can monitor usage using Key Vault logs

The primary difference (besides price) with a software-protected key is when cryptographic operations are performed, they are done in software using Azure compute services while for HSM-protected keys the cryptographic operations are performed within the HSM.

For production use, it's recommended to use HSM-protected keys and use software-protected keys in only test/pilot scenarios.

Secrets

Secrets are small (less than 10K) data blobs protected by a HSM-generated key created with the Key Vault. Secrets exist to simplify the process of persisting sensitive settings that almost every application has: storage account keys, .PFX files, SQL connection strings, data encryption keys, etc.

Key vault uses

With these three elements, an Azure Key Vault helps address the following issues:

- **Secrets management.** Azure Key Vault can securely store (with HSMs) and tightly control access to tokens, passwords, certificates, API keys, and other secrets.
- **Key management.** Azure Key Vault is a cloud-based key management solution, making it easier to create and control the encryption keys used to encrypt your data. Azure services such as App Service integrate directly with Azure Key Vault and can decrypt secrets without knowledge of the encryption keys.
- **Certificate management.** Azure Key Vault is also a service that lets you easily provision, manage, and deploy public and private SSL/TLS certificates for use with Azure and your internal connected resources. It can also request and renew TLS certificates through partnerships with certificate authorities, providing a robust solution for certificate lifecycle management.

Key Vault access has two facets: the management of the Key Vault itself, and accessing the data contained in the Key Vault. Documentation refers to these facets as the management plane and the data plane.

These two areas are separated because the creation of the Key Vault (a management operation) is a different role than storing and retrieving a secret stored in the Key Vault. To access a key vault, all users or applications must have proper authentication to identify the caller, and authorization to determine the operations the caller can perform.

Authentication

Azure Key Vault uses Azure Active Directory to authenticate users and applications that try to access a vault. Authentication is always performed by associated the Azure AD tenant of the subscription that the Key Vault is part of and every user or app making a request must be known to the AAD. There is no support for anonymous access to a Key Vault.

Authorization

Management operations (creating a new Azure Key Vault) use role-based access control (RBAC). There is a built-in role **Key Vault Contributor** that provides access to management features of key vaults, but doesn't allow access to the key vault data. This is the recommended role to use. There's also a **Contributor** role that includes full administration rights - including the ability to grant access to the data plane.

Restricting network access

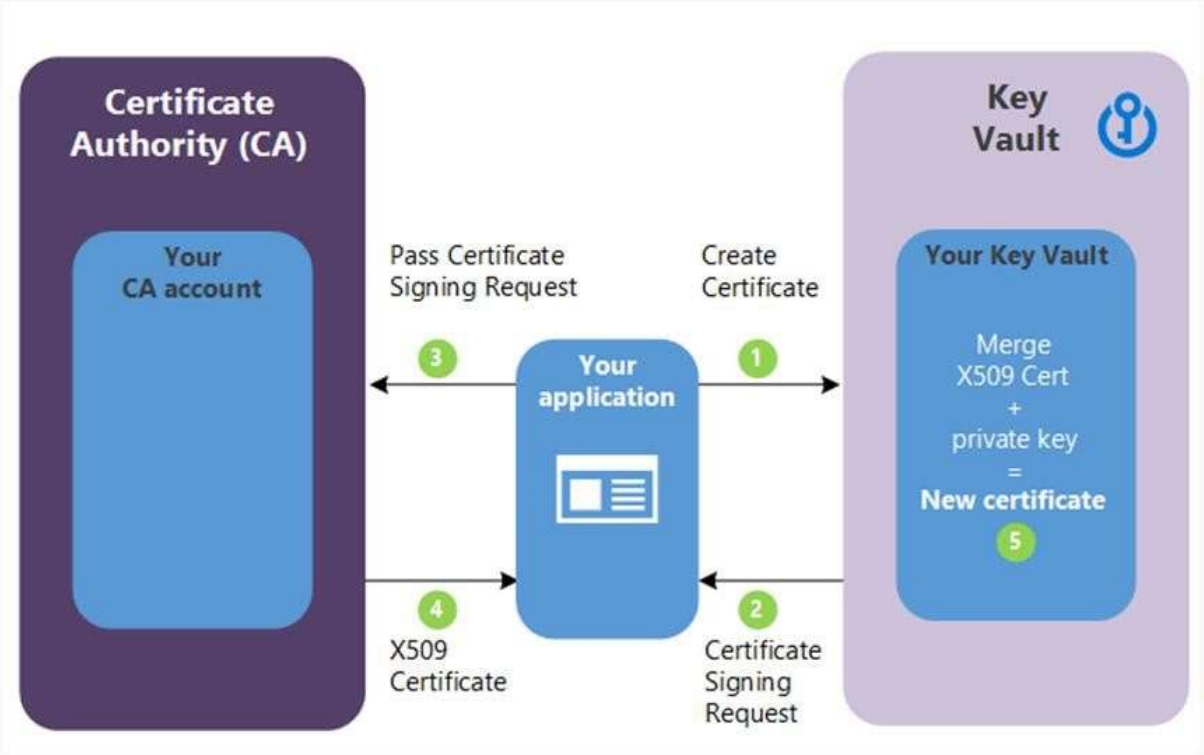
Another point to consider with Azure Key Vault is what services in your network can access the vault. In most cases, the network endpoints don't need to be open to the Internet. You should determine the minimum network access required - for example you can restrict Key Vault endpoints to specific Azure Virtual Network subnets, specific IP addresses, or trusted Microsoft services including Azure SQL, Azure App Service, and various data and storage services that use encryption keys.

Adding certificates to a Key Vault

Azure Key Vault manages X.509 based certificates that can come from several sources.

First, you can create self-signed certificates directly in the Azure portal. This process creates a public/private key pair and signs the certificate with its own key. These certificates can be used for testing and development.

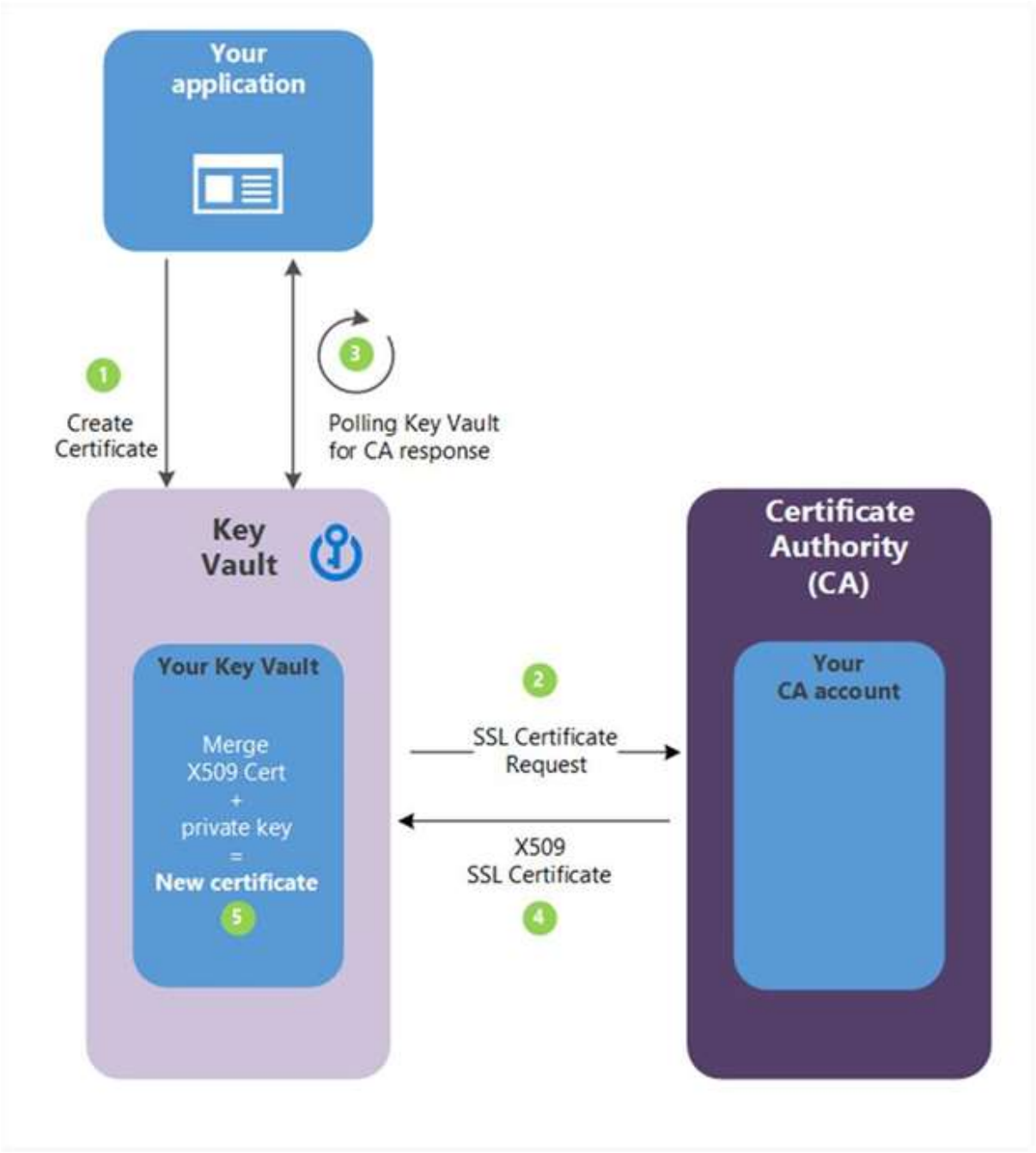
Second, you can create an X.509 certificate signing request (CSR). This creates a public/private key pair in Key Vault along with a CSR you can pass over to your certification authority (CA). The signed X.509 certificate can then be merged with the held key pair to finalize the certificate in Key Vault as shown in the following diagram.



1. In the diagram above, your application is creating a certificate which internally begins by creating a key in your Azure Key Vault.
2. Key Vault returns to your application a Certificate Signing Request (CSR).
3. Your application passes the CSR to your chosen CA.
4. Your chosen CA responds with an X.509 Certificate.
5. Your application completes the new certificate creation with a merger of the X.509 Certificate from your CA.

This approach works with any certificate issuer and provides better security than handling the CSR directly because the private key is created and secured in Azure Key Vault and never revealed.

Third, you can connect your Key Vault with a trusted certificate issuer (referred to as an *integrated* CA) and create the certificate directly in Azure Key Vault. This approach requires a one-time setup to connect the certificate authority. You can then request to create a certificate and the Key Vault will interact directly with the CA to fulfill the request in a similar process to the manual CSR creation process shown above. The full details of this process are presented in the following diagram.



1. In the diagram above, your application is creating a certificate which internally begins by creating a key in your key vault.
2. Key Vault sends a SSL Certificate Request to the CA.
3. Your application polls, in a loop and wait process, for your Key Vault for certificate completion. The certificate creation is complete when Key Vault receives the CA's response with x509 certificate.
4. The CA responds to Key Vault's SSL Certificate Request with an X509 SSL Certificate.
5. Your new certificate creation completes with the merger of the X509 Certificate for the CA.

This approach has several distinct advantages. Because the Key Vault is connected to the issuing CA, it can manage and monitor the lifecycle of the certificate. That means it can automatically renew the certificate, notify you about expiration, and monitor events such as whether the certificate has been revoked.

Finally, you can import existing certificates - this allows you to add certificates to Key Vault that you are already using. The imported certificate can be in either PFX or PEM format and must contain the private key.

Azure App Service integration

Once you have a public/private key pair certificate in your Azure Key Vault, you can easily associate it to your web app through the Azure portal.

1. Select TLS/SSL settings under Settings.
2. Select the Private Key Certificate (.pfx) tab.
3. Select + Import Key Vault Certificate as shown in the following screenshot.
4. You can then select the vault, which must be in the same subscription, and the secret containing the certificate.
    a. The certificate must be an X.509 cert with a content type of application/x-pkcs12 and cannot have a password.

Links

- What is Azure Key Vault?
- Azure Key Vault pricing
- Certificate operations
- Implementing bring your own key (BYOK) for Azure Key Vault

Secure your Azure resources with role-based access control (RBAC)

Organizations that are considering using the public cloud are concerned about two things:

1. Ensuring that when people leave the organization, they lose access to resources.
2. Striking the right balance between autonomy and central governance - giving project teams the ability to create and manage virtual machines in the cloud while centrally controlling the networks those VMs use to communicate with other resources.

Azure Active Directory (Azure AD) and Role-Based Access Control (RBAC) work together to make it simple to carry out these goals.

Azure subscriptions

Each Azure subscription is associated with a single Azure AD directory. Users, groups, and applications in that directory can manage resources in the Azure subscription. The subscriptions use Azure AD for single sign-on (SSO) and access management. You can extend your on-premises Active Directory to the cloud by using **Azure AD Connect**. This feature allows your employees to manage their Azure subscriptions by using their existing work identities. When you disable an on-premises Active Directory account, it automatically loses access to all Azure subscriptions connected with Azure AD.

What is RBAC?

Role-based access control (RBAC) is an authorization system built on Azure Resource Manager that provides fine-grained access management of resources in Azure. With RBAC, you can grant the exact access that users need to do their jobs.

What can I do with RBAC?

RBAC allows you to grant access to Azure resources that you control. Suppose you need to manage access to resources in Azure for the development, engineering, and marketing teams. You've started to receive access requests, and you need to quickly learn how access management works for Azure resources.

Here are some scenarios you can implement with RBAC.

- Allow one user to manage virtual machines in a subscription and another user to manage virtual networks
- Allow 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

How does RBAC work?

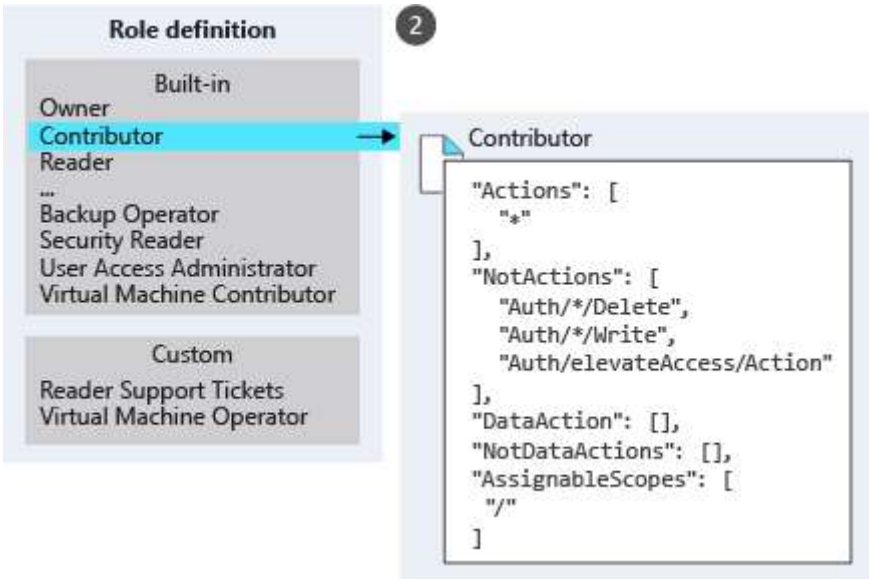
You control access to resources using RBAC by creating role assignments, which control how permissions are enforced. To create a role assignment, you need three elements: a security principal, a role definition, and a scope. You can think of these elements as "who", "what", and "where".

1. Security principal (who)

A *security principal* is just a fancy name for a user, group, or application that you want to grant access to.

2. Role definition (what you can do)

A *role definition* is a collection of permissions. It's sometimes just called a role. A role definition lists the permissions that can be performed, such as read, write, and delete. Roles can be high-level, like Owner, or specific, like Virtual Machine Contributor.



Azure includes several built-in roles that you can use. The following lists four fundamental built-in roles:

- Owner - Has full access to all resources, including the right to delegate access to others.
- Contributor - Can create and manage all types of Azure resources, but can't grant access to others.
- Reader - Can view existing Azure resources.
- User Access Administrator - Lets you manage user access to Azure resources.

If the built-in roles don't meet the specific needs of your organization, you can create your own custom roles.

3. Scope (where)

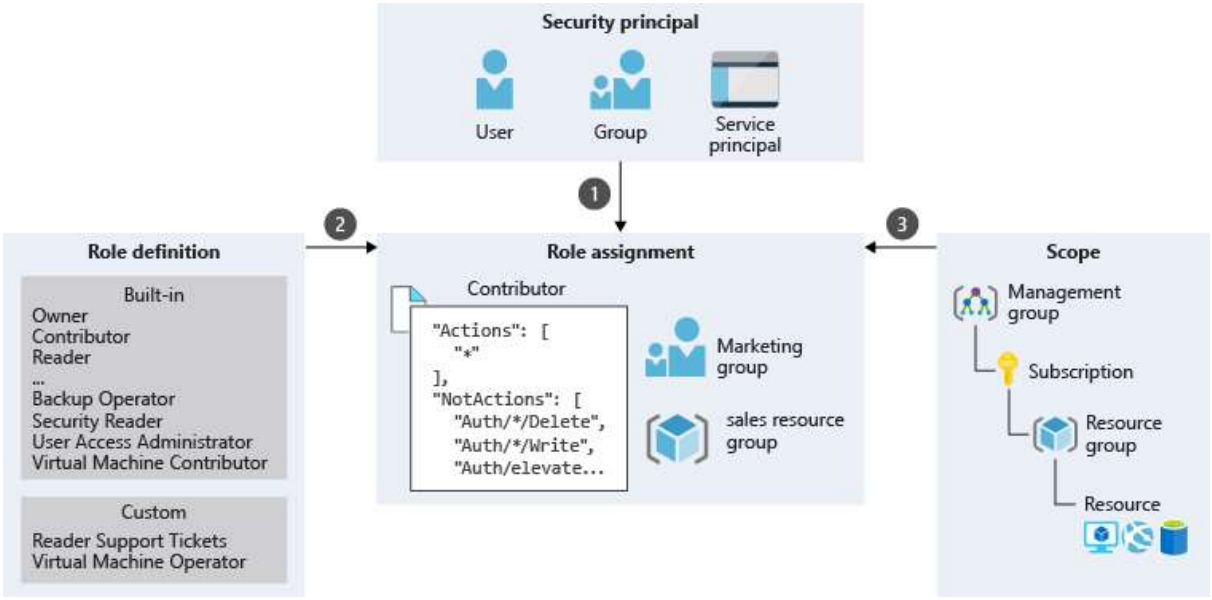
Scope is where the access applies to. This is helpful if you want to make someone a Website Contributor, but only for one resource group.

In Azure, you can specify a scope at multiple levels: management group, subscription, resource group, or resource. Scopes are structured in a parent-child relationship. When you grant access at a parent scope, those permissions are inherited by the child scopes. For example, if you assign the Contributor role to a group at the subscription scope, that role is inherited by all resource groups and resources in the subscription.

Role assignment

Once you have determined the who, what, and where, you can combine those elements to grant access. A *role assignment* is the process of binding a role to a security principal at a particular scope, for the purpose of granting access. To grant access, you create a role assignment. To revoke access, you remove a role assignment.

The following example shows how the Marketing group has been assigned the Contributor role at the sales resource group scope.



RBAC is an allow model

RBAC is an allow model. What this means is that when you are assigned a role, RBAC allows you to perform certain actions, such as read, write, or delete. So, if one role assignment grants you read permissions to a resource group and a different role assignment grants you write permissions to the same resource group, you will have read and write permissions on that resource group.

RBAC has something called NotActions permissions. Use NotActions to create a set of allowed permissions. The access granted by a role, the effective permissions, is computed by subtracting the NotActions operations from the Actions operations. For example, the Contributor role has both Actions and NotActions. The wildcard (*) in Actions indicates that it can perform all operations on the control plane. Then you subtract the following operations in NotActions to compute the effective permissions:

- Delete roles and role assignments
- Create roles and role assignments
- Grants the caller User Access Administrator access at the tenant scope
- Create or update any blueprint artifacts
- Delete any blueprint artifacts

Links

Azure built-in roles

Summary

1. True or false: A role definition in Azure is a collection of permissions?
True
A role definition in Azure is a collection of permissions with a name that you can assign to a user, group, or application.
2. Suppose you want to assign a role to allow a user to create and manage Azure resources but not be able to grant access to others. Which of the following built-in roles would support this?
Contributor
A contributor can create and manage all types of Azure resources, but they can't grant access to other users.
3. What is the inheritance order for scope in Azure?
Management group, Subscription, Resource group, Resource
The inheritance order for scope is Management group, Subscription, Resource group, *Resource. For example, if you assigned a Contributor role to a group at the Subscription scope level, it will be inherited by all Resource groups and Resources.*
1. True or false: To grant a user access to Azure resources, you create a role assignment?
True
A role assignment is the process of binding a role to a security principal at a particular scope for the purpose of granting access.
False
2. Suppose a developer needs full access to a resource group. If you are following least-privilege best practices, what scope should you specify?
Resource group
Following least-privilege best practices, you grant only the access the user needs to do their job. In this case, you should set the scope to the resource group.

Secure your Azure SQL Database

Firewall rules

Azure SQL Database has a built-in firewall that is used to allow and deny network access to both the database server itself, as well as individual databases. Initially, all public access to your Azure SQL Database is blocked by the SQL Database firewall. To access a database server, you must specify one or more server-level IP firewall rules that enable access to your Azure SQL Database. You use the IP firewall rules to specify which IP address ranges from the Internet are allowed, and whether Azure applications can attempt to connect to your Azure SQL Database.

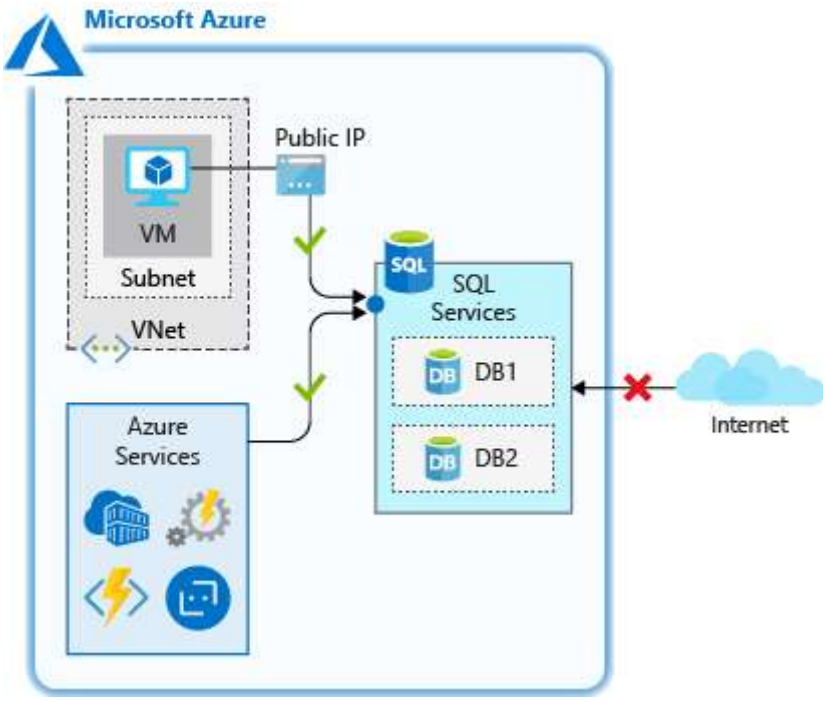
Firewall rules are configured at the server and/or database level, and will specifically state which network resources are allowed to establish a connection to the database. Depending on the level, the rules you can apply will be as follows:

- **Server-level firewall rules**
  - Allow access to Azure services
  - IP address rules
  - Virtual network rules
- **Database-level firewall rules**
  - IP address rules

Server-level firewall rules

These rules enable clients to access your entire Azure SQL server, that is, all the databases within the same logical server. There are three types of rules that can be applied at the server level.

The Allow access to Azure services rule allows services within Azure to connect to your Azure SQL Database. When enabled, this setting allows communications from all Azure public IP addresses. This includes all Azure Platform as a Service (PaaS) services, such as Azure App Service and Azure Container Service, as well as Azure VMs that have outbound Internet access. This rule can be configured through the ON/OFF option in the firewall pane in the portal, or by an IP rule that has 0.0.0.0 as the start and end IP addresses.

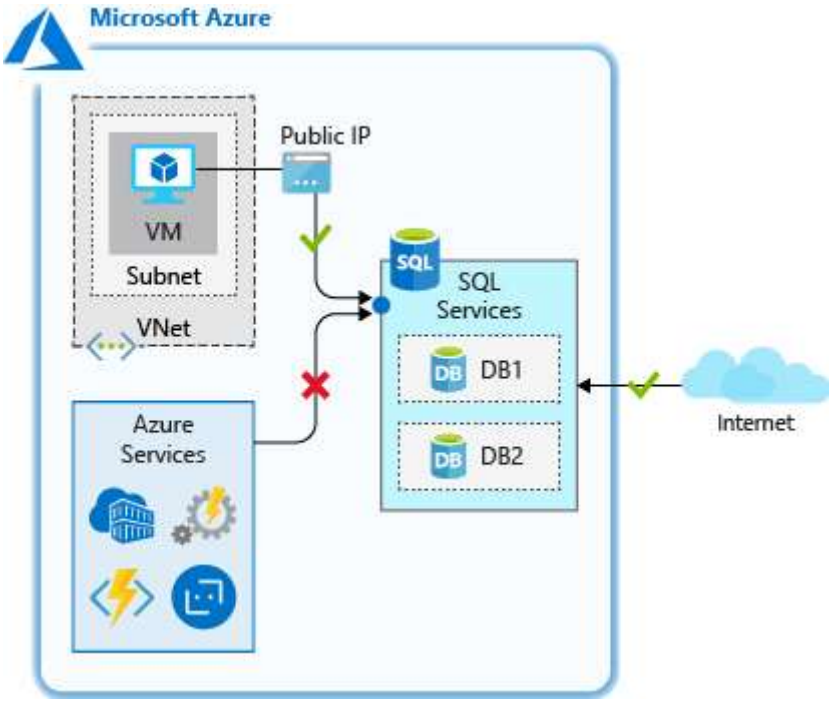


This rule is used when you have applications running on PaaS services in Azure, such as Azure Logic Apps or Azure Functions, that need to access your Azure SQL Database. Many of these services don't have a static IP address, so this rule is needed to ensure they are able to connect to the database.

Important

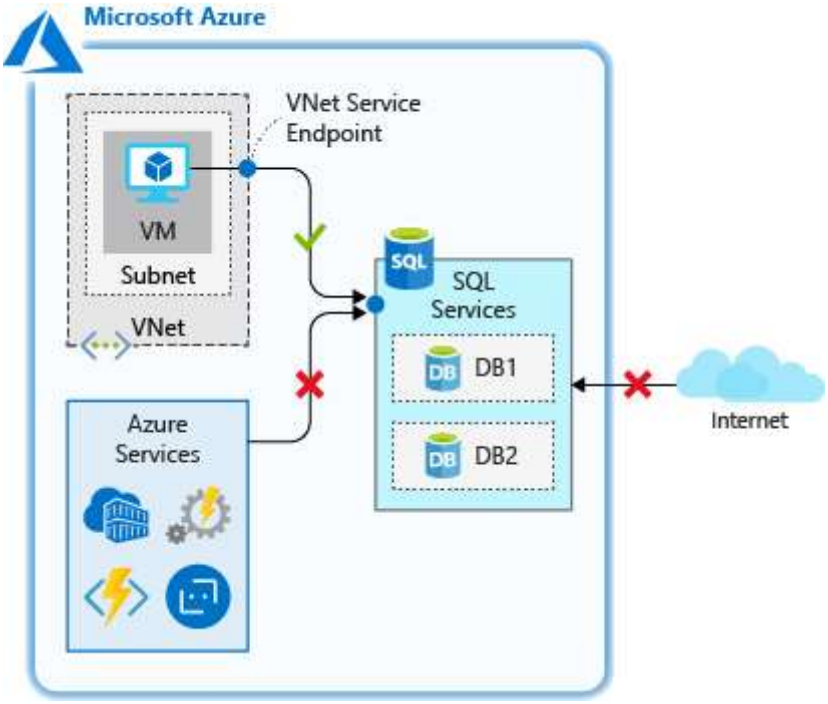
This option configures the firewall to allow all connections from Azure including connections from the subscriptions of other customers. When selecting this option, make sure your login and user permissions limit access to only authorized users.

IP address rules are rules that are based on specific public IP address ranges. IP addresses connecting from an allowed public IP range will be permitted to connect to the database.



These rules can be used when you have a static public IP address that needs to access your database.

Virtual network rules allow you to explicitly allow connection from specified subnets inside one or more Azure virtual networks (VNets). Virtual network rules can provide greater access control to your databases and can be a preferred option depending on your scenario. Since Azure VNet address spaces are private, you can effectively eliminate exposure to public IP addresses and secure connectivity to those addresses you control.

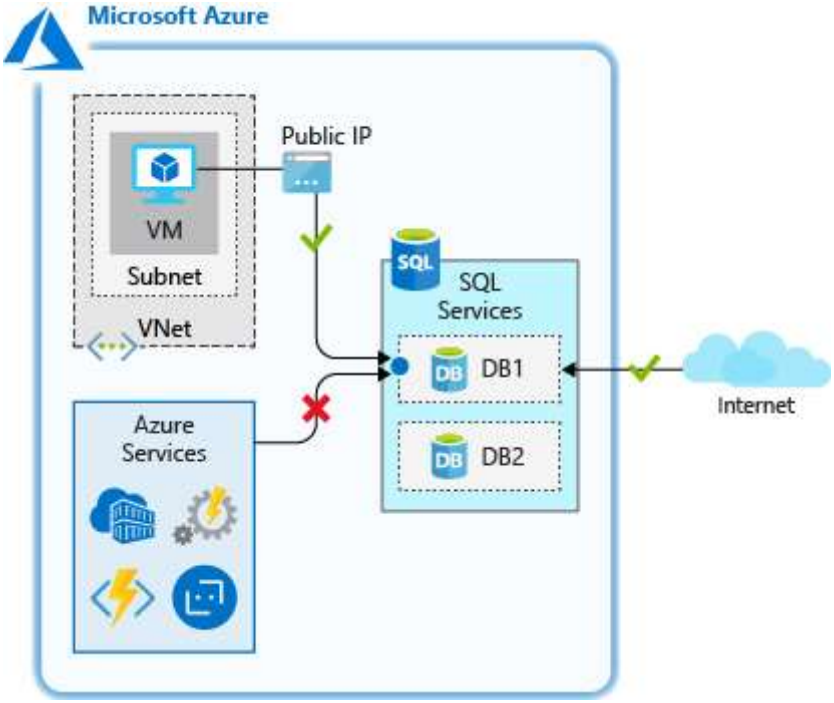


Virtual network rules are used when you have Azure VMs that need to access your database.

For server-level rules, all of the above can be created and manipulated through the portal, PowerShell, the CLI and through Transact-SQL (T-SQL).

Database-level firewall rules

These rules allow access to an individual database on a logical server and are stored in the database itself. For database-level rules, only IP address rules can be configured. They function the same as when applied at the server-level, but are scoped to the database only.



The benefits of database-level rules are their portability. When replicating a database to another server, the database-level rules will be replicated, since they are stored in the database itself.

The downside to database-level rules is that you can only use IP address rules. This may limit the flexibility you have and can increase administrative overhead.

Lastly, database-level firewall rules can be created and manipulated only through T-SQL.

SQL authentication

SQL authentication method uses a username and password. User accounts can be created in the master database and can be granted permissions in all databases on the server, or they can be created in the database itself (called contained users) and given access to only that database. When you created the logical server for your database, you specified a

"server admin" login with a username and password. Using these credentials, you can authenticate to any database on that server as the database owner, or "dbo".

Azure Active Directory authentication

This authentication method uses identities managed by Azure Active Directory (AD) and is supported for managed and integrated domains. Use Azure AD authentication (integrated security) whenever possible. With Azure AD authentication, you can centrally manage the identities of database users and other Microsoft services in one central location. Central ID management provides a single place to manage database users and simplifies permission management. If you want to use Azure AD authentication, you must create another server admin called the "Azure AD admin," which is allowed to administer Azure AD users and groups. This admin can also perform all operations that a regular server admin can.

Authorization

Authorization refers to what an identity can do within an Azure SQL Database. This is controlled by permissions granted directly to the user account and/or database role memberships. A database role is used to group permissions together to ease administration, and a user is added to a role to be granted the permissions the role has. These permissions can grant things such as the ability to log in to the database, the ability to read a table, and the ability to add and remove columns from a database. As a best practice, you should grant users the least privileges necessary. The process of granting authorization to both SQL and Azure AD users is the same.

TLS network encryption

Azure SQL Database enforces Transport Layer Security (TLS) encryption at all times for all connections, which ensures all data is encrypted "in transit" between the database and the client. By using TLS encryption, you can ensure that anyone who may have intercepted the traffic between the app server and database would not be able to read the data. TLS encryption is a standard of securing traffic over the internet, and in this case ensures your network traffic to and from your Azure SQL database is secure by default.

Transparent data encryption

Azure SQL Database protects your data at rest using transparent data encryption (TDE). TDE performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. Using a database encryption key, transparent data encryption performs real-time I/O encryption and decryption of the data at the page level. Each page is decrypted when it's read into memory and then encrypted before being written to disk.

By default, TDE is enabled for all newly deployed Azure SQL databases. It's important to check that data encryption hasn’t been turned off, and older Azure SQL Server databases may not have TDE enabled.

Dynamic data masking

You might have noticed when we ran our query in the previous unit that some of the information in the database is sensitive; there are phone numbers, email addresses, and other information that we may not want to fully display to everyone with access to the data.

Maybe we don't want our users to be able to see the full phone number or email address, but we'd still like to make a portion of the data available for customer service representatives to identify a customer. By using the dynamic data masking feature of Azure SQL Database, we can limit the data that is displayed to the user. Dynamic data masking is a policy-based security feature that hides the sensitive data in the result set of a query over designated database fields, while the data in the database is not changed.

Data masking rules consist of the column to apply the mask to, and how the data should be masked. You can create your own masking format, or use one of the standard masks such as:

- Default value, which displays the default value for that data type instead.
- Credit card value, which only shows the last four digits of the number, converting all other numbers to lower case x's.
- Email, which hides the domain name and all but the first character of the email account name.
- Number, which specifies a random number between a range of values. For example, on the credit card expiry month and year, you could select random months from 1 to 12 and set the year range from 2018 to 3000.
- Custom string, which allows you to set the number of characters exposed from the start of the data, the number of characters exposed from the end of the data, and the characters to repeat for the remainder of the data.

When querying the columns, database administrators will still see the original values, but non-administrators will see the masked values. You can allow other users to see the non-masked versions by adding them to the SQL users excluded from masking list.

Azure SQL Database auditing

By enabling auditing, operations that occur on the database are stored for later inspection or to have automated tools analyze them. Auditing is also used for compliance management or understanding how your database is used. Auditing is also required if you wish to use Azure threat detection on your Azure SQL database.

You can use SQL database auditing to:

- Retain an audit trail of selected events. You can define categories of database actions to be audited.

- Report on database activity. You can use pre-configured reports and a dashboard to get started quickly with activity and event reporting.
- Analyze reports. You can find suspicious events, unusual activity, and trends.

Audit logs are written to Append Blobs in an Azure Blob storage account that you designate. Audit policies can be applied at the server-level or database-level. Once enabled, you can use the Azure portal to view the logs, or send them to Log Analytics or Event Hub for further processing and analysis.

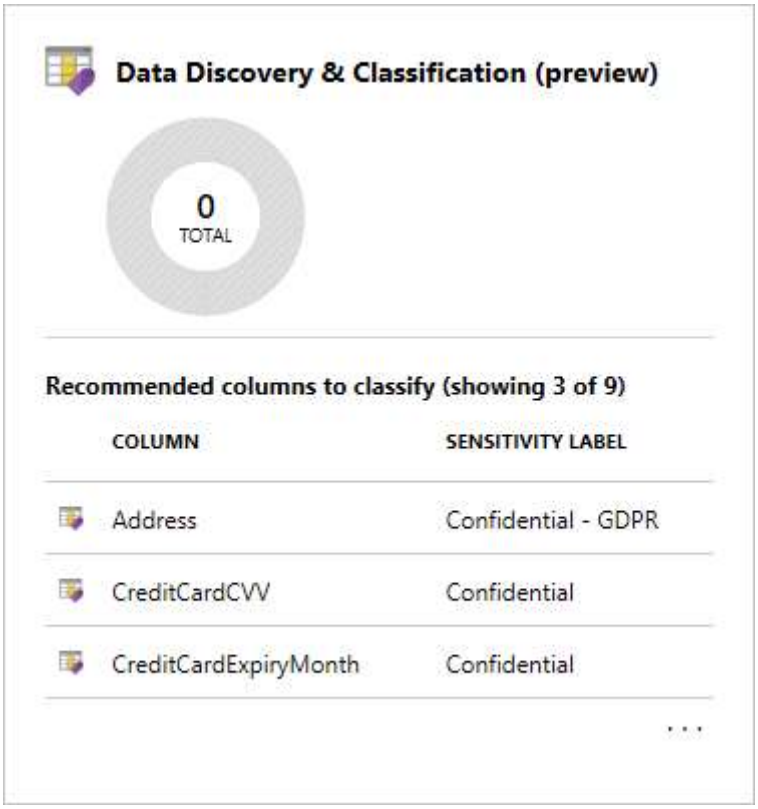
Advanced Data Security for Azure SQL Database

Advanced Data Security (ADS) provides a set of advanced SQL security capabilities, including data discovery & classification, vulnerability assessment, and Advanced Threat Protection.

- **Data discovery & classification** (currently in preview) provides capabilities built into Azure SQL Database for discovering, classifying, labeling & protecting the sensitive data in your databases. It can be used to provide visibility into your database classification state, and to track the access to sensitive data within the database and beyond its borders.
- **Vulnerability assessment** is an easy to configure service that can discover, track, and help you remediate potential database vulnerabilities. It provides visibility into your security state, and includes actionable steps to resolve security issues, and enhance your database fortifications.
- **Advanced Threat Protection** detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit your database. It continuously monitors your database for suspicious activities, and provides immediate security alerts on potential vulnerabilities, SQL injection attacks, and anomalous database access patterns. Advanced Threat Protection alerts provide details of the suspicious activity and recommend action on how to investigate and mitigate the threat.

Data Discovery & Classification

The Data Discovery & Classification panel shows columns within your tables that need to be protected. Some of the columns may have sensitive information or may be considered classified in different countries or regions.



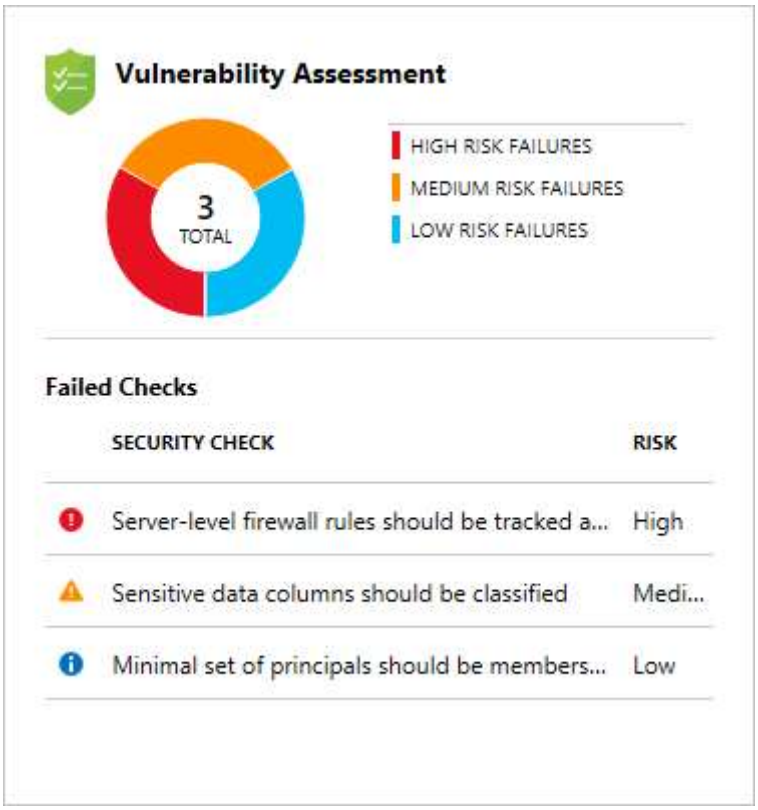
A message will be displayed if any columns need protection configured. This message will be formatted like *"We have found 10 columns with classification recommendations"*. You can click on the text to view the recommendations.

Select the columns that you want to classify by clicking the checkmark next to the column, or select the checkbox to the left of the schema header. Select the Accept selected recommendations options to apply the classification recommendations.

Next, you'll edit the columns and then define the information type and the sensitivity label for the database. Click on the Save button to save the changes.

No active recommendations should be listed once you've managed the recommendations successfully.

Vulnerability Assessment

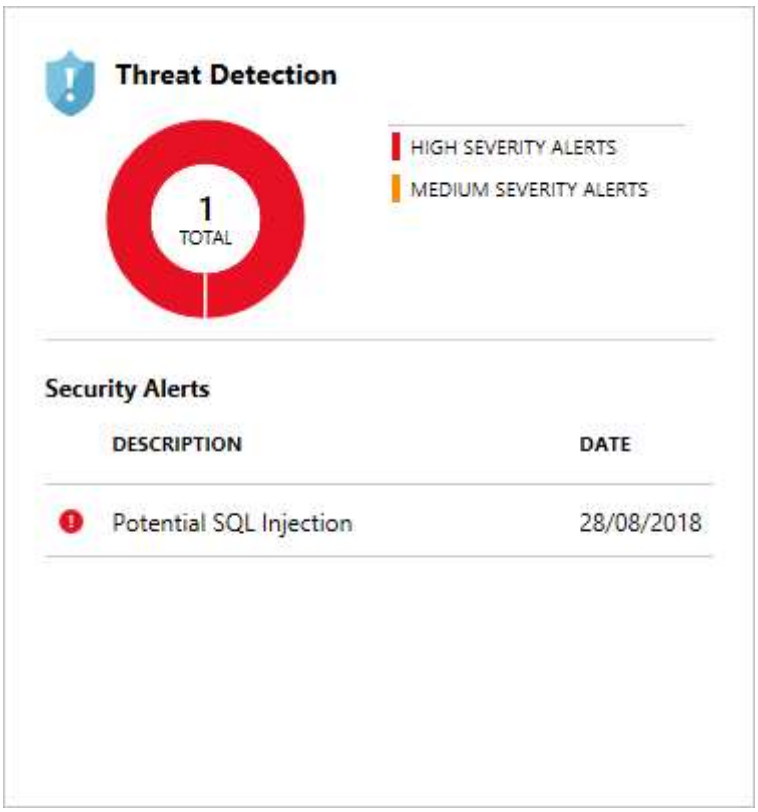


The Vulnerability Assessment lists configuration issues on your database and the associated risk. For example, in the image above, you can see the server-level firewall needs to be set up.

Click on the Vulnerability Assessment panel to review a full list of vulnerabilities. From here, you can click on each individual vulnerability.

On the vulnerability page you will see the details such as the risk level, which database it applies to, a description of the vulnerability, and the recommended remediation to fix the issue. Apply the remediation to fix the issue or issues. Make sure to address all the vulnerabilities.

Threat Detection



Address any issues by following the recommendations. For issues such as the SQL injection warnings, you'll be able to look at the query and work backward to where the query is being executed in code. Once found, you should rewrite the code so it will no longer have the issue.

Links

- Manage IP firewall rules for Azure SQL Server
- Manage Advanced Data Security settings for a SQL database

Summary

1. Which of the following is the most efficient way to secure a database to allow only access from a VNet while restricting access from the internet?
A server-level virtual network rule
A server-level virtual network rule will allow you to allow connectivity from specific Azure VNet subnets, and will block access from the internet. This is the most efficient manner to secure this configuration.

2. A mask has been applied to a column in the database that holds a user's email address, laura@contoso.com. From the list of options, what would the mask display for a database administrator account?
laura@contoso.com

When database administrator accounts access data that have a mask applied, the mask is removed, and the original data is visible.

3. Transparent Data Encryption will encrypt which database files?
Database files, log files, and backup files
Transparent Data Encryption encrypts all database, log, and backup files. When new Azure SQL databases are created, Transparent Data Encryption will be enabled by default.

4. Is encrypted communication turned on automatically when connecting to an Azure SQL Server?
Yes
Azure SQL Database enforces encryption (SSL/TLS) at all times for all connections