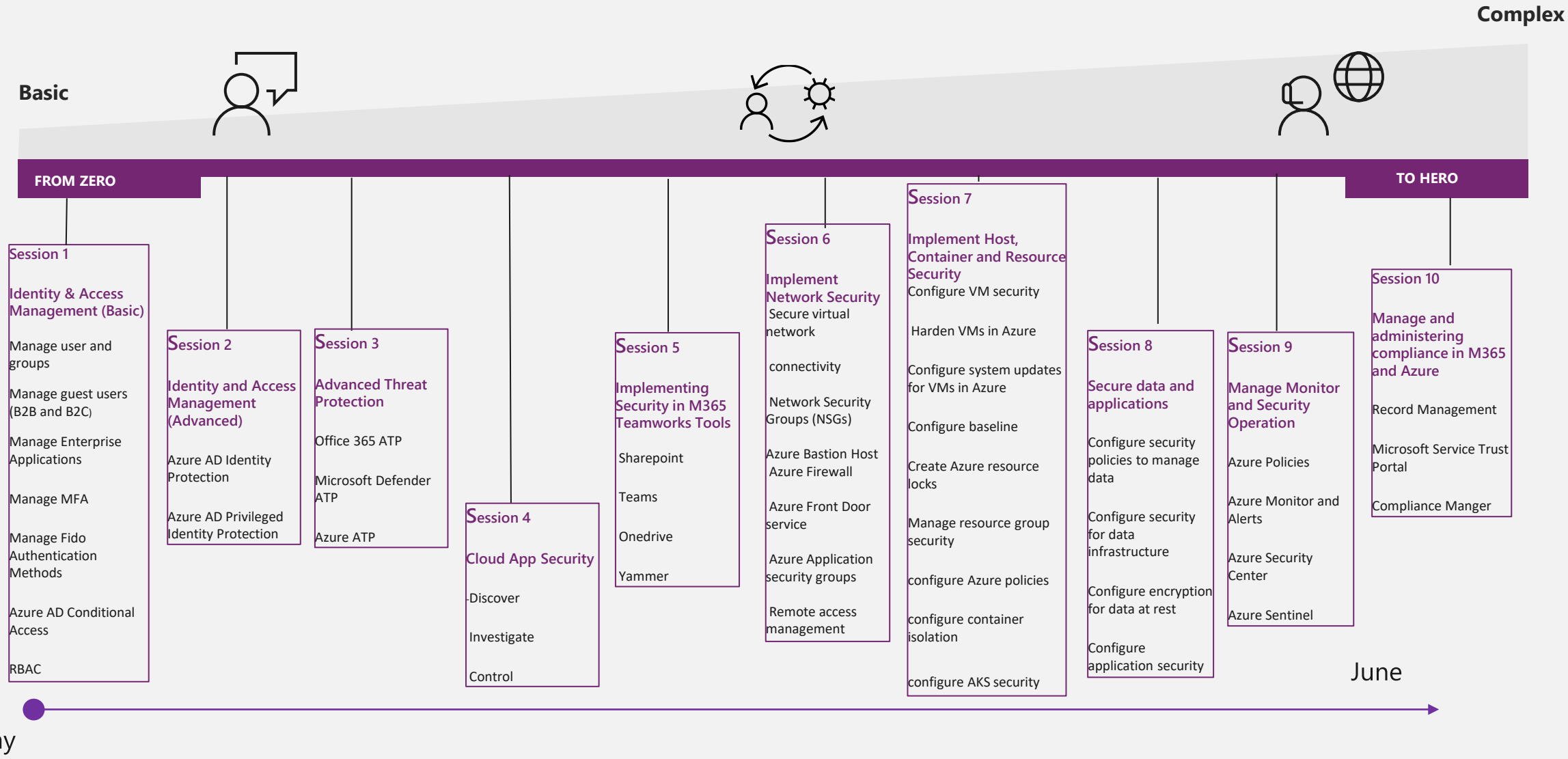
Security Sailing

Session 8 – Secure data and applications

Michele

- Senior Consultant – Speaker – Trainer (22 anni)
- Dipendente 50% su tecnologie Microsoft Dipartimento di Informatica – Università degli Studi di Milano
- Freelance 50/70%
- Mi occupo di: AD, SCCM, W10, Win Server, AzureAD, O365, M365, Azure, Enterprise Mobility & Security
- Speaker da 12 anni di WPC e da 5 responsabile agenda ITPRO e Security
- Certificato MCT, MCSE, MCSA, MCITP
- Contatti:
  - michele@sensalari.com
  - michele.sensalari@overneteducation.it
  - Twitter: @ilsensa7
  - Linkedin: https://www.linkedin.com/in/michele-sensalari-4988b7/

Content and Timeline Details



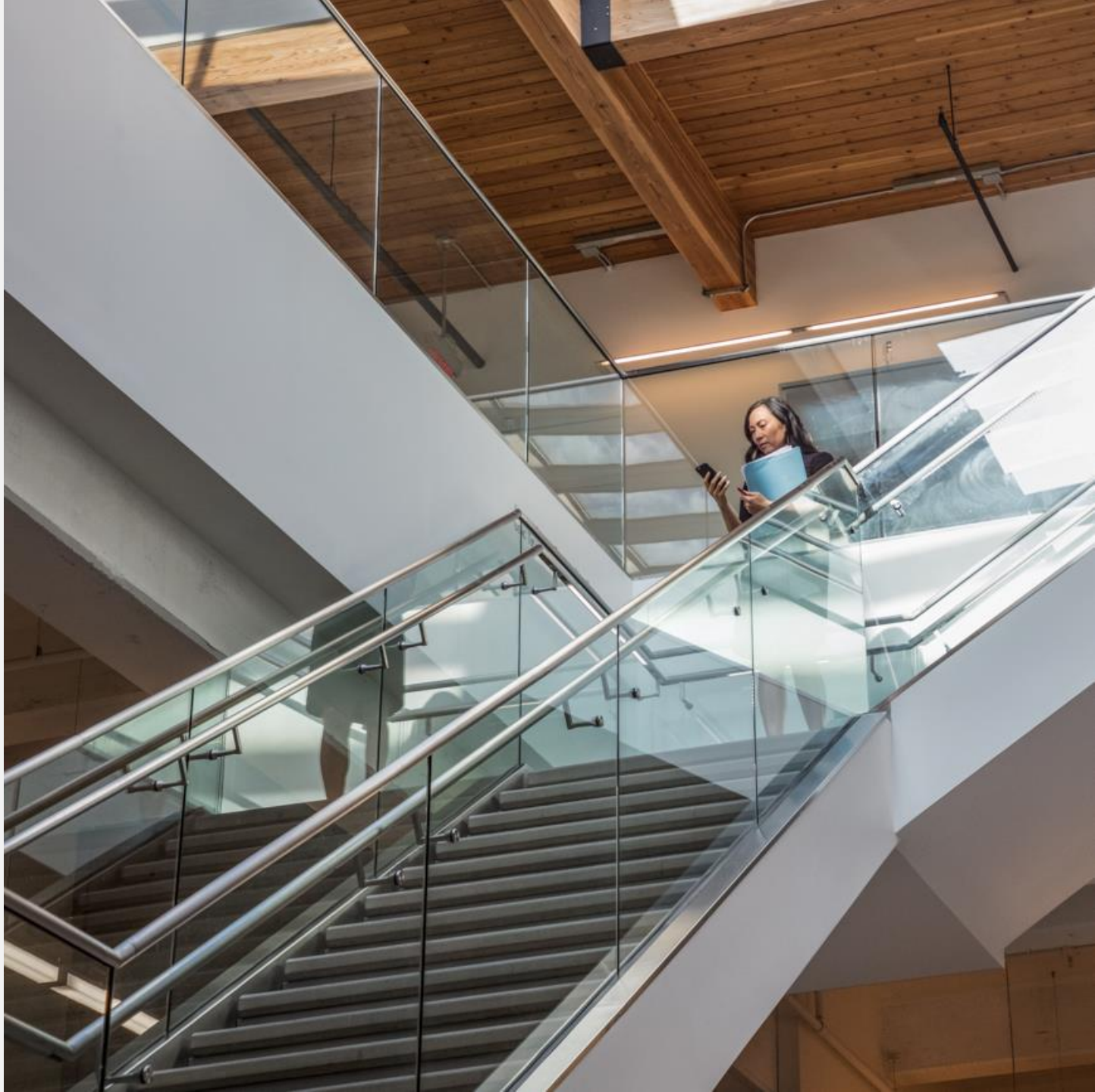
Microsoft Security

Agenda

Storage account security

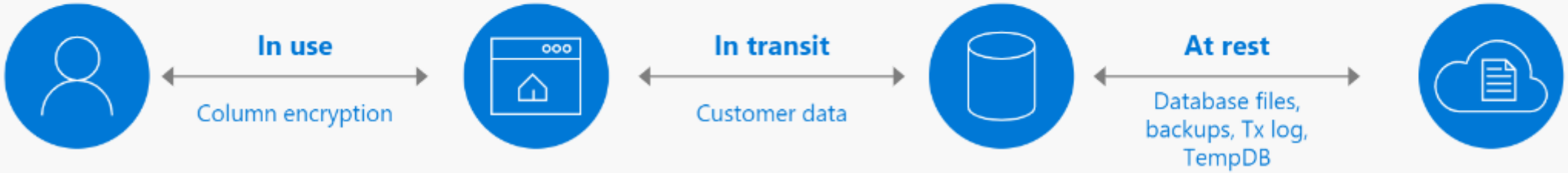
Azure SQL Security

Big Data Security



Types of data Encryption

Data encryption	Encryption technology	Customer value
In transit	Transport Layer Security (TLS) from the client to the server	Protects data between client and server against snooping and man-in-the-middle attacks *Azure SQL Database is phasing out Secure Sockets Layer (SSL) 3.0 and TLS 1.0 in favor of TLS 1.2
At rest	Transparent Data Encryption (TDE) for Azure SQL Database	Protects data on the disk Key management is done by Azure, which makes it easier to obtain compliance
In use (end-to-end)	Always Encrypted for client-side column encryption	Data is protected end-to-end, but the application is aware of encrypted columns This is used in the absence of data masking and TDE for compliance-related scenarios



Microsoft Security

Storage Account Security

Storage account overview

- Azure Storage Data objects: blobs, files, queues, tables and disks
- Unique namespace for Azure Storage Data
- It’s accessible from anywhere with HTTP or HTTPS
- Different Storage Account types:
  - **General-purpose v2 accounts**: Basic storage account type for blobs, files, queues, and tables. Recommended for most scenarios using Azure Storage.
  - **General-purpose v1 accounts**: Legacy account type for blobs, files, queues, and tables. Use general-purpose v2 accounts instead when possible.
  - **BlockBlobStorage accounts**: Storage accounts with premium performance characteristics for block blobs and append blobs. Recommended for scenarios with high transactions rates, or scenarios that use smaller objects or require consistently low storage latency.
  - **FileStorage accounts**: Files-only storage accounts with premium performance characteristics. Recommended for enterprise or high performance scale applications.
  - **BlobStorage accounts**: Legacy Blob-only storage accounts. Use general-purpose v2 accounts instead when possible.

Types of Storage Accounts and their capabilities

Storage account type	Supported services	Supported performance tiers	Supported access tiers	Replication options	Deployment model[1]	Encryption[2]
General-purpose V2	Blob, File, Queue, Table, Disk, and Data Lake Gen2[6]	Standard, Premium[5]	Hot, Cool, Archive[3]	LRS, GRS, RA-GRS, ZRS, GZRS (preview), RA-GZRS (preview)[4]	Resource Manager	Encrypted
General-purpose V1	Blob, File, Queue, Table, and Disk	Standard, Premium[5]	N/A	LRS, GRS, RA-GRS	Resource Manager, Classic	Encrypted
BlockBlobStorage	Blob (block blobs and append blobs only)	Premium	N/A	LRS, ZRS[4]	Resource Manager	Encrypted
FileStorage	File only	Premium	N/A	LRS, ZRS[4]	Resource Manager	Encrypted
BlobStorage	Blob (block blobs and append blobs only)	Standard	Hot, Cool, Archive[3]	LRS, GRS, RA-GRS	Resource Manager	Encrypted

Storage Account Access Control

- **Azure Active Directory:** Use Azure Active Directory (Azure AD) credentials to authenticate a user, group, or other identity for access to blob and queue data. If authentication of an identity is successful, then Azure AD returns a token to use in authorizing the request to Azure Blob storage or Queue storage.
- **Shared Key authorization:** Use your storage account access key to construct a connection string that your application uses at runtime to access Azure Storage. The values in the connection string are used to construct the *Authorization* header that is passed to Azure Storage.
- **Shared access signature:** Use a shared access signature to delegate access to resources in your storage account, if you aren't using Azure AD authorization. A shared access signature is a token that encapsulates all of the information needed to authorize a request to Azure Storage on the URL. You can specify the storage resource, the permissions granted, and the interval over which the permissions are valid as part of the shared access signature.

Authorizing access to data in Azure Storage

	SHARED KEY (STORAGE ACCOUNT KEY)	SHARED ACCESS SIGNATURE (SAS)	AZURE ACTIVE DIRECTORY (AZURE AD)	ON-PREMISES ACTIVE DIRECTORY DOMAIN SERVICES (PREVIEW)	ANONYMOUS PUBLIC READ ACCESS
Azure Blobs	Supported	Supported	Supported	Not supported	Supported
Azure Files (SMB)	Supported	Not supported	Supported, only with AAD Domain Services	Supported, credentials must be synced to Azure AD	Not supported
Azure Files (REST)	Supported	Supported	Not supported	Not supported	Not supported
Azure Queues	Supported	Supported	Supported	Not Supported	Not supported
Azure Tables	Supported	Supported	Not supported	Not supported	Not supported

Azure Files overview

Azure Files can be deployed in two main ways:

- by directly mounting the serverless Azure file shares
- by caching Azure file shares on-premises using Azure File Sync.

There are two main types of storage accounts you will use for Azure Files deployments:

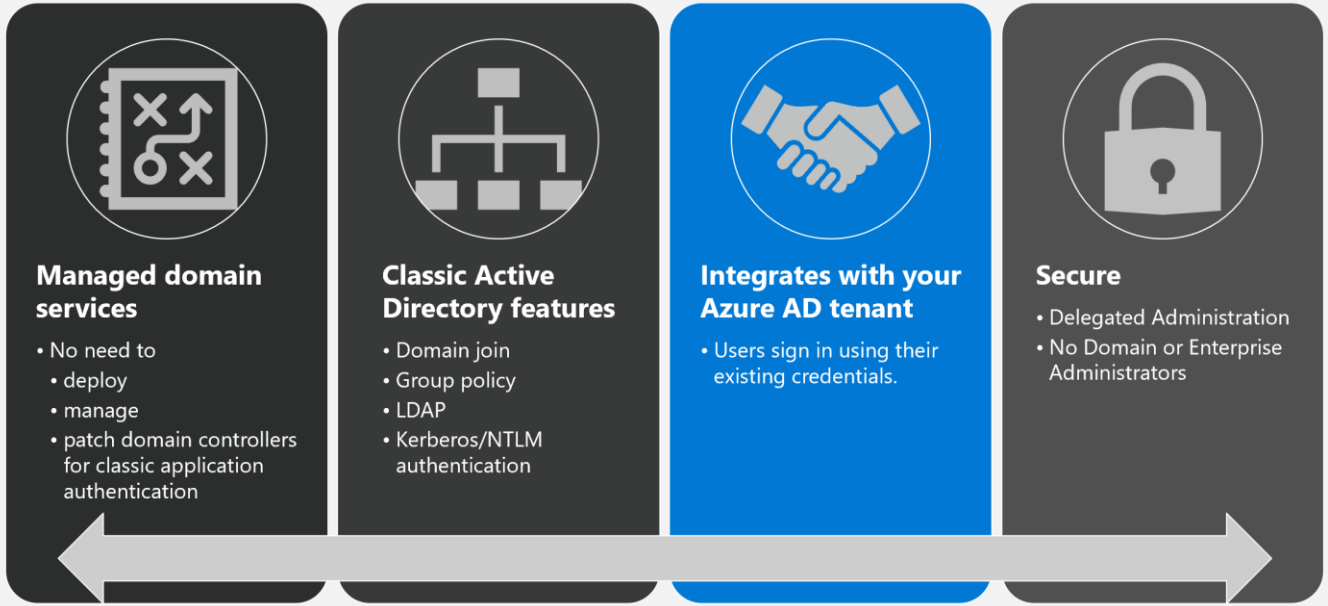
- General purpose version 2 (GPv2) storage accounts
- FileStorage storage accounts

Identity Management:

- On-premises Active Directory Domain Services
- Azure Active Directory Domain Services (Azure AD DS)
- Azure Storage account key

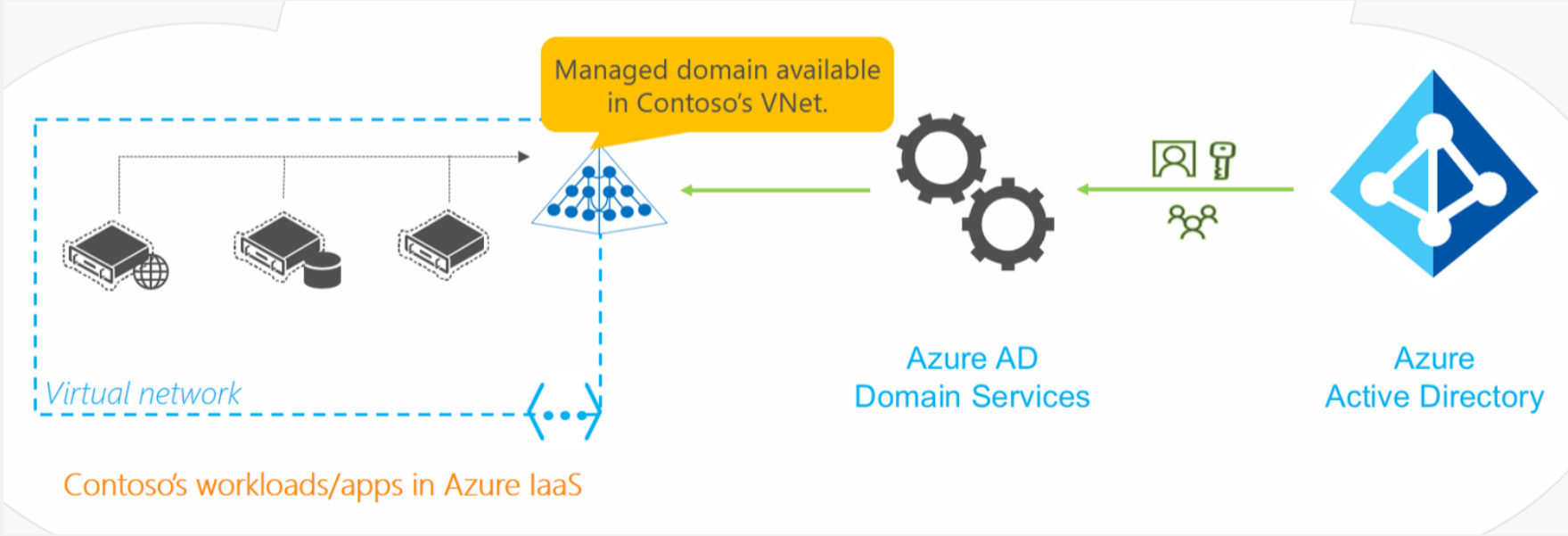
Azure Active Directory (AD) Domain Services

- Azure AD Domain Services provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that are fully compatible with Windows Server Active Directory.
- You can consume these domain services without the need for you to deploy, manage, and patch domain controllers in the cloud.
- Azure AD Domain Services integrates with your existing Azure AD tenant, thus making it possible for users to log in using their corporate credentials. Additionally, you can use existing groups and user accounts to secure access to resources, thus ensuring a smoother 'lift-and-shift' of on-premises resources to Azure Infrastructure Services.
- Azure AD Domain Services functionality works seamlessly regardless of whether your Azure AD tenant is cloud-only or synced with your on-premises Active Directory



Azure Domain Services for Cloud Only Organizations

A cloud-only Azure AD tenant (often referred to as 'managed tenants') does not have any on-premises identity footprint. In other words, user accounts, their passwords, and group memberships are all native to the cloud - that is, created and managed in Azure AD

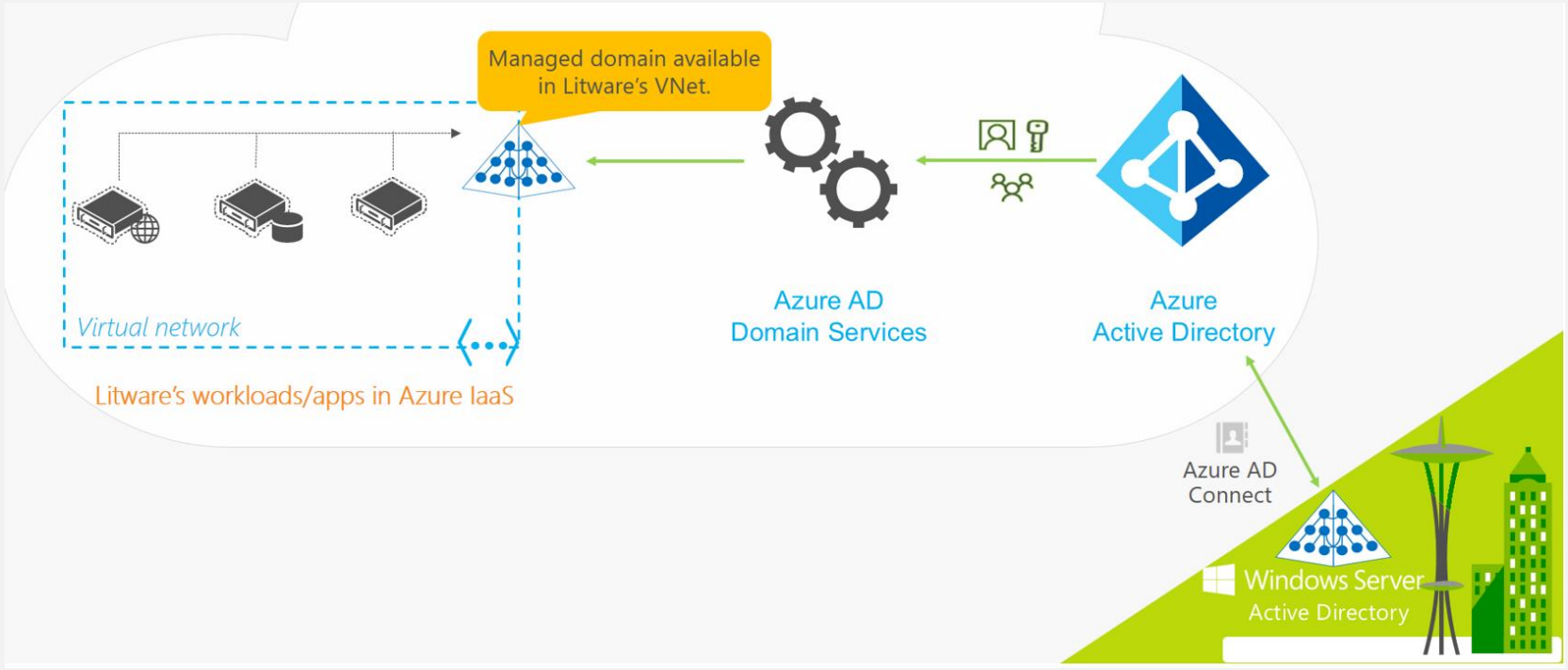


Azure Domain Services for Cloud Only Organizations

- Company’s administrator does not need to manage, patch, or monitor this domain or any domain controllers for this managed domain.
- There is no need to manage AD replication for this domain. User accounts, group memberships, and credentials from Contoso's Azure AD tenant are automatically available within this managed domain.
- Since the domain is managed by Azure AD Domain Services, Contoso's IT administrator does not have Domain Administrator or Enterprise Administrator privileges on this domain

Azure Domain Services for Hybrid Organizations

Organizations with a hybrid IT infrastructure consume a mix of cloud resources and on-premises resources. Such organizations synchronize identity information from their on-premises directory to their Azure AD tenant. As hybrid organizations look to migrate more of their on-premises applications to the cloud, especially legacy directory-aware applications, Azure AD Domain Services can be useful to them.



Azure Domain Services – Benefits

- Simple – You can satisfy the identity needs of virtual machines deployed to Azure Infrastructure services with a few simple clicks. You do not need to deploy and manage identity infrastructure in Azure or setup connectivity back to your on-premises identity infrastructure.

- Integrated – Azure AD Domain Services is deeply integrated with your Azure AD tenant. You can now use Azure AD as an integrated cloud-based enterprise directory that caters to the needs of both your modern applications and traditional directory-aware applications.

- Compatible – Azure AD Domain Services is built on the proven enterprise grade infrastructure of Windows Server Active Directory. Therefore, your applications can rely on a greater degree of compatibility with Windows Server Active Directory features. Not all features available in Windows Server AD are currently available in Azure AD Domain Services. However, available features are compatible with the corresponding Windows Server AD features you rely on in your on-premises infrastructure. The LDAP, Kerberos, NTLM, Group Policy, and domain join capabilities constitute a mature offering that has been tested and refined over various Windows Server releases.

- Cost-effective – With Azure AD Domain Services, you can avoid the infrastructure and management burden that is associated with managing identity infrastructure to support traditional directory-aware applications. You can move these applications to Azure Infrastructure Services and benefit from greater savings on operational expenses.

Azure Active Directory Domain Services Authentication on Azure Files

Azure Files enforces authorization on user access to both the share and the directory/file levels. Share-level permission assignment can be performed on Azure Active Directory (Azure AD) users or groups managed through the role-based access control (RBAC) model. With RBAC, the credentials you use for file access should be available or synced to Azure AD. You can assign built-in RBAC roles like:

- Storage File Data SMB Share Reader – This allows read access in Azure Storage file shares over SMB.
- Storage File Data SMB Share Contributor – This allows read, write, and delete access in Azure Storage file shares over SMB.
- Storage File Data SMB Share Elevated Contributor – allows read, write, delete and modify NTFS permissions in Azure Storage file shares over SMB.

At the directory/file level, Azure Files supports preserving, inheriting, and enforcing Windows DACLs just like any Windows file servers. You can choose to keep Windows DACLs when copying data over SMB between your existing file share and your Azure file shares. Whether you plan to enforce authorization or not, you can use Azure file shares to back up ACLs along with your data

.

Active Directory Domain Services Authentication on Azure Files

To enable AD DS authentication over SMB for Azure file shares, you need to register your storage account with AD DS and then set the required domain properties on the storage account.

To register your storage account with AD DS, create an account representing it in your AD DS. You can think of this process as if it were like creating an account representing an on-premises Windows file server in your AD DS. When the feature is enabled on the storage account, it applies to all new and existing file shares in the account.

The Join-AzStorageAccountForAuth cmdlet performs the equivalent of an offline domain join on behalf of the specified storage account. The script uses the cmdlet to create a computer account in your AD domain

Share-level permissions and directory and file level permissions same as Azure AD Domain Service

Security reccomandation for Blob Storage

- Data Protection
  - Use Azure Resource Manager
  - Enable the Secure transfer required options
  - Enable advanced threat protection
  - Turn on soft delete for blob data
  - *Store business-critical data in immutable blobs*
  - Limit SAS tokens to HTTPs connection only
- Identity and access management
  - *Use Azure Active Directory (Azure AD) to authorize access to blob data*
  - RBAC
  - *Secure your account access keys with Azure Key Vault*
  - Keep in mind the principal of leaast privilege when assign permissions to SAS
  - Limiti anonymous public read access to containers and blob
- Logging and Monitoring

Authorize access to blobs and queues using Azure Active Directory

- With Azure AD, you can use role-based access control (RBAC) to grant permissions to a security principal, which may be a user, group, or application service principal
- Authorizing requests against Azure Storage with Azure AD provides superior security and ease of use over Shared Key authorization
- Only storage accounts created with the Azure Resource Manager deployment model support Azure AD authorization.
- Blob storage additionally supports creating shared access signatures (SAS) that are signed with Azure AD credentials
- When a security principal (a user, group, or application) attempts to access a blob or queue resource, therequest must be authorized, unless it is a blob available for anonymous access. With Azure AD, access to a resource is a two-step process. First, the security principal's identity is authenticated and an OAuth 2.0 token is returned. Next, the token is passed as part of a request to the Blob or Queue service and used by the service to authorize access to the specified resource
- The authorization step requires that one or more RBAC roles be assigned to the security principal. Azure Storage provides RBAC roles that encompass common sets of permissions for blob and queue data. The roles that are assigned to a security principal determine the permissions that the principal will have.

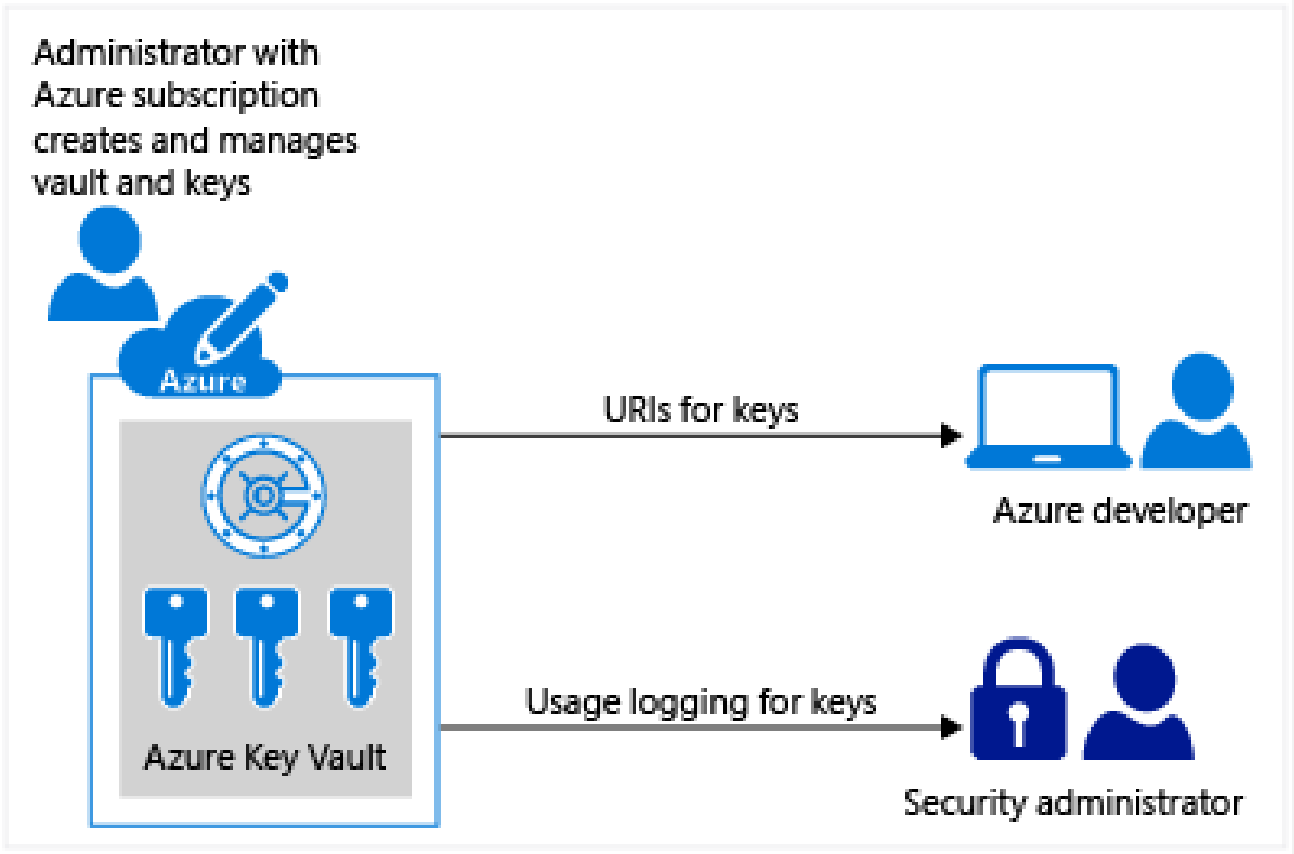
Key management for Storage Account: Azure Key Vault

Safeguard cryptographic keys and secrets used by cloud applications and services
Simplify and automate tasks for SSL/TLS certificate

Keys are stored in a vault and invoked by URI when needed

Key Vaults performs cryptographic operations on behalf of the application

The application does not see the customers keys
Customers can import their own keys into Azure and manage them



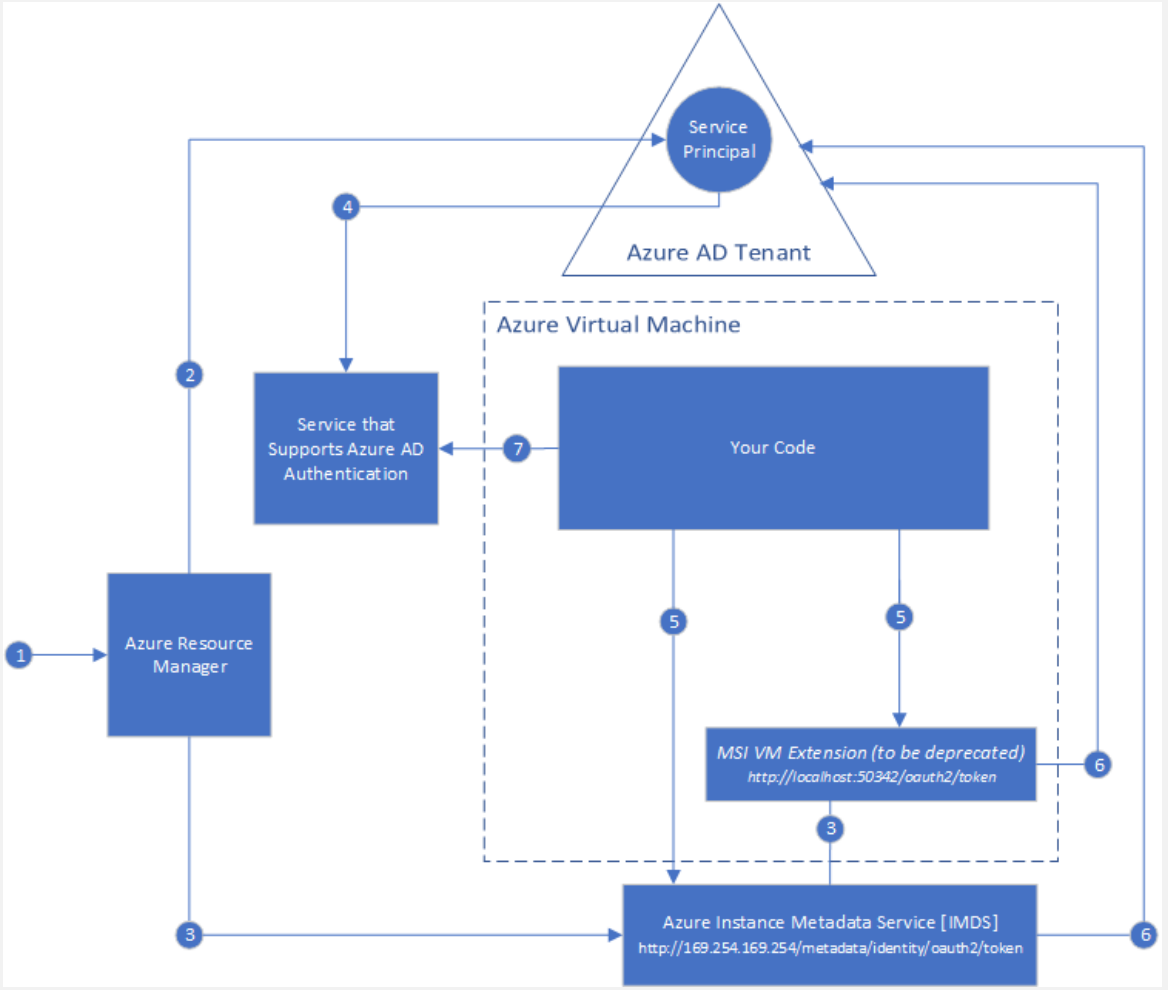
Managed Identity for Azure Resources

Azure Key Vault provides a way to securely store credentials, secrets, and other keys, but your code has to authenticate to Key Vault to retrieve them.

The managed identities for Azure resources feature in Azure Active Directory (Azure AD) solves this problem. The feature provides Azure services with an automatically managed identity in Azure AD. You can use the identity to authenticate to any service that supports Azure AD authentication, including Key Vault, without any credentials in your code.

There are two types of managed identity:

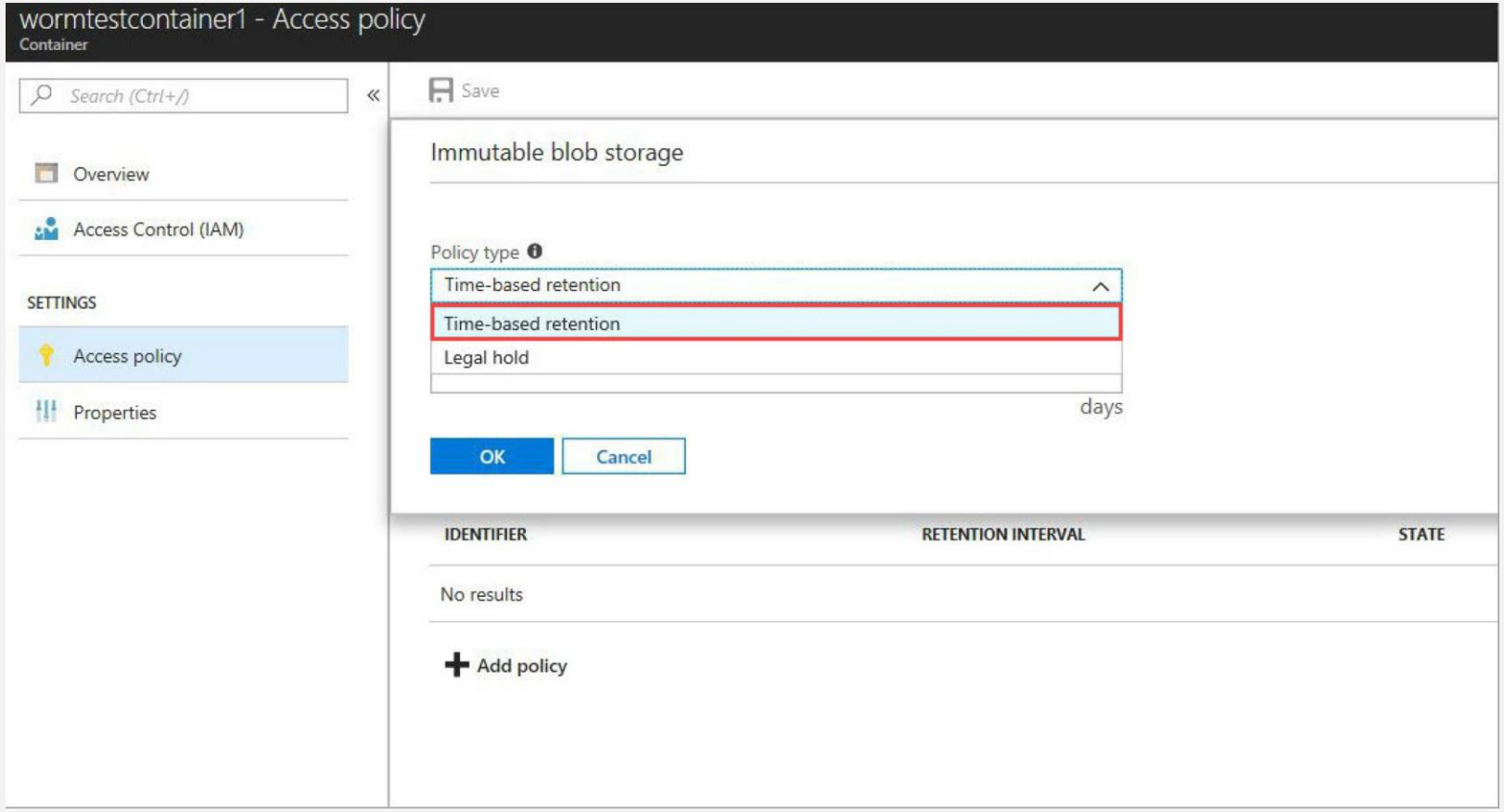
- A **system-assigned managed identity** is enabled directly on an Azure service instance. When the identity is enabled, Azure creates an identity for the instance in the Azure AD tenant that's trusted by the subscription of the instance. After the identity is created, the credentials are provisioned onto the instance. The life cycle of a system-assigned identity is directly tied to the Azure service instance that it's enabled on. If the instance is deleted, Azure automatically cleans up the credentials and the identity in Azure AD.
- A **user-assigned managed identity** is created as a standalone Azure resource. Through a create process, Azure creates an identity in the Azure AD tenant that's trusted by the subscription in use. After the identity is created, the identity can be assigned to one or more Azure service instances. The life cycle of a user-assigned identity is managed separately from the life cycle of the Azure service instances to which it's assigned.



Manage Storage Access Keys with Key Vault

- When you create a storage account, Azure generates two 512-bit storage account access keys. These keys can be used to authorize access to data in your storage account via Shared Key authorization.
- Microsoft recommends that you use Azure Key Vault to manage your access keys, and that you regularly rotate and regenerate your keys. Using Azure Key Vault makes it easy to rotate your keys without interruption to your applications. You can also manually rotate your keys
- You can use the Key Vault managed storage account key feature to list (sync) keys with an Azure storage account, and regenerate (rotate) the keys periodically
- You can also ask Key Vault to generate shared access signature tokens
- https://docs.microsoft.com/en-us/azure/key-vault/secrets/overview-storage-keys-powershell

Set and manage immutability policies for Blob storage



Immutable storage for Azure Blob storage enables users to store business-critical data objects in a WORM (Write Once, Read Many) state. This state makes the data non-erasable and non-modifiable for a user-specified interval. For the duration of the retention interval, blobs can be created and read, but cannot be modified or deleted.

Immutable storage is available for general-purpose v2 and Blob storage accounts in all Azure regions

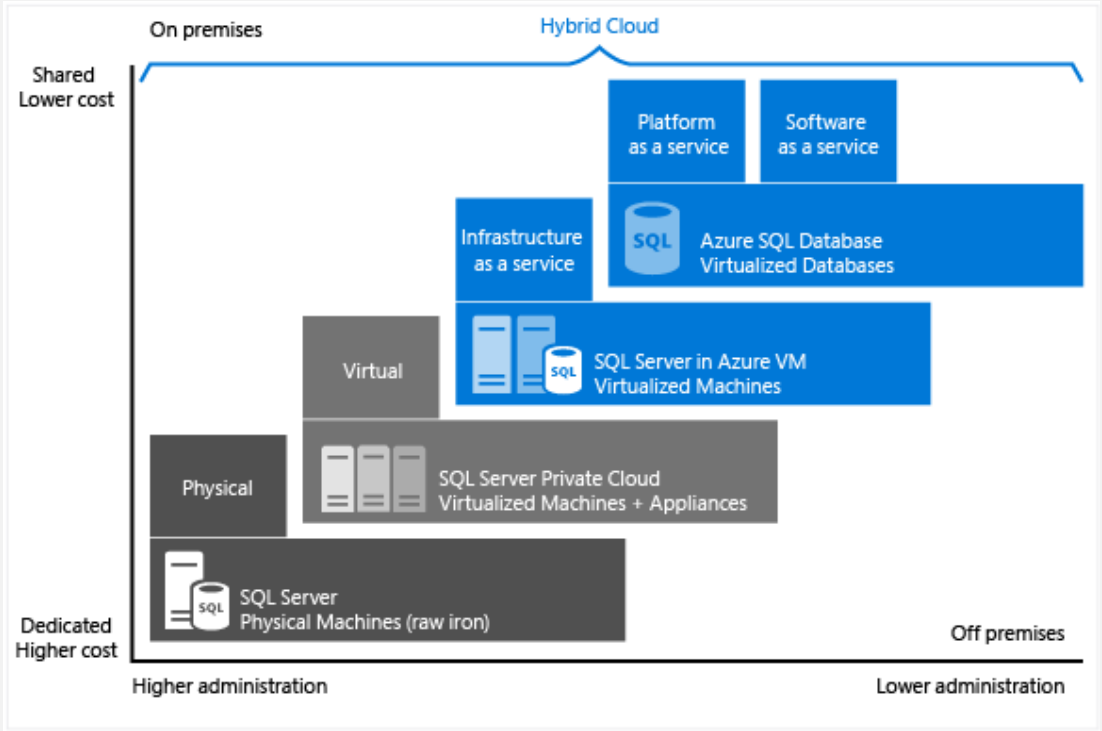
Microsoft Security

Azure SQL Security

Azure SQL Overview

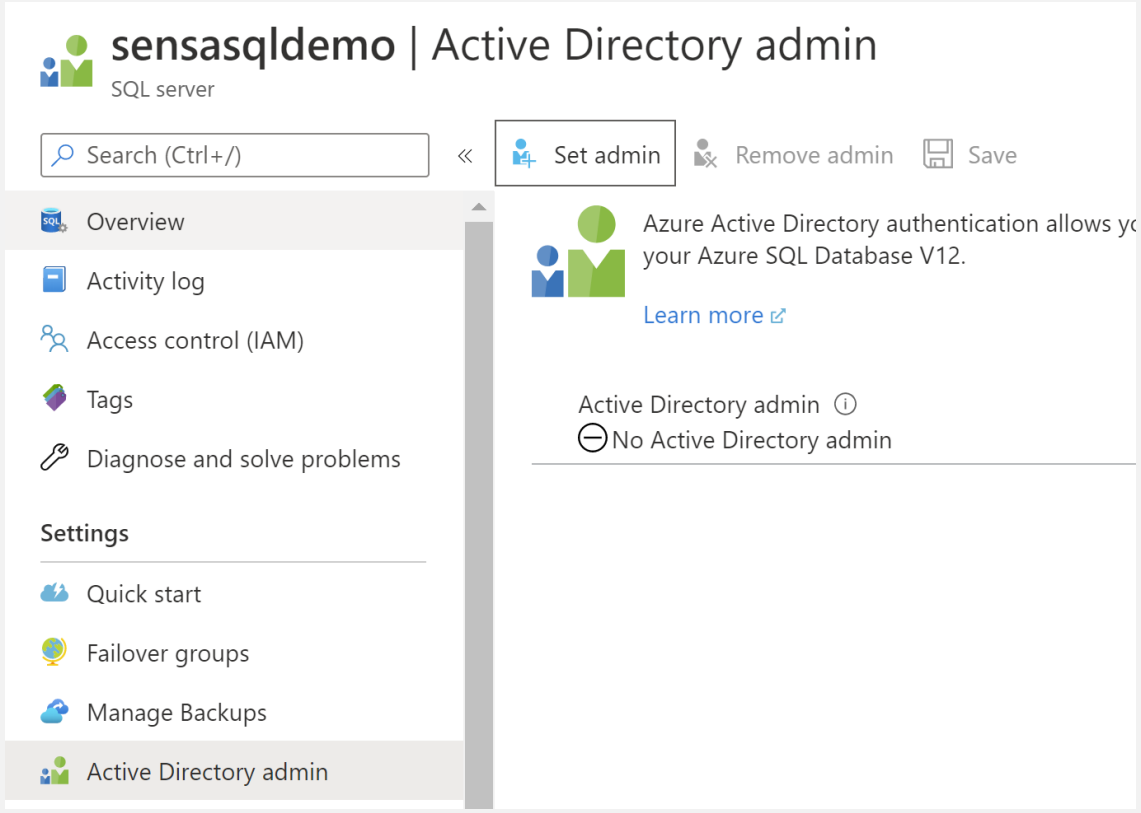
Azure SQL is a family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud.

- **Azure SQL Database**: Support modern cloud applications on an intelligent, managed database service, that includes serverless compute.
- **Azure SQL Managed Instance**: Modernize your existing SQL Server applications at scale with an intelligent fully managed instance as a service, with almost 100% feature parity with the SQL Server database engine. Best for most migrations to the cloud.
- **SQL Server on Azure VMs**: Lift-and-shift your SQL Server workloads with ease and maintain 100% SQL Server compatibility and operating system-level access.



Azure SQL – Database Authentication

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



Azure SQL Database Firewall

When you create a new server in Azure SQL Database a server-level firewall blocks all access to the public endpoint for the server.

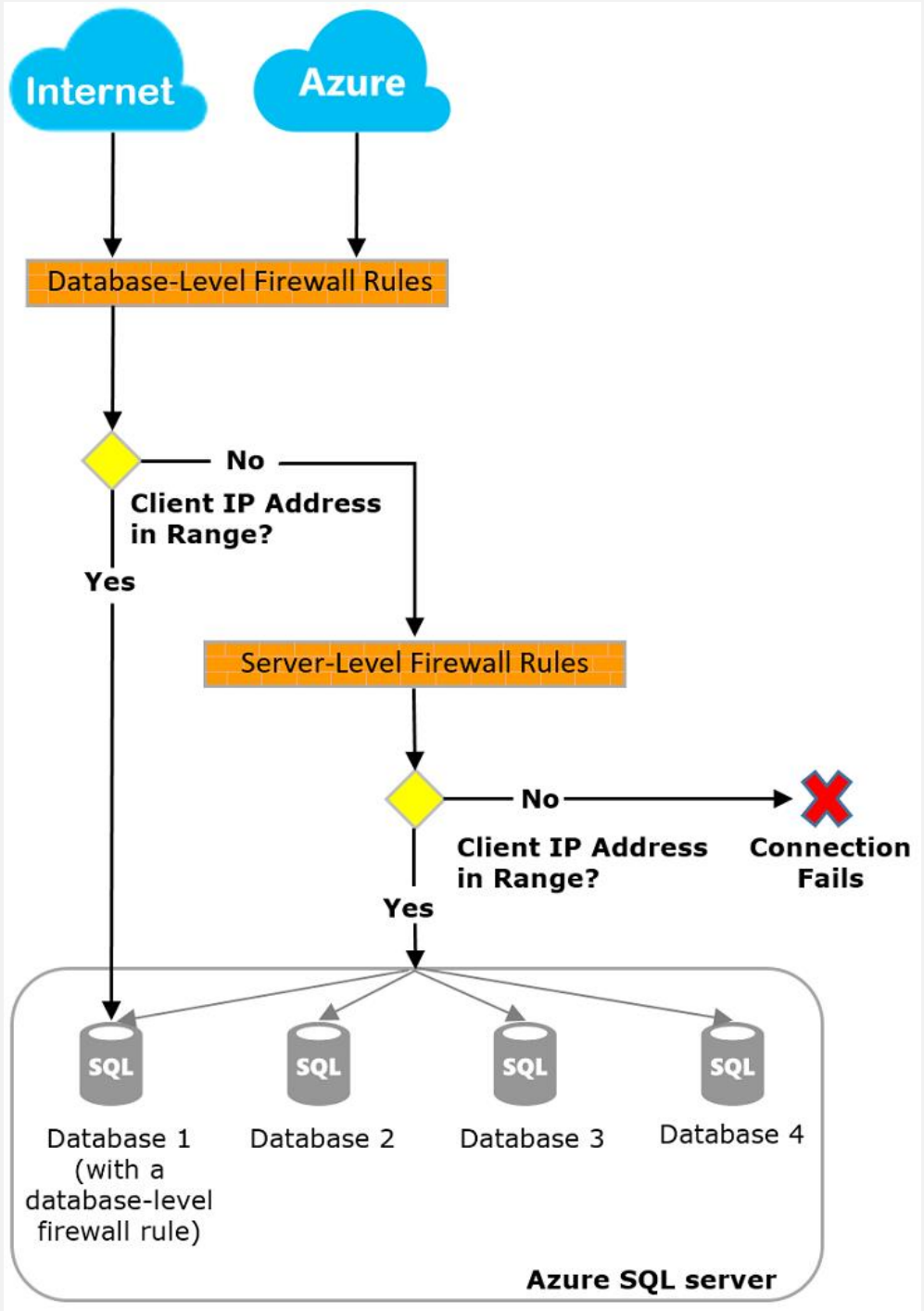
Connection attempts from the internet and Azure must pass through the firewall before they reach your server or database.

Server-level IP firewall rules

These rules enable clients to access your entire server, that is, all the databases managed by the server. The rules are stored in the *master* database. You can have a maximum of 128 server-level IP firewall rules for a server.

Database-level IP firewall rules

Database-level IP firewall rules enable clients to access certain (secure) databases. You create the rules for each database (including the master database), and they're stored in the individual database.



Azure SQL Encryption

Transport Layer Security (Encryption-in-transit)

SQL Database and SQL Managed Instance secure customer data by encrypting data in motion with Transport Layer Security (TLS). This ensures all data is ncrypted "in transit" between the client and server irrespective of the setting of **Encrypt** or **TrustServerCertificate** in the connection string.

Transparent Data Encryption (Encryption-at-rest)

Transparent Data Encryption (TDE) for Azure SQL Database and SQL Managed Instance adds a layer of security to help protect data at rest from unauthorized or offline access to raw files or backups. Common scenarios include data center theft or unsecured disposal of hardware or media such as disk drives and backup tapes. TDE encrypts the entire database using an AES encryption algorithm, which doesn't require application developers to make any changes to existing applications

Always Encrypted (Encryption-in-use)

Always Encrypted is a feature designed to protect sensitive data stored in specific database columns from access (for example, credit card numbers, national identification numbers, or data on a need to know basis). This includes database administrators or other privileged users who are authorized to access the database to perform management tasks, but have no business need to access the particular data in the encrypted columns. The data is always encrypted, which means the encrypted data is decrypted only for processing by client applications with access to the encryption key. The encryption key is never exposed to SQL Database or SQL Managed Instance and can be stored either in the Windows Certificate Store or in Azure Key Vault.

Azure SQL Advanced data security

Advanced Data Security (ADS) is a unified package for advanced SQL security capabilities. ADS is available for Azure SQL Database and Azure SQL Managed Instance.

It includes functionality for discovering and classifying sensitive data, surfacing and mitigating potential database vulnerabilities, and detecting anomalous activities that could indicate a threat to your database. It provides a single go-to location for enabling and managing these capabilities.

Data Discovery & Classification provides capabilities built into Azure SQL Database and Azure SQL Managed Instance, for discovering, classifying, labeling, and reporting 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 it includes actionable steps to resolve security issues and enhance your database fortification.

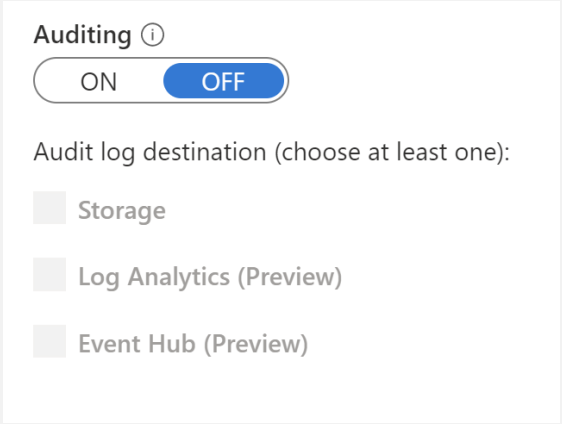
Advanced Threat Protection detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit your database. It continuously monitors your database for suspicious activities, and it provides immediate security alerts on potential vulnerabilities, Azure SQL injection 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.

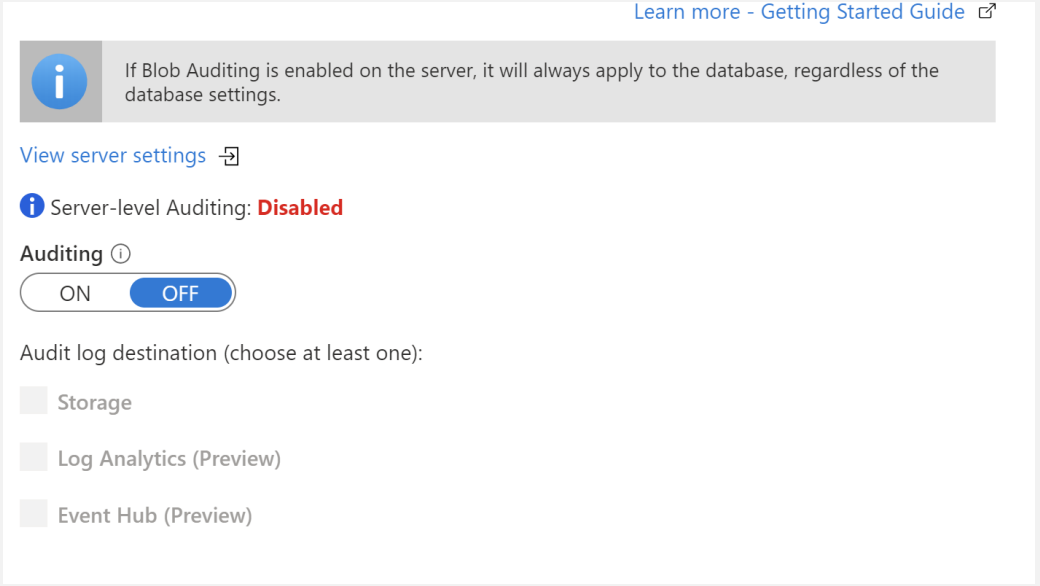
Azure SQL - Auditing

Primary characteristics of Azure SQL Database auditing:

- Facilitates tracking of designated events, reporting on database activities, and event analysis
- Is configurable via server and database audit policies

- Enable a server policy to audit all existing and newly created databases
- Enable a database policy to:
  - Retain an audit trail of selected events
  - Report on DB Activity
  - Analyze reports
- Server vs DB Auditing
  - Server policy applies to existing and new database
  - DB-level settings do not override server-level settings





Microsoft Security

Big Data Security

Overview Big data

Big data is collected in escalating volumes, at higher velocities, and in a greater variety of formats than ever before. It can be historical (meaning stored) or real time (meaning streamed from the source).

Azure Data Lake (Storage / Repository)

HD Insight (Process Massive Data)

Azure HD Insight Overview

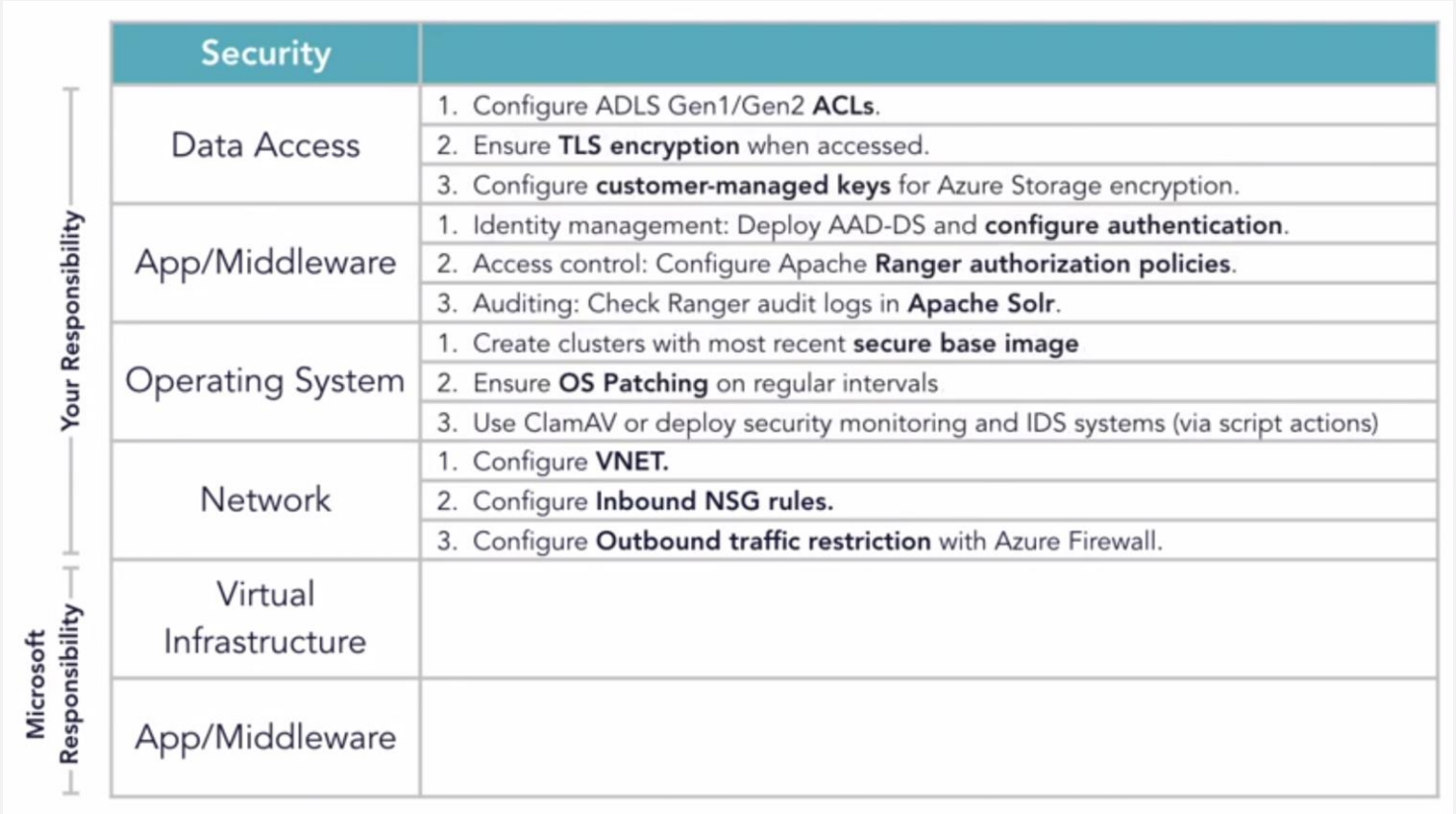
Azure HDInsight is a cloud distribution of Hadoop components. Azure HDInsight makes it easy, fast, and cost-effective to process massive amounts of data. You can use the most popular open-source frameworks such as Hadoop, Spark, Hive, LLAP, Kafka, Storm, R, and more. With these frameworks, you can enable a broad range of scenarios such as extract, transform, and load (ETL), data warehousing, machine learning, and IoT.

Azure HDInsight can be used for a variety of scenarios in big data processing:

- Batch processing (ETL)
- Data warehousing
- Internet of Things (IoT)
- Data science
- Hybrid

HD Insight Security Measures

1. HDInsight Data Encryption (at rest and in motion)
2. Virtual Networks
3. Azure AD Authentication
4. Role Based Access Control

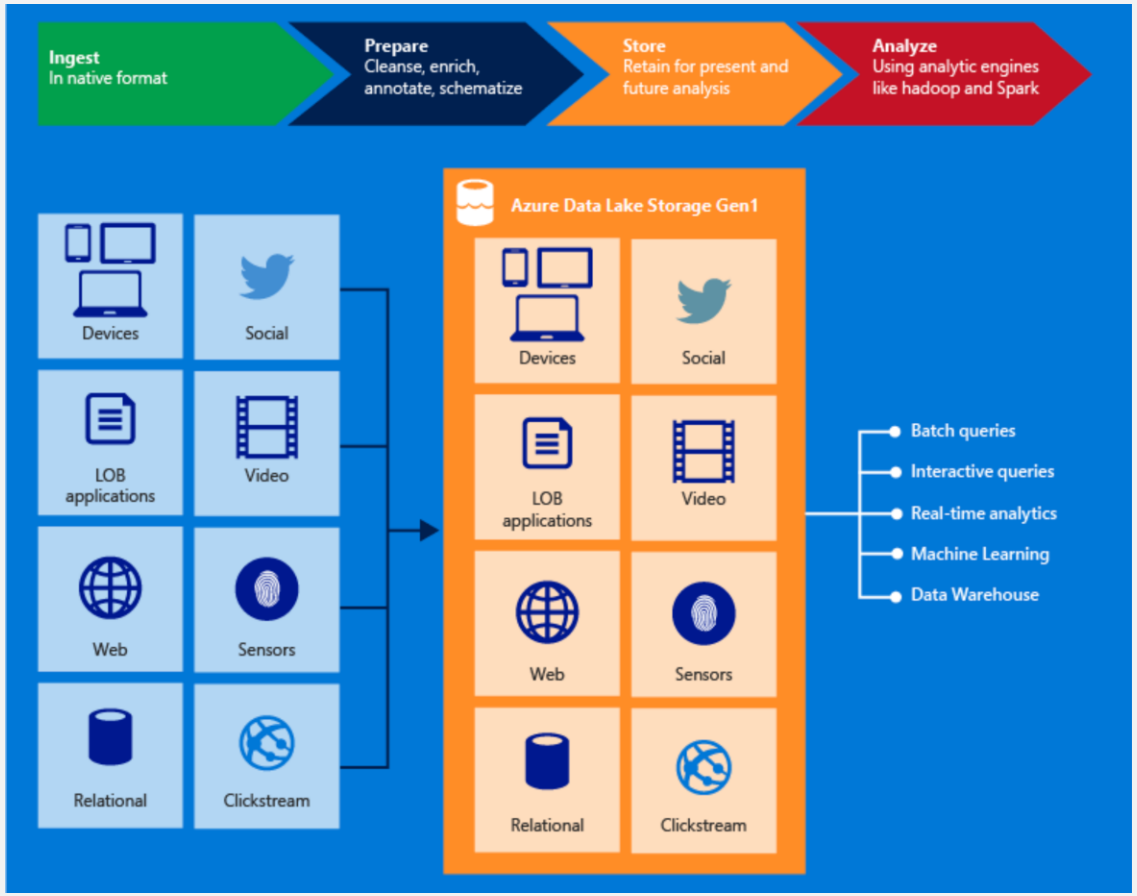


	Security	
Your Responsibility	Data Access	1. Configure ADLS Gen1/Gen2 **ACLs**.
		2. Ensure **TLS encryption** when accessed.
		3. Configure **customer-managed keys** for Azure Storage encryption.
	App/Middleware	1. Identity management: Deploy AAD-DS and **configure authentication**.
		2. Access control: Configure Apache **Ranger authorization policies**.
		3. Auditing: Check Ranger audit logs in **Apache Solr**.
	Operating System	1. Create clusters with most recent **secure base image**
		2. Ensure **OS Patching** on regular intervals
		3. Use ClamAV or deploy security monitoring and IDS systems (via script actions)
	Network	1. Configure **VNET.**
		2. Configure **Inbound NSG rules.**
		3. Configure **Outbound traffic restriction** with Azure Firewall.
Microsoft Responsibility	Virtual Infrastructure	
	App/Middleware	

Azure Data Lake Overview

Azure Data Lake Storage Gen1 is an enterprise-wide hyper-scale repository for big data analytic workloads. Azure Data Lake enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics.

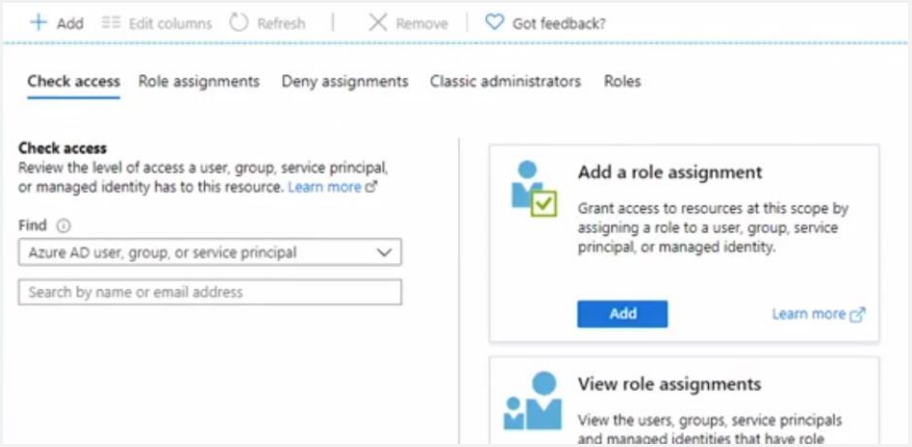
Data Lake Storage Gen1 can be accessed from Hadoop (available with HDInsight cluster) using the WebHDFS-compatible REST APIs. It's designed to enable analytics on the stored data and is tuned for performance for data analytics scenarios. Data Lake Storage Gen1 includes all enterprise-grade capabilities: security, manageability, scalability, reliability, and availability.



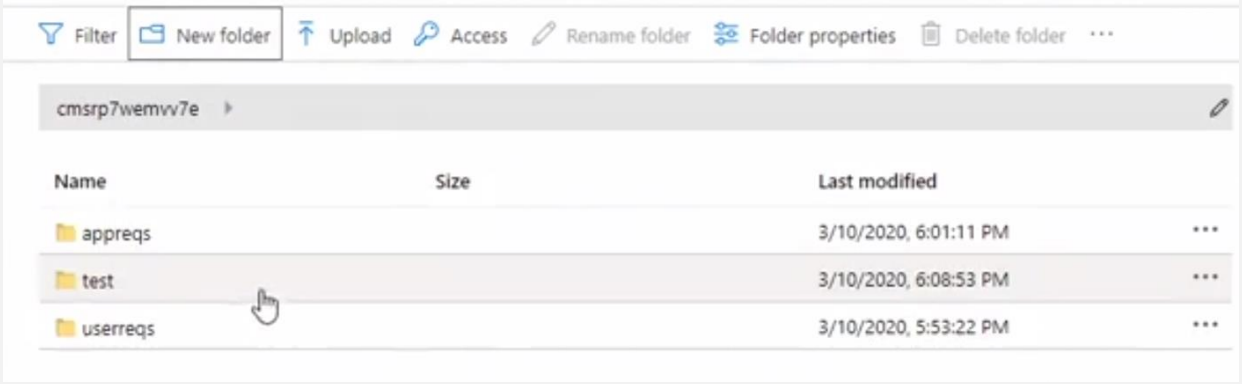
Azure Data Lake Security

Permissions in Azure Data Lake:

Role-based access control (RBAC) at the resource level



POSIX access control list)ACL) at the data level



Overview Azure Cosmos DB

Today's applications are required to be highly responsive and always online. To achieve low latency and high availability, instances of these applications need to be deployed in datacenters that are close to their users. Applications need to respond in real time to large changes in usage at peak hours, store ever increasing volumes of data, and make this data available to users in milliseconds.

Azure Cosmos DB is Microsoft's globally distributed, multi-model database service. With a click of a button, Cosmos DB enables you to elastically and independently scale throughput and storage across any number of Azure regions worldwide. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access using your favorite API including: SQL, MongoDB, Cassandra, Tables, or Gremlin

Cosmos DB enables you to build highly responsive and highly available applications worldwide. Cosmos DB transparently replicates your data wherever your users are, so your users can interact with a replica of the data that is closest to them.

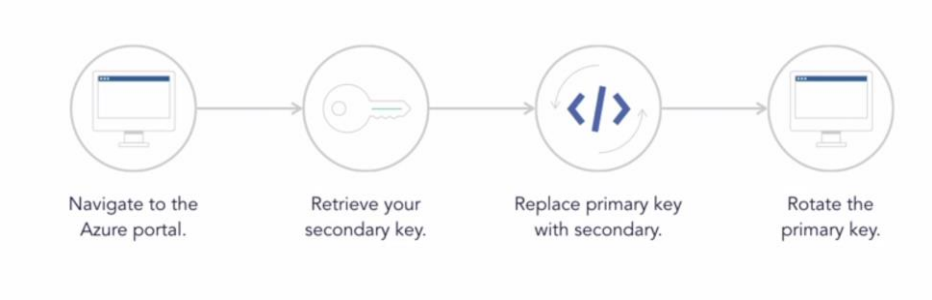
By virtue of deep integration with Azure infrastructure and transparent multi-master replication, Cosmos DB provides 99.999% high availability for both reads and write

https://docs.microsoft.com/en-us/azure/cosmos-db/introduction

Azure Cosmos DB Security

Azure Cosmos DB uses two types of keys

- Master key used for administrative resources.
  - Two Master keys: this ensures you can regenerate or roll keys without impacting access to your data
  - Rotate Master Keys



- Resource tokens used for application resources
  - Provide access to the application resources within a database
  - You can use a resource token to provice access to a client that cannot be trusted with the master key

Microsoft Security

Azure Backup

What can I back up?

On-premises - Back up files, folders, system state using the Microsoft Azure Recovery Services (MARS) agent. Or use the DPM or Azure Backup Server (MABS) agent to protect on-premises VMs (Hyper-V and VMWare) and other on-premises workloads

Azure VMs - Back up entire Windows/Linux VMs (using backup extensions) or back up files, folders, and system state using the MARS agent.

Azure Files shares - Back up Azure File shares to a storage account

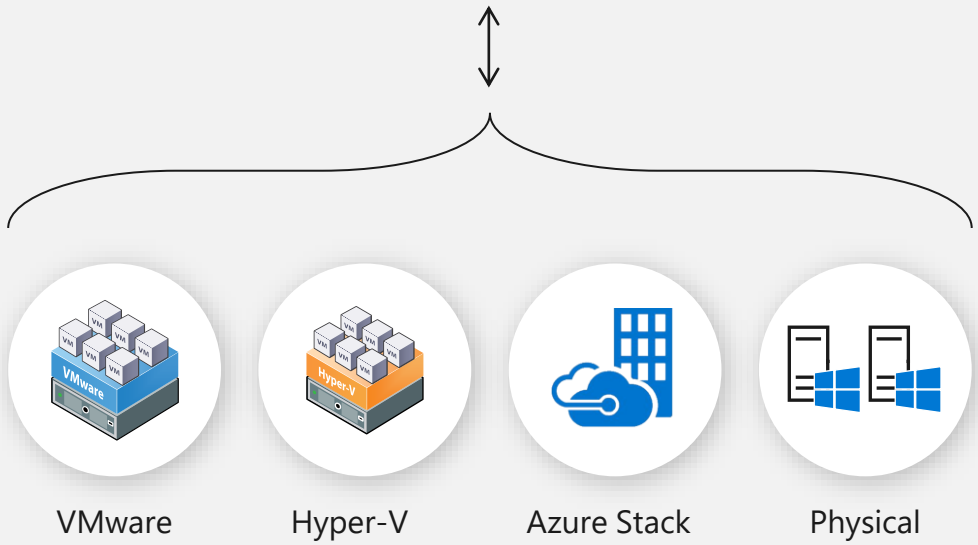
SQL Server in Azure VMs - Back up SQL Server databases running on Azure VMs

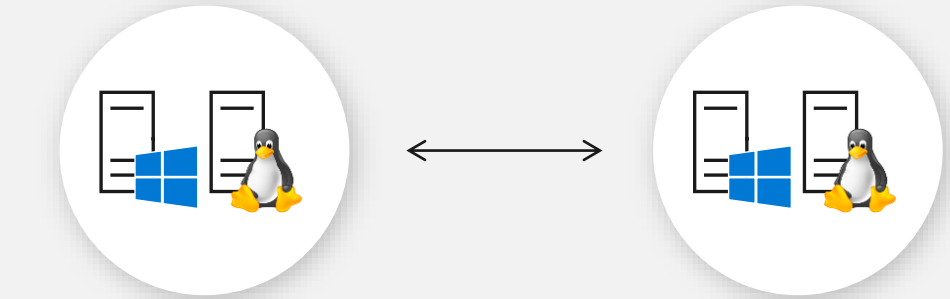
SAP HANA databases in Azure VMs - Backup SAP HANA databases running on Azure VMs

Azure Site Recovery

The complete disaster recovery solution

Private cloud to Azure | **Any cloud** | **Azure to Azure**





Azure to Azure

Region to Region

NEW Availability Zone to Availability Zone

Windows | **Any OS** | **Linux**

Grazie! – Q&A