Microsoft

Security Architecture for Azure Data Services

Azure Data Services – Security Reference Guide

Prepared by

DM Jumpstart Engineering Team (askdmjfordmtools@microsoft.com)

Disclaimer

The High-Level Architecture, Migration Dispositions and guidelines in this document is developed in consultation and collaboration with Microsoft Corporation technical architects. Because Microsoft must respond to changing market conditions, this document should not be interpreted as an invitation to contract or a commitment on the part of Microsoft.

Microsoft has provided generic high-level guidance in this document with the understanding that MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE INFORMATION CONTAINED HEREIN.

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Note: The detail provided in this document has been harvested as part of a customer engagement sponsored through the Azure Data Services Jumpstart Program.

Table of Contents

1 Introduction

This whitepaper outlines general guidelines for application and data security in Microsoft Azure Platform for organizations looking to leverage benefits of big data on Microsoft Cloud Platform offerings such as Azure HDInsight, Azure Cosmos DB, Azure Databricks, Azure SQL Data Warehouse, Azure SQL MI and Azure SQL DB. Azure Databases comes with industry-leading innovations such as built-in security with single sign-on and multi-factor authentication with Azure Active Directory as well automatic monitoring and threat detection. Security is a top concern for managing data in cloud, and it has always been a priority for Azure Data Services. For example, Azure SQL Database/Data Warehouse supports connection security with firewall rules and connection encryption. It supports authentication with username and password as well as integrated Azure Active Directory (Azure AD) authentication, which uses identities managed by Azure Active Directory. Authorization uses Role-Based Access Control (RBAC).

Azure SQL DB and DW support transparent data encryption by performing real-time encryption and decryption of databases, associated backups, and transaction log files at rest without requiring changes to the application.

This whitepaper also talks about Azure Key Vault and how it can be leveraged for securely storing keys/certificates in security modules, which can used with several other services, for example like Azure Databricks.

2 Security Architecture

Security in Azure SQL DB, DW, Azure Databricks, Azure HDInsight & Azure Cosmos DB follows Defense-in-Depth or peripheral security. There is a multiple level of security checks in place to protect your data from various kind of threats. Data encryption, multi-level authorization and authentication, network security is provided by Azure. In this whitepaper we are going to explain security architecture, and different options you have for Azure data platforms and how you can leverage those in your Azure Cloud Data Platform implementations.

3 Data Protection

3.1 Encryption

Azure supports various encryption models, including server-side encryption that uses service-managed keys, customer-managed keys in Key Vault, or customer-managed keys on customer-controlled hardware. With client-side encryption, you can manage and store keys on-premises or in another secure location.

3.2 Client-side encryption

Client-side encryption is performed outside of Azure. It includes:

- Data encrypted by an application that's running in the customer's datacenter or by a service application.
- Data that is already encrypted when it is received by Azure.

With client-side encryption, cloud service providers don't have access to the encryption keys and cannot decrypt this data. You maintain complete control of the keys.

3.3 Server-side encryption

The three server-side encryption models offer different key management characteristics, which you can choose according to your requirements:

- **Service-managed keys**: Provides a combination of control and convenience with low overhead.
- **Customer-managed keys**: Gives you control over the keys, including Bring Your Own Keys (BYOK) support, or allows you to generate new ones.
- **Service-managed keys in customer-controlled hardware**: Enables you to manage keys in your proprietary repository, outside of Microsoft control. This characteristic is called Host Your Own Key (HYOK). However, configuration is complex, and most Azure services don't support this model as it imposes several other challenges.

Azure SQL DB (and for that matter other services like Azure DW, Azure Data Lake Store, Cosmos DB etc.) has Transparent Data Encryption (TDE) which helps protect against the threat of malicious activity by encrypting and decrypting your data at rest. When you encrypt your database, associated backups and transaction log files are encrypted without requiring any changes to your applications. TDE encrypts the storage of an entire database by using a symmetric key called the database encryption key.

In SQL DB, the database encryption key is protected by a built-in server certificate. The built-in server certificate is unique for each SQL Database server. Microsoft automatically rotates these certificates at least every 90 days. The encryption algorithm used by SQL Data Warehouse is AES-256. For a general description of TDE, see Transparent Data Encryption.

You can encrypt your database using the Azure portal or T-SQL. This encryption policy applies to Azure Cosmos DB, Azure SQL Database.

Azure Databricks supports HIPAA compliant deployment to process PHI data. In this deployment mode, all PHI data will be encrypted at rest and through transit. Azure Databricks also supports GDPR. To read more about it, refer to this article

In Azure HDInsight for HDInsight clusters -- Azure Blob storage and Azure Data Lake Storage Gen1/Gen2 – which support transparent server-side encryption of data at rest. Secure HDInsight clusters will seamlessly work with this capability of server-side encryption of data at rest.

3.4 Encryption of data in transit

Azure offers many mechanisms for keeping data private as it moves from one location to another.

3.5 TLS/SSL encryption in Azure

Microsoft uses the Transport Layer Security (TLS) protocol to protect data when it's traveling between the cloud services and customers. Microsoft datacenters negotiate a TLS connection with client systems that connect to Azure services. TLS provides strong authentication, message privacy, and integrity (enabling detection of message tampering, interception, and forgery), interoperability, algorithm flexibility, and ease of deployment and use.

Perfect Forward Secrecy (PFS) protects connections between customers' client systems and Microsoft cloud services by unique keys. Connections also use RSA-based 2,048-bit encryption key lengths. This combination makes it difficult for someone to intercept and access data that is in transit.

3.6 Microsoft Azure Key Vault

Azure Key Vault offers an easy, cost-effective way to safeguard keys and other secrets used by cloud apps and services. With Key Vault, customers can streamline key management and maintain control of keys used to access and encrypt their data.

Key management lifecycle

Security Operations - Supplies keys

- Creates a Key Vault in Azure

- Adds keys / secrets to the Vault
- Grants permission to specific application(s) to perform specific operations using keys e.g. decrypt, unwrap
- Enables usage logs

Developer/IT Pro - Deploys application

- Tells application the URI of the key / secret
- Application programmatically uses key / secret (and may abuse)

Auditor - Monitors access to keys

- Reviews usage logs to confirm proper key use and compliance with data security standards

Secure storage access keys with Azure Key Vault

- Store access keys and SAS tokens in Key Vault
- Use Azure Automation job to periodically rotate keys, generate SAS Tokens, Update Key Vault
- Give applications permission in Key Vault to read secrets
- Applications read keys and tokens from Key Vault
    - Cache secrets in app for time less than rotation period
- More details here:
    - http://www.dushyantgill.com/blog/2015/04/26/say-goodbye-to-key-management-manage-access-to-azure-storage-data-using-azure-ad/

Azure Key Vault is used to Protect sensitive data and store encryption keys. You can read more on how Azure Key Vault can be used with Azure SQL Database here.

Likewise, Azure Key Vault can be used with Azure Cosmos DB, Azure HDInsight, Azure Databricks

For added assurance, when you use Azure Key Vault, you can import or generate keys in hardware security modules (HSMs) that never leave the HSM boundary. This scenario is often referred to as bring your own key, or BYOK. The HSMs are FIPS 140-2 Level 2 validated. Azure Key Vault uses Thales n Shield family of HSMs to protect your keys.

Use the information in this topic to help you plan for, generate, and then transfer your own HSM-protected keys to use with Azure Key Vault.

This functionality is not available for Azure China at the time of writing this paper. Read More

3.7 Encryption – at different services

Azure Key Vault offers an easy, cost-effective way to safeguard keys and other secrets used by cloud apps and services.

3.7.1 Azure Blob Storage

Data at rest in Azure Blob storage and Azure file shares can be encrypted in both server-side and client-side scenarios.

Azure Storage Service Encryption (SSE) can automatically encrypt data before it is stored, and it automatically decrypts the data when you retrieve it. The process is completely transparent to users. Storage Service Encryption uses 256-bit Advanced Encryption Standard (AES) encryption, which is one of the strongest block ciphers available. AES handles encryption, decryption, and key management transparently.

To read more on Azure Blob Storage Encryption, click here

3.7.2 Azure Data Lake Store

Azure Data Lake is an enterprise-wide repository of every type of data collected in a single place prior to any formal definition of requirements or schema. Data Lake Store supports "on by default," transparent encryption of data at rest, which is set up during the creation of your account. By default, Azure Data Lake Store manages the keys for you, but you have the option to manage them yourself.

Three types of keys are used in encrypting and decrypting data: the Master Encryption Key (MEK), Data Encryption Key (DEK), and Block Encryption Key (BEK). The MEK is used to encrypt the DEK, which is stored on persistent media, and the BEK is derived from the DEK and the data block. If you are managing your own keys, you can rotate the MEK. To read more on Azure Data Lake Store Encryption, click here

3.7.3 Azure Event Hub / Event Grid

Encryption in Azure Event Hub is achieved using Server-Side Using Service-Managed Key, read more

3.7.4 Azure SQL Database or SQL MI or SQL DW or SQL for PostgreSQL/MySQL/MariaDB

Transparent data encryption (TDE) helps protect Azure SQL Database, Azure SQL Managed Instance, 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 databases. TDE cannot be used to encrypt the logical master database in SQL Database. The master database contains objects that are needed to perform the TDE operations on the user databases.

TDE needs to be manually enabled for Azure SQL Managed Instance, older databases of Azure SQL Database, or Azure SQL Data Warehouse. To read more on TDE, click here

3.7.5 Azure Cosmos DB

All client-to-service Azure Cosmos DB interactions are SSL/TLS 1.2 capable. Also, all intra datacenter and cross datacenter replication is SSL/TLS 1.2 enforced.

Encryption at rest is a phrase that commonly refers to the encryption of data on nonvolatile storage devices, such as solid-state drives (SSDs) and hard disk drives (HDDs). Cosmos DB stores its primary databases on SSDs. Its media attachments and backups are stored in Azure Blob storage, which is generally backed up by HDDs. With the release of encryption at rest for Cosmos DB, all your databases, media attachments, and backups are encrypted. Your data is now encrypted in transit (over the network) and at rest (nonvolatile storage), giving you end-to-end encryption.

As a PaaS service, Cosmos DB is very easy to use. Because all user data stored in Cosmos DB is encrypted at rest and in transport, you don't have to take any action. Another way to put this is that encryption at rest is "on" by default. There are no controls to turn it off or on. Read More

3.7.6 Azure HDInsight

Azure HDInsight provides Encryption of data at rest. In most Hadoop distributions, HDFS is backed by local storage on the machines in the cluster. Using local storage can be costly for a cloud-based solution where you are charged hourly or by minute for compute resources.

HDInsight uses either blobs in Azure Storage or Azure Data Lake Store as the default store.

Encryption of data at rest helps you protect and safeguard your data to meet your organizational security and compliance commitments. With this feature, Azure Storage or Azure Data Lake Store automatically encrypts your data prior to persisting to storage and decrypts prior to retrieval. The encryption, decryption, and key management are totally transparent to users.

Data processed in Azure HDInsight can now be secured at rest via server-side encryption in Azure Storage or the Azure Data Lake Store.

It is possible also to achieve a node to node encryption in HDInsight for additional security but comes with a performance penalty, here are the details. Refer to the article to know about node to node encryption.

Disclaimer: *Microsoft does not recommend using this approach of node to node encryption as the HDInsight cluster already have VNet and other security provided features enabled by Azure Services.*

3.7.7 Azure Databricks

Like Azure HDInsight, Azure Databricks uses either blobs in Azure Storage or Azure Data Lake Store as storage layer. Data encryption of at rest for these services are transparent to Azure Databricks and its users.

Encryption of data at rest helps you protect and safeguard your data to meet your organizational security and compliance commitments. With this feature, Azure Storage or Azure Data Lake Store automatically encrypts your data prior to persisting to storage and decrypts prior to retrieval. The encryption, decryption, and key management are totally transparent to users.

3.8 Azure Data Box

Azure Data Box cloud solution lets you send terabytes of data into Azure in a quick, inexpensive, and reliable way. The secure data transfer is accelerated by shipping you a proprietary Data Box storage device. Each storage device has a maximum usable storage capacity of 80 TB and is transported to your datacenter through a regional carrier. The device has a rugged casing to protect and secure data during the transit.

Data Box has built-in security protections for the device, data, and the service.

- The device has a rugged casing secured by tamper-resistant screws and tamper-evident stickers.

- The data on the device is secured with an AES 256-bit encryption at all times.
- The device can only be unlocked with a password provided in the Azure portal.
- The service is protected by the Azure security features.
- Once your data is uploaded to Azure, the disks on the device are wiped clean, in accordance with NIST 800-88r1 standards. Read More

4 Access Control

To provide security, Azure SQL Database, SQL Data Warehouse and other Azure Data Services control access with firewall rules limiting connectivity by VNET/IP address, authentication mechanisms requiring users to prove their identity, and authorization mechanisms limiting users to specific actions and data.

4.1 Azure SQL Database - Firewall and firewall rules

Microsoft Azure SQL Database provides a relational database service for Azure and other Internet-based applications. To help protect your data, firewalls prevent all access to your database server until you specify which computers have permission. The firewall grants access to databases based on the originating IP address of each request. For more information, see Overview of Azure SQL Database firewall rules

The Azure SQL Database service is only available through TCP port 1433. To access a SQL Database from your computer, ensure that your client computer firewall allows outgoing TCP communication on TCP port 1433. If not needed for other applications, block inbound connections on TCP port 1433.

As part of the connection process, connections from Azure virtual machines are redirected to a different IP address and port, unique for each worker role. The port number is in the range from 11000 to 11999. For more information about TCP ports, see Ports beyond 1433 for ADO.NET 4.5 and SQL Database.

4.2 Authorization

4.2.1 Azure Blob Storage

Azure Storage supports authentication and authorization with Azure Active Directory (AD) for the Blob and Queue services. With Azure AD, you can use role-based access control (RBAC) to grant access to users, groups, or application service principals. Read more.

4.2.2 Azure Data Lake Store Gen 1

Azure Data Lake Storage Gen1 uses Azure Active Directory for authentication. Before authoring an application that works with Data Lake Storage Gen1, you must decide how to authenticate your application with Azure Active Directory (Azure AD).

Authentication options:

- **End-user authentication** - An end user's Azure credentials are used to authenticate with Data Lake Storage Gen1. The application you create to work with Data Lake Storage Gen1 prompts for these user credentials. As a result, this authentication mechanism is *interactive* and the application runs in the logged in user's context. For more information and instructions, see End-user authentication for Data Lake Storage Gen1.
- **Service-to-service authentication** - Use this option if you want an application to authenticate itself with Data Lake Storage Gen1. In such cases, you create an Azure Active Directory (AD) application and use the key from the Azure AD application to authenticate with Data Lake Storage Gen1. As a result, this authentication mechanism is *non-interactive*. For more information and instructions, see Service-to-service authentication for Data Lake Storage Gen1.

4.2.3 Azure Data Lake Store Gen 2

With the introduction of Azure Data Lake Storage Gen2, the long-awaited support for POSIX like ACL authentication.

Access control lists specify exactly which data objects a user may read, write, or execute (execute is required to browse the directory structure). ACLs are POSIX-compliant, thus familiar to those with a Unix or Linux background.

POSIX does not operate on a security inheritance model, which means that access ACLs are specified for every object. The concept of default ACLs is critical for new files within a directory to obtain the correct security settings, but it should not be thought of as inheritance. Because of the overhead assigning ACLs to every object, and because there is a limit of 32 ACLs for every object, it is extremely important to manage data-level security in ADLS Gen1 or Gen2 via Azure Active Directory groups.

Fortunately, both the ACLs for both directories and files are enforced regardless of which multi-protocol access point is used to access the data.

4.2.4 Azure Event Hub / Event Grid

The Event Hubs security model is based on a combination of Shared Access Signature (SAS) tokens and *event publishers*. An event publisher defines a virtual endpoint for an event hub. The publisher can only be used to send messages to an event hub. It is not possible to receive messages from a publisher. Read more

4.2.5 Azure SQL Database or SQL MI or SQL DW or SQL for PostgreSQL/MySQL/MariaDB

By using identities in Azure Active Directory (Azure AD) users can be authenticated in Azure SQL Database or SQL MI or SQL DW.

Azure Database for PostgreSQL/MySQL server supports native PostgreSQL/MySQL authentication. You can connect and authenticate to server with the server's admin login.

Azure Database for MariaDB server supports native MySQL authentication.

Note: *By the time, this whitepaper was written, AAD for Azure SQL MI was in preview.*

4.2.6 Azure Cosmos DB

Azure Cosmos DB uses hash-based message authentication code (HMAC) for authorization. Each request is hashed using the secret account key, and the subsequent base-64 encoded hash is sent with each call to Azure Cosmos DB. To validate the request, the Azure Cosmos DB service uses the correct secret key and properties to generate a hash, then it compares the value with the one in the request. If the two values match, the operation is authorized successfully and the request is processed, otherwise there is an authorization failure and the Request is rejected.

You can use either a master key, or a resource token allowing fine-grained access to a resource such as a document.

Learn more in Securing access to Azure Cosmos DB resources.

4.2.7 Azure HDInsight

HDInsight relies on a popular identity provider--Active Directory--in a managed way. By integrating HDInsight with Azure Active Directory Domain Services (Azure AD DS), you can access the clusters by using your domain credentials.

4.2.8 Azure Databricks

For authentication in Azure Databricks refer to section 5.4.1 and 5.6.1

5 Authentication

SQL Database supports two types of authentication:

5.1 SQL Authentication:

This authentication method uses a username and password. When you created the SQL Database 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."

5.2 Azure Active Directory Authentication

This authentication method uses identities managed by Azure Active Directory and is supported for managed and integrated domains. Use Active Directory authentication (integrated security) whenever possible. If you want to use Azure Active Directory Authentication, you must create another server admin called the "Azure AD admin," which can administer Azure AD users and groups. This admin can also perform all operations that a regular server admin can. See Connecting to SQL Database by Using Azure Active Directory Authentication for a walkthrough of how to create an Azure AD admin to enable Azure Active Directory Authentication.

To read more on authentication please refer to this article.

5.3 Azure Databricks

5.3.1 Azure Databricks Security | Data Protection

- Encryption-At-Rest – Service Managed Keys, User Managed Keys
- Encryption-in-flight (Transport Layer Security TLS)

5.3.1.1

- File/Folder Level ACLs for AAD Users, Groups, Service Principals
- ACLs for Clusters, Folders, Notebooks, Tables, Jobs
- Secrets with Azure Key Vault

Data Protection | Encryption | At-Rest

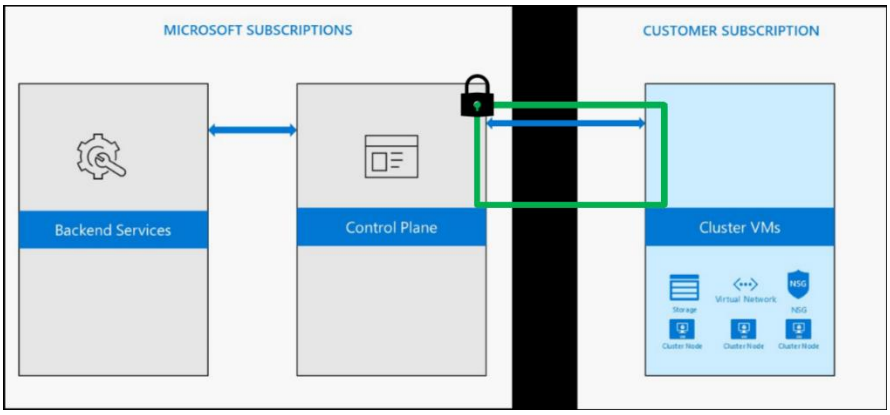
- Azure Databricks has separation of compute and storage
- Storage Services such as Azure Blob Store, Azure Data Lake Storage Provide
  - Encryption of Data

- Customer Managed Keys
- File/Folder Level ACLs (Azure Data Lake Storage)

Data Protection | Encryption | In-Transit

All the traffic from the Control Plane to the Clusters in the customer subscription is always encrypted with TLS.

5.3.1.2



5.3.2 Data Protection | Access Control | ADLS Passthru

- Authenticate automatically to Azure Data Lake Storage (ADLS) from Azure Databricks clusters using the same Azure Active Directory (Azure AD) identity that one uses to log into Azure Databricks.
- Commands running on a configured cluster will be able to read and write data in Azure Data Lake Storage without requiring one to configure service principal credentials.

5.3.3 Data Protection | Secrets

- Using our Secrets APIs, Secrets can be securely stored including in a Azure Key Vault or Databricks backend
- Authorized users can consume the secrets to access services

5.3.4 Azure Databricks Security | IAM/Auth

- Azure Active Directory (AAD) Authentication (w/ MFA)
- AAD Groups (using SCIM)
- AAD Conditional Access
- AAD Access Tokens

5.3.5 Azure Databricks Security | Network Security

- Managed VNets
- VNet Peering
- VNET Injection*
    - On-Premises Data Access
    - Single-IP SNAT and Firewall-based filtering via custom routing
    - Service Endpoint

5.3.6 Azure Databricks Rest APIs

To authenticate and access Azure Databricks REST APIs, you use personal access tokens. Tokens are like passwords; you should treat them with care. Tokens expire and can be revoked.

5.3.7 Azure Virtual Network for Azure Databricks

The default deployment of Azure Databricks is a fully managed service on Azure: all data plane resources, including a virtual network (VNet, section 6.1) that all clusters will be associated with, are deployed to a locked resource group. If you require network customization, however, you can deploy Azure Databricks data plane resources in your own virtual network (sometimes called VNet injection), enabling you to:

- Connect Azure Databricks to other Azure services (such as Azure Storage) in a more secure manner using service endpoints.
- Connect to on-premises data sources for use with Azure Databricks, taking advantage of user-defined routes.
- Connect Azure Databricks to a network virtual appliance to inspect all outbound traffic and take actions according to allow and deny rules.
- Configure Azure Databricks to use custom DNS.
- Configure network security group (NSG) rules to specify egress traffic restrictions.
- Deploy Azure Databricks clusters in your existing virtual network.

To read more on Azure Databricks VNet Injection click here

6 Network Security

6.1 Azure Virtual Network

An Azure Virtual Network (VNet) is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can use VNets to provision and manage virtual private networks (VPNs) in Azure and, optionally, link the VNets with other VNets in Azure, or with your on-premises IT infrastructure to create hybrid or cross-premises solutions. Each VNet you create has its own CIDR block and can be linked to other VNets and on-premises networks as long as the CIDR blocks do not overlap. You also have control of DNS server settings for VNets, and segmentation of the VNet into subnets.

Use VNets to:

- Create a dedicated private cloud-only VNet Sometimes you don't require a cross-premises configuration for your solution. When you create a VNet, your services and VMs within your VNet can communicate directly and securely with each other in the cloud. You can still configure endpoint connections for the VMs and services that require Internet communication, as part of your solution.
- Securely extend your data center With VNets, you can build traditional site-to-site (S2S) VPNs to securely scale your datacenter capacity. S2S VPNs use IPSEC to provide a secure connection between your corporate VPN gateway and Azure.
- Enable hybrid cloud scenarios VNets give you the flexibility to support a range of hybrid cloud scenarios. You can securely connect cloud-based applications to any type of on-premises system such as mainframes and Unix systems.

To read more about Azure VNet refer to this article

6.2 Service Endpoints (Virtual Network)

Virtual Network (VNet) service endpoints extend your virtual network private address space and the identity of your VNet to the Azure services, over a direct connection. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Traffic from your VNet to the Azure service always remains on the Microsoft Azure backbone network.
Virtual network rules are one firewall security feature that controls whether the database server for your single databases and elastic pool in Azure SQL Database or for your databases in SQL Data Warehouse accepts communications that are sent from particular subnets in virtual networks. This article explains why the virtual network rule feature is sometimes your best option for securely allowing communication to your Azure SQL Database and SQL Data Warehouse.
To read more on VNet Service Endpoints, refer to this article

For Azure Databricks, refer to section 5.1.5

To read more on how to use VNet Service Endpoints with various Azure Services, refer to this article.

6.3 Point-to-site VPN

A Point-to-Site (P2S) VPN gateway connection lets you create a secure connection to your virtual network from an individual client computer. A P2S connection is established by starting it from the client computer. This solution is useful for telecommuters who want to connect to Azure VNets from a remote location, such as from home or a conference. P2S VPN is also a useful solution to use instead of S2S VPN when you have only a few clients that need to connect to a VNet.

You use a Point-to-Site (P2S) VPN gateway to create a secure connection to your virtual network from an individual client computer. Point-to-Site VPN connections are useful when you want to connect to your VNet from a remote location. When you have only a few clients that need to connect to a VNet, a P2S VPN is a useful solution to use instead of a Site-to-Site VPN. A P2S VPN connection is established by starting it from the client computer.

To learn more about Point-to-site VPN, refer to this article

6.4 Site-to-Site VPN

A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it. Refer to this article to learn more about Site-to-Site VPN.

6.5 Express Route

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, Office 365, and Dynamics 365.

Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility. ExpressRoute

connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

6.6 Azure Cosmos DB | Network Security

Using an IP firewall is the first layer of protection to secure your database. Azure Cosmos DB supports policy driven IP-based access controls for inbound firewall support. The IP-based access controls are similar to the firewall rules used by traditional database systems, but they are expanded so that an Azure Cosmos DB database account is only accessible from an approved set of machines or cloud services.

Azure Cosmos DB enables you to enable a specific IP address (168.61.48.0), an IP range (168.61.48.0/8), and combinations of IPs and ranges.

All requests originating from machines outside this allowed list are blocked by Azure Cosmos DB. Requests from approved machines and cloud services then must complete the authentication process to be given access control to the resources.

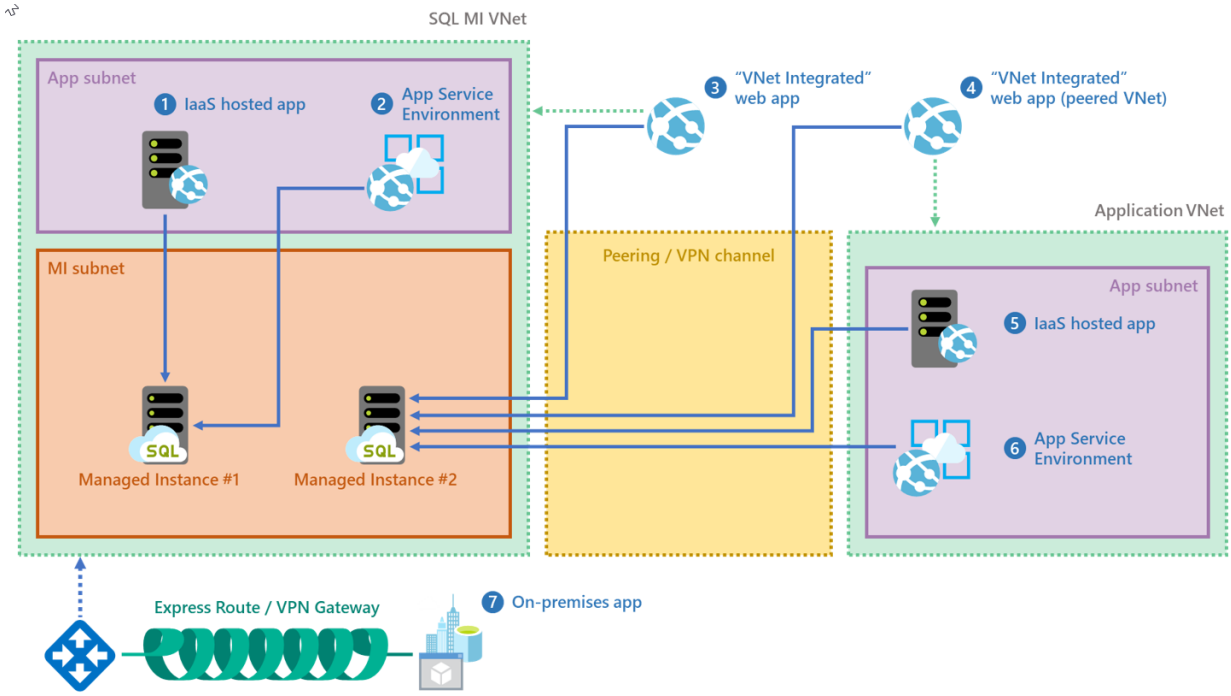
Learn more in Azure Cosmos DB firewall support.

6.7 Private Virtual Network

6.7.1 SQL Server Managed Instance | Network Security

A managed instance provides additional security isolation from other tenants in the Azure cloud. Security isolation includes:

- Native virtual network implementation and connectivity to your on-premises environment using Azure Express Route or VPN Gateway.
- SQL endpoint is exposed only through a private IP address, allowing safe connectivity from private Azure or hybrid networks.
- Single tenant with dedicated underlying infrastructure (compute, storage).



To learn more details about VNet integration and networking policy enforcement at the subnet level, see [VNet architecture for managed instances] and [Connect your application to a managed instance].

6.8 Azure Private link

Azure Private Link is a secure and scalable way for Azure customers to consume Azure Services like Azure Storage or SQL, Microsoft Partner Services or their own services privately from their Azure Virtual Network (VNet). The technology is based on a provider and consumer model where the provider and the consumer are both hosted in Azure.

Azure Private Link is useful to privately connect to Paas, Saas and own services. [Read More]

7 Threat Protection

7.1 Azure Blob Storage | Threat Protection

Advanced Threat Protection for Azure Storage provides an additional layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit storage accounts. This layer of protection allows you to address threats without the need to be a security expert or manage security monitoring systems.

Security alerts are triggered when anomalies in activity occur. These security alerts are integrated with Azure Security Center, and are also sent via email to subscription administrators, with details of suspicious activity and recommendations on how to investigate and remediate threats. Read more

7.2 Azure Data Lake Store | Thread Protection

7.3 Azure Event Hub / Event Grid | Threat Protection

Protect your real-time data. Event Hubs is certified by CSA STAR, ISO, SOC, GxP, HIPAA, HITRUST and PCI.

7.4 SQL auditing in Azure Monitor logs and Event Hubs

SQL Database auditing tracks database activities and helps to maintain compliance with security standards by recording database events to an audit log in a customer-owned Azure storage account. Auditing allows users to monitor ongoing database activities, as well as analyze and investigate historical activity to identify potential threats or suspected abuse and security violations. For more information, see Get started with SQL Database Auditing.

Auditing can be also enabled for security related events in server level or at a database level. This feature is supported in Azure SQL Database and Azure Managed Instance database.

7.5 Azure Cosmos DB | Threat Protection

https://docs.microsoft.com/en-us/azure/storage/common/storage-advanced-threat-protection

7.6 Azure HDInsight | Threat Protection

https://docs.microsoft.com/en-us/azure/storage/common/storage-advanced-threat-protection

7.7 Azure Databricks | Threat Protection

https://docs.microsoft.com/en-us/azure/storage/common/storage-advanced-threat-protection

7.8 Threat detection

Threat detection enhances auditing by analyzing audit logs for unusual behavior and potentially harmful attempts to access or exploit databases. Alerts are created for suspicious activities or anomalous access patterns such as SQL injection attacks, potential data infiltration, and brute force password attacks. Threat detection alerts are viewed from the Azure Security Center, where the details of the suspicious activities are provided and recommendations for further investigation given along with actions to mitigate the threat. For more information, see Get started with SQL Database Threat detection.

You can read more on Azure Threat Detection in this article

7.8.1 Azure SQL Database Managed Instance | Threat detection

Threat detection for a managed instance detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit databases. Threat detection can identify Potential SQL injection, Access from unusual location or data center, Access from unfamiliar principal or potentially harmful application, and Brute force SQL credentials - see more details in threat detection alerts.

You can receive notifications about the detected threats via email notifications or Azure portal

Threat detection is part of the advanced data security (ADS) offering, which is a unified package for advanced SQL security capabilities. Threat detection can be accessed and managed via the central SQL ADS portal.

7.8.2 Azure SQL Database threat detection

SQL Database secures customer data by providing auditing and threat detection capabilities.

Read More

7.8.3 Azure Cosmos DB

By using audit logging and activity logs, you can monitor your account for normal and abnormal activity. You can view what operations were performed on your resources, who initiated the operation, when the operation occurred, the status of the operation.

8 Reference

Azure SQL VM: https://docs.microsoft.com/en-in/azure/virtual-machines/windows/sql/virtual-machines-windows-sql-server-iaas-overview

https://docs.microsoft.com/en-in/azure/virtual-machines/linux/sql/sql-server-linux-virtual-machines-overview

Azure database security overview: https://docs.microsoft.com/en-us/azure/security/azure-database-security-overview

Azure SQL Database Access Control : https://docs.microsoft.com/en-us/azure/sql-database/sql-database-control-access

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-security-overview

Transparent Data Encryption: https://docs.microsoft.com/en-us/azure/sql-database/transparent-data-encryption-azure-sql

https://azure.microsoft.com/en-in/blog/bring-your-own-keys-for-apache-kafka-on-hdinsight/

Azure SQL MI P2S: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-configure-p2s

ExpressRoute: https://docs.microsoft.com/en-us/azure/expressroute/expressroute-introduction

Virtual network service endpoints: https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-service-endpoints-overview

Virtual Network for Azure Services: https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-for-azure-services

Azure Security Services: https://docs.microsoft.com/en-us/azure/security/azure-security-services-technologies

Azure SQL Database VNet Service Endpoint https://docs.microsoft.com/en-us/azure/sql-database/sql-database-vnet-service-endpoint-rule-overview

Azure SQL Database Security Features: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#azure-sql-database-security-features

Azure Data Platform: https://azure.microsoft.com/en-in/overview/data-platform/

Azure Key Vault BYOK: https://docs.microsoft.com/en-us/azure/key-vault/key-vault-hsm-protected-keys

https://docs.microsoft.com/en-us/azure/sql-database/transparent-data-encryption-byok-azure-sql

Azure HDInsight VNet: https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-extend-hadoop-virtual-network

Azure HDInsight Port Settings: https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-port-settings-for-services

Azure HDInsight Enterprise Security Package: https://docs.microsoft.com/en-us/azure/hdinsight/domain-joined/apache-domain-joined-introduction

Azure Cosmos DB Data Security: https://docs.microsoft.com/en-us/azure/cosmos-db/database-security

Azure Security: https://docs.microsoft.com/en-us/azure/security/azure-security-encryption-atrest

Suggested articles:

Azure Security: https://docs.microsoft.com/en-us/azure/security/

Azure security best practices and patterns: https://docs.microsoft.com/en-us/azure/security/security-best-practices-and-patterns

9 Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data Migration Jumpstart Team (askdmjfordmtools@microsoft.com). Thanks for your support!

Note: For additional information about migrating various source databases to Azure, see the Azure Database Migration Guide.