Security in Azure AI Search

Azure AI Search provides comprehensive security controls across network access, data access, and data protection to meet enterprise requirements. As a solution architect, you should understand three key security domains:

- **Network traffic patterns and network security:** Inbound, outbound, and internal traffic.
- **Access control mechanisms:** API keys or Microsoft Entra ID with roles.
- **Data residency and protection:** Encryption in transit, in use with optional confidential computing, and at rest with optional double encryption.

A search service supports multiple network security topologies, from IP firewall restrictions for basic protection to private endpoints for complete network isolation. Optionally, use a network security perimeter to create a logical boundary around your Azure PaaS resources. For enterprise scenarios requiring granular permissions, you can implement document-level access controls. All security features integrate with Azure's compliance framework and support common enterprise patterns like multitenancy and cross-service authentication using managed identities.

This article details the implementation options for each security layer to help you design appropriate security architectures for development and production environments.

Network traffic patterns

An Azure AI Search service can be hosted in the Azure public cloud, an Azure private cloud, or a sovereign cloud (such as Azure Government). By default, for all cloud hosts, the search service is typically accessed by client applications over public network connections. While that pattern is predominant, it's not the only traffic pattern that you need to care about. Understanding all points of entry as well as outbound traffic is necessary background for securing your development and production environments.

Azure AI Search has three basic network traffic patterns:

- Inbound requests made by a user or client to the search service (the predominant pattern)
- Outbound requests issued by the search service to other services on Azure and elsewhere
- Internal service-to-service requests over the secure Microsoft backbone network

Inbound traffic

Inbound requests that target a search service endpoint include:

- Create, read, update, or delete indexes and other objects on the search service
- Load an index with search documents
- Query an index
- Run indexer or skillset jobs

The REST APIs describe the full range of inbound requests that are handled by a search service.

At a minimum, all inbound requests must be authenticated using either of these options:

- Key-based authentication (default). Inbound requests provide a valid API key.
- Role-based access control. Authorization is through Microsoft Entra identities and role assignments on your search service.

Additionally, you can add network security features to further restrict access to the endpoint. You can create either inbound rules in an IP firewall, or create private endpoints that fully shield your search service from the public internet.

Outbound traffic

Outbound requests can be secured and managed by you. Outbound requests originate from a search service to other applications. These requests are typically made by indexers for text-based and multimodal indexing, custom skills-based AI enrichment, and vectorizations at query time. Outbound requests include both read and write operations.

The following list is a full enumeration of the outbound requests for which you can configure secure connections. A search service makes requests on its own behalf, and on the behalf of an indexer or custom skill.

Operation	Scenario
Indexers	Connect to external data sources to retrieve data (read access). For more information, see Indexer access to content protected by Azure network security.
Indexers	Connect to Azure Storage for write operations to knowledge stores, cached enrichments, debug sessions.
Custom skills	Connect to Azure functions, Azure web apps, or other apps running external code that's hosted off-service. The request for external processing is sent during skillset execution.
Indexers and integrated vectorization	Connect to Azure OpenAI and a deployed embedding model, or it goes through a custom skill to connect to an embedding model that you provide. The search service sends text to embedding models for vectorization during indexing.
Vectorizers	Connect to Azure OpenAI or other embedding models at query time to convert user text strings to vectors for vector search.
Knowledge bases	Connect to chat completion models for agentic retrieval query planning, and also for formulating answers grounded in search results. If you're implementing a basic RAG pattern, your query logic calls an external chat completion model for formulating an answer grounded in search results. For this pattern, the connection to the model uses the identity of your client or user. The search service identity isn't used for the connection. In contrast, if you use knowledge bases in a RAG retrieval pattern, the outbound request is made by the search service managed identity.
Search service	Connect to Azure Key Vault for customer-managed encryption keys used to encrypt and decrypt sensitive data.

Outbound connections can generally be made using a resource's full access connection string that includes a key or a database login, or a managed identity if you're using Microsoft Entra ID and role-based access.

To reach Azure resources behind a firewall, create inbound rules on other Azure resources that admit search service requests.

To reach Azure resources protected by Azure Private Link, create a shared private link that an indexer uses to make its connection.

Exception for same-region search and storage services

If Azure Storage and Azure AI Search are in the same region, network traffic is routed through a private IP address and occurs over the Microsoft backbone network. Because private IP addresses are used, you can't configure IP firewalls or a private endpoint for network security.

Configure same-region connections using either of the following approaches:

- Trusted service exception

- Resource instance rules

Internal traffic

Internal requests are secured and managed by Microsoft. You can't configure or control these connections. If you're locking down network access, no action on your part is required because internal traffic isn't customer-configurable.

Internal traffic consists of:

- Service-to-service calls for tasks like authentication and authorization through Microsoft Entra ID, resource logging sent to Azure Monitor, and private endpoint connections that utilize Azure Private Link.
- Requests for built-in skills processing, with same-region requests directed to an internally hosted Microsoft Foundry resource used exclusively for built-in skills processing by Azure AI Search.
- Requests made to the various models that support semantic ranking.

Network security

Network security protects resources from unauthorized access or attack by applying controls to network traffic. Azure AI Search supports networking features that can be your frontline of defense against unauthorized access.

Inbound connection through IP firewalls

A search service is provisioned with a public endpoint that allows access using a public IP address. To restrict which traffic comes through the public endpoint, create an inbound firewall rule that admits requests from a specific IP address or a range of IP addresses. All client connections must be made through an allowed IP address, or the connection is denied.

[[IMG:inbound-firewall-ip-restrictions.png]]

You can use the Azure portal to configure firewall access.

Alternatively, you can use the management REST APIs. Starting with API version 2020-03-13, with the IpRule parameter, you can restrict access to your service by identifying IP addresses, individually or in a range, that you want to grant access to your search service.

Inbound connection to a private endpoint (network isolation, no Internet traffic)

For more stringent security, you can establish a private endpoint for Azure AI Search allows a client on a virtual network to securely access data in a search index over a Private Link.

The private endpoint uses an IP address from the virtual network address space for connections to your search service. Network traffic between the client and the search service traverses over the virtual network and a private link on the Microsoft backbone network, eliminating exposure from the public internet. A virtual network allows for secure communication among resources, with your on-premises network as well as the Internet.

[[IMG:inbound-private-link-azure-cog-search.png]]

While this solution is the most secure, using more services is an added cost so be sure you have a clear understanding of the benefits before diving in. For more information about costs, see the pricing page. For instructions on how to set up the endpoint, see Create a Private Endpoint for Azure AI Search.

Network security perimeter

A network security perimeter is a logical network boundary around your platform-as-a-service (PaaS) resources that are deployed outside of a virtual network. It establishes a perimeter for controlling public network access to resources like Azure AI Search, Azure Storage, and Azure OpenAI. You can grant exceptions through explicit access rules for inbound and outbound traffic. This approach helps prevent data exfiltration while maintaining necessary connectivity for your applications.

Inbound client connections and service-to-service connections occur within the boundary, which simplifies and strengthens your defenses against unauthorized access. It's common in Azure AI Search solutions to use multiple Azure resources. The following resources can all be joined to an existing network security perimeter:

- Azure AI Search
- Azure OpenAI
- Azure Storage
- Azure Monitor

For a complete list of eligible services, see Onboarded private link resources.

Authentication

Once a request is admitted to the search service, it must still undergo authentication and authorization that determines whether the request is permitted. Azure AI Search supports two approaches:

- Microsoft Entra authentication establishes the caller (and not the request) as the authenticated identity. An Azure role assignment determines authorization.
- Key-based authentication is performed on the request (not the calling app or user) through an API key, where the key is a string composed of randomly generated numbers and letters that prove the request is from a trustworthy source. Keys are required on every request. Submission of a valid key is considered proof the request originates from a trusted entity.

Reliance on API key-based authentication means that you should have a plan for regenerating the admin key at regular intervals, per Azure security best practices. There are a maximum of two admin keys per search service. For more information about securing and managing API keys, see Create and manage api-keys.

Key-based authentication is the default for data plane operations (creating and using objects on the search service). You can use both authentication methods, or disable an approach that you don't want available on your search service.

Authorization

Azure AI Search provides authorization models for service management and content management.

Privileged access

On a new search service, existing role assignments at the subscription level are inherited by the search service, and only Owners and User Access Administrators can grant access.

Control plane operations (service or resource creation and management) tasks are exclusively authorized through role assignments, with no ability to use key-based authentication for service administration.

Control plane operations include create, configure, or delete the service, and manage security. As such, Azure role assignments will determine who can perform those tasks, regardless of whether they're using the portal, PowerShell, or the Management REST APIs.

Three basic roles (Owner, Contributor, Reader) apply to search service administration.

> **Note:** Using Azure-wide mechanisms, you can lock a subscription or resource to prevent accidental or unauthorized deletion of your search service by users with admin rights. For more information, see Lock resources to prevent unexpected deletion.

Authorize access to content

Data plane operations refers to the objects created and used on a search service.

- For role-based authorization, use Azure role assignments to establish read-write access to operations.
- For key-based authorization, an API key and a qualified endpoint determine access. An endpoint might be the service itself, the indexes collection, a specific index, a documents collection, or a specific document. When chained together, the endpoint, the operation (for example, a create request) and the type of key (admin or query) authorize access to content and operations.

Restricting access to indexes

Using Azure roles, you can set permissions on individual indexes as long as it's done programmatically.

Using keys, anyone with an admin key to your service can read, modify, or delete any index in the same service. For protection against accidental or malicious deletion of indexes, your in-house source control for code assets is the solution for reversing an unwanted index deletion or modification. Azure AI Search has failover within the cluster to ensure availability, but it doesn't store or execute your proprietary code used to create or load indexes.

For multitenancy solutions requiring security boundaries at the index level, it's common to handle index isolation in the middle tier in your application code. For more information about the multitenant use case, see Design patterns for multitenant SaaS applications and Azure AI Search.

Restricting access to documents

User permissions at the document level, also known as *row-level security*, is available as a preview feature and depends on the data source. If content originates from Azure Data Lake Storage (ADLS) Gen2 or Azure blobs, user permission metadata that originates in Azure Storage is preserved in indexer-generated indexes and enforced at query time so that only authorized content is included in search results.

For other data sources, you can push a document payload that includes user or group permission metadata, and those permissions are retained in indexed content and also enforced at query time. This capability is also in preview.

If you can't use preview features and you require permissioned access over content in search results, there's a technique for applying filters that include or exclude documents based on user identity. This workaround adds a string field in the data source that represents a group or user identity, which you can make filterable in your index. For more information about this pattern, see Security trimming based on identity filters. For more information about document access, see Document-level access control.

Data residency

When you set up a search service, you choose a region that determines where customer data is stored and processed. Each region exists within a geography (Geo) that often includes multiple regions (for example, Switzerland is a Geo that contains Switzerland North and Switzerland West). Azure AI Search might replicate your data to another region within the same Geo for durability and high availability. The service won't store or process customer data outside of your specified Geo unless you configure a feature that has a dependency on another Azure resource, and that resource is provisioned in a different region.

Currently, the only external resource that a search service writes to is Azure Storage. The storage account is one that you provide, and it could be in any region. A search service writes to Azure Storage if you use any of the following features:

- enrichment cache
- debug session
- knowledge store

For more information about data residency, see data residency in Azure.

Exceptions to data residency commitments

Object names appear in the telemetry logs used by Microsoft to provide support for the service. Object names are stored and processed outside of your selected region or location. Object names include the names of indexes and index fields, aliases, indexers, data sources, skillsets, synonym maps, resources, containers, and key vault store. Customers shouldn't place any sensitive data in name fields or create applications designed to store sensitive data in these fields.

Telemetry logs are retained for one and a half years. During that period, Microsoft might access and reference object names under the following conditions:

- Diagnose an issue, improve a feature, or fix a bug. In this scenario, data access is internal only, with no third-party access.
- During support, this information might be used to provide quick resolution to issues and escalate product team if needed.

Data protection

At the storage layer, data encryption is built in for all service-managed content saved to disk, including indexes, synonym maps, and the definitions of indexers, data sources, and skillsets. Service-managed encryption applies to both long-term data storage and temporary data storage.

Optionally, you can add customer-managed keys (CMK) for supplemental encryption of indexed content for double encryption of data at rest. For services created after August 1, 2020, CMK encryption extends to short-term data on temporary disks.

Data in transit

For search service connections over the public internet, Azure AI Search listens on HTTPS port 443.

Azure AI Search supports TLS 1.2 and 1.3 for client-to-service channel encryption:

- TLS 1.3 is the default on newer client operating systems and versions of .NET.
- TLS 1.2 is the default on older systems, but you can explicitly set TLS 1.3 on a client request.

Earlier versions of TLS (1.0 or 1.1) aren't supported.

For more information, see TLS support in .NET Framework.

Data in use

By default, Azure AI Search deploys your search service on standard Azure infrastructure. This infrastructure encrypts data at rest and in transit, but it doesn't protect data while it's being actively processed in memory.

Optionally, you can use the Azure portal or Services - Create or Update (REST API) to configure confidential computing during service creation. Confidential computing protects data in use from unauthorized access, including from Microsoft, through hardware attestation and encryption. For more information, see Confidential computing use cases.

The following table compares both compute types.

Compute type	Description	Limitations	Cost	Availability
Default	Processes data on standard VMs with built-in encryption for data at rest and in transit. No hardware-based isolation for data in use.	No limitations.	No change to the base cost of free or billable tiers.	Available in all regions.
Confidential	Processes data on confidential VMs (DCasv5 or DCesv5) in a hardware-based trusted execution environment. Isolates computations and memory from the host operating system and other VMs.	Disables or restricts agentic retrieval, semantic ranker, query rewrite, skillset execution, and indexers that run in the multitenant environment. When you enable this compute type, indexers can only run in the private execution environment, meaning they run from the search clusters hosted on confidential computing.	Adds a 10% surcharge to the base cost of billable tiers. For more information, see the pricing page.	Available in some regions. For more information, see the list of supported regions.

> **Important:** We only recommend confidential computing for organizations whose compliance or regulatory requirements necessitate data-in-use protection. For daily usage, the default compute type suffices.

Data at rest

For data handled internally by the search service, the following table describes the data encryption models. Some features, such as knowledge store, incremental enrichment, and indexer-based indexing, read from or write to data structures in other Azure Services. Services that have a dependency on Azure Storage can use the encryption features of that technology.

Model	Keys	Requirements	Restrictions	Applies to
server-side encryption	Microsoft-managed keys	None (built-in)	None, available on all tiers, in all regions, for content created after January 24, 2018.	Content (indexes and synonym maps) and definitions (indexers, data sources, skillsets) on data disks and temporary disks
server-side encryption	customer-managed keys	Azure Key Vault	Available on billable tiers, in specific regions, for content created after August 1, 2020.	Content (indexes and synonym maps) and definitions (indexers, data sources, skillsets) on data disks
server-side full encryption	customer-managed keys	Azure Key Vault	Available on billable tiers, in all regions, on search services after May 13, 2021.	Content (indexes and synonym maps) and definitions (indexers, data sources, skillsets) on data disks and temporary disks

When you introduce CMK encryption, you're encrypting content twice. For the objects and fields noted in the previous section, content is first encrypted with your CMK, and secondly with the Microsoft-managed key. Content is doubly encrypted on data disks for long-term storage, and on temporary disks used for short-term storage.

Service-managed keys

Service-managed encryption is a Microsoft-internal operation that uses 256-bit AES encryption. It occurs automatically on all indexing, including on incremental updates to indexes that aren't fully encrypted (created before January 2018).

Service-managed encryption applies to all content on long-term and short-term storage.

Customer-managed keys (CMK)

Customers use CMK for two reasons: extra protection, and the ability to revoke keys, preventing access to content.

Customer-managed keys require another billable service, Azure Key Vault, which can be in a different region, but under the same Azure tenant, as Azure AI Search.

CMK support was rolled out in two phases. If you created your search service during the first phase, CMK encryption was restricted to long-term storage and specific regions. Services created in the second phase can use CMK encryption in any region. As part of the second wave rollout, content is CMK-encrypted on both long-term and short-term storage.

- The first rollout was on August 1, 2020 and included the following five regions. Search services created in the following regions supported CMK for data disks, but not temporary disks:
  - West US 2
  - East US
  - South Central US
  - US Gov Virginia
  - US Gov Arizona
- The second rollout on May 13, 2021 added encryption for temporary disks and extended CMK encryption to all supported regions.

If you're using CMK from a service created during the first rollout and you also want CMK encryption over temporary disks, you need to create a new search service in your region of choice and redeploy your content. To determine your service creation date, see Check your service creation or upgrade date.

Enabling CMK encryption will increase index size and degrade query performance. Based on observations to date, you can expect to see an increase of 30-60 percent in query times, although actual performance will vary depending on the index definition and types of queries. Because of the negative performance impact, we recommend that you only enable this feature on indexes that really require it. For more information, see Configure customer-managed encryption keys in Azure AI Search.

Logging and monitoring

Azure AI Search doesn't log user identities so you can't refer to logs for information about a specific user. However, the service does log create-read-update-delete operations, which you might be able to correlate with other logs to understand the agency of specific actions.

Using alerts and the logging infrastructure in Azure, you can pick up on query volume spikes or other actions that deviate from expected workloads. For more information about setting up logs, see Collect and analyze log data and Monitor query requests.

Compliance and governance

Azure AI Search participates in regular audits, and has been certified against many global, regional, and industry-specific standards for both the public cloud and Azure Government. For the complete list, download the **Microsoft Azure Compliance Offerings** whitepaper from the official Audit reports page.

We recommend that you regularly review Azure AI Search compliance certifications and documentation to ensure alignment with your regulatory requirements.

Use Azure Policy

For compliance, you can use Azure Policy to implement the high-security best practices of Microsoft cloud security benchmark. The Microsoft cloud security benchmark is a collection of security recommendations, codified into security controls that map to key actions you should take to mitigate threats to services and data. There are currently 12 security controls, including Network Security, Logging and Monitoring, and Data Protection.

Azure Policy is a capability built into Azure that helps you manage compliance for multiple standards, including those of Microsoft cloud security benchmark. For well-known benchmarks, Azure Policy provides built-in definitions that provide both criteria and an actionable response that addresses noncompliance.

For Azure AI Search, there's currently one built-in definition. It's for resource logging. You can assign a policy that identifies search services that are missing resource logging, and then turn it on. For more information, see Azure Policy Regulatory Compliance controls for Azure AI Search.

Use tags

Apply metadata tags to categorize search services based on data sensitivity and compliance requirements. This facilitates proper governance and security controls. For more information, see Use tags to organize your Azure resources.