# Azure AI Search documentation

Information retrieval at scale for vector and text content in traditional or generative search scenarios.

## About Azure AI Search

CONCEPT

What is Azure AI Search?

AI enrichment

Semantic reranking

QUICKSTART

Create a search service

Create a search index

## Vector stores

CONCEPT

Vectors in Azure AI Search

Integrated vectorization (preview)

Retrieval Augmented Generation (RAG)

QUICKSTART

Create a vector store

Chat with your data

Query a vector store

SAMPLE

Vector samples

## Azure AI Studio

HOW-TO GUIDE

Create a vector store in AI Studio

Build a question and answer copilot

## Index data

CONCEPT

What's a search index?

Importing data

Indexer overview

HOW-TO GUIDE

Index from Azure Blob Storage

Index from Azure SQL Database

Index from Azure Cosmos DB

Index any data

## Develop apps

TUTORIAL

Add search to web apps

HOW-TO GUIDE

Develop in .NET

Develop in REST

REFERENCE

Azure REST API Reference

Azure SDK for .NET

Azure SDK for Python

Azure SDK for Java

Azure SDK for JavaScript

## Query data

CONCEPT

Query types and composition

Create a simple query

Create advanced queries

REFERENCE

Simple syntax (default)

OData language reference

Search Documents (REST)

# What's Azure AI Search?

Article • 11/22/2023

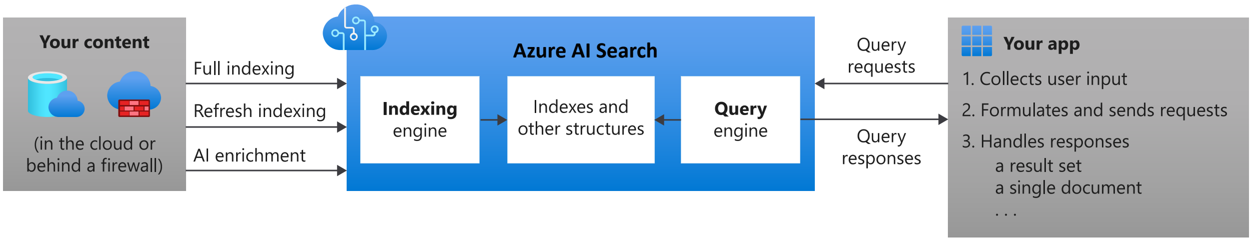
Azure AI Search (formerly known as "Azure Cognitive Search") provides secure information retrieval at scale over user-owned content in traditional and conversational search applications.

Information retrieval is foundational to any app that surfaces text and vectors. Common scenarios include catalog or document search, data exploration, and increasingly chat-style search modalities over proprietary grounding data. When you create a search service, you'll work with the following capabilities:

- A search engine for full text and vector search over a search index
- Rich indexing, with integrated data chunking and vectorization (preview), lexical analysis for text, and optional AI enrichment for content extraction and transformation
- Rich query syntax for vector queries, text search, hybrid search, fuzzy search, autocomplete, geo-search and others
- Azure scale, security, and reach
- Azure integration at the data layer, machine learning layer, Azure AI services and Azure OpenAI

Create a search service

Architecturally, a search service sits between the external data stores that contain your un-indexed data, and your client app that sends query requests to a search index and handles the response.



In your client app, the search experience is defined using APIs from Azure AI Search, and can include relevance tuning, semantic ranking, autocomplete, synonym matching, fuzzy matching, pattern matching, filter, and sort.

Across the Azure platform, Azure AI Search can integrate with other Azure services in the form of *indexers* that automate data ingestion/retrieval from Azure data sources, and *skillsets* that incorporate consumable AI from Azure AI services, such as image and natural language processing, or custom AI that you create in Azure Machine Learning or wrap inside Azure Functions.

## Inside a search service

On the search service itself, the two primary workloads are *indexing* and *querying*.

- **Indexing** is an intake process that loads content into your search service and makes it searchable. Internally, inbound text is processed into tokens and stored in inverted indexes, and inbound vectors are stored in vector indexes. The document format that Azure AI Search can index is JSON. You can upload JSON documents that you've assembled, or use an indexer to retrieve and serialize your data into JSON.

  AI enrichment through cognitive skills is an extension of indexing. If you have images or large unstructured text in source document, you can attach skills that perform OCR, describe images, infer structure, translate text and more. You can also attach skills that perform data chunking and vectorization.

- **Querying** can happen once an index is populated with searchable content, when your client app sends query requests to a search service and handles responses. All query execution is over a search index that you control.

  Semantic ranking is an extension of query execution. It adds language understanding to search results processing, promoting the most semantically relevant results to the top.

## Why use Azure AI Search?

Azure AI Search is well suited for the following application scenarios:

- Search over your vector and text content, isolated from the internet.

- Consolidate heterogeneous content into a user-defined and populated search index composed of vectors and text.

- Integrate data chunking and vectorization for generative AI and RAG apps.

- Apply granular access control at the document level.

- Offload indexing and query workloads onto a dedicated search service.

- Easily implement search-related features: relevance tuning, faceted navigation, filters (including geo-spatial search), synonym mapping, and autocomplete.

- Transform large undifferentiated text or image files, or application files stored in Azure Blob Storage or Azure Cosmos DB, into searchable chunks. This is achieved during indexing through cognitive skills that add external processing from Azure AI.

- Add linguistic or custom text analysis. If you have non-English content, Azure AI Search supports both Lucene analyzers and Microsoft's natural language processors. You can also configure analyzers to achieve specialized processing of raw content, such as filtering out diacritics, or recognizing and preserving patterns in strings.

For more information about specific functionality, see Features of Azure AI Search

## How to get started

Functionality is exposed through the Azure portal, simple REST APIs, or Azure SDKs like the Azure SDK for .NET. The Azure portal supports service administration and content management, with tools for prototyping and querying your indexes and skillsets.

An end-to-end exploration of core search features can be accomplished in four steps:

1. **Decide on a tier** and region. One free search service is allowed per subscription. All quickstarts can be completed on the free tier. For more capacity and capabilities, you'll need a billable tier.

2. **Create a search service** in the Azure portal.

3. **Start with Import data wizard**. Choose a built-in sample or a supported data source to create, load, and query an index in minutes.

4. **Finish with Search Explorer**, using a portal client to query the search index you just created.

Alternatively, you can create, load, and query a search index in atomic steps:

1. **Create a search index** using the portal, REST API, .NET SDK, or another SDK. The index schema defines the structure of searchable content.

2. **Upload content** using the "push" model to push JSON documents from any source, or use the "pull" model (indexers) if your source data is of a supported type.

3. **Query an index** using Search explorer in the portal, REST API, .NET SDK, or another SDK.

💡 **Tip**

For help with complex or custom solutions, **contact a partner** with deep expertise in Azure AI Search technology.

## Compare search options

Customers often ask how Azure AI Search compares with other search-related solutions. The following table summarizes key differences.

| Compared to | Key differences |
|---|---|
| Microsoft Search | Microsoft Search is for Microsoft 365 authenticated users who need to query over content in SharePoint. Azure AI Search pulls in content across Azure and any JSON dataset. |
| Bing | Bing APIs query the indexes on Bing.com for matching terms. Azure AI Search searches over indexes populated with your content. You control data ingestion and the schema. |
| Database search | SQL Server has full text search and Azure Cosmos DB and similar technologies have queryable indexes. Azure AI Search becomes an attractive alternative when you need features like lexical analyzers and relevance tuning, or content from heterogeneous sources. Resource utilization is another inflection point. Indexing and queries are computationally intensive. Offloading search from the DBMS preserves system resources for transaction processing. |
| Dedicated search solution | Assuming you've decided on dedicated search with full spectrum functionality, a final categorical comparison is between search technologies. Among cloud providers, Azure AI Search is strongest for vector, keyword, and hybrid workloads over content on Azure, for apps that rely primarily on search for both information retrieval and content navigation. |

Key strengths include:

- Relevance tuning through semantic ranking and scoring profiles.
- Data integration (crawlers) at the indexing layer.
- Azure AI integration for transformations that make content text and vector searchable.
- Microsoft Entra security for trusted connections, and Azure Private Link for private connections in no-internet scenarios.
- Full search experience: Linguistic and custom text analysis in 56 languages. Faceting, autocomplete queries and suggested results, and synonyms.

- Azure scale, reliability, and global reach.

# What's new in Azure AI Search

Article • 02/21/2024

**Azure Cognitive Search is now Azure AI Search**. Learn about the latest updates to Azure AI Search functionality, docs, and samples.

## February 2024

⛶ Expand table

| Item | Type | Description |
|---|---|---|
| **New dimension limits** | Feature | For vector fields, maximum dimension limits are now `3072`, up from `2048`. Next-generation embedding models support more dimensions. Limits have been increased accordingly. |

## November 2023

⛶ Expand table

| Item | Type | Description |
|---|---|---|
| **Vector search, generally available** | Feature | Vector search is now supported for production workloads. The previous restriction on customer-managed keys (CMK) is now lifted. Prefiltering and exhaustive K-nearest neighbor algorithm are also now generally available. |
| **Semantic ranking, generally available** | Feature | Semantic ranking (formerly known as "semantic search") is now supported for production workloads. |
| **Integrated vectorization (preview)** | Feature | Adds data chunking and text-to-vector conversions during indexing, and also adds text-to-vector conversions at query time. |
| **Import and vectorize data wizard (preview)** | Feature | A new wizard in the Azure portal that automates data chunking and vectorization. It targets the 2023-10-01-Preview REST API. |
| **Index projections (preview)** | Feature | A component of a skillset definition that defines the shape of a secondary index. Index projections are used for a one-to-many index pattern, where content from an enrichment pipeline can target multiple indexes. You can define index projections using the 2023-10-01-Preview REST API, the |

| Item | Type | Description |
|---|---|---|
| | | Azure portal, and any Azure SDK beta packages that are updated to use this feature. |
| 2023-11-01 Search REST API | API | New stable version of the Search REST APIs for vector fields, vector queries, and semantic ranking. See Upgrade REST APIs for migration steps to generally available features. |
| 2023-11-01 Management REST API | API | New stable version of the Management REST APIs for control plane operations. This version adds APIs that enable or disable semantic ranking. |
| Azure OpenAI Embedding skill (preview) | Skill | Connects to a deployed embedding model on your Azure OpenAI resource to generate embeddings during skillset execution. This skill is available through the 2023-10-01-Preview REST API, the Azure portal, and any Azure SDK beta packages that are updated to use this feature. |
| Text Split skill (preview) | Skill | Updated in 2023-10-01-Preview to support native data chunking. |
| How vector search and semantic ranking improve your GPT prompts | Video | Watch this short video to learn how hybrid retrieval gives you optimal grounding data for generating useful AI responses and enables search over both concepts and keywords. |
| Access Control in Generative AI applications | Blog | Explains how to use Microsoft Entra ID and Microsoft Graph API to roll out granular user permissions on chunked content in your index. |

ⓘ Note

Looking for preview features? Previews are announced here, but we also maintain a preview features list so you can find them in one place.

## October 2023

Expand table

| Item | Type | Description |
|---|---|---|
| "Chat with your data" solution accelerator | Sample | End-to-end RAG pattern that uses Azure AI Search as a retriever. It provides indexing, data chunking, orchestration and chat based on Azure OpenAI GPT. |

| Item | Type | Description |
|---|---|---|
| Exhaustive K-Nearest Neighbors (KNN) | Feature | Exhaustive K-Nearest Neighbors (KNN) is a new scoring algorithm for similarity search in vector space. It performs an exhaustive search for the nearest neighbors, useful for situations where high recall is more important than query performance. Available in the 2023-10-01-Preview REST API only. |
| Prefilters in vector search | Feature | Evaluates filter criteria before query execution, reducing the amount of content that needs to be searched. Available in the 2023-10-01-Preview REST API only, through a new `vectorFilterMode` property on the query that can be set to `preFilter` (default) or `postFilter`, depending on your requirements. |
| 2023-10-01-Preview Search REST API | API | New preview version of the Search REST APIs that changes the definition for vector fields and vector queries. This API version introduces breaking changes from **2023-07-01-Preview**, otherwise it's inclusive of all previous preview features. We recommend creating new indexes for **2023-10-01-Preview**. You might encounter an HTTP 400 on some features on a migrated index, even if you migrated correctly. |

## August 2023

Expand table

| Item | Type | Description |
|---|---|---|
| Enhanced semantic ranking | Feature | Upgraded models are rolling out for semantic reranking, and availability is extended to more regions. Maximum unique token counts doubled from 128 to 256. |

## July 2023

Expand table

| Item | Type | Description |
|---|---|---|
| Vector demo (Azure SDK for JavaScript) | Sample | Uses Node.js and the **@azure/search-documents 12.0.0-beta.2** library to generate embeddings, create and load an index, and run several vector queries. |
| Vector demo (Azure SDK for .NET) | Sample | Uses the **Azure.Search.Documents 11.5.0-beta.3** library to generate embeddings, create and load an index, and run |

| Item | Type | Description |
|---|---|---|
| | | several vector queries. You can also try this sample from the Azure SDK team. |
| Vector demo (Azure SDK for Python) | Sample | Uses the latest beta release of the **azure.search.documents** to generate embeddings, create and load an index, and run several vector queries. Visit the azure-search-vector-samples/demo-python repo for more vector search demos. |

## June 2023

Expand table

| Item | Type | Description |
|---|---|---|
| Vector search public preview | Feature | Adds vector fields to a search index for similarity search over vector representations of data. |
| 2023-07-01-Preview Search REST API | API | New preview version of the Search REST APIs that adds support for vector search. This API version is inclusive of all preview features. If you're using earlier previews, switch to **2023-07-01-preview** with no loss of functionality. |
| Semantic search availability | Feature | Semantic search is now available on the Basic tier. |

## May 2023

Expand table

| Item | Type | Description |
|---|---|---|
| Azure RBAC (role-based access control) | Feature | Announcing general availability. |
| 2022-09-01 Management REST API | API | New stable version of the Management REST APIs, with support for configuring search to use Azure RBAC. The **Az.Search** module of Azure PowerShell and **Az search** module of the Azure CLI are updated to support search service authentication options. You can also use the Terraform provider to configure authentication options (see this Terraform quickstart for details). |

## April 2023

⛶ Expand table

| Item | Type | Description |
|---|---|---|
| Multi-region deployment of Azure AI Search for business continuity and disaster recovery | Sample | Deployment scripts that fully configure a multi-regional solution for Azure AI Search, with options for synchronizing content and request redirection if an endpoint fails. |

# March 2023

⛶ Expand table

| Item | Type | Description |
|---|---|---|
| ChatGPT + Enterprise data with Azure OpenAI and Azure AI Search (GitHub) | Sample | Python code and a template for combining Azure AI Search with the large language models in OpenAI. For background, see this Tech Community blog post: Revolutionize your Enterprise Data with ChatGPT .<br>Key points:<br>Use Azure AI Search to consolidate and index searchable content.<br>Query the index for initial search results.<br>Assemble prompts from those results and send to the gpt-35-turbo (preview) model in Azure OpenAI.<br>Return a cross-document answer and provide citations and transparency in your customer-facing app so that users can assess the response. |

# 2022 announcements

⛶ Expand table

| Month | Item |
|---|---|
| November | **Add search to websites** series, updated versions of React and Azure SDK client libraries:<br>• C#<br>• Python<br>• JavaScript |

| Month | Item |
|---|---|
| | "Add search to websites" is a tutorial series with sample code available in three languages. If you're integrating client code with a search index, these samples demonstrate an end-to-end approach to integration. |
| November | **Retired** - Visual Studio Code extension for Azure AI Search ↗. |
| November | Query performance dashboard ↗. This Application Insights sample demonstrates an approach for deep monitoring of query usage and performance of an Azure AI Search index. It includes a JSON template that creates a workbook and dashboard in Application Insights and a Jupyter Notebook that populates the dashboard with simulated data. |
| October | Compliance risk analysis using Azure AI Search. On Azure Architecture Center, this guide covers the implementation of a compliance risk analysis solution that uses Azure AI Search. |
| October | Beiersdorf customer story using Azure AI Search ↗. This customer story showcases semantic search and document summarization to provide researchers with ready access to institutional knowledge. |
| September | Event-driven indexing for Azure AI Search ↗. This C# sample is an Azure Function app that demonstrates event-driven indexing in Azure AI Search. If you've used indexers and skillsets before, you know that indexers can run on demand or on a schedule, but not in response to events. This demo shows you how to set up an indexing pipeline that responds to data update events. |
| August | Tutorial: Index large data from Apache Spark. This tutorial explains how to use the SynapseML open-source library to push data from Apache Spark into a search index. It also shows you how to make calls to Azure AI services to get AI enrichment without skillsets and indexers. |
| June | Semantic search (preview). New support for Storage Optimized tiers (L1, L2). |
| June | **General availability** - Debug Sessions. |
| May | **Retired** - Power Query connector preview. |
| February | Index aliases. An index alias is a secondary name that can be used to refer to an index for querying, indexing, and other operations. When index names change, for example if you version the index, instead of updating the references to an index name in your application, you can just update the mapping for your alias. |

# Previous year's announcements

- 2021 announcements
- 2020 announcements
- 2019 announcements

## Service rebrand

This service has had multiple names over the years. Here they are in reverse chronological order:

- **Azure AI Search** (November 2023) Renamed to align with Azure AI services and customer expectations.
- **Azure Cognitive Search** (October 2019) Renamed to reflect the expanded (yet optional) use of cognitive skills and AI processing in service operations.
- **Azure Search** (March 2015) The original name.

## Service updates

Service update announcements for Azure AI Search can be found on the Azure web site.

## Feature rename

Semantic search was renamed to semantic ranking in November 2023 to better describe the feature, which provides L2 ranking of an existing result set.

# Features of Azure AI Search

Article • 12/12/2023

Azure AI Search provides information retrieval and uses optional AI integration to extract more text and structure content.

The following table summarizes features by category. For more information about how Azure AI Search compares with other search technologies, see Compare search options.

There's feature parity in all Azure public, private, and sovereign clouds, but some features aren't supported in specific regions. For more information, see product availability by region.

> ⓘ **Note**
>
> Looking for preview features? See the **preview features list**.

## Indexing features

Expand table

| Category | Features |
|---|---|
| Data sources | Search indexes can accept text from any source, provided it's submitted as a JSON document.<br>**Indexers** are a feature that automates data import from supported data sources to extract searchable content in primary data stores. Indexers handle JSON serialization for you and most support some form of change and deletion detection. You can connect to a variety of data sources, including Azure SQL Database, Azure Cosmos DB, or Azure Blob storage. |
| Hierarchical and nested data structures | **Complex types** and collections allow you to model virtually any type of JSON structure within a search index. One-to-many and many-to-many cardinality can be expressed natively through collections, complex types, and collections of complex types. |
| Linguistic analysis | Analyzers are components used for text processing during indexing and search operations. By default, you can use the general-purpose Standard Lucene analyzer, or override the default with a language analyzer, a custom analyzer that you configure, or another predefined analyzer that produces tokens in the format you require. |

| Category | Features |
|---|---|
| | **Language analyzers** from Lucene or Microsoft are used to intelligently handle language-specific linguistics including verb tenses, gender, irregular plural nouns (for example, 'mouse' vs. 'mice'), word de-compounding, word-breaking (for languages with no spaces), and more.<br>**Custom lexical analyzers** are used for complex query forms such as phonetic matching and regular expressions. |

# Vector and hybrid search

Expand table

| Category | Features |
|---|---|
| Vector indexing | Within a search index, add vector fields to support **vector search** scenarios. Vector fields can co-exist with nonvector fields in the same search document. |
| Vector queries | Formulate single and multiple vector queries. |
| Vector search algorithms | Use Hierarchical Navigable Small World (HNSW) or exhaustive K-Nearest Neighbors (KNN) to find similar vectors in a search index. |
| Vector filters | Apply filters before or after query execution for greater precision during information retrieval. |
| Hybrid information retrieval | Search for concepts and keywords in a single hybrid query request.<br>**Hybrid search** consolidates vector and text search, with optional semantic ranking and relevance tuning for best results. |
| Integrated data chunking and vectorization (preview) | Native data chunking through Text Split skill and native vectorization through vectorizers and the AzureOpenAIEmbeddingModel skill.<br>**Integrated vectorization** (preview) provides an end-to-end indexing pipeline from source files to queries. |
| **Import and vectorize data** (preview) | A new wizard in the Azure portal that creates a full indexing pipeline that includes data chunking and vectorization. The wizard creates all of the objects and configuration settings. |

## AI enrichment and knowledge mining

⛶ Expand table

| Category | Features |
|---|---|
| AI processing during indexing | **AI enrichment** refers to embedded image and natural language processing in an indexer pipeline that extracts text and information from content that can't otherwise be indexed for full text search. AI processing is achieved by adding and combining skills in a skillset, which is then attached to an indexer. AI can be either built-in skills from Microsoft, such as text translation or Optical Character Recognition (OCR), or custom skills that you provide. |
| Storing enriched content for analysis and consumption in non-search scenarios | **Knowledge store** is persistent storage of enriched content, intended for non-search scenarios like knowledge mining and data science processing. A knowledge store is defined in a skillset, but created in Azure Storage as objects or tabular rowsets. |
| Cached enrichments | **Incremental enrichment (preview)** refers to cached enrichments that can be reused during skillset execution. Caching is particularly valuable in skillsets that include OCR and image analysis, which are expensive to process. |

## Query and user experience

⛶ Expand table

| Category | Features |
|---|---|
| Free-form text search | **Full-text search** is a primary use case for most search-based apps. Queries can be formulated using a supported syntax.<br>**Simple query syntax** provides logical operators, phrase search operators, suffix operators, precedence operators.<br>**Full Lucene query syntax** includes all operations in simple syntax, with extensions for fuzzy search, proximity search, term boosting, and regular expressions. |
| Relevance | **Simple scoring** is a key benefit of Azure AI Search. Scoring profiles are used to model relevance as a function of values in the documents themselves. For example, you might want newer products or discounted products to appear higher in the search results. You can also build scoring profiles using tags for |

| Category | Features |
|---|---|
| | personalized scoring based on customer search preferences you've tracked and stored separately.<br>Semantic ranker is premium feature that reranks results based on semantic relevance to the query. Depending on your content and scenario, it can significantly improve search relevance with almost minimal configuration or effort. |
| Geospatial search | Geospatial functions filter over and match on geographic coordinates. You can match on distance or by inclusion in a polygon shape. |
| Filters and facets | Faceted navigation is enabled through a single query parameter. Azure AI Search returns a faceted navigation structure you can use as the code behind a categories list, for self-directed filtering (for example, to filter catalog items by price-range or brand).<br>Filters can be used to incorporate faceted navigation into your application's UI, enhance query formulation, and filter based on user- or developer-specified criteria. Create filters using the OData syntax. |
| User experience | Autocomplete can be enabled for type-ahead queries in a search bar.<br>Search suggestions also works off of partial text inputs in a search bar, but the results are actual documents in your index rather than query terms.<br>Synonyms associates equivalent terms that implicitly expand the scope of a query, without the user having to provide the alternate terms.<br>Hit highlighting applies text formatting to a matching keyword in search results. You can choose which fields return highlighted snippets.<br>Sorting is offered for multiple fields via the index schema and then toggled at query-time with a single search parameter.<br>Paging and throttling your search results is straightforward with the finely tuned control that Azure AI Search offers over your search results. |

## Security features

Expand table

| Category | Features |
|---|---|
| Data encryption | Microsoft-managed encryption-at-rest is built into the internal storage layer and is irrevocable.<br>Customer-managed encryption keys that you create and manage in Azure Key Vault can be used for supplemental encryption of indexes and synonym maps. For services created after August 1 2020, CMK encryption extends to data on temporary disks, for full double encryption of indexed content. |
| Endpoint protection | IP rules for inbound firewall support allows you to set up IP ranges over which the search service will accept requests.<br>Create a private endpoint using Azure Private Link to force all requests through a virtual network. |
| Inbound access | Azure role-based access control assigns roles to users and groups in Microsoft Entra ID for controlled access to search content and operations. You can also use key-based authentication if you don't have an Azure tenant. |
| Outbound security (indexers) | Data access through private endpoints allows an indexer to connect to Azure resources that are protected through Azure Private Link.<br>Data access using a trusted identity means that connection strings to external data sources can omit user names and passwords. When an indexer connects to the data source, the resource allows the connection if the search service was previously registered as a trusted service. |

# Portal features

Expand table

| Category | Features |
|---|---|
| Tools for prototyping and inspection | Add index is an index designer in the portal that you can use to create a basic schema consisting of attributed fields and a few other settings. After saving the index, you can populate it using an SDK or the REST API to provide the data.<br>Import data wizard creates indexes, indexers, skillsets, and data source definitions. If your data exists in Azure, this wizard can save you significant time and effort, especially for proof-of-concept |

| Category | Features |
|---|---|
| | investigation and exploration.<br>Search explorer is used to test queries and refine scoring profiles.<br>Create demo app is used to generate an HTML page that can be used to test the search experience.<br>Debug Sessions is a visual editor that lets you debug a skillset interactively. It shows you dependencies, output, and transformations. |
| Monitoring and diagnostics | Enable monitoring features to go beyond the metrics-at-a-glance that are always visible in the portal. Metrics on queries per second, latency, and throttling are captured and reported in portal pages with no extra configuration required. |

# Programmability

Expand table

| Category | Features |
|---|---|
| REST | Service REST API is for data plane operations, including all operations related to indexing, queries, and AI enrichment. You can also use this client library to retrieve system information and statistics.<br>Management REST API is for service creation and provisioning through Azure Resource Manager. You can also use this API to manage keys and capacity. |
| Azure SDK for .NET | Azure.Search.Documents is for data plane operations, including all operations related to indexing, queries, and AI enrichment. You can also use this client library to retrieve system information and statistics.<br>Microsoft.Azure.Management.Search is for service creation and provisioning through Azure Resource Manager. You can also use this API to manage keys and capacity. |
| Azure SDK for Java | com.azure.search.documents is for data plane operations, including all operations related to indexing, queries, and AI enrichment. You can also use this client library to retrieve system information and statistics.<br>com.microsoft.azure.management.search is for service creation |

| Category | Features |
|---|---|
| | and provisioning through Azure Resource Manager. You can also use this API to manage keys and capacity. |
| Azure SDK for Python | azure-search-documents is for data plane operations, including all operations related to indexing, queries, and AI enrichment. You can also use this client library to retrieve system information and statistics.<br>azure-mgmt-search is for service creation and provisioning through Azure Resource Manager. You can also use this API to manage keys and capacity. |
| Azure SDK for JavaScript/TypeScript | azure/search-documents is for data plane operations, including all operations related to indexing, queries, and AI enrichment. You can also use this client library to retrieve system information and statistics.<br>azure/arm-search is for service creation and provisioning through Azure Resource Manager. You can also use this API to manage keys and capacity. |

## See also

- What's new in Azure AI Search
- Preview features in Azure AI Search

# Azure AI Search Frequently Asked Questions

FAQ

Find answers to commonly asked questions about Azure AI Search.

## General

### What is Azure AI Search?

Azure AI Search provides a dedicated search engine and persistent storage of your searchable content for full text and vector search scenarios. It also includes optional, integrated AI to extract more text and structure from raw content, and to chunk and vectorize content for vector search.

### How do I work with Azure AI Search?

The primary workflow is create, load, and query an index. Although you can use the portal for most tasks, Azure AI Search is intended to be used programmatically, handling requests from client code. Programmatic support is provided through REST APIs and client libraries in .NET, Python, Java, and JavaScript SDKs for Azure.

### Are "Azure Search" and "Azure Cognitive Search" and "Azure AI Search" the same product?

Azure Search was renamed to Azure Cognitive Search in October 2019 to reflect the expanded (yet optional) use of cognitive skills and AI processing in service operations. Azure Cognitive Search was renamed to Azure AI Search in October 2023 to align with Azure AI services.

### What languages are supported?

The default analyzer used for tokenization is standard Lucene and it is language agnostic. Otherwise, language support is expressed through language analyzers that apply linguistic rules to inbound (indexing) and outbound (queries) content. Some features, such as speller, are limited to a subset of languages.

## How do I integrate search into my solution?

Client code should call the client libraries or REST APIs to connect to a search index, formulate queries, and handle responses. You can also write code that builds and refreshes an index, or runs indexers programmatically or by script.

## Is there functional parity across the various APIs?

Not always. The REST API is always the first to implement new features in preview API versions. The client libraries in Azure SDKs will pick up new features over time, but are released on their own schedule.

Although the REST APIs are first out with newest features, the Azure SDKs provide more coding support, and are recommended over REST unless a required feature is unavailable.

## Can I pause the service and stop billing?

You can't pause a search service. In Azure AI Search, computing resources are allocated when the service is created. It's not possible to release and reclaim those resources on-demand.

## Can I upgrade, downgrade, rename or move the service?

Service tier, name, and region are fixed for the lifetime of the service.

## If I migrate my search service to another subscription or resource group, should I expect any downtime?

As long as you follow the checklist before moving resources and make sure each step is completed, there shouldn't be any downtime.

# Indexing

## What does "indexing" mean in Azure AI Search?

It refers to the ingestion, parsing, and storing of textual content and tokens that populate a search index. Indexing creates inverted indexes and other physical data structures that support information retrieval. It creates vector indexes if the schema includes vector fields.

## Can I move, backup, and restore indexes?

There's no native support for porting indexes. Search indexes are considered downstream data structures, accepting content from other data sources that collect operational data. As such, there's no built-in support for backing up and restoring indexes because the expectation is that you would rebuild an index from source data if you deleted it, or wanted to move it.

However, if you want to move an index between search services, you can try the **index-backup-restore** sample code in this Azure AI Search .NET sample repo ↗. There's also a Python version of backup and restore ↗.

## Can I restore my index or service once it's deleted?

No, if you delete an Azure AI Search index or service, it can't be recovered. When you delete a search service, all indexes in the service are deleted permanently.

## Can I index from SQL Database replicas?

If you're using the search indexer for Azure SQL Database, there are no restrictions on the use of primary or secondary replicas as a data source when building an index from scratch. However, refreshing an index with incremental updates (based on changed records) requires the primary replica. This requirement comes from SQL Database, which guarantees change tracking on primary replicas only. If you try using secondary replicas for an index refresh workload, there's no guarantee you get all of the data.

# Vectors

## What is vector search?

Vector search is a technique that finds the most similar documents by comparing their vector representations. Since the goal of a vector representation is to capture the essential characteristics of an item in a numerical format, vector queries can identify similar content even if there are no explicit matches based on keywords or tags. When a user performs a search, the query is summarized into a vector representation and the vector search engine identifies the most similar documents. To improve efficiency on large databases, vector search often provides the approximate nearest neighbors for a query vector. See Vector search overview for the specifics of Azure AI Search's vector offering.

## Does Azure AI Search support vector search?

Azure AI Search supports vector indexing and retrieval. It can vectorize query strings and content if you use the preview and beta libraries.

## How does vector search work in Azure AI Search?

With standalone vector search, you first use an embedding model to transform content into a vector representation within an embedding space. You can then provide these vectors in a document payload to the search index for indexing. To serve search requests, you use the same DNN from indexing to transform the search query into a vector representation, and vector search finds the most similar vectors and return the corresponding documents.

In Azure AI Search, you can index vector data as fields in documents alongside textual and other types of content. The data type for a vector field is `Collection(Edm.Single)`.

Vector queries can be issued standalone or in combination with other query types, including term queries and filters in the same search request.

## Can Azure AI Search vectorize my content or queries?

Integrated vectorization is now in public preview.

## Does my search service support vector search?

Most existing services support vector search. If you're using a package or API that supports vector search and index creation fails, the underlying search service doesn't support vector search, and a new service must be created. This can occur for a small subset of services created prior to January 1, 2019.

## Can I add vector search to an existing index?

If your search service supports vector search, both existing and new indexes can accommodate vector fields.

## Why do I see different vector index size limits between my new search services and existing search services?

We're rolling out improved vector index size limits worldwide for new search services, but we're still building out infrastructure capacity in certain regions. New search services created in supported regions will see increased vector index size limits. Unfortunately, we can't migrate existing services to the new limits.

## How do I enable vector search on a search index?

To enable vector search in an index, you should:

- Add one or more vector fields to a field collection.
- Add a "vectorSearch" section to the index schema specifying the configuration used by vector search fields, including the parameters of the Approximate Nearest Neighbor algorithm used, like HNSW.
- Use 2023-11-01 or an Azure SDK to create or update the index, load documents, and issue queries.

# Queries

## Where does query execution occur?

Queries execute over a single search index that's hosted on your search service. You can't join multiple indexes to search content in two or more indexes, but you can query same-name indexes in multiple search services.

## Why are there zero matches on terms I know to be valid?

The most common case isn't knowing that each query type supports different search behaviors and levels of linguistic analyses. Full text search, which is the predominant workload, includes a language analysis phase that breaks down terms to root forms. This aspect of query parsing casts a broader net over possible matches, because the tokenized term matches a greater number of variants.

Wildcard, fuzzy and regex queries, however, aren't analyzed like regular term or phrase queries and can lead to poor recall if the query doesn't match the analyzed form of the word in the search index. For more information on query parsing and analysis, see query architecture.

## Why are my wildcard searches slow?

Most wildcard search queries, like prefix, fuzzy and regex, are rewritten internally with matching terms in the search index. This extra processing adds to latency. Further, broad search queries, like `a*` for example, are likely to be rewritten with many terms, which can be slow. For performant wildcard searches, consider defining a custom analyzer.

## Can I search across multiple indexes?

No, a query is always scoped to a single index.

## Why is the search score a constant 1.0 for every match?

Search scores are generated for full text search queries, based on the statistical properties of matching terms, and ordered high to low in the result set. Query types that aren't full text search (wildcard, prefix, regex) aren't ranked by a relevance score. This behavior is by design. A constant score allow matches found through query expansion to be included in the results, without affecting the ranking.

For example, suppose an input of "tour*" in a wildcard search produces matches on "tours", "tourettes", and "tourmaline". Given the nature of these results, there's no way to reasonably infer which terms are more valuable than others. For this reason, term frequencies are ignored when scoring results in queries of types wildcard, prefix, and regex. Search results based on a partial input are given a constant score to avoid bias towards potentially unexpected matches.

# Security

## Where does Azure AI Search store customer data?

It stores your data wherever your service is deployed. Azure AI Search doesn't store customer data outside of the deployment region.

## Does Azure AI Search send customer data to other services for processing?

Yes, if you use the built-in skills based on Azure AI services, the indexer sends requests to Azure AI services over the internal network. If you add a custom skill, the indexer sends content to the URI provided in the custom skill over the public network.

## Can I control access to search results based on user identity?

Not exactly. Typically, users who are authorized to run your application are also authorized to see all search results. Azure AI Search doesn't have built-in support for row-level or document-level permissions, but you can implement security filters as a workaround.

## Can I control access to operations based on user identity?

Yes, you can use role-based authorization for data plane operations over content.

## Can I use the Azure portal to view and manage search content if the search service is behind an IP firewall or a private endpoint?

You can use the Azure portal on a network-protected search service if you create a network exception that allows client and portal access. For more information, see connect through an IP firewall or connect through a private endpoint.

## Next steps

If your question isn't answered here, you can refer to the following sources for more questions and answers.

Stack Overflow: Azure AI Search ↗

How full text search works in Azure AI Search

What is Azure AI Search?