Azure Cognitive Search documentation

Cloud search over private heterogeneous content, with options for AI enrichment if your content is unstructured or unsearchable in raw form.

About Cognitive Search

OVERVIEW

What is Azure Cognitive Search?

What is semantic search?

TRAINING

Learning module

10-minute quickstarts

QUICKSTART

Create a search service

Create a search index

Create a demo app

Create a skillset

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

Index data

OVERVIEW

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

Query data

CONCEPT

Query types and composition

Simple syntax (default)

Full Lucene syntax

HOW-TO GUIDE

Create a simple query

Create advanced queries

REFERENCE

OData language reference

Search Documents (REST)

Enrich with AI

CONCEPT

What is AI enrichment?

Skillsets

Knowledge stores

What's Azure Cognitive Search?

Article • 12/02/2022 • 7 minutes to read

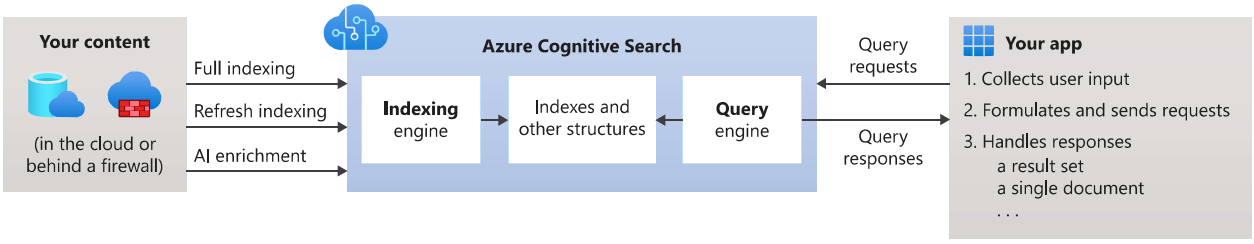
Azure Cognitive Search (formerly known as "Azure Search") is a cloud search service that gives developers infrastructure, APIs, and tools for building a rich search experience over private, heterogeneous content in web, mobile, and enterprise applications.

Search is foundational to any app that surfaces text to users, where common scenarios include catalog or document search, online retail apps, or data exploration over proprietary content. When you create a search service, you'll work with the following capabilities:

- A search engine for full text search over a search index containing user-owned content
- Rich indexing, with lexical analysis and optional AI enrichment for content extraction and transformation
- Rich query syntax for text search, fuzzy search, autocomplete, geo-search and more
- Programmability through REST APIs and client libraries in Azure SDKs
- Azure integration at the data layer, machine learning layer, and AI (Cognitive Services)

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 Cognitive Search, and can include relevance tuning, semantic ranking, autocomplete, synonym matching, fuzzy matching, pattern matching, filter, and sort.

Across the Azure platform, Cognitive 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 Cognitive 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 for fast scans. You can upload JSON documents, or use an indexer to serialize your data into JSON.

  AI enrichment through cognitive skills is an extension of indexing. If your content needs image or language analysis before it can be indexed, AI enrichment can extract text embedded in application files, translate text, and also infer text and structure from non-text files by analyzing the content.

- **Querying** can happen once an index is populated with searchable text, 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 search 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 Cognitive Search?

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

- Consolidate heterogeneous content into a private, user-defined search index.
- 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.
- Add linguistic or custom text analysis. If you have non-English content, Azure Cognitive 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 Cognitive 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 Cognitive Search technology.

Compare search options

Customers often ask how Azure Cognitive 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. It's offered as a ready-to-use search experience, enabled and configured by administrators, with the ability to accept external content through connectors from Microsoft and other sources. If this describes your scenario, then Microsoft Search with Microsoft 365 is an attractive option to explore. In contrast, Azure Cognitive Search executes queries over an index that you define, populated with data and documents you own, often from diverse sources. Azure Cognitive Search has crawler capabilities for some Azure data sources through indexers, but you can push any JSON document that conforms to your index schema into a single, consolidated searchable resource. You can also customize the indexing pipeline to include machine learning and lexical analyzers. Because Cognitive Search is built to be a plug-in component in larger solutions, you can integrate search into almost any app, on any platform.
Bing	Bing Web Search API searches the indexes on Bing.com for matching terms you submit. Indexes are built from HTML, XML, and other web content on public sites. Built on the same foundation, Bing Custom Search offers the same crawler technology for web content types, scoped to individual web sites. In Cognitive Search, you can define and populate the index. You can use indexers to crawl data on Azure data sources, or push any index-conforming JSON document to your search service.

Compared to	Key differences
Database search	Many database platforms include a built-in search experience. SQL Server has full text search. Azure Cosmos DB and similar technologies have queryable indexes. When evaluating products that combine search and storage, it can be challenging to determine which way to go. Many solutions use both: DBMS for storage, and Azure Cognitive Search for specialized search features. Compared to DBMS search, Azure Cognitive Search stores content from heterogeneous sources and offers specialized text processing features such as linguistic-aware text processing (stemming, lemmatization, word forms) in 56 languages. It also supports autocorrection of misspelled words, synonyms, suggestions, scoring controls, facets, and custom tokenization. The full text search engine in Azure Cognitive Search is built on Apache Lucene, an industry standard in information retrieval. However, while Azure Cognitive Search persists data in the form of an inverted index, it isn't a replacement for true data storage and we don't recommend using it in that capacity. For more information, see this forum post. Resource utilization is another inflection point in this category. Indexing and some query operations are often computationally intensive. Offloading search from the DBMS to a dedicated solution in the cloud preserves system resources for transaction processing. Furthermore, by externalizing search, you can easily adjust scale to match query volume.
Dedicated search solution	Assuming you've decided on dedicated search with full spectrum functionality, a final categorical comparison is between on premises solutions or a cloud service. Many search technologies offer controls over indexing and query pipelines, access to richer query and filtering syntax, control over rank and relevance, and features for self-directed and intelligent search. A cloud service is the right choice if you want a turn-key solution with minimal overhead and maintenance, and adjustable scale. Within the cloud paradigm, several providers offer comparable baseline features, with full-text search, geospatial search, and the ability to handle a certain level of ambiguity in search inputs. Typically, it's a specialized feature, or the ease and overall simplicity of APIs, tools, and management that determines the best fit.

Among cloud providers, Azure Cognitive Search is strongest for full text search workloads over content stores and databases on Azure, for apps that rely primarily on search for both information retrieval and content navigation.

Key strengths include:

- Data integration (crawlers) at the indexing layer.
- AI and machine learning integration with Azure Cognitive Services, useful if you need to make unsearchable content full text-searchable.

- Security integration with Azure Active Directory for trusted connections, and with Azure Private Link integration to support private connections to a search index in no-internet scenarios.
- Linguistic and custom text analysis in 56 languages.
- Full search experience: rich query language, relevance tuning and semantic ranking, faceting, autocomplete queries and suggested results, and synonyms.
- Azure scale, reliability, and world-class availability.

Among our customers, those able to apply the widest range of features in Azure Cognitive Search include online catalogs, line-of-business programs, and document discovery applications.

What's new in Azure Cognitive Search

Article • 05/15/2023

Learn about the latest updates to Azure Cognitive Search functionality, docs, and samples.

> **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.

May 2023

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

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

March 2023

Item	Type	Description

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

2022 announcements

Month	Item
November	**Add search to websites** series, updated versions of React and Azure SDK client libraries: • C# • Python • JavaScript "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 Cognitive Search.
November	Query performance dashboard. This Application Insights sample demonstrates an approach for deep monitoring of query usage and performance of an Azure Cognitive 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 Cognitive Search. Published on Azure Architecture Center, this guide covers the implementation of a compliance risk analysis solution that uses Azure Cognitive Search.

Month	Item
October	Beiersdorf customer story using Azure Cognitive Search. This customer story showcases semantic search and document summarization to provide researchers with ready access to institutional knowledge.
September	Azure Cognitive Search Lab. This C# sample provides the source code for building a web front-end that accesses all of the REST API calls against an index. This tool is used by support engineers to investigate customer support issues. You can try this demo site before building your own copy.
September	Event-driven indexing for Cognitive Search. This C# sample is an Azure Function app that demonstrates event-driven indexing in Azure Cognitive 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 Cognitive 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 re-brand

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.

Service updates

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

Features of Azure Cognitive Search

Article • 05/05/2023

Azure Cognitive Search provides a full-text search engine, persistent storage of search indexes, integrated AI used during indexing to extract more text and structure, and APIs and tools.

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

Indexing features

Category	Features
Data sources	Search indexes can accept text from any source, provided it's submitted as a JSON document. **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. **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. **Custom lexical analyzers** are used for complex query forms such as phonetic matching and regular expressions.

AI enrichment and knowledge mining

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

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. **Simple query syntax** provides logical operators, phrase search operators, suffix operators, precedence operators. **Full Lucene query syntax** includes all operations in simple syntax, with extensions for fuzzy search, proximity search, term boosting, and regular expressions.

Category	Features
Relevance	**Simple scoring** is a key benefit of Azure Cognitive 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 personalized scoring based on customer search preferences you've tracked and stored separately. **Semantic search (preview)** 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 Cognitive 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). **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.

Category	Features
User experience	Autocomplete can be enabled for type-ahead queries in a search bar. 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. Synonyms associates equivalent terms that implicitly expand the scope of a query, without the user having to provide the alternate terms. Hit highlighting applies text formatting to a matching keyword in search results. You can choose which fields return highlighted snippets. Sorting is offered for multiple fields via the index schema and then toggled at query-time with a single search parameter. Paging and throttling your search results is straightforward with the finely tuned control that Azure Cognitive Search offers over your search results.

Security features

Category	Features
Data encryption	Microsoft-managed encryption-at-rest is built into the internal storage layer and is irrevocable. 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. Create a private endpoint using Azure Private Link to force all requests through a virtual network.

Category	Features
Inbound access	Azure role-based access control assigns roles to users and groups in Azure Active Directory 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. 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

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. 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 investigation and exploration. Search explorer is used to test queries and refine scoring profiles. Create demo app is used to generate an HTML page that can be used to test the search experience. 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

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. 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. 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. com.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 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. 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. 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 Cognitive Search
- Preview features in Cognitive Search

Cognitive Search Frequently Asked Questions

FAQ

Find answers to commonly asked questions about Azure Cognitive Search.

General

What is Azure Cognitive Search?

Azure Cognitive Search is a service on Azure that provides a dedicated search engine and persistent storage of your searchable content for full text search scenarios. It also includes optional, integrated AI used during indexing to extract more text and structure from raw content.

How do I work with Cognitive Search?

The primary workflow is create, load, and query an index. Although you can use the portal for most tasks, Cognitive 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" the same service?

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.

What languages are supported?

The default analyzer used for tokenization is standard Lucene and it's 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 semantic search and 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, and the generally available versions support all programmatic operations. 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 Cognitive 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 Cognitive Search?

It refers to the ingestion, parsing, and storing of textual content and tokens that populate a search index. Indexing creates inverted indices and other physical data

structures that support information retrieval.

Can I move, backup, and restore indexes?

There's no native support for index management. 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 Cognitive Search .NET sample repo.

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

No, if you delete an Azure Cognitive 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.

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 indices 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, that are likely to be rewritten with many terms 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 Cognitive Search store customer data?

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

Does Cognitive Search send customer data to other services for processing?

Yes, if you use the built-in skills based on Cognitive Services, the indexer sends requests to Cognitive Services over the internal network. If you add a custom skill, the indexer will send 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. Cognitive 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 Cognitive Search

How full text search works in Azure Cognitive Search

What is Azure Cognitive Search?