

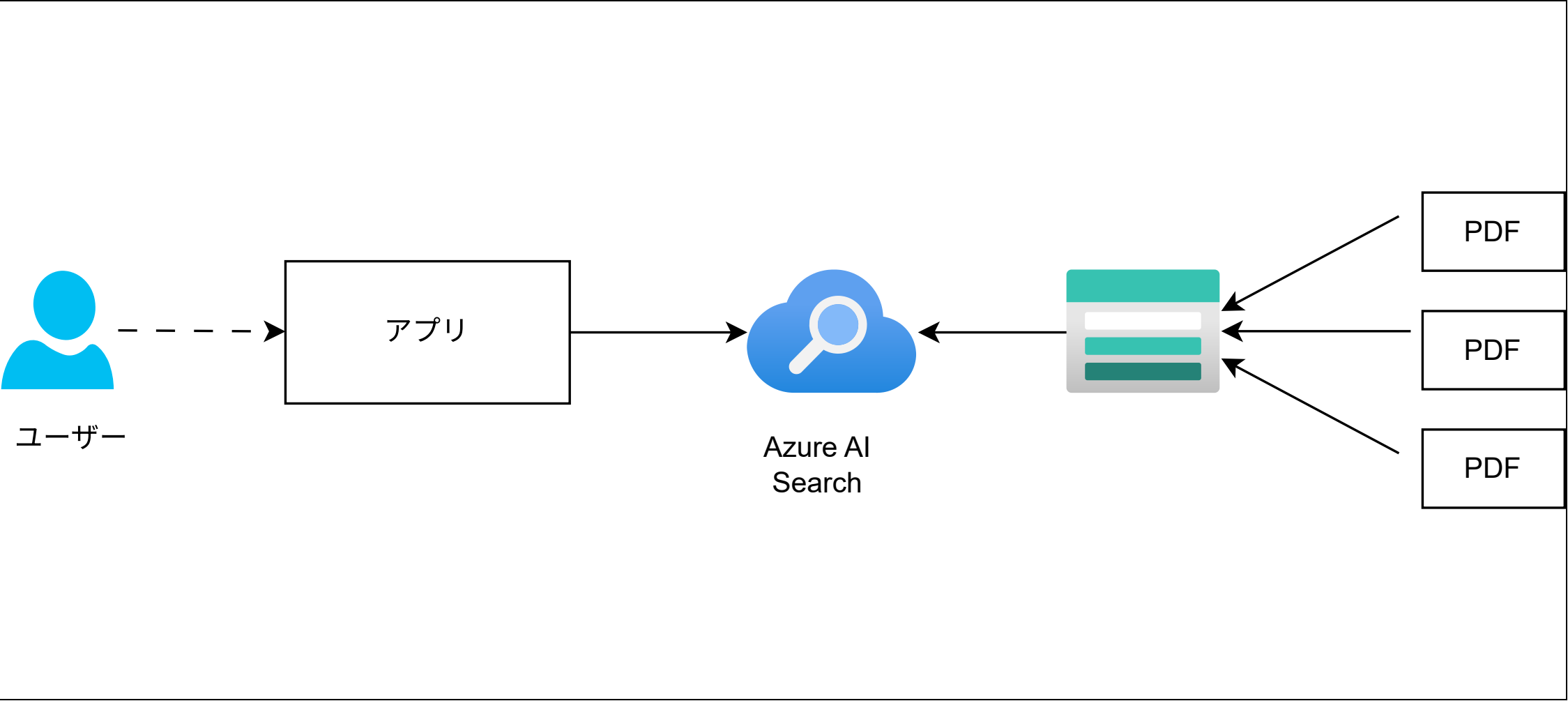
Azure AI Search

- 公式ドキュメント
- 大量のデータに対する高速な検索を実現するためのサービス
- 旧名称「Azure Cognitive Search」「Azure Search」
- さまざまな「データソース」（データの格納場所）に対応
- サーバー、ネットワーク、ストレージなどの設計、運用、メンテナンスはマイクロソフトが実施。

AIエンリッチメント

- AIを使用したデータ処理（インデックス作成）が可能（「AIエンリッチメント」）
 - PDFに含まれる画像から「画像のキャプション」（説明文）を生成することで、画像（の説明文）に対するキーワード検索が可能
 - PDFに含まれる文章から「エンティティ抽出」により価格などを認識し、抽出することで「5,000円以上, 10,000円以下」といった価格の商品データの検索などが可能

Azure AI Search の運用イメージ



クエリ

Azure AI Searchは、通常、アプリの内部での「クエリ」（検索）の実行に利用される。

- データの検索を行うことを「クエリ」（query、問い合わせ）という
- 開発者はユーザー向けのアプリ（Webアプリ、モバイルアプリなど）を開発
- ユーザーがアプリに検索キーワードや検索条件を入力
 - 例: 東京在住の「山本」さん
- アプリはAzure AI SearchのAPIに、クエリの実行を依頼
- Azure AI Searchはクエリを実行
 - Azure AI Searchに接続された「データソース」（から作成した「インデックス」）を使用して検索を行う
- Azure AI Searchは検索結果（JSON）をアプリに返す

主なデータソース(1)

Azure AI Search で利用可能なデータソース

※データソース = Azure AI Searchで検索できるデータの置き場所

- ストレージアカウント (Blob, Table, Azure Files)
- Azure Data Lake Storage Gen2
- Azure SQL Database (Azure SQL Database, Azure SQL Database Managed Instance, SQL Server on VM)
- Azure Cosmos DB (for NoSQL, for MongoDB, for Apache Gremlin)
- SharePoint サイト (SharePoint Online)
- Fabric OneLake レイクハウス

主なデータソース(2)

サードパーティから提供される「**データソースコネクタ**」を使用することで、更に多くのデータソースに接続できる

- Amazon S3、Amazon RDS
- ファイルシステム（Windows, Linux, Mac）, Googleドライブ, OneDrive
- MySQL, Oracle, PostgreSQL
- Salesforce, ServiceNow
- SAP ERP, SAP HANA
- Twitter, Slack
- その他多数

サポートされる主なファイル・ドキュメント形式

※データソースがストレージアカウントの **Blob** や **Azure Files** の場合

- DOCX/DOC/DOCM、XLSX/XLS/XLSM、PPTX/PPT/PPTM、MSG
- PDF
- テキストファイル, HTML, CSV, XML
- [Markdown](#) (プレビュー)
- JSON
 - 1つのJSONファイルの全体を1データとして取り出す
 - 1つのJSONファイル内の配列に含まれる複数のデータを取り出す
- JSONL
 - 1つのJSONファイルから複数のデータを取り出す

データの例（複数の社員の情報）

ストレージアカウントのBlobコンテナーにJSONファイルとして保存する。

```
[
  {
    "Id": 1,
    "Name": "宮城 直樹",
    "Department": "開発部",
    "Profile": "C#が得意です。山口県出身で現在は東京に住んでいます",
    "Age": 35,
    "HireDate": "2015-04-01",
    "Location": "東京",
    "Certifications": ["Microsoft Certified Azure AI Engineer"],
    "Seniority": "上級",
    "Languages": ["日本語", "英語"],
    "RemoteWork": true
  },
  ...
]
```

インデックス

インデックス（検索インデックス、search indexes とも）

- Azure AI Searchでは、データソースのデータに対して事前に**インデックス**（索引）を作る
 - 大量のデータをあらかじめスキャンして、どのファイルのどの位置にどの単語が出現するか、という情報をあらかじめインデックスとして保存しておく。
 - すると、ユーザーがクエリを実行する際に、インデックスを利用して、検索結果をすばやく提供することができる。
- **インデックス**は Azure AI Search リソースの内部に置かれる
 - インデックスの最大サイズには制限があることに注意（後述）

インデックスの作成方法

プルモデルと**プッシュモデル**の2つの作成方法がある

- **プル モデル**: Azure AI Searchの中に「**インデクサー**」を作り、データソースのデータをスキャンし、インデックスを作る
 - インデクサー作成直後に初回のスキャンが実行される
 - その後のスキャンの**スケジュール設定**も可能。毎日、毎時、など。
- **プッシュ モデル**: **ユーザーのアプリケーション側のコードで、データをスキャンし、Azure AI Searchのインデックスを作成・更新する**

※プッシュモデルは比較的高度な方式（インデックスを書き込むプログラムコードの開発が必要）。ラボではプルモデルを使用。

インデクサーの作成

- Azure portalから比較的簡単にインデクサーを作成できる
- 基本的な流れ
 - データソースを選択
 - データの**フィールド属性**（各検索項目のデータ型や検索オプション）を設定
 - インデクサーの名前や実行スケジュールを設定

Microsoft Azure

リソース、サービス、ドキュメントの

ホーム > searchservice-1747232760262 | 概要 > ai902837425 >

データのインポート ...

データに接続します ログニティブ スキルを追加します (省略可能) *対象インデックスをカスタマイズします

インデクサー

名前 *

azureblob-indexer

スケジュール ⓘ

1 度 毎時間 毎日 カスタム

説明

(省略可能)

インデクサーの作成

検索エクスペローラー

検索エクスペローラー

- Azure portal に組み込まれているクエリ（検索）ツール
- 開発者向けのツール
 - 通常エンドユーザー向けには別途、検索画面を持つアプリを開発し、そこから検索を行ってもらう
- **クエリ ビュー** と **JSON ビュー** という2つの検索モードがある
 - クエリビューでは単純な全文検索を実行。つまり指定されたキーワードを含むデータを検索
 - JSONビューは、専用の形式を使用して、高度な検索条件の指定が可能

検索エクスプローラーの「クエリビュー」での検索の例

※ Name と Profile と Location に対して Searchable と設定されているとする

クエリ オプション 表示

宮城

検索

結果

```
1 {
2   "@odata.context": "https://ai902837425.search.windows.net/indexes('index1')/$metadata#docs(*)",
3   "@odata.count": 1,
4   "value": [
5     {
6       "@search.score": 2.367833,
7       "Id": 1,
8       "Name": "宮城 直樹",
9       "Department": "開発部",
10      "Profile": "C#が得意です。山口県出身で現在は東京に住んでいます",
11      "Age": 35,
12      "HireDate": "2015-04-01T00:00:00Z",
13      "Location": "東京",
14      "Certifications": [
15        "Microsoft Certified: Azure AI Engineer"
16      ],
17      "Seniority": "上級",
```

検索エクスプローラーの「JSON ビュー」での検索の例

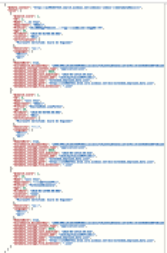
JSON クエリ エディター

```
1 {
2   "filter": "Certifications/any(c: c eq 'Microsoft Certified: Azure AI Engineer')"
```

検索

結果

```
1 {
2   "@odata.context": "https://ai902837425.search.windows.net/indexes('index1')/$metadata#docs(*)",
3   "value": [
4     {
5       "@search.score": 1,
6       "Id": 1,
7       "Name": "宮城 直樹",
8       "Department": "開発部",
9       "Profile": "c#が得意です。山口県出身で現在は東京に住んでいます",
10      "Age": 35,
11      "HireDate": "2015-04-01T00:00:00Z",
12      "Location": "東京",
13      "Certifications": [
14        "Microsoft Certified: Azure AI Engineer"
15      ],
16      "Seniority": "上級",
```



リソースの作成

Microsoft Azure

リソース、サービス、ドキュメントの検索 (G+)

ホーム > Azure AI services | AI Search >

検索サービスを作成する ...

基本情報

スケーリング

ネットワーク

タグ

確認して作成

プロジェクトの詳細

サブスクリプション *

MOC Subscription-lod50579408

リソース グループ *

ResourceGroup1

新規作成

インスタンスの詳細

サービス名 * ⓘ

aisearch0928734825

✓

場所 *

(US) Central US

価格レベル * ⓘ

Standard

160 GB/パーティション、最大 12 個のレプリカ、最大 12 個のパーティション、最大 36 個の検索ユニット

価格レベルの変更

価格レベルの選択

Microsoft Azure

リソース、サービス、ドキュメントの検索 (G+)

Copilot

User1-51283208@L...
LODS-PROD-MCA (LODSPR...

ホーム > Azure AI services | AI Search > 検索サービスを作成する >

価格レベルの選択 ...

利用可能な SKU とその機能を参照する

SKU	オファリング	インデックス	インデクサー	ベクター クォータ	ストレージの合計	検索単位	レプリカ	パーティション	検索ユニットのコスト/月 (推定)
F	Free	3	3	25 MB	50 MB	1	1	1	\$0.00
B	基本	15	15	5 GB/パーティション	15 GB/パーティション	9	3	3	\$75.14
S	Standard	50	50	35 GB/パーティション	160 GB/パーティション	36	12	12	\$249.98
S2	Standard	200	200	150 GB/パーティション	512 GB/パーティション	36	12	12	\$999.94
S3	Standard	200	200	300 GB/パーティション	1 TB/パーティション	36	12	12	\$1,999.87
S3HD	高密度	1000	0	300 GB/パーティション	1 TB/パーティション	36	12	3	\$1,999.87
L1	ストレージ最適化	10	10	150 GB/パーティション	2 TB/パーティション	36	12	12	\$2,856.22
L2	ストレージ最適化	10	10	300 GB/パーティション	4 TB/パーティション	36	12	12	\$5,711.69

このリージョンの新規サービスでは、追加コストなしでより高いストレージ上限を利用できます。

Select

示されている価格は、現地通貨での推定価格であり、Azure インフラストラクチャのコストと、サブスクリプションおよび場所に対する割引のみが含まれます。価格には、適用されるソフトウェアのコストは含まれません。最終的な料金は、コスト分析と課金のビューに現地通貨で表示されます。 [Azure 料金計算ツールを表示します。](#)

17 / 25

価格レベル

- リソース作成時に、Free、Basic、Standardなどの**価格レベル**を選択する
- 価格レベルにより以下のようなものが決まる
 - 「インデックス」の上限数、「インデクサー」の上限数
 - 「検索ユニット」の性能
 - 検索性能やインデックス作成の性能に関係
 - 「検索ユニット」1個あたりのストレージの容量
 - このストレージはインデックスデータを格納するのに使用される
 - 「検索ユニット」の上限数、「レプリカ」の上限数、「パーティション」の上限数

価格レベルの変更

- 以前は、リソース作成時に選択した価格レベルは**リソース作成後に変更することはできなかった**
- 2025/3より、価格レベルの変更が可能となった（現在プレビュー）
- Basic レベルと Standard レベル (S1、S2、S3) の間の変更がサポートされる
- Basic から S1 に移行するなど、下位レベルから上位レベルにのみ切り替えが可能
- 操作の完了には数時間かかり、取り消しはできない

スケーリングオプション（レプリカ、パーティション、検索ユニット）

- 検索ユニット: Azure AI Searchリソースを構成する内部サーバー
 - クエリの実行を担当
 - ストレージを提供
 - 検索ユニットを増やすことで、クエリ同時実行性能が向上し、トータルのストレージ容量が増加する
- パーティション: 検索ユニットの集まりのこと
 - 例えばパーティション=3 と設定すると、パーティション内に検索ユニットが3個作成される
- レプリカ: パーティションの集まりのこと
 - 可用性を向上させるしくみ
- これらのスケーリングオプションについては**リソース作成後に設定変更が可能**

スケーリングオプション（レプリカ、パーティション、検索ユニット）



Azure AI Search リソース
価格レベル: S(Standard), レプリカ=2、パーティション=3

パーティション
(可用性ゾーン1)



パーティション
(可用性ゾーン2)



ストレージの総容量 =
 $160 \text{ GB} * 3 = 480 \text{ GB}$

リソース全体の検索ユニット数 =
 $2 \text{ レプリカ} * 3 \text{ パーティション} = 6$

検索ユニットのコスト = \$250/月

全体のコスト =
 $\$250/\text{月} * 6 \text{ 検索ユニット} = \$1,500/\text{月}$

- Azure AI Searchの価格
- 検索ユニット1個あたりの単価は、リソースの価格レベルとリージョンによって決まる
 - たとえば Standard (S) の場合 \$250/月 など
- リソース全体のコストは、検索ユニット数（=パーティション * レプリカ）に比例
 - たとえば パーティション 3、レプリカ 2 の場合、 $\$250/\text{月} * 3 * 2 = \$1,500/\text{月}$ など
- 小規模な利用や検証向けの、無料で使用できる **Free 価格レベル** も選択可能。
 - ストレージが50MBまで、検索ユニット（サーバー数）が1個のみ（パーティションとレプリカは1で固定、スケーリング不可）

- SDK = Software Development Kit
- C# / Python / Java / JavaScript(TypeScript) の4言語のSDKが提供されている。
- 各機能の「クイックスタート」で、リソースの作成、プロジェクトの作成、SDKの導入、コードの記述と実行などの具体的な手順を確認できる
 - C#
 - Python
 - Java
 - JavaScript

コード例（東京在住の「山本」さんを検索する例、主要部分のみ）

```
// エンドポイント、キー、インデックス名を環境変数から読み取り
string endpoint = Environment.GetEnvironmentVariable("SEARCH_ENDPOINT") ?? "";
string key = Environment.GetEnvironmentVariable("SEARCH_KEY") ?? "";
string index = Environment.GetEnvironmentVariable("SEARCH_INDEX") ?? "";

// 検索クライアントを作成
var client = new SearchClient(new Uri(endpoint), index, new AzureKeyCredential(key));

// 検索を実行
var response = client.Search<SearchDocument>("山本", new SearchOptions()
{
    Filter = "Location eq '東京'",
});

// 検索結果を表示
await foreach (var result in response.Value.GetResultsAsync())
{
    Console.WriteLine($"Name: {result.Document["Name"]}");
}
```


このコードの検索結果は、以下のような「JSONビュー」のクエリの検索結果と同じとなる。

JSON クエリ エディター

```
1 {
2   "search": "山本",
3   "filter": "Location eq '東京'"
4 }
5
```

検索

結果

```
1 {
2   "@odata.context": "https://ai902837425.search.windows.net/indexes('index1')/$metadata#docs(*)",
3   "value": [
4     {
5       "@search.score": 1.052196,
6       "Id": 3,
7       "Name": "山本 愛",
8       "Department": "総務部",
9       "Profile": "C#、 Django、 JavaScript",
10      "Age": 30,
11      "HireDate": "2017-09-01T00:00:00Z",
12      "Location": "東京",
13      "Certifications": [
```