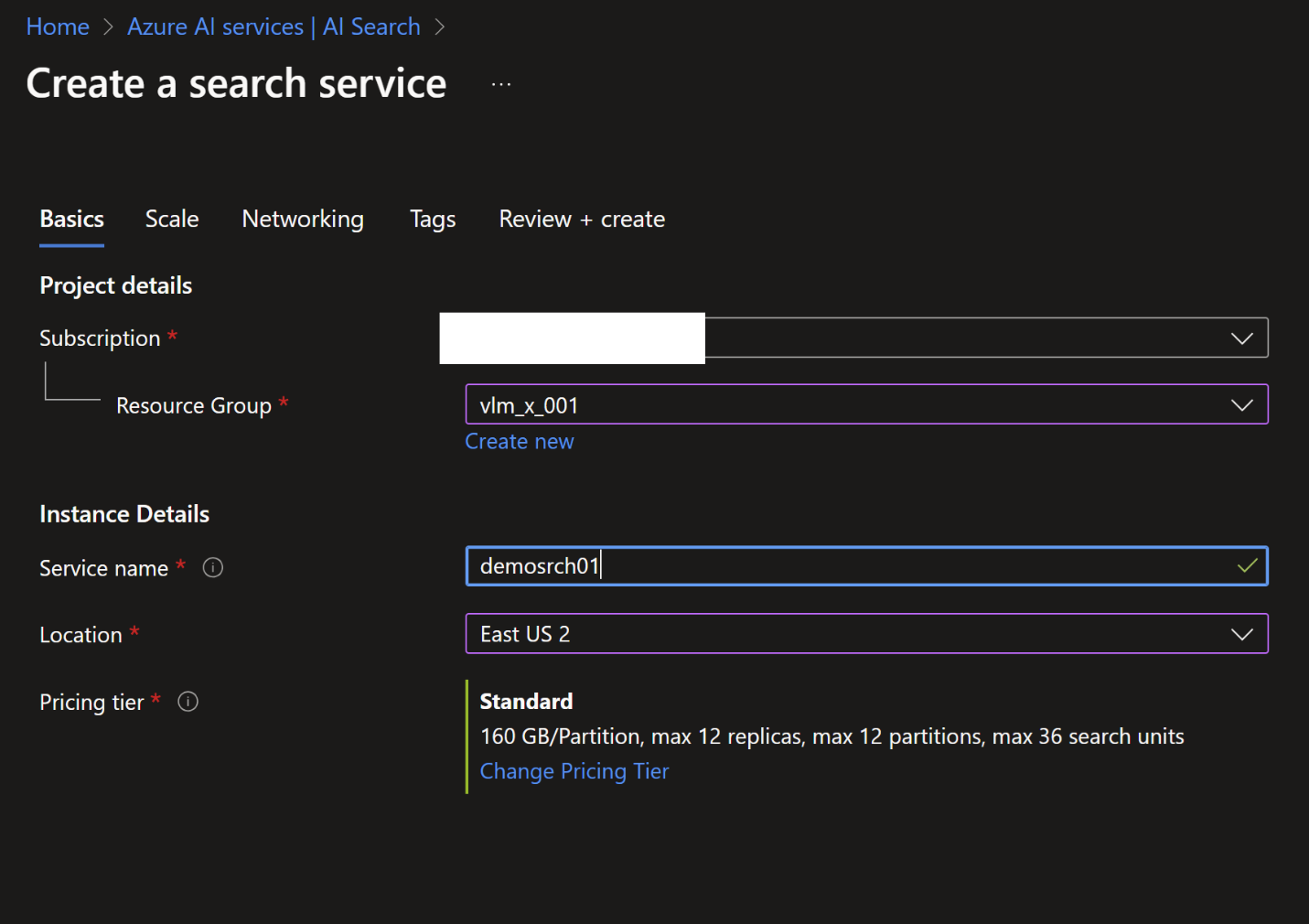
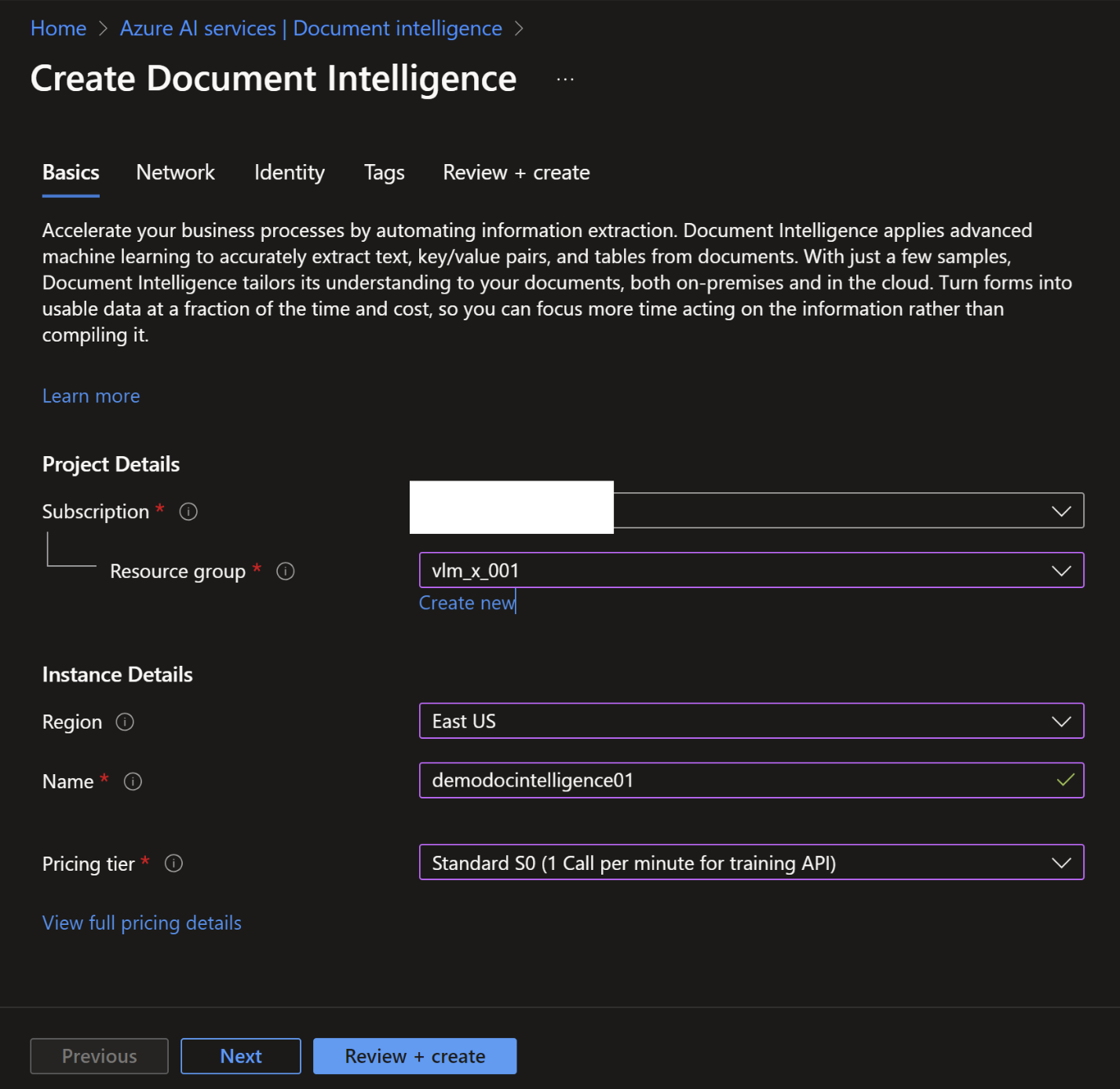
[https://portal.azure.com/#view/Microsoft\_Azure\_ProjectOxford/CognitiveServicesHub/~/CognitiveSearch](https://portal.azure.com/" \l "view/Microsoft_Azure_ProjectOxford/CognitiveServicesHub/~/CognitiveSearch)



Go with the defaults, and click Create

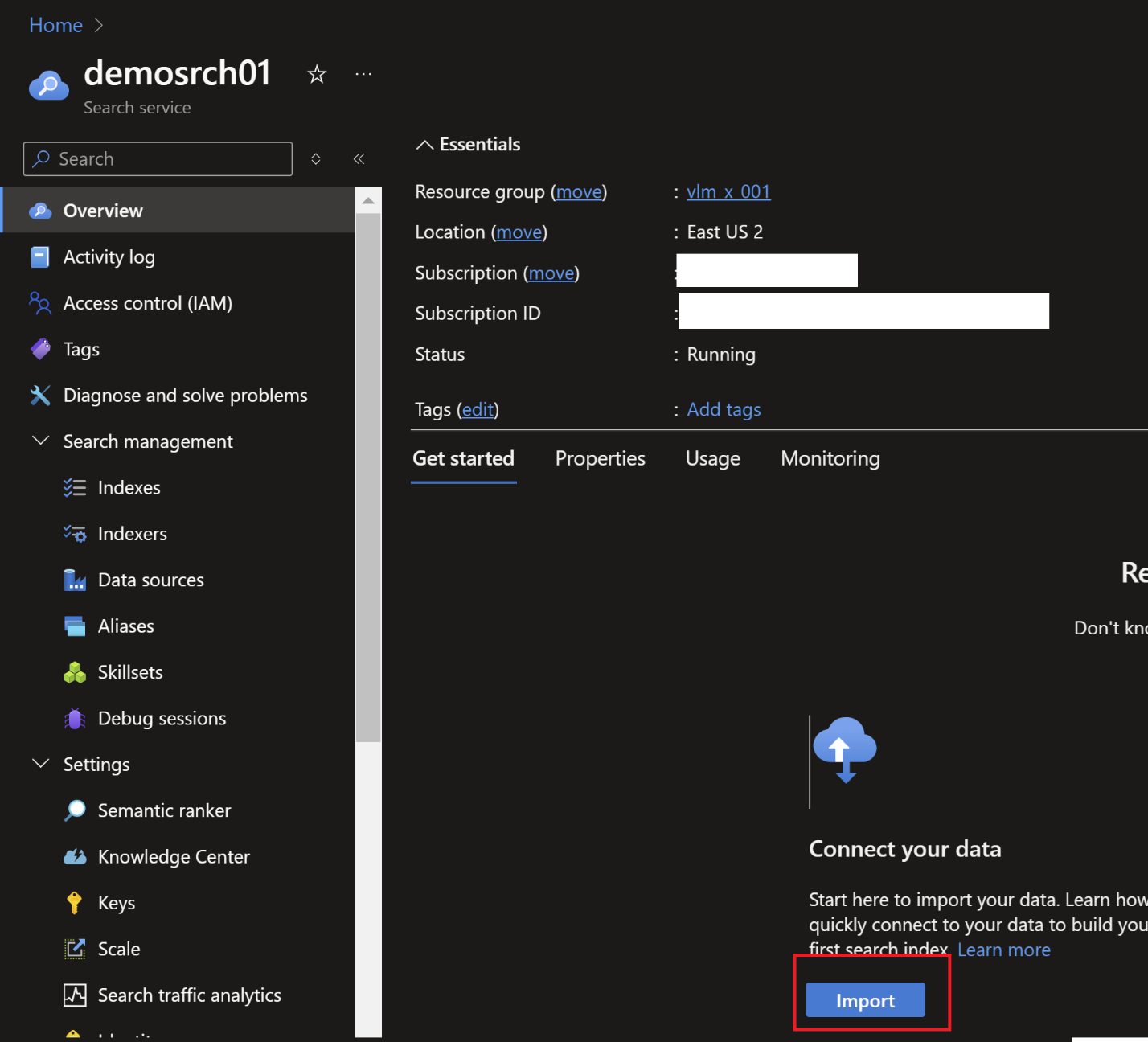
<https://azure.microsoft.com/en-us/pricing/details/ai-document-intelligence/>

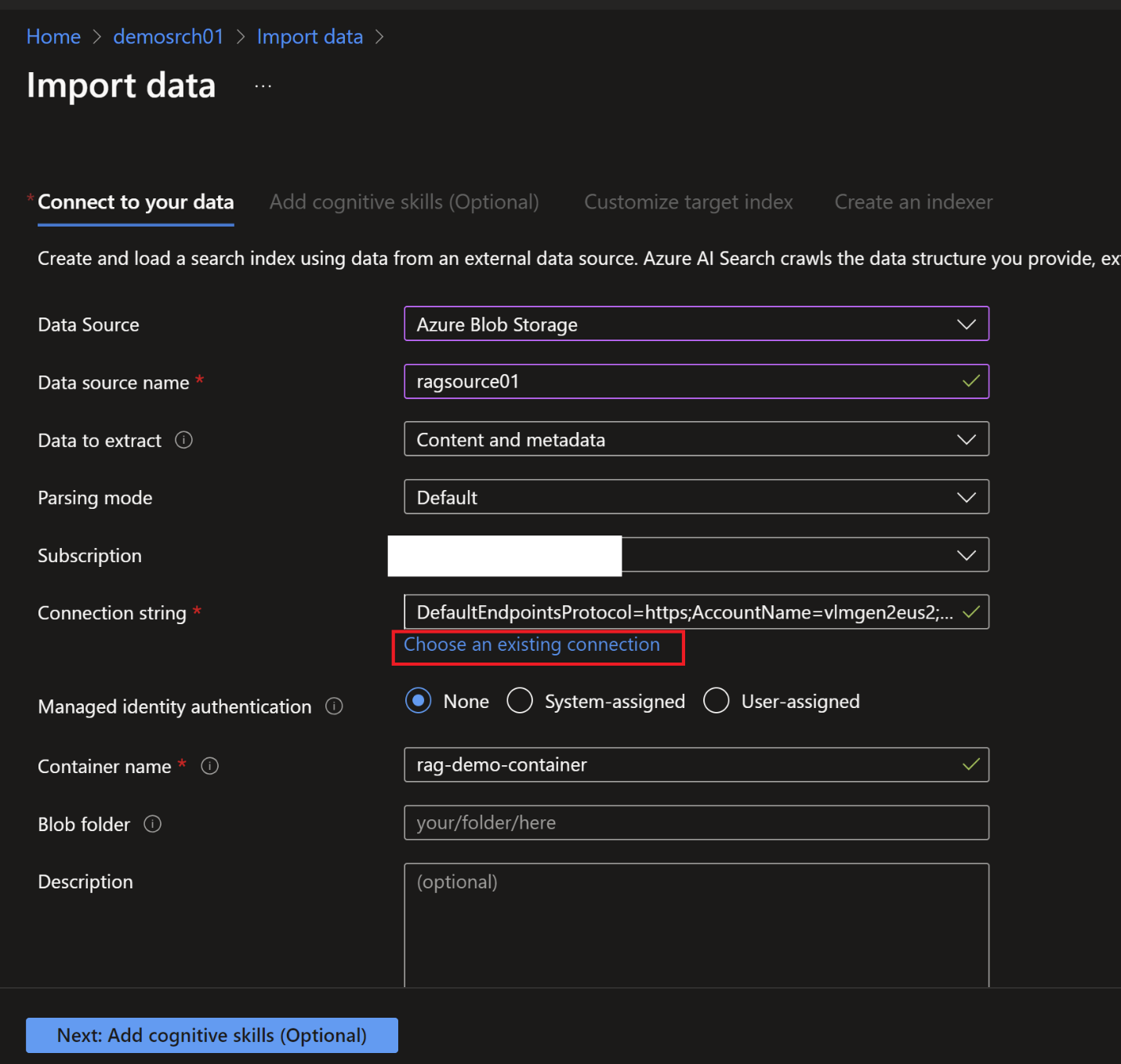


Uploaded sample data

A screenshot of a computer

Description automatically generated





A screenshot of a computer

Description automatically generated

Create a target index

A screenshot of a computer

Description automatically generated

indexer

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Import and Vectorize Data

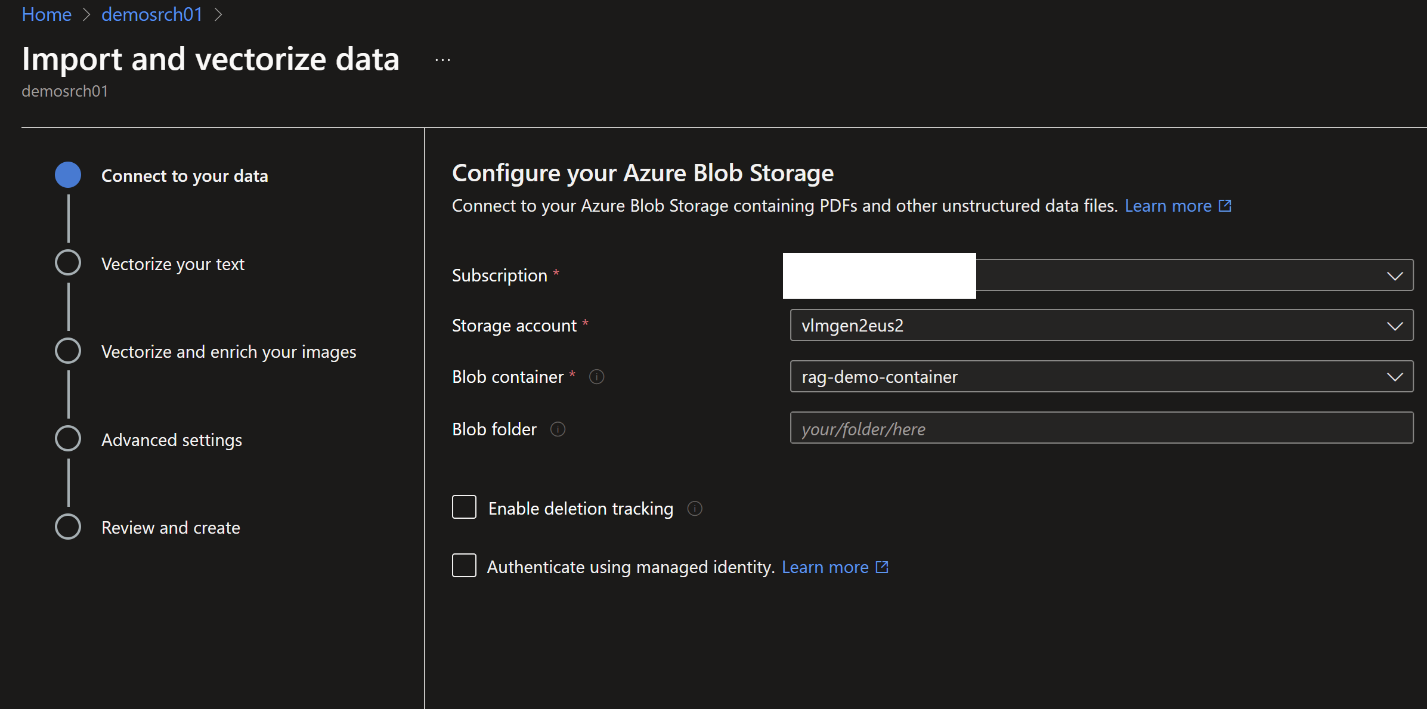
Create embedding

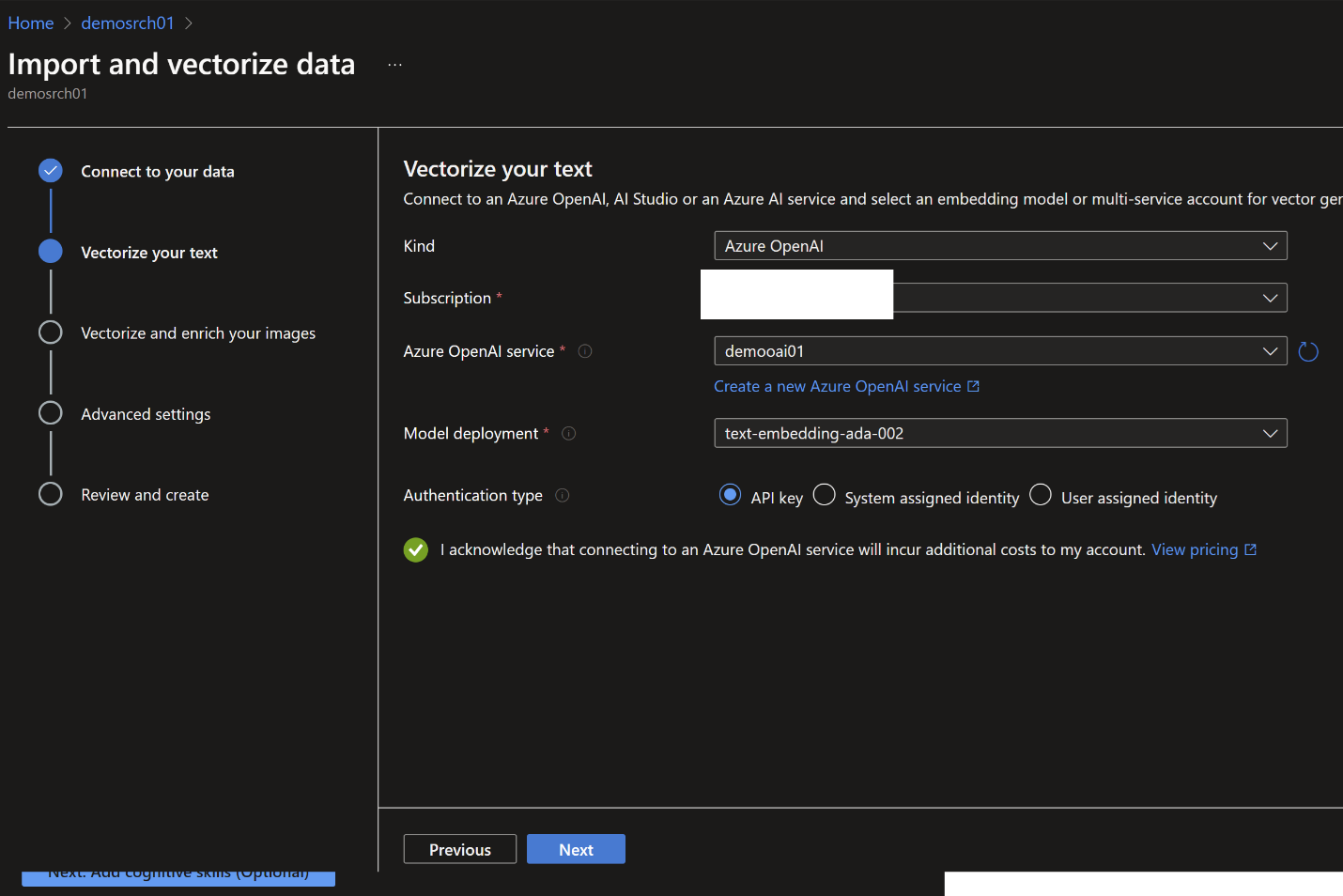
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated





A screenshot of a computer

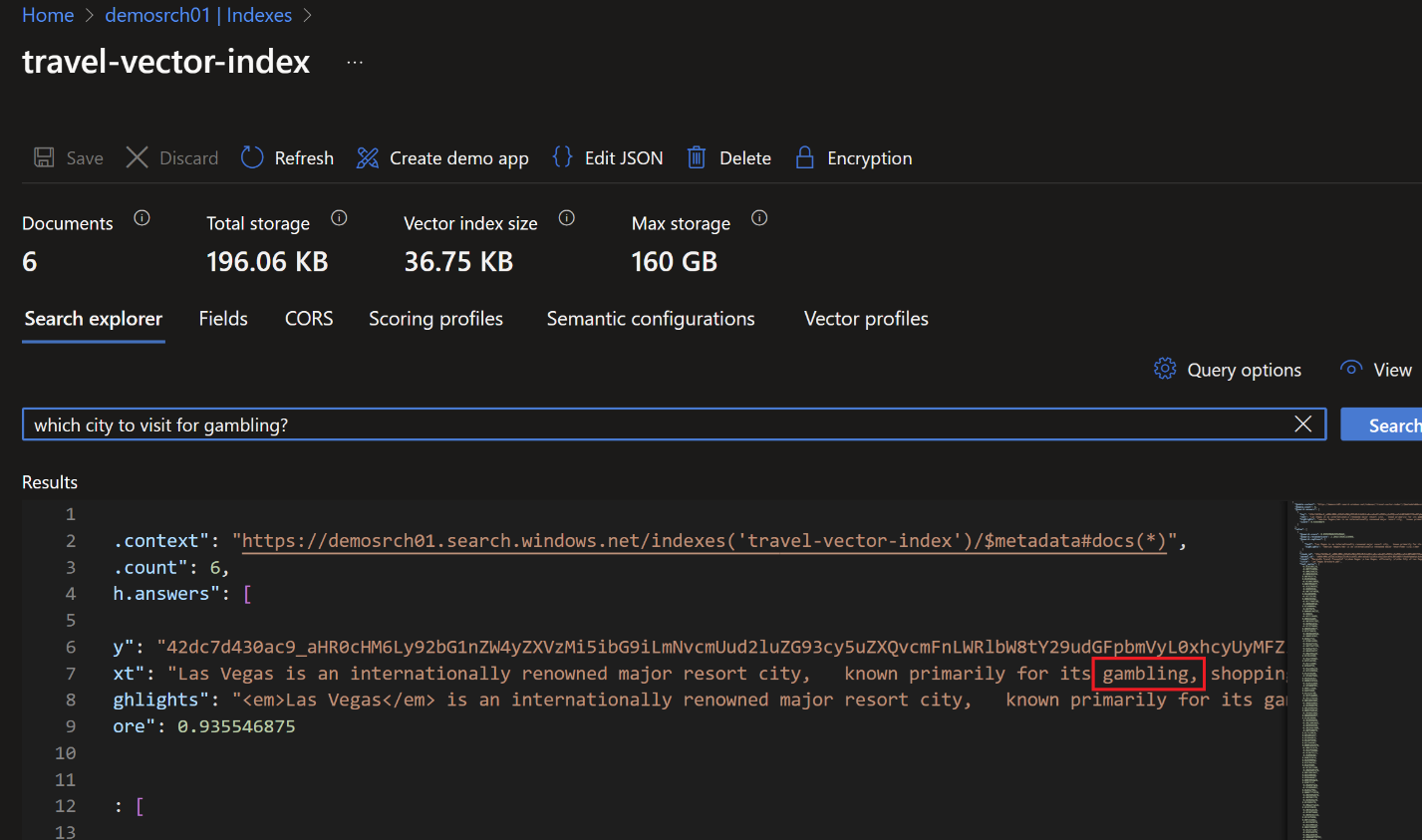
Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated



A screenshot of a computer

Description automatically generated

<https://microsoftlearning.github.io/mslearn-openai/Instructions/Exercises/06-use-own-data.html>

After 02-rag

Chunking and index creation

A screenshot of a computer

Description automatically generated

Creating a new index with custom fields

from azure.search.documents.indexes.models import (

    ScoringProfile,

    SearchableField,

    SearchField,

    SearchFieldDataType,

    SimpleField,

    TextWeights,

)

embeddings: AzureOpenAIEmbeddings = AzureOpenAIEmbeddings(

    azure\_deployment=azure\_deployment,

    openai\_api\_version=azure\_openai\_api\_version,

    azure\_endpoint=azure\_endpoint,

    api\_key=azure\_openai\_api\_key,

)

embedding\_function = embeddings.embed\_query

fields = [

    SimpleField(

        name="id",

        type=SearchFieldDataType.String,

        key=True,

        filterable=True,

    ),

    SearchableField(

        name="content",

        type=SearchFieldDataType.String,

        searchable=True,

    ),

    SearchField(

        name="content\_vector",

        type=SearchFieldDataType.Collection(SearchFieldDataType.Single),

        searchable=True,

        vector\_search\_dimensions=len(embedding\_function("Text")),

        vector\_search\_profile\_name="myHnswProfile",

    ),

    SearchableField(

        name="metadata",

        type=SearchFieldDataType.String,

        searchable=True,

    ),

    # Additional field to store the title

    SearchableField(

        name="title",

        type=SearchFieldDataType.String,

        searchable=True,

    ),

    # Additional field for filtering on document source

    SimpleField(

        name="source",

        type=SearchFieldDataType.String,

        filterable=True,

    ),

]

index\_name: str = "langchain-vector-index-custom"

vector\_store: AzureSearch = AzureSearch(

    azure\_search\_endpoint=vector\_store\_address,

    azure\_search\_key=vector\_store\_password,

    index\_name=index\_name,

    embedding\_function=embedding\_function,

    fields=fields,

)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

vector\_store.add\_texts(

    ["Test 1", "Test 2", "Test 3"],

    [

        {"title": "Title 1", "source": "A", "random": "10290"},

        {"title": "Title 2", "source": "A", "random": "48392"},

        {"title": "Title 3", "source": "B", "random": "32893"},

    ],

)

Random is ignored because that is not defined in the field

A screenshot of a computer program

Description automatically generated

Scoring profile

from azure.search.documents.indexes.models import (

    FreshnessScoringFunction,

    FreshnessScoringParameters,

    ScoringProfile,

    SearchableField,

    SearchField,

    SearchFieldDataType,

    SimpleField,

    TextWeights,

)

#  Azure OpenAI is your provider.

embeddings: AzureOpenAIEmbeddings = AzureOpenAIEmbeddings(

    azure\_deployment=azure\_deployment,

    openai\_api\_version=azure\_openai\_api\_version,

    azure\_endpoint=azure\_endpoint,

    api\_key=azure\_openai\_api\_key,

)

embedding\_function = embeddings.embed\_query

fields = [

    SimpleField(

        name="id",

        type=SearchFieldDataType.String,

        key=True,

        filterable=True,

    ),

    SearchableField(

        name="content",

        type=SearchFieldDataType.String,

        searchable=True,

    ),

    SearchField(

        name="content\_vector",

        type=SearchFieldDataType.Collection(SearchFieldDataType.Single),

        searchable=True,

        vector\_search\_dimensions=len(embedding\_function("Text")),

        vector\_search\_profile\_name="myHnswProfile",

    ),

    SearchableField(

        name="metadata",

        type=SearchFieldDataType.String,

        searchable=True,

    ),

    # Additional field to store the title

    SearchableField(

        name="title",

        type=SearchFieldDataType.String,

        searchable=True,

    ),

    # Additional field for filtering on document source

    SimpleField(

        name="source",

        type=SearchFieldDataType.String,

        filterable=True,

    ),

    # Additional data field for last doc update

    SimpleField(

        name="last\_update",

        type=SearchFieldDataType.DateTimeOffset,

        searchable=True,

        filterable=True,

    ),

]

# Adding a custom scoring profile with a freshness function

sc\_name = "custom\_scoring\_profile"

sc = ScoringProfile(

    name=sc\_name,

    text\_weights=TextWeights(weights={"title": 5}),

    function\_aggregation="sum",

    functions=[

        FreshnessScoringFunction(

            field\_name="last\_update",

            boost=100,

            parameters=FreshnessScoringParameters(boosting\_duration="P2D"),

            interpolation="linear",

        )

    ],

)

index\_name = "langchain-vector-custom-scoring-profile"

vector\_store: AzureSearch = AzureSearch(

    azure\_search\_endpoint=vector\_store\_address,

    azure\_search\_key=vector\_store\_password,

    index\_name=index\_name,

    embedding\_function=embeddings.embed\_query,

    fields=fields,

    scoring\_profiles=[sc],

    default\_scoring\_profile=sc\_name,

)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated