AI-102 Exam Preparation Notes

Adel Ghlamallah

Dec. 2nd, 2024

Table of Contents

Resources

- Study guide for Exam AI-102: https://learn.microsoft.com/en-ca/credentials/certifications/resources/study-guides/ai-102
- Microsoft Learn Track: https://learn.microsoft.com/en-us/credentials/certifications/azure-ai-engineer/?practice-assessment-type=certification
- Microsoft Official Exam Prep Videos: https://learn.microsoft.com/en-us/shows/exam-readiness-zone/preparing-for-ai-102-plan-and-manage-an-azure-ai-solution
- Microsoft Instructor series: https://learn.microsoft.com/en-us/shows/on-demand-instructor-led-training-series/?terms=AI-102
- MeasureUp AI-102 practice paid tests: https://www.measureup.com/microsoft-practice-test-ai-102-designing-and-implementing-an-azure-ai-solution.html
- John Savill's AI-102 Study Cram (https://youtu.be/I7fdWafTcPY?si=iqTt2MH7QYne8lAA)
- John Savill's AI-102 Whiteboard: https://raw.githubusercontent.com/johnthebrit/CertificationMaterials/main/whiteboards/AI-102-Whiteboard.png
- Notes from someone who passed this exam: https://areebpasha.notion.site/AI-102-Notes-dd32c9f349bb4e64a0d26ea661ba789c
- Other notes: AI-102 Notes : https://github.com/vatsprat/AI-102-AI-Engineer-Associate-Certification-Exam-/blob/main/AI_102_Notes.pdf
- Coursera course labs: https://38labs.atlassian.net/wiki/spaces/~712020c9681e0922854e8faabd2d72dc4e3702/pages/76709889/Azure+AI+Engineer+Associate+Certification+Prep+Bootcamp
- Microsoft Labs : https://github.com/MicrosoftLearning/AI-102-AIEngineer

The exam

There are 56 questions. I had 2 case studies, one at the beginning and one at the end. The 1h40 min seem like an eternally long time but without judicious use of it, you may run the risk of sprinting through the last questions.

Skills measured

- Plan and manage an Azure AI solution (15–20%)
- Implement content moderation solutions (10–15%)
- Implement computer vision solutions (15–20%)
- Implement natural language processing solutions (30–35%)

- Implement knowledge mining and document intelligence solutions (10–15%)
- Implement generative AI solutions (10–15%)

Introduction to IA

Examples of IA

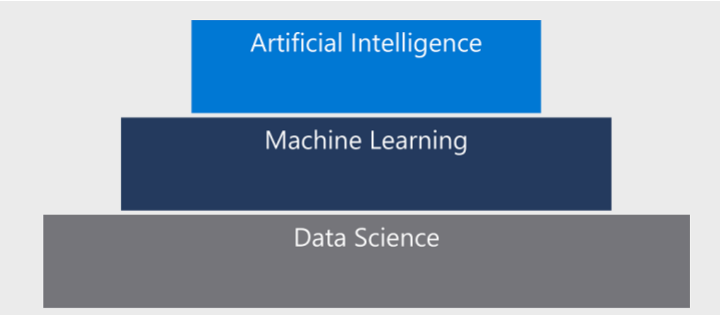
Visual perception - The ability to use *computer vision* capabilities to accept, interpret, and process input from images, video streams, and live cameras.

Text analysis and conversation - The ability to use *natural language processing (NLP)* to not only "read", but also generate realistic responses and extract semantic meaning from text.

Speech - The ability to recognize speech as input and synthesize spoken output. The combination of speech capabilities together with the ability to apply NLP analysis of text enables a form of human-compute interaction that's become known as *conversational AI*, in which users can interact with AI agents (usually referred to as *bots*) in much the same way they would with another human.

Decision making - The ability to use past experience and learned correlations to assess situations and take appropriate actions. For example, recognizing anomalies in sensor readings and taking automated action to prevent failure or system damage.

Understand AI-related terms



Data science
Data science is a discipline that focuses on the processing and analysis of data; applying statistical techniques to uncover and visualize relationships and patterns in the data, and defining experimental *models* that help explore those patterns.

Machine learning

Machine learning is a subset of data science that deals with the training and validation of *predictive* models. Typically, a data scientist prepares the data and then uses it to train a model based on an algorithm that exploits the relationships between the *features* in the data to predict values for unknown *labels*.

Artificial intelligence
Artificial intelligence usually (but not always) builds on machine learning to create software that emulates one or more characteristics of human intelligence.

Plan and manage an Azure AI solution (15-20%)

- Select the appropriate Azure AI service
- Plan, create and deploy an Azure AI service
- Manage, monitor, and secure an Azure AI service

Azure AI services are cloud-based services that encapsulate AI capabilities. Rather than a single product, you should think of AI services as a set of individual services that you can **use as building blocks to compose sophisticated**, intelligent applications.

AI services include a wide range of individual services across language, speech, vision, generative AI, and more. You can use AI services to build your own AI solutions to provide out-of-the-box solutions for common AI scenarios. A few examples of individual Azure AI services include:

- **Azure AI Vision** - Analyze content in images and videos.
- **Azure AI Language** - Build apps with industry-leading natural language understanding capabilities.
- **Azure AI Speech** - Speech to text, text to speech, translation, and speaker recognition.
- **Azure AI Document Intelligence** - An optical character recognition (OCR) solution that can extract semantic meaning from forms, such as invoices, receipts, and others.
- **Azure AI Search** - A cloud-scale search solution that uses AI services to extract insights from data and documents.
- **Azure OpenAI** - An Azure AI service that provides access to the capabilities of OpenAI.

Select the appropriate Azure AI service

- Select the appropriate service for a computer vision solution
- Select the appropriate service for a natural language processing solution
- Select the appropriate service for a speech solution

- Select the appropriate service for a generative AI solution
- Select the appropriate service for a document intelligence solution
- Select the appropriate service for a knowledge mining solution

Capabilities:

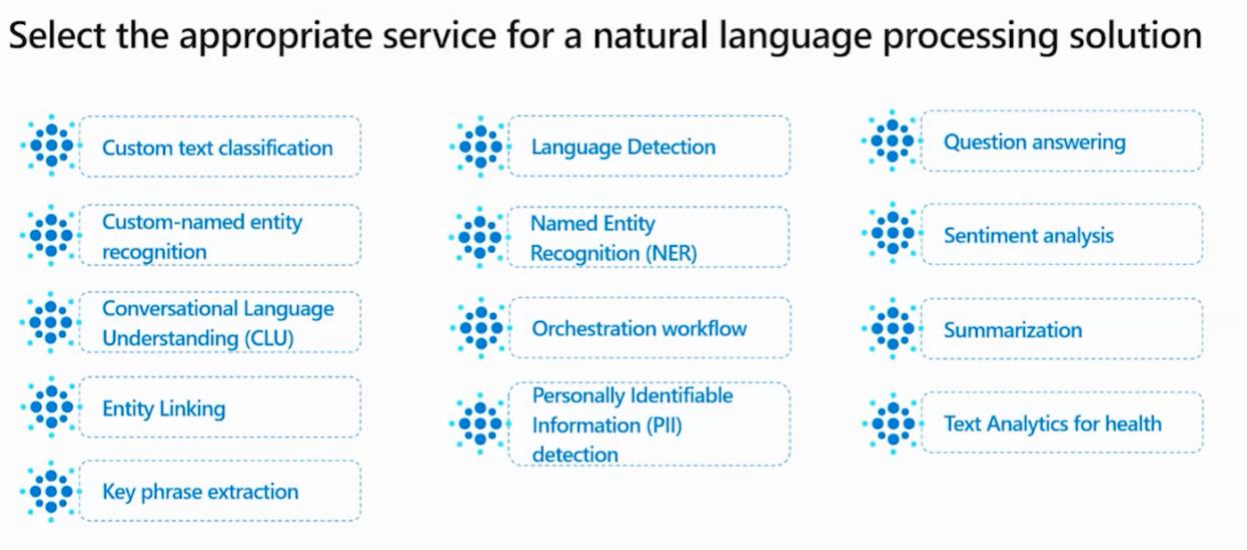
Natural language processing	Knowledge mining and document intelligence	Computer vision	Decision support	Generative AI
Text analysis	AI Search	Image analysis	Content safety	Azure OpenAI Service
Question answering	Document Intelligence	Video analysis	Content moderation	DALL-E image generation
Language understanding	Custom Document Intelligence	Image classification		
Translation	Custom skills	Object detection		
Named entity recognition		Facial analysis		
Custom text classification		Optical character recognition		
Speech		Azure AI Video Indexer		
Speech Translation				

Select the appropriate service for a computer vision solution

- **Analyze Images**: Use **AI Vision service** – Use pre-trained models to analyze images and insights and text from them
- **Video Analysis**: Azure **Video Indexer** is a service to extract insights from video, including face identification, text recognition, object labels, scene segmentations...etc.
- **Image classification**: Classify images by **training** a **custom model** with Azure AI Vision.

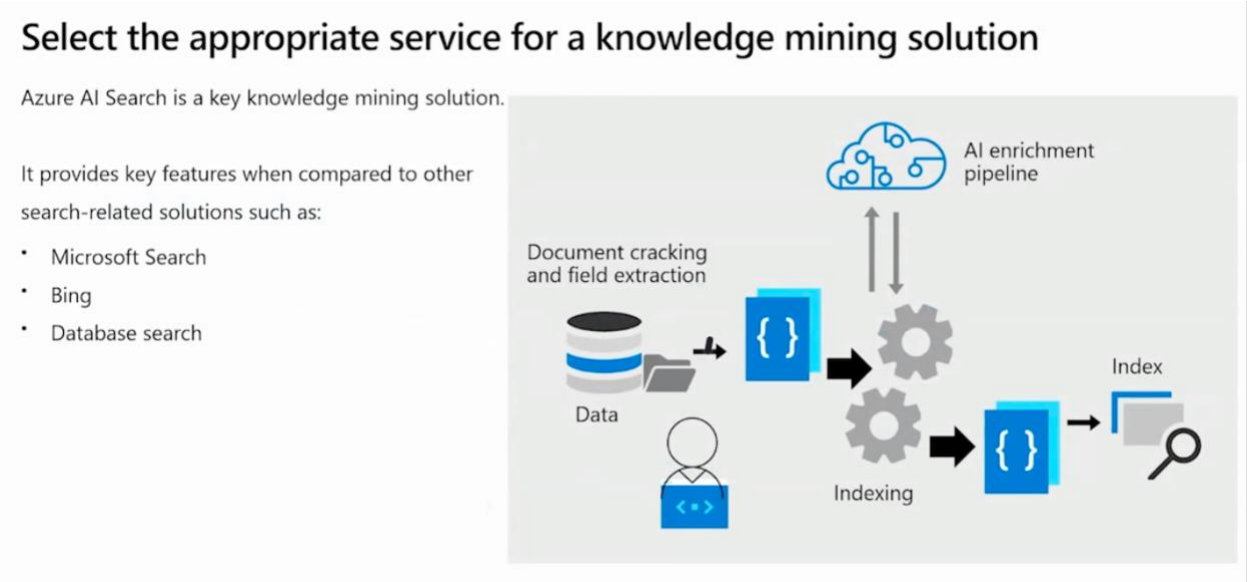
- **Object detection**: Object detection is used to locate and identify objects in images. You can use Azure AI Custom Vision to train a model to detect specific classes of object in images.
- **Detect, analyze, and recognize faces:** detect human faces, analyze facial features and emotions, and identify individuals
- **optical character recognition (OCR):** Vision service uses algorithms to process images and return information. Use the Image Analysis API for OCR.

Select the appropriate service for a natural language processing (NLP) solution



- **Language understanding:** Azure AI Language service extract semantic information from text
- **Conversational Language Understanding (CLU):** enables the **training** of a model that can be used to extract meaning from natural language
- **Question Answering solutions:** users ask questions using natural language and receive appropriate answers
- **Translation:** Translator service enables you to create intelligent apps and services that can translate text between languages
- **Speech recognition and transition**: Speech service enables you to use the speech-to-text (speech recognition) and text to speech (speech synthesis) APIs.
- **Speech translation**: Translation of speech builds on speech recognition by recognizing and transcribing spoken input in a specified language, and returning translations of the transcription in one or more other languages

Select the appropriate service for a knowledge mining solution



Azure AI Search is an Applied AI Service that enables you to **ingest and index data** from various sources, and search the index to find, filter, and sort information extracted from the source data.

In addition to basic text-based indexing, Azure AI Search enables you to define an ***enrichment pipeline*** that uses AI skills to enhance the index with insights derived from the source data - for example, by using computer vision and natural language processing capabilities to generate descriptions of images, extract text from scanned documents, and determine key phrases in large documents that encapsulate their key points.

Select the appropriate service for a generative AI solution

The Right Azure AI Service

Azure OpenAI Service

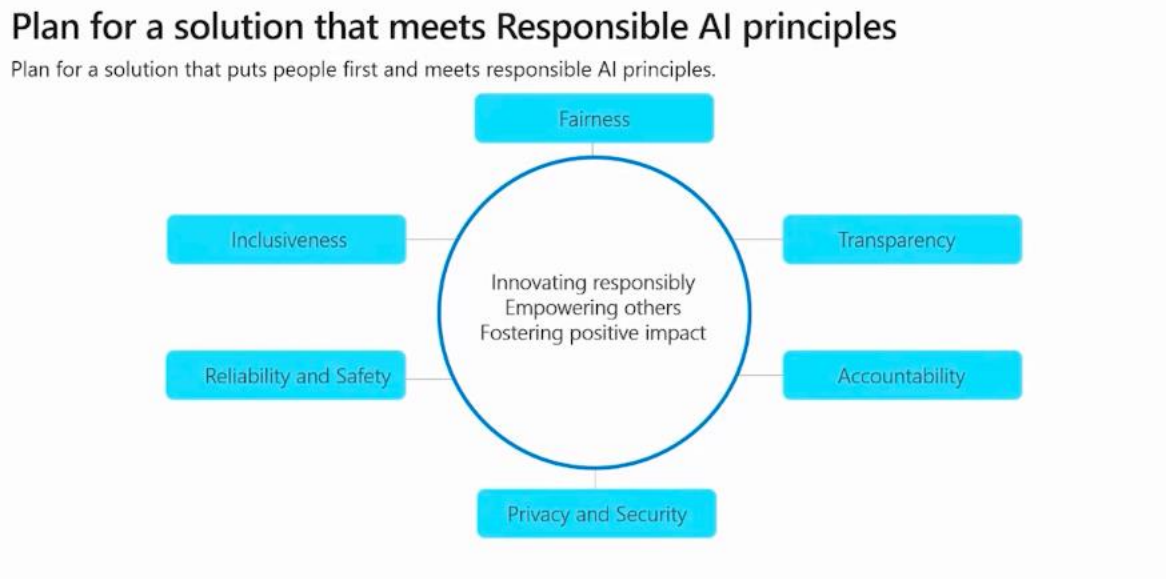
- Algorithms that generate text based on Large Language Models
- Examples includes
  - Generative AI such as ChatGPT
  - Generate content (Images, text, reports, business plan etc)
  - Retrieval Augmented Generation (RAG)

Plan, create and deploy an Azure AI service

- Plan for a solution that meets Responsible AI principles
- Create an Azure AI resource

- Determine a default endpoint for a service
- Integrate Azure AI services into a continuous integration and continuous delivery (CI/CD) pipeline
- Plan and implement a container deployment

Responsible AI principles



Fairness

AI systems should treat all people fairly. For example, suppose you create a machine learning model to support a loan approval application for a bank. The model should make predictions of whether or not the loan should be **approved without incorporating any bias based on gender**, **ethnicity**, or other factors that might result in an unfair advantage or disadvantage to specific groups of applicants.

Reliability and safety

AI systems should perform reliably and safely. For example, consider an AI-based software system for an autonomous vehicle; or a machine learning model that diagnoses patient symptoms and recommends prescriptions. **Unreliability in these kinds of system can result in substantial risk to human life**.

Privacy and security

AI systems should be secure and respect privacy. The machine learning models on which AI systems are based rely on large volumes of data, which may contain **personal details** that must be **kept private**.

Inclusiveness

AI systems should empower everyone and engage people. AI should bring benefits to all parts of society, regardless of physical ability, gender, sexual orientation, ethnicity, or other factors.

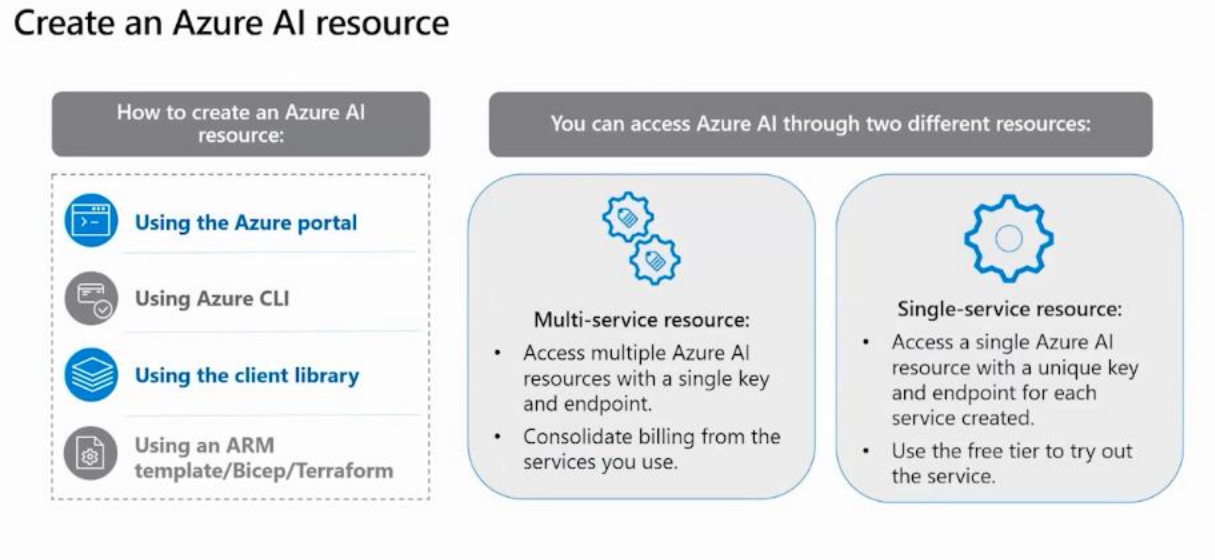
Transparency

AI systems should be **understandable**. Users should be made fully aware of the purpose of the system, how it works, and what limitations may be expected.

Accountability

People should be accountable for AI systems.

Create an Azure AI resource



For many of the available AI services, you can choose between the following provisioning options:

Multi-service resource

- You can provision an **AI services** resource that supports multiple different AI services. For example, you could create a single resource that enables you to use the **Azure AI Language**, **Azure AI Vision**, **Azure AI Speech**, and other services.
- This approach enables you to manage a **single set of access credentials** to consume multiple services at a single endpoint, and with a single point of billing for usage of all services.

Single-service resource

- Each AI service can be provisioned individually, for example by creating discrete **AI Language** and **AI Vision** resources in your Azure subscription.

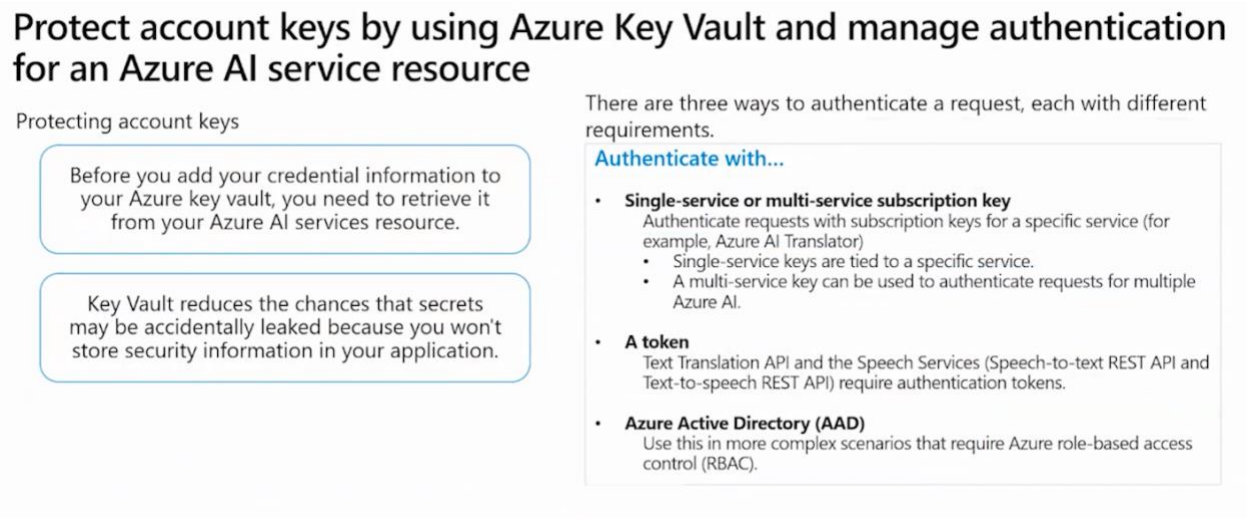
- This approach enables you to use separate endpoints for each service and to manage access credentials for each service independently.
- It also enables you to manage billing separately for each service.
- Single-service resources generally **offer a free tier** (with usage restrictions), making them a good choice to try out a service before using it in a production application.

Identify endpoints and keys

When you provision an Azure AI services service resource in your Azure subscription, you are defining an endpoint through which the service can be consumed by an application.
To consume the service through the endpoint, applications require the following information:

- **The endpoint URI**. This is the HTTP address at which the REST interface for the service can be accessed. Most AI services software development kits (SDKs) use the endpoint URI to initiate a connection to the endpoint.
- **A subscription key**. Access to the endpoint is restricted based on a subscription key. Client applications must provide a valid key to consume the service. When you provision an AI services resource, two keys are created - applications can use either key. You can also regenerate the keys as required to control access to your resource.
- **The resource location**. When you provision a resource in Azure, you generally assign it to a location, which determines the Azure data center in which the resource is defined. While most SDKs use the endpoint URI to connect to the service, some require the location.

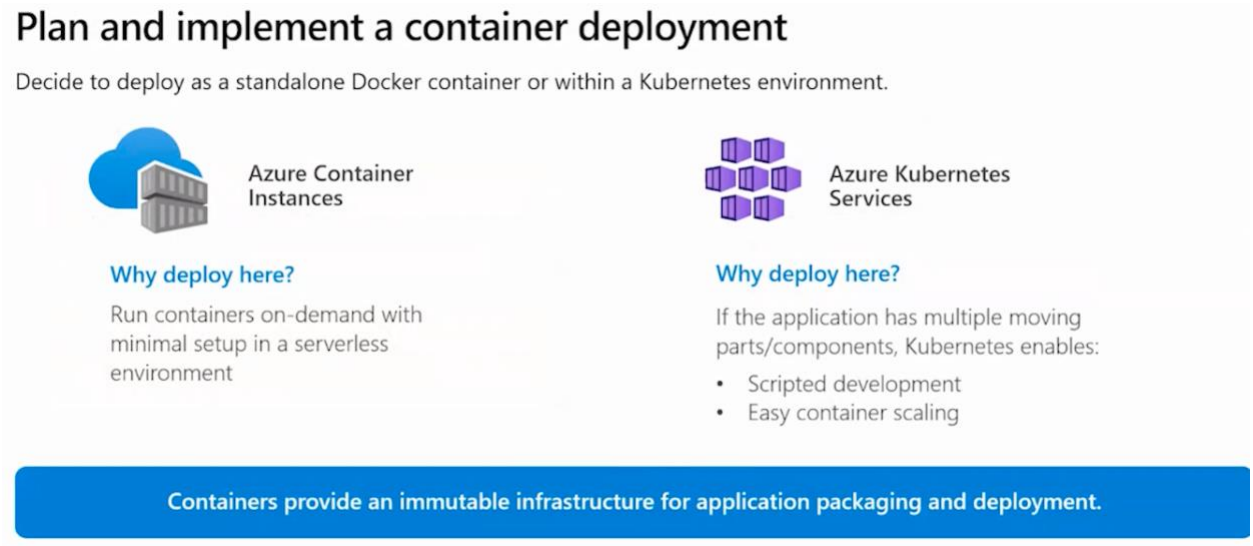
Determine a default endpoint for a service

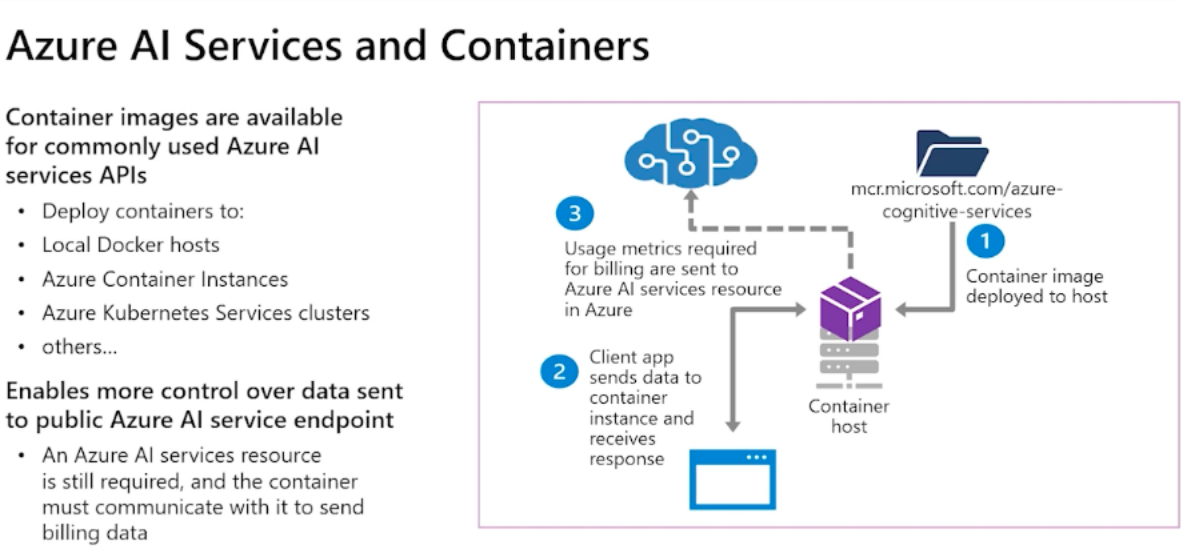


Plan and implement a container deployment

- Containers enable you to host Azure AI services either on-premises or in Azure.
- For example, if your application uses sensitive data in an on-premises SQL Server to call an Azure AI services service, you can deploy Azure AI services in containers on the same network. Your data can stay on your local network and not be passed to the cloud.

- Deploying Azure AI services in a container on-premises will also decrease the latency between the service and your local data, which can improve performance.





- Each container provides a subset of Azure AI services functionality.

- When you deploy an Azure AI services container image to a host, you must specify three settings:
    - ApiKey: Key from your deployed Azure AI service; used for billing.
    - Billing: Endpoint URI from your deployed Azure AI service; used for billing.
    - Eula: Value of accept to state you accept the license for the container.

Manage, monitor, and secure an Azure AI service

- Configure diagnostic logging
- Monitor an Azure AI resource
- Manage costs for Azure AI services
- Manage account keys
- Protect account keys by using Azure Key Vault
- Manage authentication for an Azure AI Service resource
- Manage private communications

Configure diagnostic logging

- Diagnostic logging enables you to **capture rich operational data** for an Azure AI services resource, which can be used to analyze service usage and troubleshoot problems.

- You can use **Azure Event Hubs** as a destination in order to then forward the data on to a custom telemetry solution.

- You can capture diagnostic logs; you need a destination for the log data:
    - Azure Log Analytics
    - Azure Storage

- It can take an hour or more before diagnostic data starts flowing to the destinations, but when the data has been captured, you can **view it in your Azure log Analytics** resource by running queries.

Security

Azure AI services provide multiple layers of security that you should consider when implementing a solution.

Authentication

By default, access to Azure AI services resources is restricted by using **subscription keys**.

Regenerate keys

You should regenerate keys regularly to protect against the risk of keys being shared with or accessed by unauthorized users.

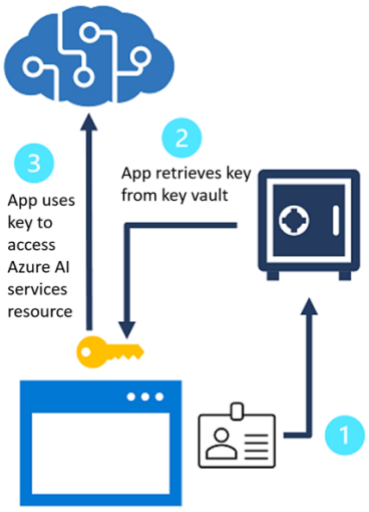
You can regenerate keys using:

- the Azure portal,
- or using the *az cognitiveservices account keys regenerate* Azure command-line interface (CLI) command.

Protect keys with Azure Key Vault

- Access to the key vault is granted to *security principals*, which you can think of user identities that are authenticated using Microsoft Entra ID.

- Administrators can assign a security principal to an application (in which case it is known as a *service principal*) to define a *managed identity* for the application.
- The application can then use this identity to access the key vault and retrieve a secret to which it has access.
- You can store the subscription keys for an AI services resource in Azure Key Vault, and assign a managed identity to client applications that need to use the service.



Token-based authentication

- When using the REST interface, some AI services support (or even *require*) token-based authentication.
- In these cases, the subscription key is presented in an initial request to obtain an authentication token, which has a valid period of 10 minutes.
- Subsequent requests must present the token to validate that the caller has been authenticated.

Microsoft Entra ID authentication

- Azure AI services support Microsoft Entra ID authentication, enabling you to grant access to specific service principals or managed identities for apps and services running in Azure.
- There are different ways you can authenticate against Azure AI services using Microsoft Entra ID, including:
    - Authenticate using service principals
        - Create a custom subdomain
        - Assign a role to a service principal
    - Authenticate using managed identities
        - System-assigned managed identity
        - User-assigned managed identity

Network Security

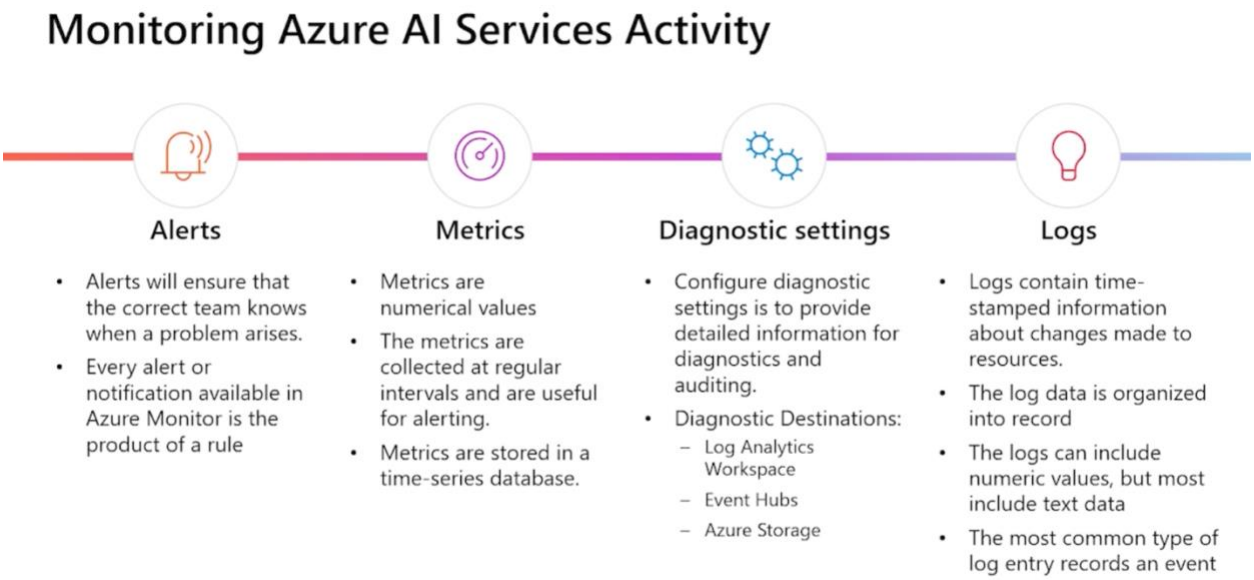
- Network security is an important measure to ensure unauthorized users can't reach the services that you are protecting.

- By default, Azure AI services are accessible from all networks. Selection the option: **Allow Access from Selected Networks and Private Endpoints**. Then in the Firewall, you **need to specify the client IP address to access the service.**

Managing Costs

- One of the main benefits of using cloud services is that you can gain cost efficiencies by only paying for services as you use them.
- Some Azure AI services resources offer a **free tier** with restrictions on use, which is useful for development and testing; and one or more billed tiers that incur charges based on transactions.
- Before deploying a solution that depends on AI services, you can **estimate** costs by using the **Azure Pricing Calculator**.
- To view costs for AI services, sign into the Azure portal and select your subscription. You can then view overall costs for the subscription by selecting the **Cost analysis** tab.
- To view only costs for AI services, add a filter that restricts the data to reflect resources with a **service name** of **Cognitive Services**.

Monitoring



Alerts

- Microsoft Azure provides alerting support for resources through the creation of *alert rules*.
- You use alert rules to configure notifications and alerts for your resources based on events or metric thresholds.
- To create an **alert rule** for an Azure AI services resource, select the resource in the Azure portal and on the **Alerts** tab, add a new alert rule and specify the following:
    - The ***scope*** of the alert rule - in other words, the **resource** you want to monitor.

    - A ***condition*** on which the alert is triggered. The specific trigger for the alert is based on a ***signal type***, which can be:
        - ***Activity Log*** (an entry in the activity log created by an action performed on the resource, such as regenerating its subscription keys)
        - or ***Metric*** (a metric threshold such as the number of errors exceeding 10 in an hour).
    - Optional ***actions***, such as sending an email to an administrator notifying them of the alert, or running an Azure Logic App to address the issue automatically.
    - *Alert* ***rule details***, such as a name for the alert rule and the resource group in which it should be defined.

View metrics

- **Azure Monitor** collects metrics for Azure resources at regular intervals so that you can **track indicators** of resource **utilization**, **health**, and **performance**.
- The specific metrics gathered depend on the Azure resource.
- In the **case of Azure AI services**, Azure Monitor collects metrics relating to endpoint **requests**, **data** submitted and returned, **errors**, and other useful measurements.

Manage diagnostic logging

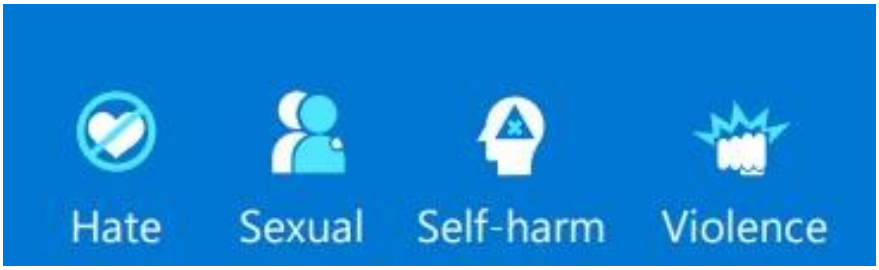
- Diagnostic logging enables you to capture rich operational data for an Azure AI services resource, which can be used to analyze service usage and troubleshoot problems.

Implement content moderation solutions (10–15%)

- Azure AI Content Safety is an AI service designed to help developers **include advanced content safety into their applications** and services.
- Azure AI Content Safety **identifies** potentially **unsafe** content and helps organizations to **comply with regulations** and meet their own quality standards.
- Azure AI **Content Safety** is a set of advanced content moderating features that can be **incorporated** into your **applications** and services.
- Azure AI Content Safety is available **as a resource** in the Azure portal.

How does Azure AI Content Safety work?

- Azure AI Content Safety works with **text** and **images**, and **AI-generated content**.
- Text analysis **uses natural language processing** techniques
- Azure AI Content Safety is multilingual and can **detect harmful content** in both short form and long form.
- Azure AI Content Safety classifies content into **four categories**:



- A **severity** level for each category is used to determine whether content should be blocked, sent to a moderator, or auto approved.

- Azure AI Content Safety features include:

    - Safeguarding text content
        - **Moderate text** scans text across four categories: violence, hate speech, sexual content, and self-harm. A **severity** level from **0 to 6** is returned for each category.
        - **Prompt shields** is a unified API to identify and **block jailbreak attacks** from **inputs to LLMs**.
        - **Protected material detection** checks AI-generated text for protected text such as recipes, copyrighted song lyrics.
        - **Groundedness detection** protects against **inaccurate responses** in AI-generated text by LLMs.

    - Safeguarding image content
        - **Moderate images** scans for inappropriate content across four categories: violence, self-harm, sexual, and hate. A **severity** level is returned: **safe**, **low**, or **high**.
        - **Moderate multimodal content** scans both images and text, including text extracted from an image using optical character recognition (**OCR**).

    - Custom safety solutions
        - **Custom categories** enable you to create your own categories by providing positive and negative examples and training the model.
        - Safety system message helps you to write effective prompts to guide an AI system's behavior.

- When evaluating how accurately Azure AI Content Safety is for your situation, compare its performance against four criteria:
    - **True positive** - correct identification of harmful content.
    - **False positive** - incorrect identification of harmful content.
    - **True negative** - correct identification of harmless content.
    - **False negative** - harmful content isn't identified.

- Azure AI Content Safety is an AI service designed to provide a more comprehensive approach to **content moderation**.
- Azure AI Content Safety helps organizations to **prioritize work for human moderators** in a growing number of situations

Create solutions for content delivery

- Implement a text moderation solution with Azure AI Content Safety

- Implement an image moderation solution with Azure AI Content Safety

- Azure AI Content Safety Studio is available as part of Azure AI Studio

- Azure AI Content Safety Studio enables you to explore and test Content Safety features for yourself.

- Azure AI content safety detects harmful user-generated and AI-generated content in applications and services. It includes text and image APIs that allow you to detect harmful or inappropriate material.

Implement computer vision solutions (15–20%)

Analyze images

- Select visual features to meet image processing requirements
- Detect objects in images and generate image tags
- Include image analysis features in an image processing request
- Interpret image processing responses
- Extract text from images using Azure AI Vision

- Convert handwritten text using Azure AI Vision

Implement custom computer vision models by using Azure AI Vision

- Choose between image classification and object detection models
- Label images
- Train a custom image model, including image classification and object detection
- Evaluate custom vision model metrics
- Publish a custom vision model
- Consume a custom vision model

Analyze videos

- Use Azure AI Video Indexer to extract insights from a video or live stream
- Use Azure AI Vision Spatial Analysis to detect presence and movement of people in video

Analyze images

Azure AI Vision – Image Analysis

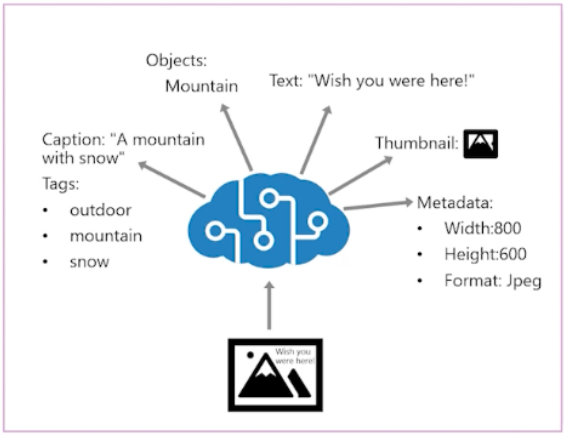
Image analysis:

- Caption and tag generation
- Object detection
- People detection
- Optical character recognition
- Smart crop thumbnails
- Background removal
- Multi-modal embeddings
- Product recognition

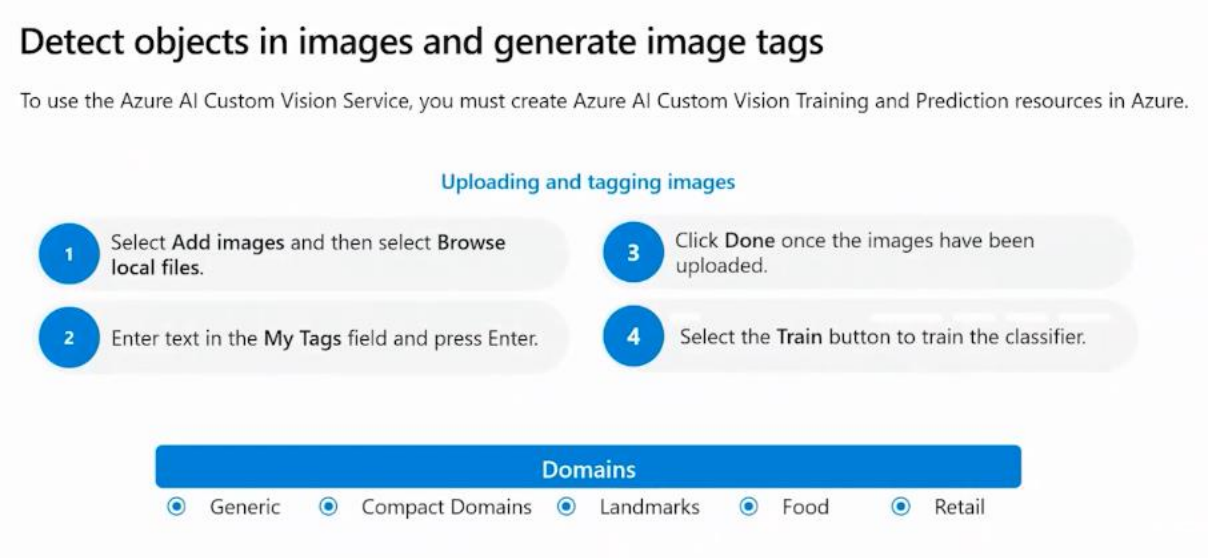
Can be used as:

- Standalone **Azure AI Vision** resource
- Multi-service **Azure AI Services** resource

* Some new features are limited to specific regions



Vision Studio: https://portal.vision.cognitive.azure.com/



The **Azure AI Vision** service is designed to help extract information from images. It provides functionality that you can use for:

- ***Description and tag generation*** - determining an appropriate caption for an image and **identifying** relevant "**tags**" that can be used as keywords to indicate its subject.
- ***Object detection*** - detecting the **presence** and **location** of specific objects within the image.
- ***People detection*** - detecting the presence, location, and features of people in the image.
- ***Image metadata****, color, and type analysis* - determining the format and size of an image, its dominant color palette, and whether it contains clip art.
- ***Category identification*** - identifying an appropriate categorization for the image, and if it contains any known landmarks.
- ***Background removal*** - detecting the background in an image and output the image with the background transparent or a greyscale alpha matte image.
- ***Moderation rating*** - determine if the image includes any adult or violent content.
- ***Optical character recognition*** - reading text in the image.
- ***Smart thumbnail generation*** - identifying the main region of interest in the image to create a smaller "thumbnail" version.

To analyze an image, you can:

- Use the **Analyze Image** REST method
- or the equivalent method in the **SDK** for your preferred programming language

Specifying the visual features you want to include in the analysis (and if you select categories, whether or not to include details of celebrities or landmarks). This method returns a JSON document containing the requested information.

Available visual features are contained in the **VisualFeatures** enum:

- VisualFeatures.**TAGS**: Identifies tags about the image, including objects, scenery, setting, and actions
- VisualFeatures.**OBJECTS**: Returns the bounding box for each detected object
- VisualFeatures.**CAPTION**: Generates a caption of the image in natural language
- VisualFeatures.**DENSE_CAPTIONS**: Generates more detailed captions for the objects detected
- VisualFeatures.**PEOPLE**: Returns the bounding box for detected people
- VisualFeatures.**SMART_CROPS**: Returns the bounding box of the specified aspect ratio for the area of interest
- VisualFeatures.**READ**: Extracts readable text

SDK Client class to access the Vision Service: **ImageAnalysisClient**

Generate a smart-cropped thumbnail

The Azure AI Vision service enables you to create a **thumbnail with different dimensions** (and aspect ratio) from the source image, and optionally to use image analysis to **determine the *region*** *of interest* in the image (its main subject) and make that the focus of the thumbnail

Remove image background

- The background removal feature can split the image into the subject in the foreground, and everything else that is considered background.
- Azure AI Vision achieves this feature by **creating an *alpha matte*** of the foreground subject, which is then used to return either the foreground or the background.

Implement custom computer vision models by using Azure AI Vision

Custom models in Azure AI Vision allow you to train an AI model to classify images or detect objects in images:

- ***Image classification*** is a common computer vision problem that requires software to analyze an image in order to categorize (or ***classify***) it.

- ***Object detection*** is another common computer vision problem that requires software to identify the location of specific classes of object in an image.

Image classification

- Image classification is a computer vision feature where a model is trained to **predict a label for an image** based on the contents of the entire image.
- Usually, the class label relates to the main *subject* of the image, however individual use cases may vary.
- Models can be trained for:
    - **multi-class classification** (where there are multiple classes, but each image can belong to only one class)

- **or multi-label classification** (where an image might be associated with multiple labels).

Object Detection

- Object detection is a form of computer vision in which a **model** is **trained** to **detect** the presence and location of one or more classes of object in an image.
- For example, an AI enabled checkout system in a grocery store might need to identify the type and location of items being purchased.

There are **two components to object detection**:

- The **class label** of each object detected in the image. For example, you might predict that an image contains one apple and two oranges.
- The **location** of each object within the image, indicated as coordinates of a bounding box that encloses the object.

Create a custom project

To create a custom Azure AI Vision model, you first need an Azure AI Services resource (or an Azure AI Vision resource). Once that resource is deployed to your subscription, you need to create a custom project.

In most cases, the steps you follow are:

1. Create your **blob storage container** and upload just the training images.
2. Create the **dataset** for your project and **connect** it to your **blob storage** container. When creating your dataset, you define what type of project it is (image classification, object detection, or product recognition).
3. Label your data in your Azure Machine Learning Data Labeling Project, which creates the **COCO** file (**json**) in your blob storage container.
4. **Connect** your completed **COCO file** for the labeled images **to your dataset**.
5. **Train** your custom model on the dataset and labels created.
6. **Verify performance** and iterate if the trained performance isn't meeting expectations.

Label and train a custom model

- Once you upload your images to blob storage and created your dataset, the next step is to label your images and connect the resulting COCO file. If you already have a COCO file for your training images, you can skip the labeling step.
- Labeling your training images is done in Azure Machine Learning studio, using the Data Labeling Project.
- Having complete and accurate labels for your training images greatly improves the performance of your trained model.
- When you label your images, be sure to accurately assign labels and completely label all instances of each class.

- In your dataset within Vision Studio, create a new Azure Machine Learning Data Labeling project or connect to an existing project if you created one in Azure Machine Learning studio.

Training your model

- With all the training images labeled, the next step is training your model.
- When training a model select the model type, specify the dataset you want to use as training data, and indicate the training budget.
- The training budget is an upper bound of time for how long the training will run; the actual time used for training is often less than the specified budget.
- Once your model is trained, selecting it allows you to view the performance of evaluation run.
- The default evaluation run takes a small set of the labeled images out of the training set, uses the trained model for predictions on that subset, and compares the predictions to the provided labels.
- From the trained model page, you can trigger new evaluation runs on a different set of images or try out your own tests in Vision Studio by selecting the tab on the top of the page.

Options for labeling images

- The easiest option for labeling images for object detection is to use the interactive interface in the **Azure AI Custom Vision portal**.

- Additionally, after tagging an initial batch of images, you can train the model.

- Subsequent labeling of new images can benefit **from the *smart labeler* tool** in the portal, which can suggest not only the regions, but the classes of object they contain.

- Alternatively, you can use a labeling tool, such as the one provided in Azure Machine Learning Studio or the Microsoft Visual Object Tagging Tool (VOTT).

- If you choose to use a labeling tool other than the Azure AI Custom Vision portal, you may need to use the bounding boxes are defined by four values that represent the left (X) and top (Y) coordinates of the top-left corner of the bounding box, and the width and height of the bounding box.

Difference between classification and Object detection

- The most **significant difference** between training an *image classification* model and training an *object detection* model is the **labeling** of the images with **tags**.
- While image classification requires one or more tags that apply to the whole image, object **detection** requires that each label consists of a **tag** and a ***region*** that defines the bounding box for each object in an image.

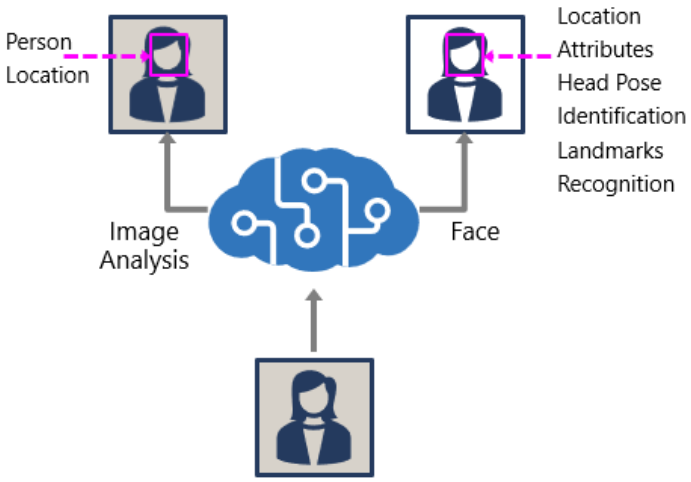
- The Azure AI Custom Vision portal provides a graphical interface that you can use to label your training images.

Detect, analyze, and recognize faces

- Face detection, analysis, and recognition are all common computer vision challenges for AI systems.
- The ability to detect **when a person is present**, identify a person's facial **location**, or **recognize** an individual based on their facial features.

There are **two Azure AI services** that you can use to build solutions that **detect faces** or people in images:

- The Azure AI **Vision** service
- The **Face** service



The Azure AI Vision service

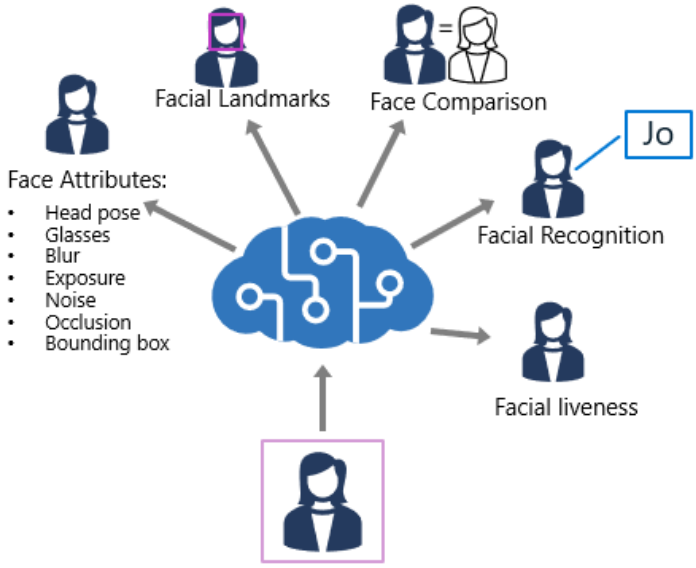
- The **Azure AI Vision** service enables you to **detect** people in an image, as well as returning a **bounding box** for its location.
- To detect and analyze faces with the Azure AI Vision service, call the **Analyze Image** function (SDK or equivalent REST method), specifying **People** as one of the visual features to be returned.
- In images that contain one or more people, the response includes details of their location in the image and the attributes of the detected person.

The Face service

The **Face** service offers **more comprehensive facial analysis** capabilities than the Azure AI Vision service, including:

- **Face detection** (with bounding box).

- Comprehensive facial feature **analysis** (including head pose, presence of spectacles, blur, facial landmarks, occlusion and others).
- Face **comparison** and **verification**.
- Facial **recognition**.



You can provision:

- **Face** as a single-service resource,
- or you can use the Face API in a multi-service **Azure AI Services** resource.

Compare and match detected faces

When a face is detected by the Face service, a **unique ID is assigned to it** and **retained** in the service resource for **24 hours**. The ID is a GUID, with no indication of the individual's identity other than their facial features.

Implement facial recognition

For scenarios where you need to **positively identify** individuals, you can train a facial recognition model using face images. This is used to recognize specific persons (E.g. employees). In this case, the faces will last more than 24 hours.

To train a facial recognition model with the Face service:

1. Create a **Person Group** that defines the set of individuals you want to identify (for example, *employees*).
2. Add a **Person** to the **Person Group** for each individual you want to identify.
3. Add detected faces from multiple images to each **person**, preferably in various poses. The IDs of these faces will no longer expire after 24 hours (so they're now referred to as *persisted* faces).
4. Train the model.

The trained model is stored in your Face (or Azure AI Services) resource, and can be used by client applications to:

- *Identify* individuals in images.
- *Verify* the identity of a detected face.
- Analyze new images to find faces that are *similar* to a known, persisted face.

Consideration (ethical) for face analysis

When building a solution that uses facial data, considerations include (but aren't limited to):

- **Data privacy and security**. Facial data is personally identifiable, and should be considered sensitive and private. You should ensure that you have implemented adequate protection for facial data used for model training and inferencing.
- **Transparency**. Ensure that users are informed about how their facial data is used, and who will have access to it.
- **Fairness and inclusiveness**. Ensure that your face-based system can't be used in a manner that is prejudicial to individuals based on their appearance, or to unfairly target individuals.

Extract text from images using Azure AI Vision (OCR)

- Suppose you are given thousands of images and asked to transfer the text on the images to a computer database.
- The scanned images have text organized in different formats and contain multiple languages.
- What are some ways you could complete the project in a reasonable time frame and make sure the data is entered with a high degree of accuracy?
- Using AI services, we can treat this project as an Azure AI Vision scenario and apply Optical Character Recognition (**OCR**).

Azure AI provides **two different features** that read text from documents and images:

- **Image Analysis** Optical character recognition (OCR)
- **Document Intelligence**

Image Analysis using OCR

- Use this feature for general, unstructured documents with **smaller amount of text**, or images that contain text.
- Results are returned immediately (synchronous) from a single API call.
    - To use the **Read OCR** feature, call the **ImageAnalysis** function (REST API or equivalent SDK method), passing the image URL or binary data, and optionally specifying a gender neutral caption or the language the text is written in (with a default value of **en** for English).
    - To make an OCR request to **ImageAnalysis**, specify the visual feature as **READ**
- Has functionality for analyzing images past extracting text, including object detection, describing or categorizing an image, generating smart-cropped thumbnails and more.
- **Examples** include: street signs, handwritten notes, and store signs.

Document Intelligence

- Use this service to **read small to large volumes of text from images and PDF documents**.
- The Read OCR model is available in Azure AI Vision and Document Intelligence
- This service uses context and structure of the document to improve accuracy.
- The initial function call returns an asynchronous operation ID, which must be used in a subsequent call to retrieve the results.
- Examples include receipts, articles, and invoices.

Some limitations

- Supported file formats are JPEG, PNG, BMP, PDF, and TIFF.
- For PDF and TIFF files, up to 2,000 pages (only the first two pages for the free tier) are processed.
- The file size of images must be less than 500 MB (4 MB for the free tier) with dimensions at least 50 x 50 pixels and at most 10,000 x 10,000 pixels. PDF files don't have a size limit.
- The minimum height of the text to be extracted is 12 pixels for a 1024 x 768 image, which corresponds to about 8-point font text at 150 DPI.

Analyze videos

Azure Video Indexer capabilities

- ***Facial recognition*** - detecting the presence of individual people in the image. This requires Limited Access approval.
- ***Optical character recognition*** - reading text in the video.
- ***Speech transcription*** - creating a text transcript of spoken dialog in the video.
- ***Topics*** - identification of key topics discussed in the video.
- ***Sentiment*** - analysis of how positive or negative segments within the video are.
- ***Labels*** - label tags that identify key objects or themes throughout the video.
- ***Content moderation*** - detection of adult or violent themes in the video.
- ***Scene segmentation*** - a breakdown of the video into its constituent scenes.

Extract custom insights

- Azure Video **Indexer** includes **predefined models** that can recognize well-known celebrities, do OCR, and transcribe spoken phrases into text.

- You can **extend** the recognition capabilities of Video Analyzer by creating **custom models**:
    - **People**: Add images of the **faces** of people you want to **recognize** in videos and train a model. Video **Indexer** will then **recognize** these people in all of your videos.

- **Language**: If your organization uses specific **terminology/language** that may not be in common usage, you can train a custom model to detect and transcribe it.
- **Brands**: You can train a model to recognize specific names as **brands**

Use Video Analyzer widgets and APIs

While you can perform all video analysis tasks in the Azure **Video Indexer portal**, you may want to **incorporate the service into custom applications**. There are two ways you can accomplish this:

- Azure Video Indexer widgets
- Azure Video Indexer API

Azure Video Indexer widgets

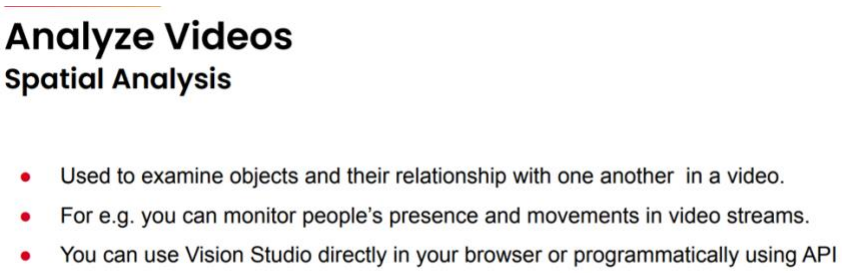
- The widgets used in the Azure Video Indexer portal to play, analyze, and edit videos can be embedded in your own custom HTML interfaces.

Azure Video Indexer API

- Azure Video Indexer provides a REST API that you can use to obtain information about your account, including an **access token**.
  https://api.videoindexer.ai/Auth/<location>/Accounts/<accountId>/AccessToken

- You can then use your token to consume the REST API and automate video indexing tasks, creating projects, retrieving insights, and creating or deleting custom models.
- For example, you can send a GET request to https://api.videoindexer.ai/<location>/Accounts/<accountId>/Videos?<accessToken>, which returns details of videos in your account, similar to the following JSON example:

Use Azure IA Vision Spatial Analysis



Implement natural language processing solutions (30–35%)

Analyze text by using Azure AI Language

- Extract key phrases
- Extract entities
- Determine sentiment of text
- Detect the language used in text
- Detect personally identifiable information (PII) in text

Process speech by using Azure AI Speech

- Implement text-to-speech
- Implement speech-to-text
- Improve text-to-speech by using Speech Synthesis Markup Language (SSML)
- Implement custom speech solutions
- Implement intent recognition
- Implement keyword recognition

Translate language

- Translate text and documents by using the Azure AI Translator service
- Implement custom translation, including training, improving, and publishing a custom model
- Translate speech-to-speech by using the Azure AI Speech service
- Translate speech-to-text by using the Azure AI Speech service
- Translate to multiple languages simultaneously

Implement and manage a language understanding model by using Azure AI Language

- Create intents and add utterances
- Create entities
- Train, evaluate, deploy, and test a language understanding model
- Optimize a language understanding model
- Consume a language model from a client application
- Backup and recover language understanding models

Create a custom question answering solution by using Azure AI Language

- Create a custom question answering project
- Add question-and-answer pairs manually
- Import sources
- Train and test a knowledge base
- Publish a knowledge base
- Create a multi-turn conversation
- Add alternate phrasing
- Add chit-chat to a knowledge base
- Export a knowledge base
- Create a multi-language question answering solution

- Natural language processing (**NLP**) solutions use **language models** to interpret the semantic meaning of written or spoken language.

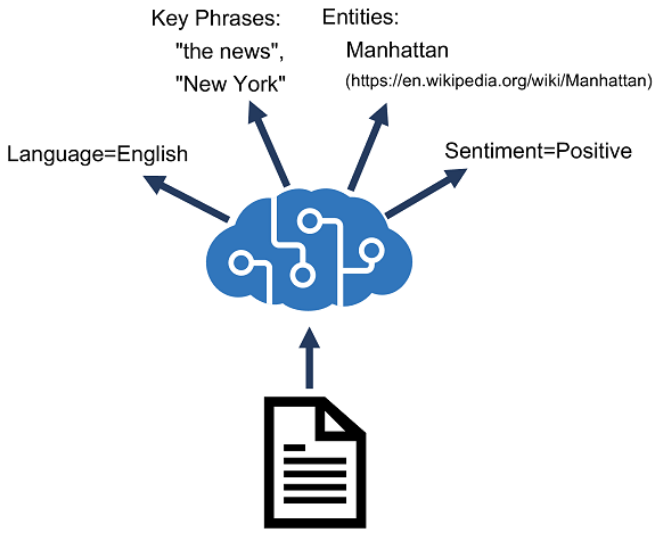
- You can use the **Language Understanding service** to build language models for your applications.
- Techniques that apply **statistical** and **semantic models** enable you to create applications that extract meaning and insights from this text-based data.

Analyze text by using Azure AI Language

Provision an Azure AI Language resource

Azure AI Language is designed to help **extract information from text**. It provides functionality for:

- ***Language detection*** - determining the language in which text is written.
- ***Key phrase extraction*** - identifying important words and phrases in the text that indicate the main points.
- ***Sentiment analysis*** - quantifying how positive or negative the text is.
- ***Named entity recognition*** - detecting references to entities, including people, locations, time periods, organizations, and more.
- ***Entity linking*** - identifying specific entities by providing reference **links to Wikipedia** articles.

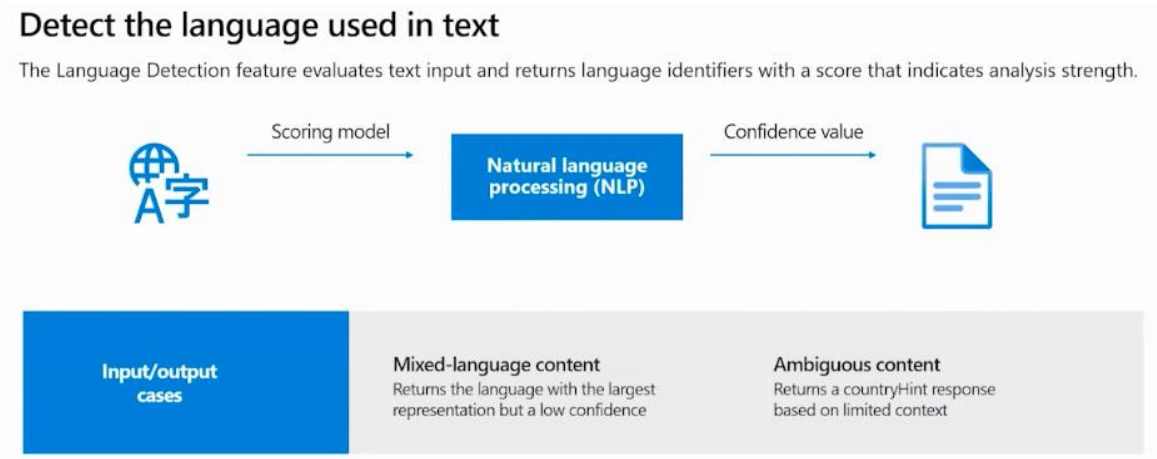


Detect language

- The Azure AI Language detection API evaluates text input and, for each document submitted, returns language identifiers with a **score** indicating the strength of the analysis.

- Language detection can work with **documents or single phrases**. It's important to note that the document size must be under 5,120 characters.

- You can use field **countryHint** to help identify the language



Example of request:

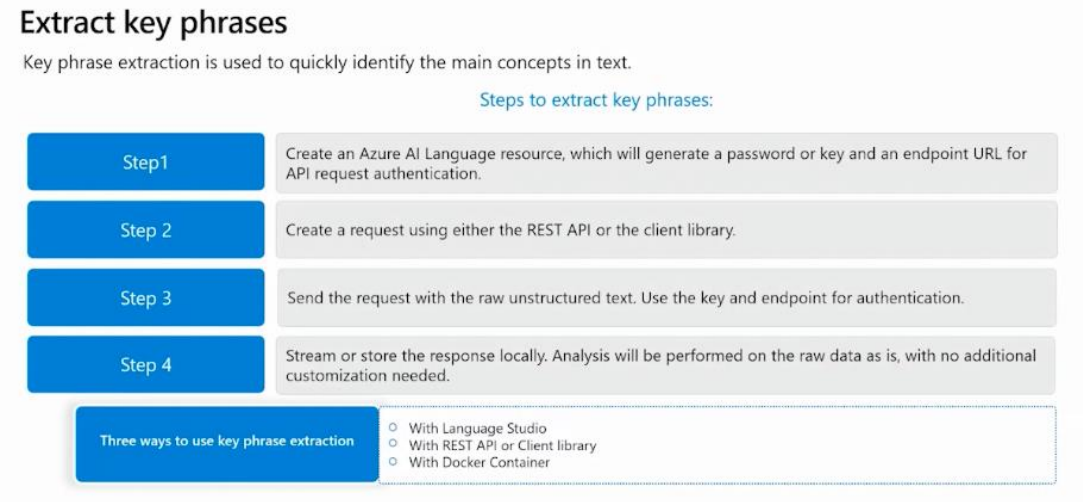
```
{
    "kind": "LanguageDetection",
    "parameters": {
        "modelVersion": "latest"
    },
    "analysisInput":{
        "documents":[
              {
                "id": "1",
                "text": "Hello world",
                "countryHint": "US"
              },
              {
                "id": "2",
                "text": "Bonjour tout le monde"
              }
        ]
    }
}
```

Example of response:

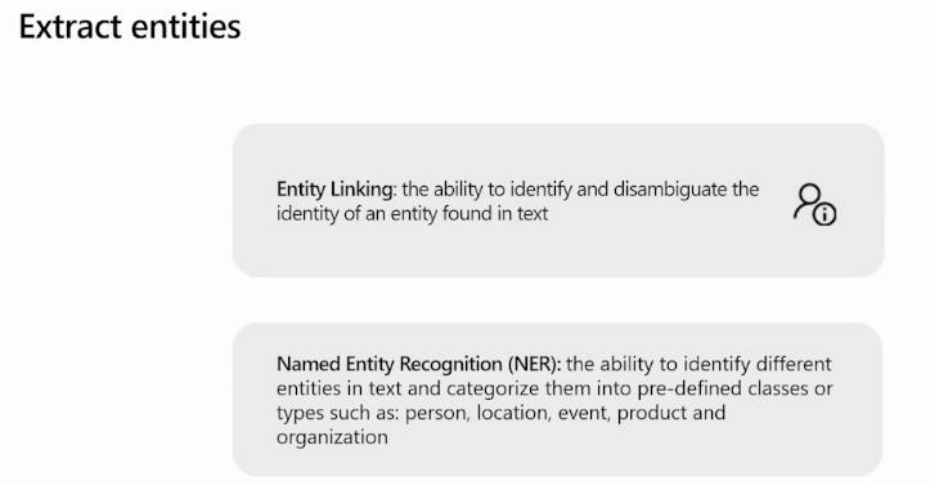
```
{   "kind": "LanguageDetectionResults",
    "results": {
        "documents": [
          {
            "detectedLanguage": {
              "confidenceScore": 1,
              "iso6391Name": "en",
              "name": "English"
            },
            "id": "1",
            "warnings": []
          },
          {
            "detectedLanguage": {
              "confidenceScore": 1,
              "iso6391Name": "fr",
              "name": "French"
            },
            "id": "2",
            "warnings": []
          }
        ],
        "errors": [],
        "modelVersion": "2022-10-01"
    }
}
```

Extract key phrases

- Key phrase extraction is the process of evaluating the text of a document, or documents, and then identifying the main points around the context of the document(s).
- Key phrase extraction works best for larger documents (the maximum size that can be analyzed is 5,120 characters).



Extract entities



Named Entity Recognition (**NER**) identifies entities that are mentioned in the text. Entities are grouped into categories and subcategories, for example:

- Person
- Location
- DateTime
- Organization
- Address
- Email

- URL

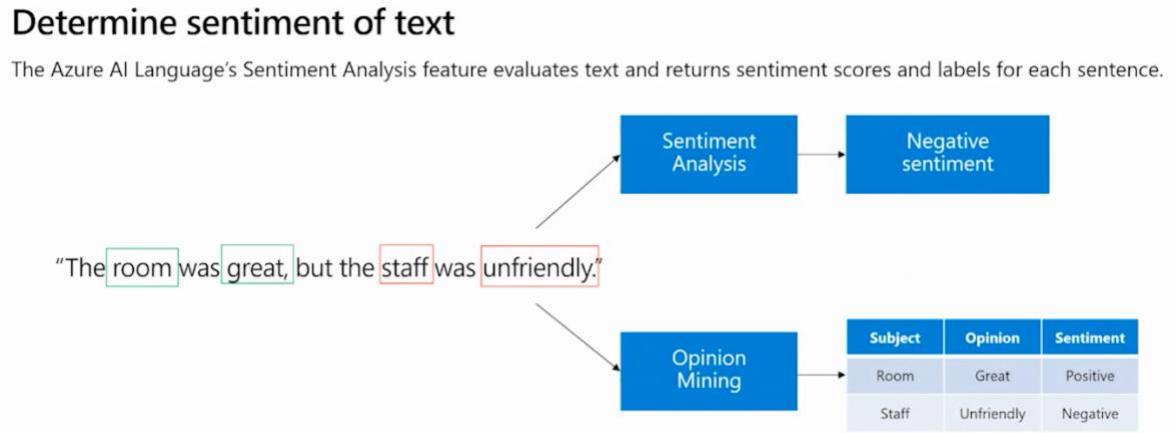
Extract linked entities

- In some cases, the same name might be applicable to more than one entity.
- For example, does an instance of the word "Venus" refer to the planet or the goddess from mythology?
- Entity linking can be used to disambiguate entities of the same name by referencing an article in a knowledge base.
- Wikipedia provides the knowledge base for the Text Analytics service. Specific article links are determined based on entity context within the text.
- For example, "I saw Venus shining in the sky" is associated with the link https://en.wikipedia.org/wiki/Venus; while "Venus, the goddess of beauty" is associated with https://en.wikipedia.org/wiki/Venus_(mythology).

Analyze sentiment

- Sentiment analysis is used to evaluate how **positive or negative a text** document is, which can be useful in various workloads, such as:
  - Evaluating a **product** by quantifying sentiment **based on reviews**.
  - Prioritizing customer service responses to correspondence received through email or social media messaging.

- When using Azure AI Language to evaluate sentiment, the response **includes overall document sentiment** and **individual sentence sentiment** for each document submitted to the service.
- Overall document sentiment is **based on sentences**:
  - If all sentences are neutral, the overall sentiment is neutral.
  - If sentence classifications include only positive and neutral, the overall sentiment is positive.
  - If the sentence classifications include only negative and neutral, the overall sentiment is negative.
  - If the sentence classifications include positive and negative, the overall sentiment is mixed.



Create question answering solutions with Azure AI Language

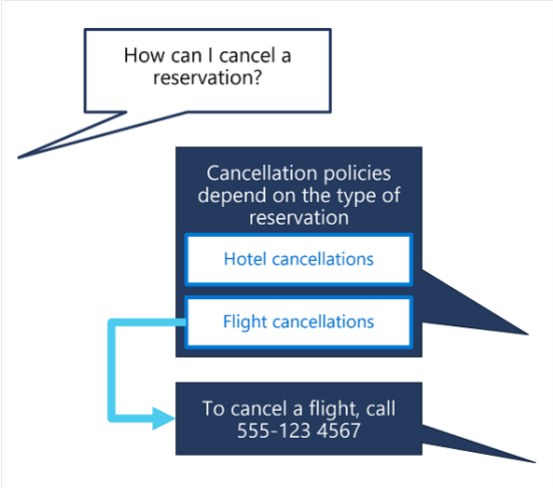
- The question answering capability of the Azure AI Language service makes it easy to build applications in which **users ask questions using natural language** and receive appropriate answers.
- This kind of solution brings conversational intelligence to a traditional frequently asked questions (FAQ) publication.
- **Azure AI Language** includes a *question answering* capability, which enables you to define a ***knowledge base*** of question-and-answer pairs that can be queried using natural language input.
- The knowledge base can be published to a REST endpoint and consumed by client applications, commonly *bots*.
- The knowledge base can be created from existing sources, including:
    - Web sites containing frequently asked question (FAQ) documentation.
    - Files containing structured text, such as brochures or user guides.
    - Built-in *chit chat* question and answer pairs that encapsulate common conversational exchanges.
- The two following services are in fact complementary. You can build comprehensive natural language solutions that combine **language understanding** models and **question answering** knowledge bases.

	Question answering	Language understanding
Usage pattern	User submits a question, expecting an answer	User submits an utterance, expecting an appropriate response or action
Query processing	Service uses natural language understanding to match the question to an answer in the knowledge base	Service uses natural language understanding to interpret the utterance, match it to an intent, and identify entities
Response	Response is a static answer to a known question	Response indicates the most likely intent and referenced entities
Client logic	Client application typically presents the answer to the user	Client application is responsible for performing appropriate action based on the detected intent

Implement multi-turn conversation

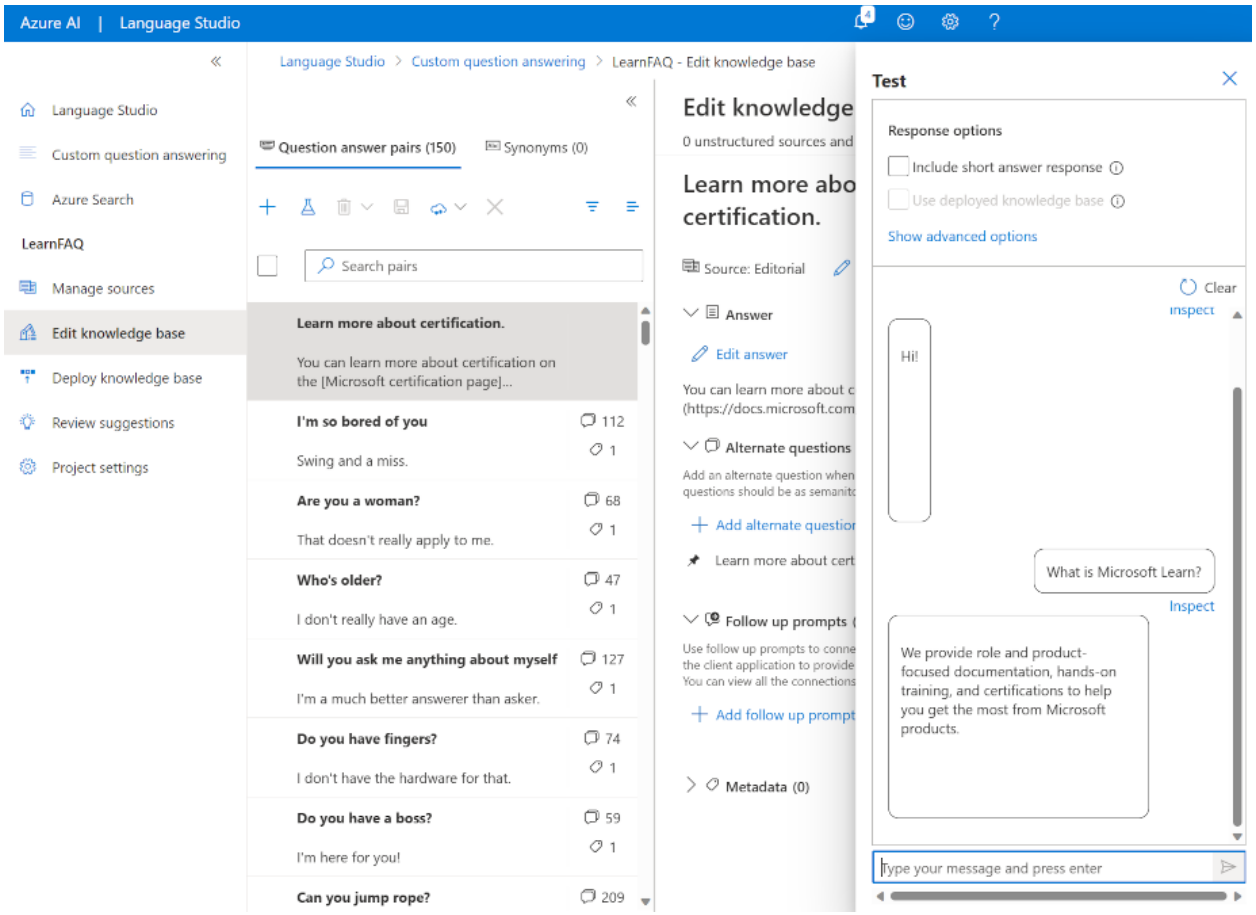
- Often, you will create an effective knowledge base that consists of individual question and answer pairs

- Sometimes you might need to ask **follow-up questions** to elicit more information from a user before presenting a definitive answer. This kind of interaction is referred to as a ***multi-turn*** conversation.



Test and publish a knowledge base

- After you have defined a knowledge base, you can train its natural language model, and test it before publishing it for use in an application or bot.
- You can **test** your model interactively in **Language Studio**



- When you are happy with the performance of your knowledge base, you can deploy it to a REST endpoint that client applications can use to submit questions and receive answers.

- You can **deploy** it directly from **Language Studio**.

- Example of request/response

```
{
  "question": "What do I need to do to cancel a reservation?",
  "top": 2,
  "scoreThreshold": 20,
  "strictFilters": [
    {
      "name": "category",
      "value": "api"
    }
  ]
}
```

```
{
  "answers": [
    {
      "score": 27.74823341616769,
      "id": 20,
      "answer": "Call us on 555 123 4567 to cancel a reservation.",
      "questions": [
        "How can I cancel a reservation?"
      ],
      "metadata": [
        {
          "name": "category",
          "value": "api"
        }
      ]
    }
  ]
}
```

Improve question answering performance

After creating and testing a knowledge base, you can **improve** its performance with:
- ***active learning***
- and by defining ***synonyms***

Use active learning

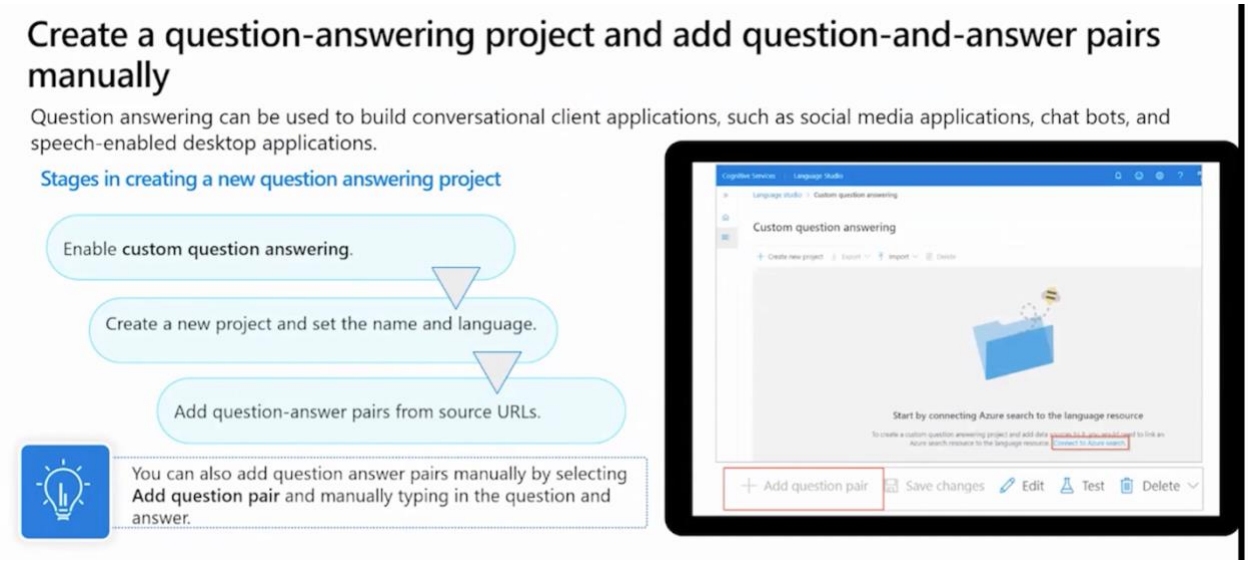
- Active learning can help you make continuous improvements to get better at answering user questions correctly over time.
- People often ask questions that are phrased differently, but ultimately have the same meaning.
- Active learning can help in situations like this because it enables you to consider alternate questions to each question and answer pair. Active learning is enabled by default.

Define synonyms

- Synonyms are useful when questions submitted by users might include multiple different words to mean the same thing.
- For example, a travel agency customer might refer to a "reservation" or a "booking".
- By defining these as synonyms, the question answering service can find an appropriate answer regardless of which term an individual customer uses.

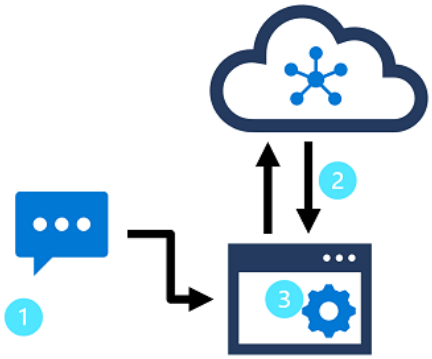
- To define synonyms, you use the REST API to submit synonyms in the following JSON format:

```
{
    "synonyms": [
        {
            "alterations": [
                "reservation",
                "booking"
                ]
        }
    ]
}
```



Build a conversational language understanding (CLU) model

- The Azure AI Language **Conversational Language Understanding** service (**CLU**) enables you to train a model that apps can use to extract meaning from natural language.
- A common design pattern for a natural language understanding solution looks like this:



  - In this design pattern:
    - An app accepts natural language input from a user.
    - A language model is used to determine semantic meaning (the **user's *intent***).
    - The app performs an appropriate action.

  - **Azure AI Language** enables developers to build apps based on language models that can be trained with a relatively small number of samples to discern a user's intended meaning.

Azure AI Language service features fall into two categories:

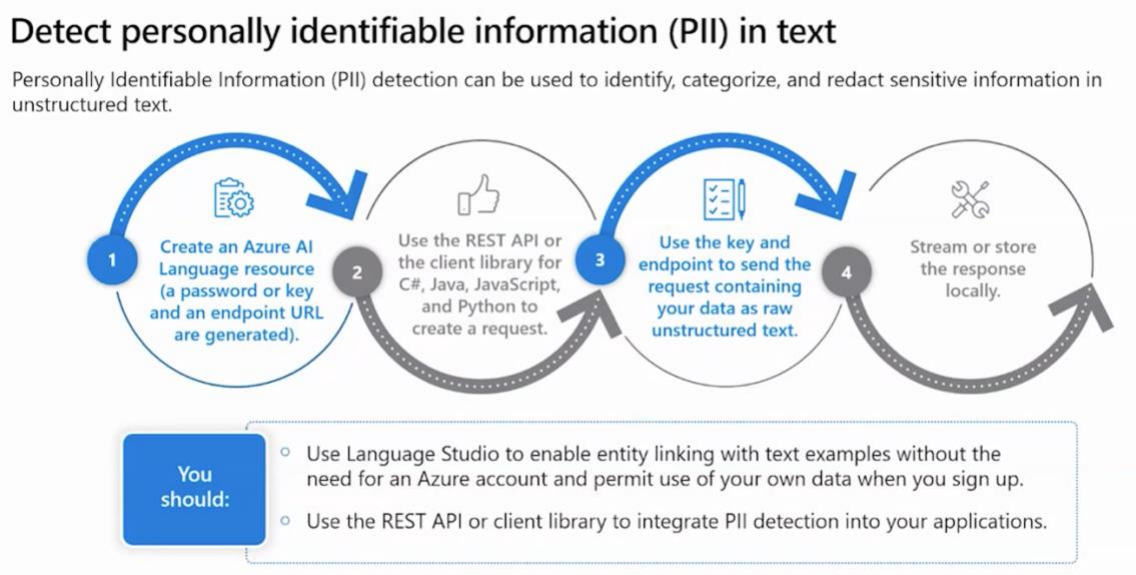
- **Pre-configured** features,
- and **Learned** features

Pre-configured features

The Azure AI Language service provides certain features without any model labeling or training. Once you create your resource, you can send your data and use the returned results within your app.

The following **features** are all **pre-configured**:

- **Summarization** is available for both documents and conversations, and will summarize the text into key sentences that are predicted to encapsulate the input's meaning.
- **Named entity** recognition can extract and identify entities, such as people, places, or companies
- **Personally identifiable information** (**PII**) detection allows you to **identify**, categorize, and redact information that could be considered sensitive, such as **email** addresses, **home** addresses, IP addresses.
- **Key phrase extraction** is a feature that quickly pulls the main concepts out of the provided text.
- **Sentiment analysis** identifies how positive or negative a string or document is.
- **Language detection** takes one or more documents, and identifies the language for each.



Learned features

- Conversational language understanding (**CLU**) is one of the **core custom features** offered by Azure AI Language.
- CLU helps users to build custom natural language understanding models to **predict overall intent and extract** important information from incoming utterances. CLU **does require data to be tagged by the user** to teach it how to predict intents and entities accurately.

- **Custom entity recognition** takes custom labeled data and extracts specified entities from unstructured text.
- **Custom text classification** enables users to classify text or documents as custom defined groups.

- **Question answering is a mostly pre-configured** feature that provides answers to questions provided as input. The data to answer these questions comes from documents like FAQs or manuals.

Define intents, utterances, and entities

- *Utterances* are the phrases that a user might enter when interacting with an application that uses your language model.
- An *intent* represents a task or action the user wants to perform, or more simply the *meaning* of an utterance.
- You create a model by defining intents and associating them with one or more utterances.

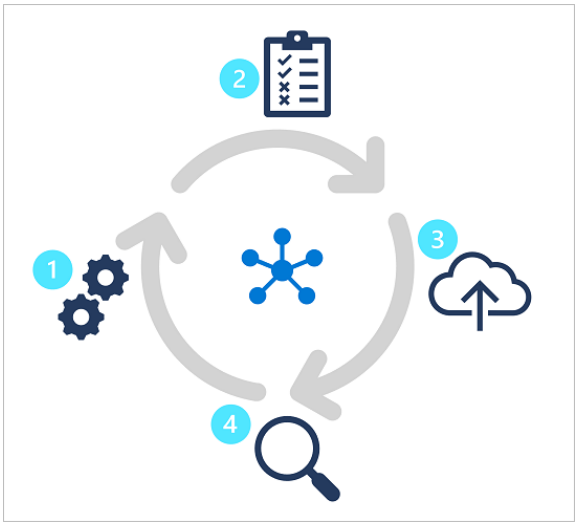
For example, consider the following intent and associated utterances:

- **GetTime**: (intent)
  - "What time is it?"
  - "What is the time?"
  - "Tell me the time"

- You can create your own language models by defining all the intents and utterances it requires,
- but often you can use prebuilt components to detect common entities such as numbers, emails, URLs, or choices.

Train, test, publish, and review a conversational language understanding model

Creating a model is an iterative process with the following activities:

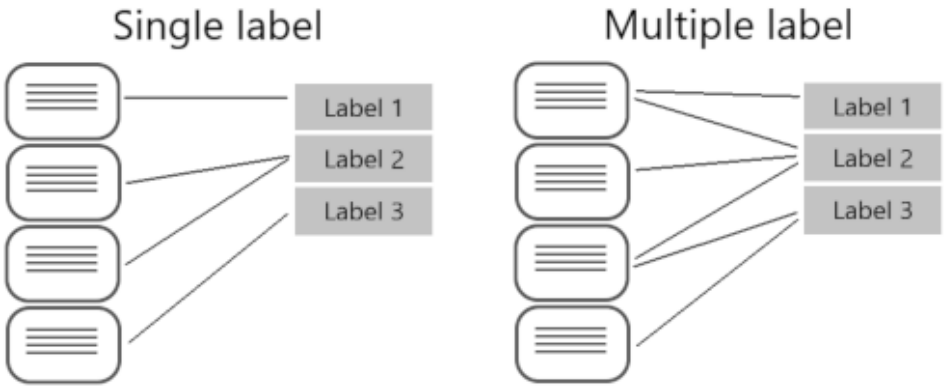


1. Train a model to learn intents and entities from sample utterances.
2. Test the model interactively or using a testing dataset with known labels
3. Deploy a trained model to a public endpoint so client apps can use it
4. Review predictions and iterate on utterances to train your model

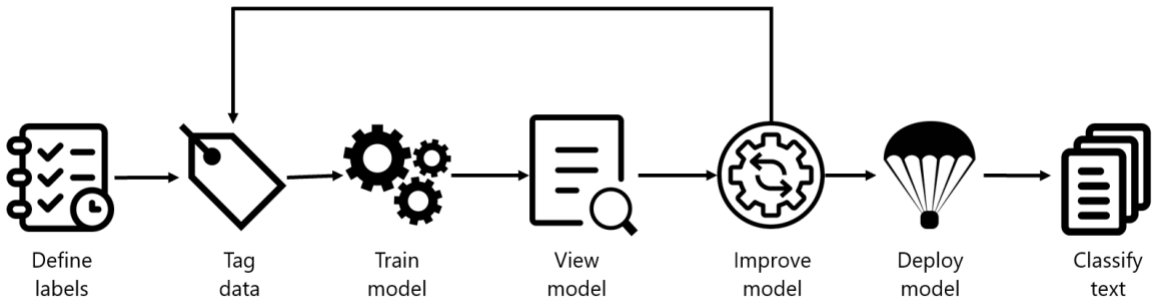
Create a custom text classification solution

- Part of **NLP** is the **ability to classify** text, and Azure provides ways to classify text including sentiment, language, and custom categories defined by the user.

- **Custom text classification** falls into two types of projects:
    - **Single label classification** - you can assign only one class to each file. Following the above example, a video game summary could only be classified as "Adventure" or "Strategy".
    - **Multiple label classification** - you can assign multiple classes to each file. This type of project would allow you to classify a video game summary as "Adventure" or "Adventure and Strategy".

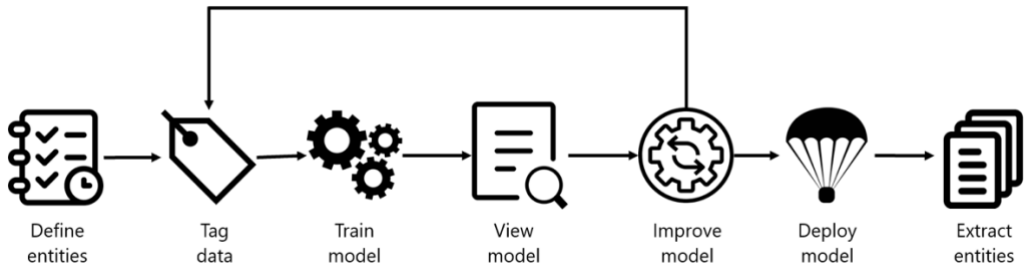


Understand how to build text classification projects



Custom named entity recognition (Custom NER)

- Custom *named entity recognition* (NER), otherwise known as custom entity extraction, offered by Azure AI Language service.
- Custom NER enables developers to **extract predefined entities** from text documents, without those documents being in a known format - such as legal agreements or online ads.
- An **entity** is a person, place, thing, event, skill, or value.
- Custom NER is an Azure API service that looks at documents, identifies, and extracts user defined entities.
- These entities could be anything from names and addresses from bank statements to knowledge mining to improve search results.
- Custom NER is part of **Azure AI Language** in Azure AI services.
- Azure AI Language provides certain **built-in entity recognition**, to recognize things such as a person, location, organization, or URL.
- **Built-in NER** allows you to set up the service with minimal configuration, and extract entities.
- To call a built-in NER, create your service and call the endpoint for that NER service like this: <YOUR-ENDPOINT>/language/**analyze-text**/jobs?api-version=<API-VERSION>



- To submit an **extraction** task, the API requires the JSON body to specify **which task to execute**.

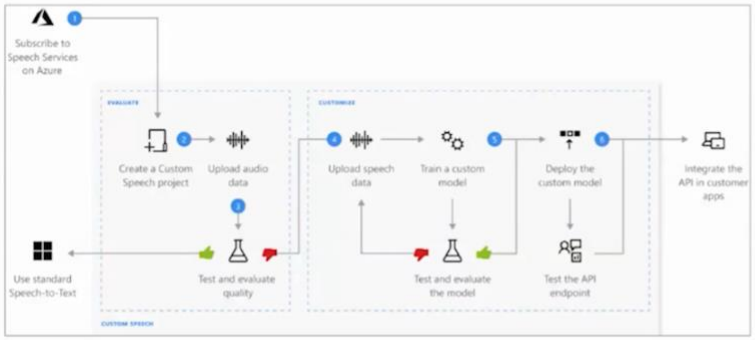
- For custom NER, the task for the JSON payload is **CustomEntityRecognition**.
- Your payload will look similar to the following JSON:

```
{
    "displayName": "string",
    "analysisInput": {
        "documents": [
            {
                "id": "doc1",
                "text": "string"
            },
            {
                "id": "doc2",
                "text": "string"
            }
        ]
    },
    "tasks": [
        {
            "kind": "CustomEntityRecognition",
            "taskName": "MyRecognitionTaskName",
            "parameters": {
            "projectName": "MyProject",
            "deploymentName": "MyDeployment"
            }
        }
    ]
}
```

- Project limits:
    - **Training** - at least 10 files, and not more than 100,000
    - **Deployments** - 10 deployment names per project
    - **APIs**
        - **Authoring** - this API creates a project, trains, and deploys your model. Limited to 10 POST and 100 GET per minute
        - **Analyze** - this API does the work of actually extracting the entities; it requests a task and retrieves the results. Limited to 20 GET or POST
    - **Projects** - only 1 storage account per project, 500 projects per resource, and 50 trained models per project
    - **Entities** - each entity can be up to 500 characters. You can have up to 200 entity types.
- **Labeling**, or tagging, your data correctly is an important part of the process to create a custom entity extraction model. **Labels identify examples of specific entities** in text used to train the model.
- **Training and evaluating your model** is an **iterative** process of adding data and labels to your training dataset to teach the model more accurately.
- To know what types of data and labels need to be improved, Language Studio provides scoring in the **View model details** page on the left hand pane.
- **Scores** are available both per entity and for the model as a whole. You may find an entity scores well, but the whole model doesn't.
- Ideally we want our model **to score well in both precision and recall**, which means the entity recognition works well.
- ***Confusion matrix*** provides a visual table of all the entities and how each performed, giving a complete view of the model and where it's falling short.

Implement custom speech solutions

Test a baseline or custom language model for Word Error Rate (WER) using accuracy tests or a custom acoustic model.



Reduce WER score each iteration

$$WER = \frac{I + D + S}{N} * 100\%$$

Ready to use	5-10%
Acceptable	20%
Needs additional testing	30%+

Implement intent recognition

Intent recognition can be used to determine what the user wants to initiate or do.

Use **Pattern matching** for a quick offline solution.

Steps

1. Create code and speech configuration.
2. Initialize an intent recognizer and declare entities as intents.
3. Enable recognition of intent.
4. Instruct code to stop upon intent recognition.
5. Display recognition results.
6. Build and run the application.

Use **CLU** to build a custom natural language understanding model to predict the intent of incoming instances.

Steps

1. Create a project by importing a JSON file.
2. Train your model.
3. Choose your training mode and data splitting method and select **Train**.
4. Deploy your model.
5. Use the model to recognize intents from a microphone.

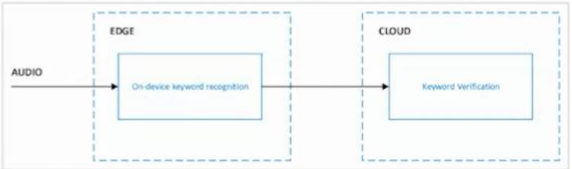
Implement keyword recognition

Keyword recognition can be used to detect a word or short phrase within any audio.

To implement keyword recognition

1. Create a new project in **Speech Studio** and enter the project details.
2. To create a custom keyword, select **Create a new model** and enter the **Name**, **Description**, and **Keyword** and select **Next**.
3. Listen to the candidate pronunciations generated by the portal and remove incorrect ones, if any. Select **Next**.
4. Select a model type and select **Create**.
5. Select **Tune** from the collapsible menu to download the model.
6. This model can now be used on any file to detect the keyword.

A typical keyword system consists of:



Keyword models can be:

- **Basic**: They may not have optimal accuracy characteristics and are suited for rapid prototyping purposes.
- **Advanced**: They improve accuracy characteristics and are suited for product integration purposes.

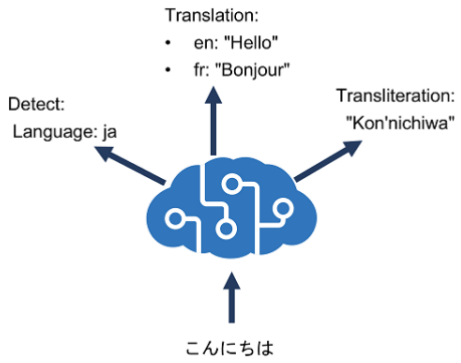
Translate language

Translate text with Azure AI Translator service

- The Translator service enables you to create intelligent apps and services that can translate text between languages.
- The Azure AI Translator provides an API for translating text between 90 supported languages.

Azure AI Translator provides a multilingual text translation API that you can use for:

- Language detection.
- One-to-many translation.
- Script transliteration (converting text from its native script to an alternative script).



Provision an Azure AI Translator resource

- To use the Azure AI Translator service, you can provision a single-service Azure AI **Translator resource**, or you can use the **Text Analytics API in a multi-service** Azure AI Services resource.
- After you have provisioned a suitable resource in your Azure subscription, you can use the **location** where you deployed the resource and one of its **subscription keys** to call the Azure AI Translator APIs from your code.
- You can call the APIs by submitting requests in JSON format to the **REST** interface, or by using any of the available programming language-specific **SDKs**.

Azure AI Translator capabilities

Language detection

- You can use the **Detect** function of the REST API to detect the language in which text is written.
- For example, you could submit the following text to the https://api.cognitive.microsofttranslator.com/**detect**?api-version=3.0 endpoint using curl.
-

Here's the text we want to translate: { 'Text' : 'こんにちは' }
And the response:

```
[
  {
    "language": "ja",
    "score": 1.0,
    "isTranslationSupported": true,
    "isTransliterationSupported": true


  }
]
```


Translation

- To translate text from one language to another, use the **Translate** function; specifying a single **from** parameter to indicate the source language, and one or more **to** parameters to specify the languages into which you want the text translated.

Transliteration

- we can submit the Japanese text to the **Transliterate** function with a **fromScript** parameter of **Jpan** and a **toScript** parameter of **Latn**:

Translation options

- Word alignment: specify the **includeAlignment** parameter with a value of **true**
- Sentence length: get this information by setting the **includeSentenceLength** parameter to **true**.
- Profanity filtering: specifying the **profanityAction** parameter, which can have one of the following values:
    - **NoAction**: Profanities are translated along with the rest of the text.
    - **Deleted**: Profanities are omitted in the translation.
    - **Marked**: Profanities are indicated using the technique indicated in the **profanityMarker** parameter

Define custom translations

- You may need to develop a translation solution for businesses or industries in that have specific vocabularies of terms that require custom translation.
- To solve this problem, you can create a custom model that maps your own sets of source and target terms for translation.
- To create a custom model, use the Custom Translator portal

Process speech by using Azure AI Speech

- The Azure AI Speech service enables you to build speech-enabled applications.
- Azure AI Speech provides APIs that you can use to build speech-enabled applications:
    - **Speech to text**: An API that enables *speech recognition* in which your application can accept spoken input.
    - **Text to speech**: An API that enables *speech synthesis* in which your application can provide spoken output.

- **Speech Translation**: An API that you can use to translate spoken input into multiple languages.
- **Speaker Recognition**: An API that enables your application to recognize individual speakers based on their voice.
- **Intent Recognition**: An API that uses conversational language understanding to determine the semantic meaning of spoken input.

Provision an Azure resource for speech

- Before you can use Azure AI Speech, you need to create an Azure AI Speech resource in your Azure subscription.
- You can use either a dedicated Azure **AI Speech resource** or a **multi-service Azure AI** Services resource.

After you create your resource, you'll need the following information to use it from a client application through one of the supported SDKs:

- The ***location*** in which the resource is deployed (for example, *eastus*)
- One of the ***keys*** assigned to your resource.

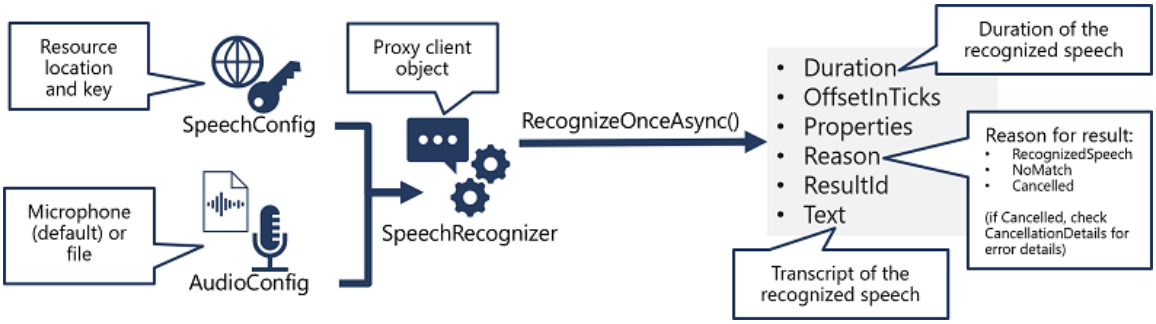
Use the Azure AI Speech to Text API

- The Azure AI Speech service supports speech recognition through two REST APIs:

  - The **Speech to text** API, which is the primary way to perform speech recognition.
  - The **Speech to text Short Audio** API, which is optimized for short streams of audio (up to 60 seconds).

- You can use either API for interactive speech recognition, depending on the expected length of the spoken input.

- You can also use the **Speech to text** API for ***batch transcription***, transcribing multiple audio files to text as a batch operation.



1. **SpeechConfig**: Contains the location and key for your Azure AI Speech resource.
2. **AudioConfig**: Defines the audio input source, which can be the system microphone or an audio file.

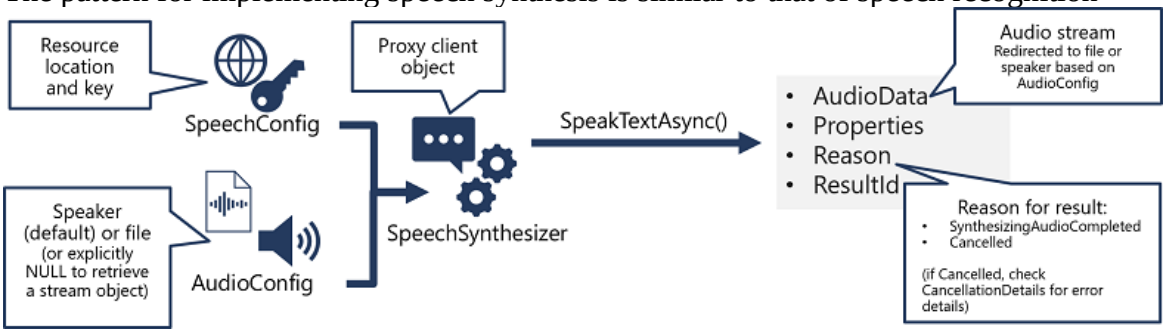
3. **SpeechRecognizer**: Created using SpeechConfig and AudioConfig, it acts as a client for the Speech-to-Text API.
4. **RecognizeOnceAsync()**: Asynchronously transcribes a single spoken utterance.
5. **SpeechRecognitionResult**: Includes properties like Duration, OffsetInTicks, Properties, Reason, ResultId, and Text. The Reason property indicates if the transcription was successful.

Use the text to speech API

Similarly to its **Speech to text** APIs, the Azure AI Speech service offers other REST APIs for speech synthesis:

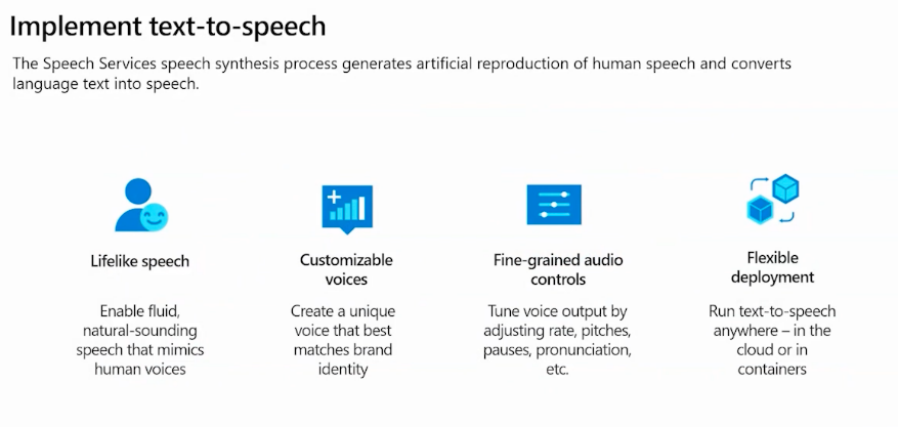
- The **Text to speech** API, which is the primary way to perform speech synthesis.
- The **Batch synthesis** API, which is designed to support batch operations that convert large volumes of text to audio - for example to generate an audio-book from the source text.

The pattern for implementing speech synthesis is similar to that of speech recognition



1. **SpeechConfig**: Contains the location and key for connecting to your Azure AI Speech resource.
2. **AudioConfig**: Defines the audio input source, which can be the system microphone or an audio file.
3. **SpeechRecognizer**: Created using SpeechConfig and AudioConfig, it acts as a client for the Speech-to-Text API.
4. **RecognizeOnceAsync()**: Asynchronously transcribes a single spoken utterance.
5. **SpeechRecognitionResult**: Includes properties like Duration, OffsetInTicks, Properties, Reason, ResultId, and Text. The Reason property indicates if the transcription was successful.

When speech has been successfully synthesized, the **Reason** property is set to the **SynthesizingAudioCompleted** enumeration and the **AudioData** property contains the audio stream

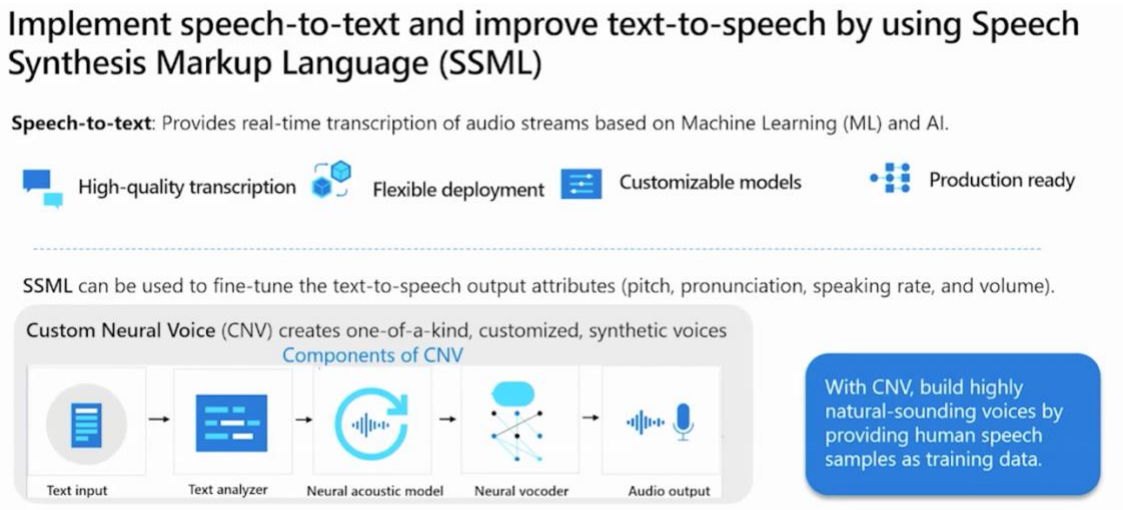


Configure audio format and voices

- When synthesizing speech, you can use a **SpeechConfig** object to customize the audio that is returned by the Azure AI Speech service.
- The Azure AI Speech service supports multiple **output formats for the audio** stream that is generated by speech synthesis:
  - Audio file type
  - Sample-rate
  - Bit-depth
- There are two kinds of voice that you can use:
  - *Standard voices* - synthetic voices created from audio samples.
  - *Neural voices* - more natural sounding voices created using deep neural networks.
- Voices are identified by names that indicate a locale and a person's name - for example en-GB-George.
- To specify a voice for speech synthesis in the **SpeechConfig**, set its **SpeechSynthesisVoiceName** property to the voice you want to use

Use Speech Synthesis Markup Language

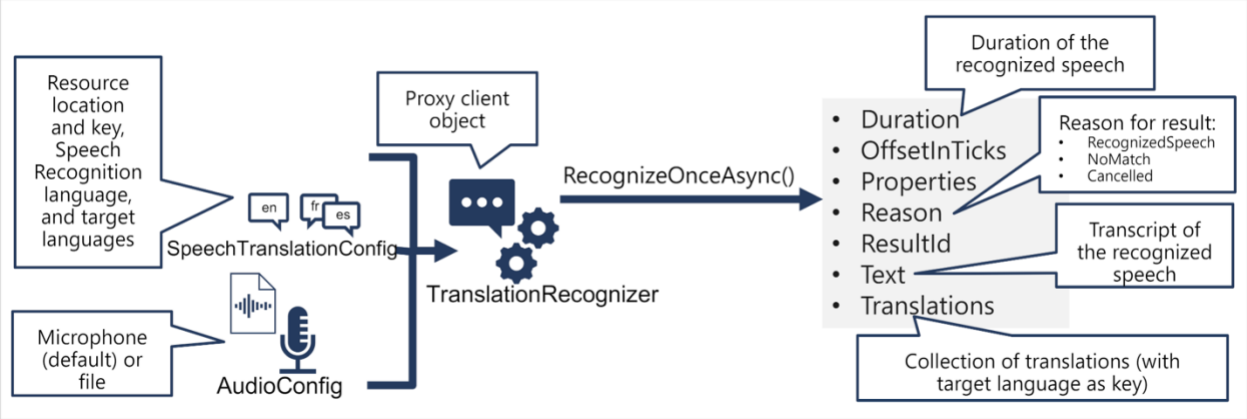
- Azure AI Speech SDK enables you to submit plain text to be synthesized into speech (for example, by using the **SpeakTextAsync()** method),
- The Azure AI Speech service also supports an XML-based syntax for describing characteristics of the speech you want to generate.
- This **Speech Synthesis Markup Language** (SSML) syntax offers greater control over how the spoken output sounds, enabling you to:
  - Specify a speaking style, such as "excited" or "cheerful" when using a neural voice.
  - Insert pauses or silence.
  - Specify *phonemes* (phonetic pronunciations), for example to pronounce the text "SQL" as "sequel".
  - Adjust the *prosody* of the voice (affecting the pitch, timbre, and speaking rate).
  - Use common "say-as" rules, for example to specify that a given string should be expressed as a date, time, telephone number, or other form.
  - Insert recorded speech or audio, for example to include a standard recorded message or simulate background noise.



Translate speech with the Azure AI Speech service

- Translation of speech builds on speech recognition by recognizing and transcribing spoken input in a specified language and returning translations of the transcription in one or more other languages.
- You can use either a dedicated Azure AI **Speech resource** or **a multi-service** Azure AI Services resource.
- After creating your Azure resource, you'll need the following information to use it from a client application through one of the supported SDKs:
    - The **location** in which the resource is deployed (for example, eastus)
    - One of the **keys** assigned to your resource.

Translate speech to text

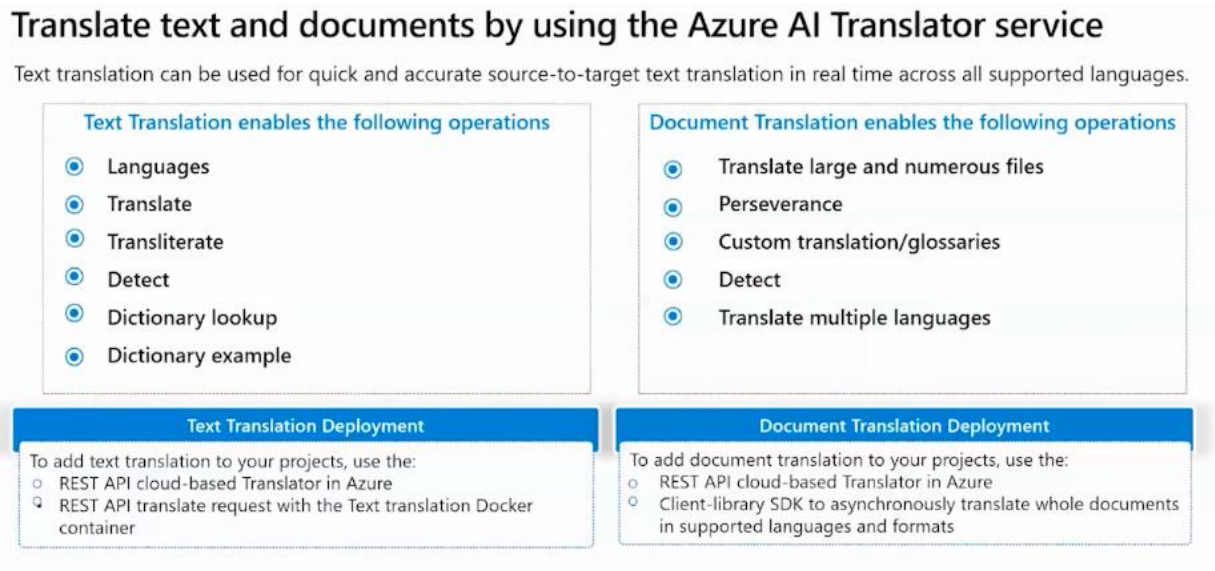


Synthesize translations

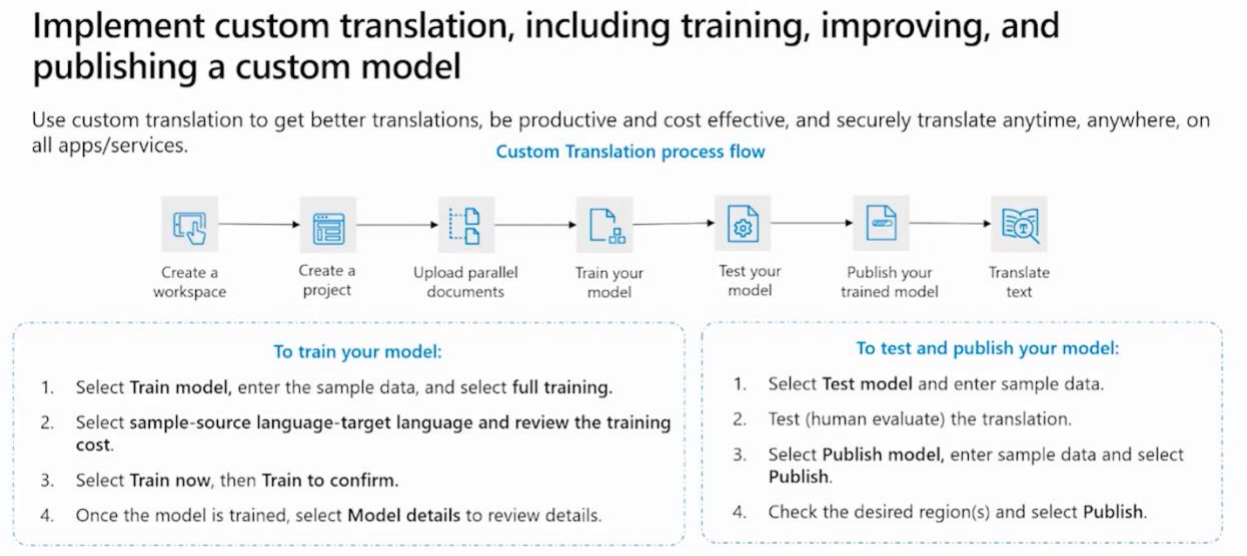
- The **TranslationRecognizer** returns translated transcriptions of spoken input - essentially translating audible speech to text.

- You can also synthesize the translation as speech to create **speech-to-speech** translation solutions. There are two ways you can accomplish this:

    - **Event-based** synthesis: When you want to perform **1:1 translation** (translating from one source language into a single target language), you can use event-based synthesis to capture the translation as an audio stream.
    - **Manual** synthesis: You can use manual synthesis to generate audio **translations for one or more target languages**.

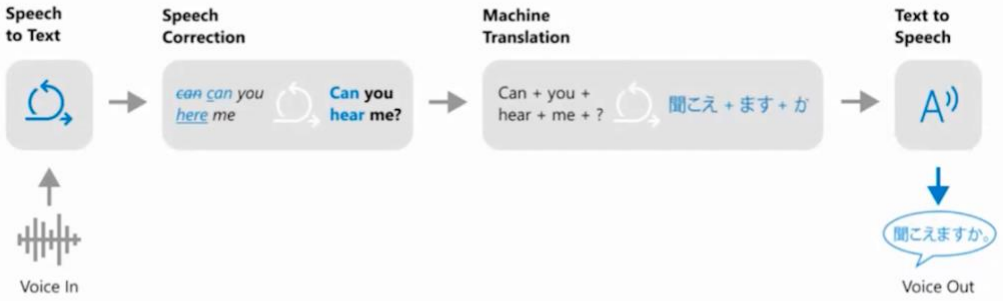


Implement and manage a language understanding model by using Azure AI Language



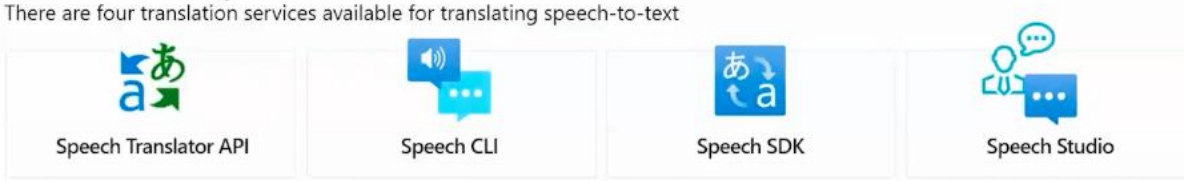
Translate speech-to-speech by using the Azure AI Speech service

The speech-to-speech service can accept audio input and translate that audio to a different language.



Translate speech-to-text and multiple languages simultaneously by using the Azure AI Speech service

There are four translation services available for translating speech-to-text



Translating documents from and to multiple languages

- Use the Document Translation feature of Azure AI Translator to translate entire documents, or batches of documents, in various file formats.
- Use it to **translate documents into multiple languages** with just a single request.
- Use the **Autodetect** feature to **translate documents with content in multiple languages into your target language.** (The Azure blob storage container stores the documents within a specific folder.)

Create intents, entities and add utterances

Conversational Language Understanding (CLU) makes use of three key aspects for understanding language.

Intents (required)
A task or action the user wants to do

Entities
A word or phrase inside the utterance to extract

Utterance
A spoken word or statement

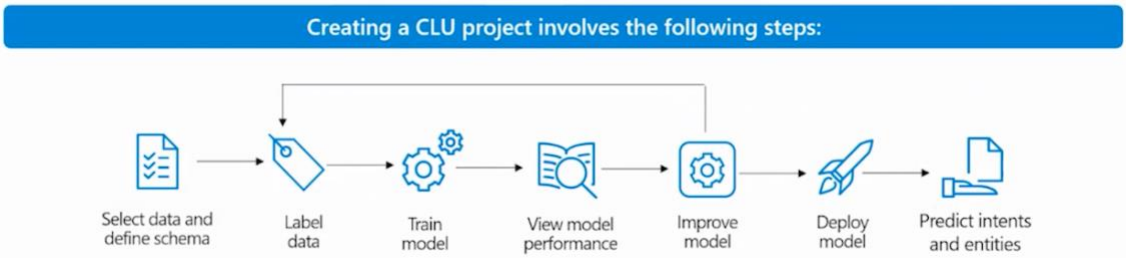
Entity component types	
Learned component	Enables prediction based on context learned through labeling utterances
List component	Represents a fixed set of related words with their synonyms
Prebuilt component	Allows selection from a library of common types like numbers, date-times, and names
Regex component	Matches regular expressions to capture consistent patterns

To add entities:

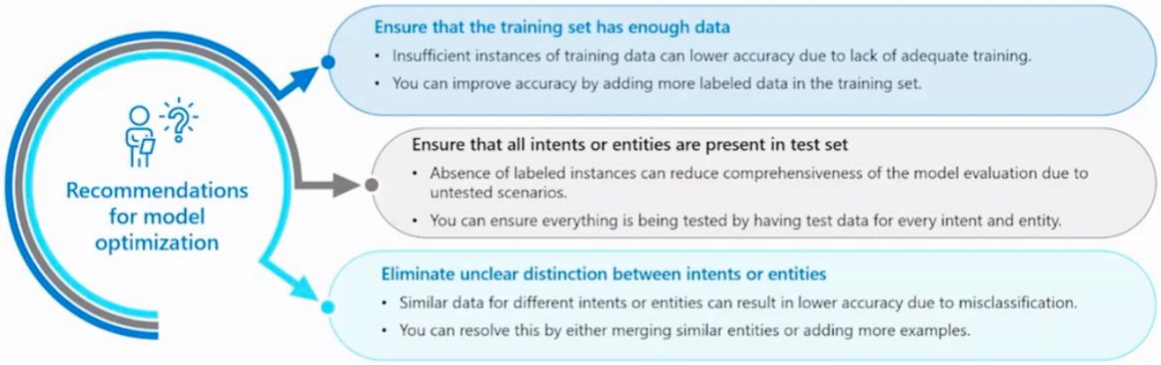
- Navigate to the **Entities** pivot.
- Select **Add** and type the entity name.
- In the entity details page, define the composition settings for the entity.
- You can add a **Learned**, **Prebuilt**, or **List** component to your entity.

Train, evaluate, deploy, and test a language understanding model

Conversational language understanding (CLU) can be used to build custom natural language understanding models which predict the intention of an incoming utterance and extract any important information from it.

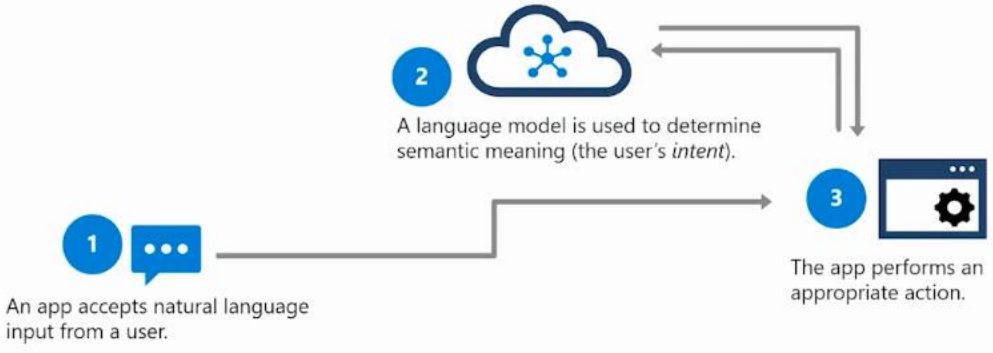


Optimize a Language Understanding model



Consume a language model from a client application

Azure AI-Language provides Natural Language Processing (NLP) features for understanding and analyzing text.



Import and export language understanding models

Export your language understanding models:

- Create a **POST** request to export your project.
- Use the Request URL to create your API request.
- Use the **Ocp-Apim-Subscription-Key** header to authenticate your request.
- Use the **GET** query to query the status of the export job.
- Submit a **GET** request to view the results of the export job.

Import your language understanding models:

- Create a **POST** request to export your project.
- Use the Request URL to create your API request.
- Use the **Ocp-Apim-Subscription-Key** header to authenticate your request.
- Enter the body text in your JSON file.
- Use the **GET** query to query the status of the import job.
- Keep polling this endpoint until the status parameter changes to **succeeded**.

Import sources

You can use Azure Question Answering to add question and answer pairs from different documents:

PRODUCT MANUALS. BROCHURES, GUIDELINES, PAPERS AND FILES

When question answering processes a document such as a manual or a brochure, it extracts the headings and subheadings as questions and the subsequent content as answers.

UNSTRUCTURED DOCUMENT SUPPORT AND MORE

Unstructured documents having free flowing content, structured question answering document (TXT, TSV and XLS) and existing projects are also supported by custom question answering.

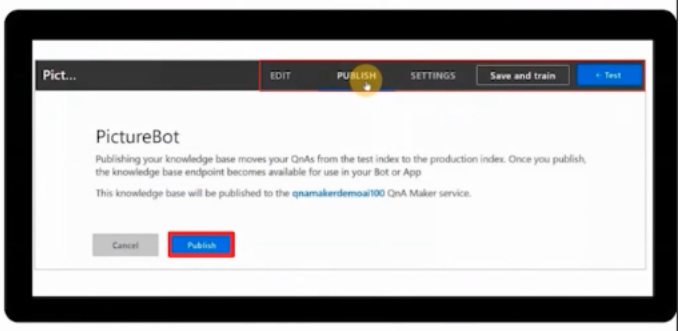
STRUCTURED QUESTION ANSWER DOCUMENT, AND OTHER FILE FORMATS

- For DOC files, question answer form of alternating questions and answers per line, one question per line followed by its answer in the following line.
- .txt, .tsv or .xls files can either be plain text or can have content in RTF or HTML.

Train, test, and publish a knowledge base

Train and test a knowledge base

1. In the **Knowledge base** page, the contents of the source document will be imported under **Questions**. You may add, delete, or edit these contents.
2. Select the **Save and Train** button, and subsequently select **Test**.
3. A test version of the Knowledge base will open, in which you can Inspect each element by using the **Inspect** button.
4. Once you are satisfied with the results, you can publish the knowledge base by selecting the **Publish** tab and then selecting **Publish**.



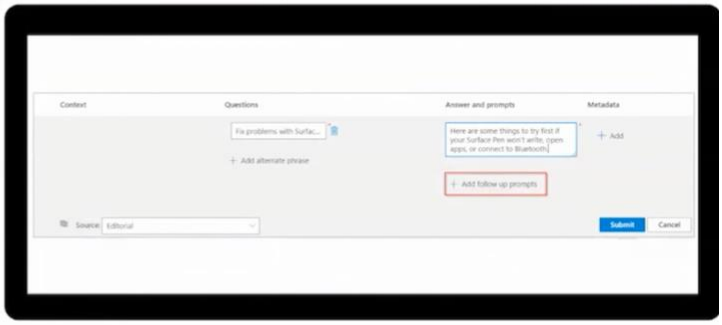
Create a custom question answering solution by using Azure AI Language

Create a multi-turn conversation

Question answering provides multi-turn prompts, which can connect question and answer pairs.

You can add a follow-up prompt to a newly created question pair by following these steps:

1. Select **Add follow-up prompts**.
2. Fill in the details of the prompt.
3. Select **Create link to new pair** and then select **Done**.
4. Select **Save changes**.
5. You can add multiple prompts to the same question by repeating the process.

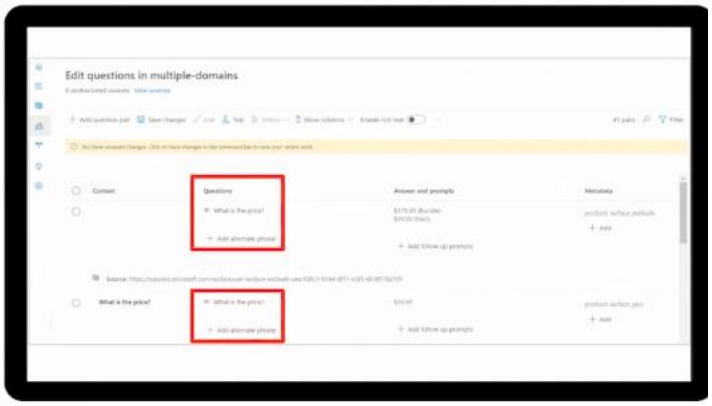


Add alternate phrasing

- You can add alternate questions with differences in the sentence structure and word-style to improve the likelihood of a match with a user query.
- These questions are useful when the same question may be asked in multiple ways.

Examples:

Original Query	Alternate query	Change
Is parking available?	Do you have a car park?	sentence structure
Hi	Hey there Yo	word-style or slang

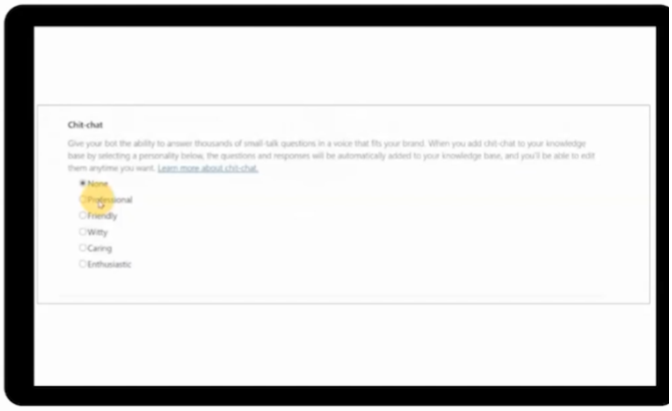


Add chit-chat to a knowledge base

You can use chit-chat to make your bot conversational and engaging by selecting a personality for your conversation.

Give your bot the ability to answer questions in a way that fits your brand.

Set a personality for your conversation and have question and answers automatically added to your knowledge base.



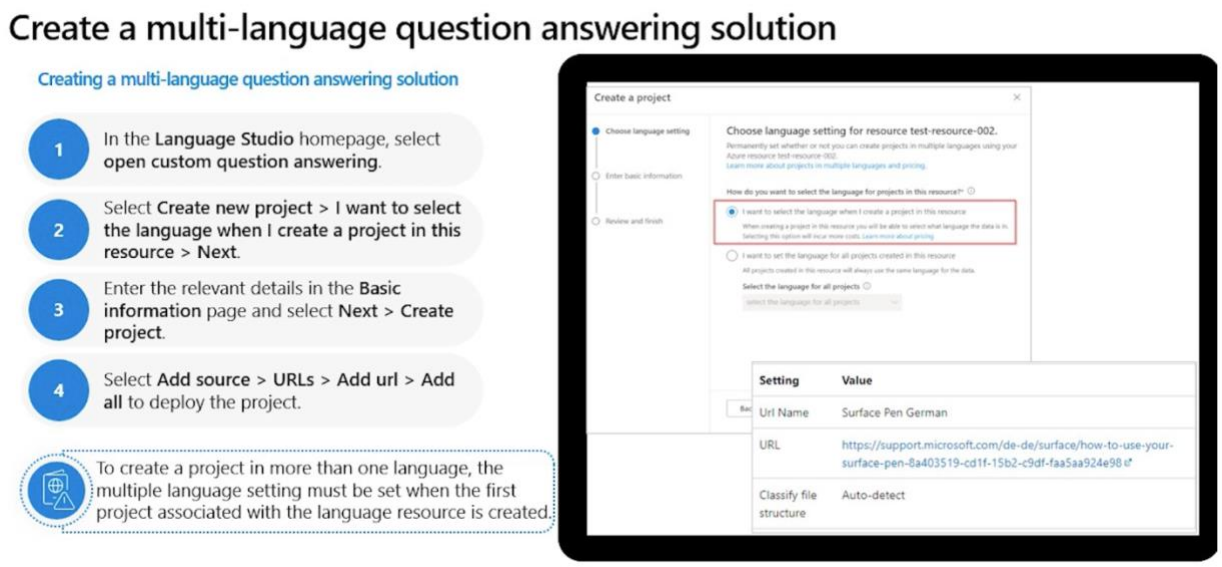
Export a knowledge base

You may want to export a knowledge base for several reasons:

- To implement a backup and restore process
- To integrate with your CI/CD pipeline
- To move your data to different regions

Steps to export a knowledge base

Step 1	From the **Answer questions** section, select **Open custom question answering**.
Step 2	Select the project you want to export and then select **Export.**
Step 3	Select if you want to export the file as an Excel file or a TSV file. The file is saved locally as a zip file.



Implement knowledge mining and document intelligence solutions (10-15%)

Implement an Azure AI Search solution

- Provision an Azure AI Search resource
- Create data sources
- Create an index
- Define a skillset
- Implement custom skills and include them in a skillset
- Create and run an indexer
- Query an index, including syntax, sorting, filtering, and wildcards
- Manage Knowledge Store projections, including file, object, and table projections

Implement an Azure AI Document Intelligence solution

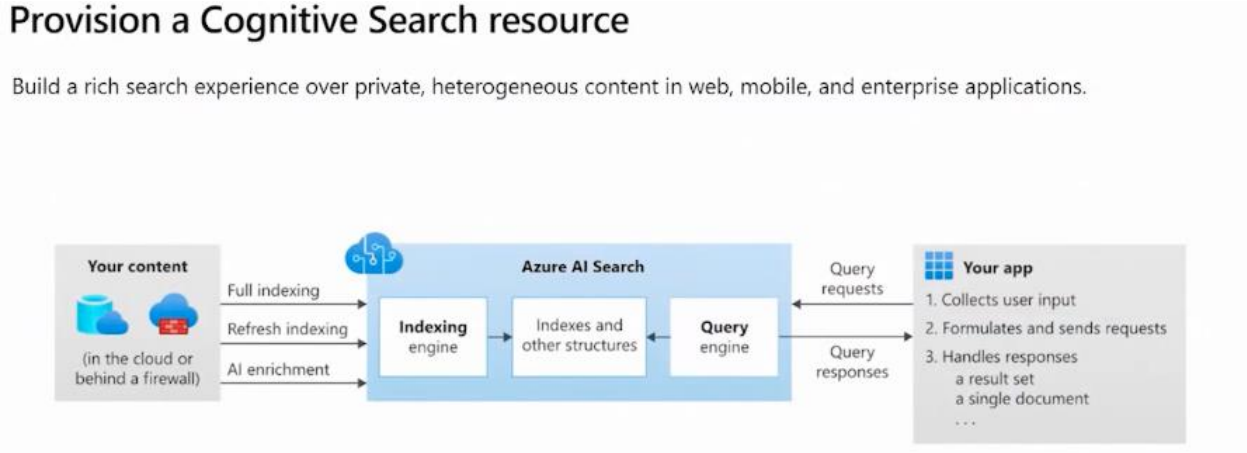
- Provision a Document Intelligence resource
- Use prebuilt models to extract data from documents
- Implement a custom document intelligence model
- Train, test, and publish a custom document intelligence model
- Create a composed document intelligence model
- Implement a document intelligence model as a custom Azure AI Search skill

Implement an Azure AI Search solution

- Azure AI Search provides a cloud-based solution for indexing and querying a wide range of data sources, and creating comprehensive and high-scale search solutions.

- With Azure AI Search, you can:
  - Index documents and data from a range of sources.
  - Use cognitive skills to enrich index data.
  - Store extracted insights in a knowledge store for analysis and integration.

- You need to create an **Azure AI Search** resource in your Azure subscription.
- Depending on the specific solution you intend to build, you may also need Azure resources for data storage and other application services.



Replicas and partitions

- Depending on the pricing tier you select, you can optimize your solution for scalability and availability by creating *replicas* and *partitions*:

  - ***Replicas*** are instances of the search service - you can think of them as nodes in a cluster. Increasing the number of replicas can help ensure there is sufficient capacity to service multiple concurrent query requests while managing ongoing indexing operations.

- ***Partitions*** are used to divide an index into multiple storage locations, enabling you to split I/O operations such as querying or rebuilding an index.

- The combination of replicas and partitions you configure determines the *search units* used by your solution.
- Put simply, the number of **search units** is the number of replicas multiplied by the number of partitions (**R x P = SU**). For example, a resource with four replicas and three partitions is using 12 search units.

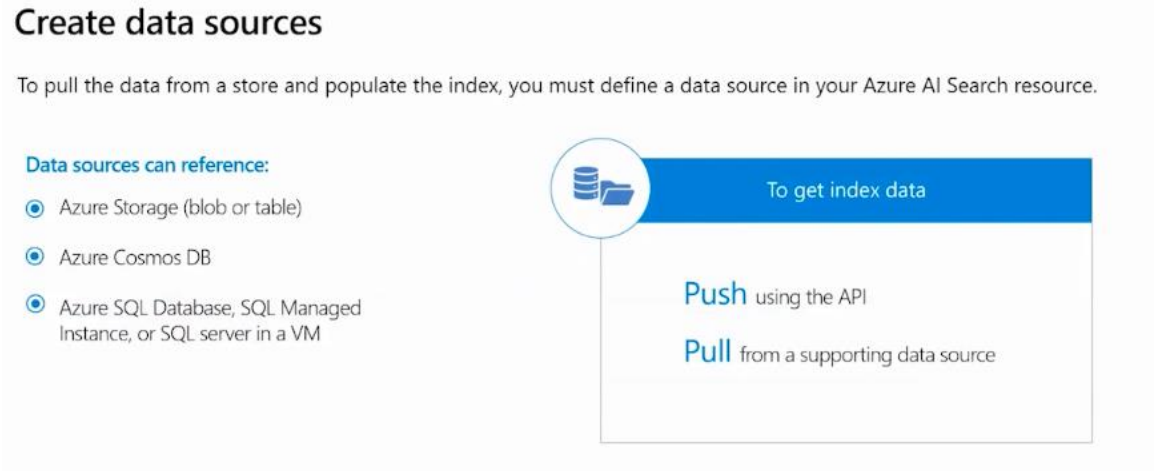
Search Components

An AI Search solution consists of multiple components, each playing an important part in the process of extracting, enriching, indexing, and searching data.

Data source

- Azure AI Search supports multiple types of data source, including:
    - Unstructured files in Azure blob storage containers.
    - Tables in Azure SQL Database.
    - Documents in Cosmos DB.

- Azure AI Search can pull data from these data sources for indexing.
- Alternatively, applications can push JSON data directly into an index, without pulling it from an existing data store.

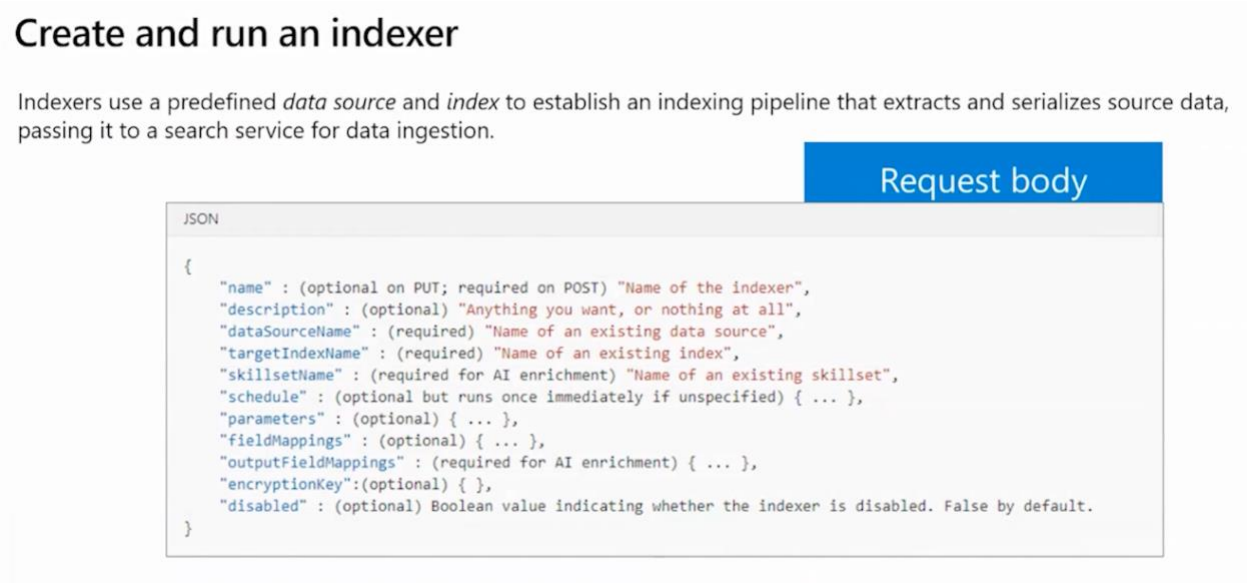


Skillset

- In Azure AI Search, you can apply artificial intelligence (AI) ***skills*** as part of the indexing process to **enrich** the source data with new information, which can be mapped to index fields.
- The skills used by an **indexer** are encapsulated in a ***skillset*** that defines an enrichment pipeline in which each step enhances the source data with insights obtained by a specific AI skill.
- Examples of the kind of information that can be extracted by an **AI skill include**: the document language, AI-generated descriptions of images.

Indexer

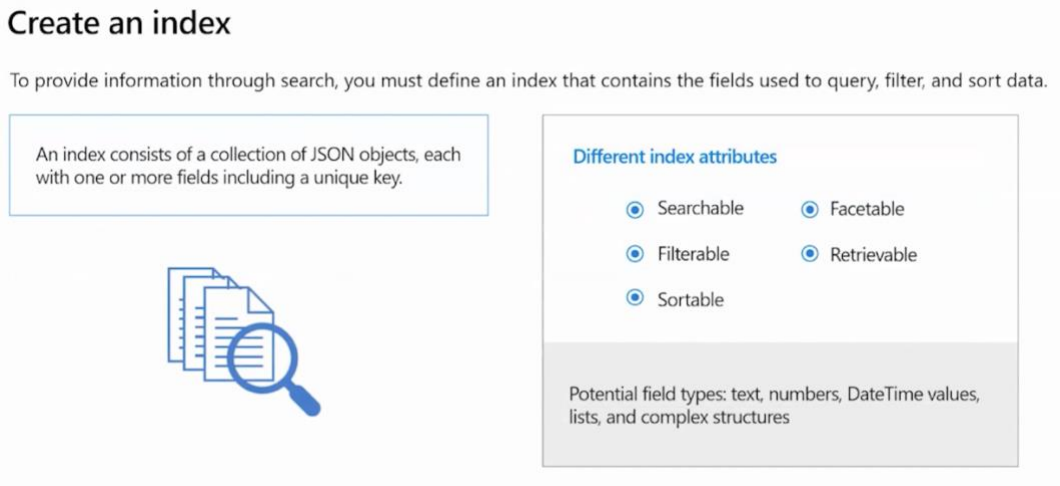
- The ***indexer*** is the **engine that drives the overall indexing** process.
- It takes the **outputs extracted** using the **skills** in the skillset**, along with the data and metadata** values extracted from the original data source, and **maps** them to fields in the **index**.
- An indexer is **automatically run** when it is created, and **can be scheduled** to run at regular intervals or run on demand to add more documents to the index.
- In some cases, such as when you add new fields to an index or new skills to a skillset, you **may need to reset the index** before re-running the indexer.



Index

- The index is the searchable result of the indexing process.
- It consists of a collection of JSON documents, with fields that contain the values extracted during indexing.
- Client applications can query the index to retrieve, filter, and sort information.

- Each index field can be configured with the following attributes: **key, searchable, filterable, sortable, facetable.**



Understand the indexing process

- The indexing process works by creating a **document** for each indexed entity.
- During indexing, an ***enrichment pipeline*** iteratively builds the documents that **combine metadata** from the data source **with enriched fields** extracted by cognitive **skills**.
- You can think of each indexed document as a JSON structure, which initially consists of a **document** with the index fields you have mapped to fields extracted directly from the source data,

Search an Index

- After you have created and populated an index, you can query it to search for information in the indexed document content.
- While you could retrieve index entries based on simple field value matching, most search solutions use ***full text search*** semantics to query an index.
- **Full text search** describes search solutions that parse text-based document contents to find query terms.
- Full text search queries in Azure AI Search are **based on the *Lucene* query syntax**, which provides a rich set of query operations for searching, filtering, and sorting data in indexes.
- Azure AI Search supports **two variants of the Lucene syntax**:
    - **Simple** - An intuitive syntax that makes it easy to perform basic searches that match literal query terms submitted by a user.
    - **Full** - An extended syntax that supports complex filtering, regular expressions, and other more sophisticated queries.

Filtering and sorting

- Users can refine query results by **filtering** and **sorting** based on field values.
- Azure AI Search supports both of these capabilities through the **search query API**.

Filtering results

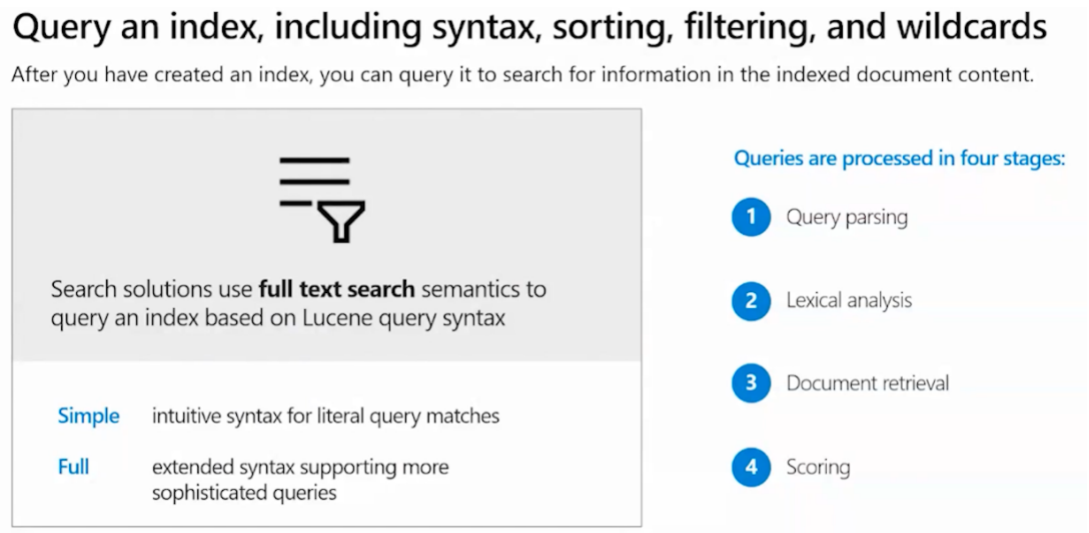
- You can apply filters to queries in two ways:
    - By including filter criteria in a *simple* **search** expression.
    - By providing an **OData** filter expression as a **$filter** parameter with a *full* syntax search expression.
- You can also filter with **facets**. They work best when a field has a small number of discrete values that can be displayed as links or options in the user interface.

Sorting

- By **default**, results are sorted based on the **relevancy score** assigned by the query process, with the **highest** scoring matches listed first.
- You can override this sort order by including an **OData orderby** parameter that specifies one or more ***sortable*** fields and a **sort order** (*asc* or *desc*).

Enhancing an index

- Azure AI Search supports several ways to enhance an index to provide a better user experience:
    - Search-as-you-type by adding a ***suggester*** to an index, you can enable two forms of search-as-you-type experience to help users find relevant results more easily: ***Suggestions*** and **Autocomplete**.
    - You can customize the way the **relevance score** is calculated by defining a ***scoring profile*** that applies a weighting value to specific fields - essentially increasing the search score for documents when the search term is found in those fields.
    - Synonyms:



Manage Knowledge Store projections, including file, object, and table projections

- Azure AI Search supports these scenarios by enabling you to define a ***knowledge store*** in the **skillset** that encapsulates your enrichment pipeline.

- The **knowledge store** consists of ***projections*** of the **enriched data**, which can be JSON objects, tables, or image files.

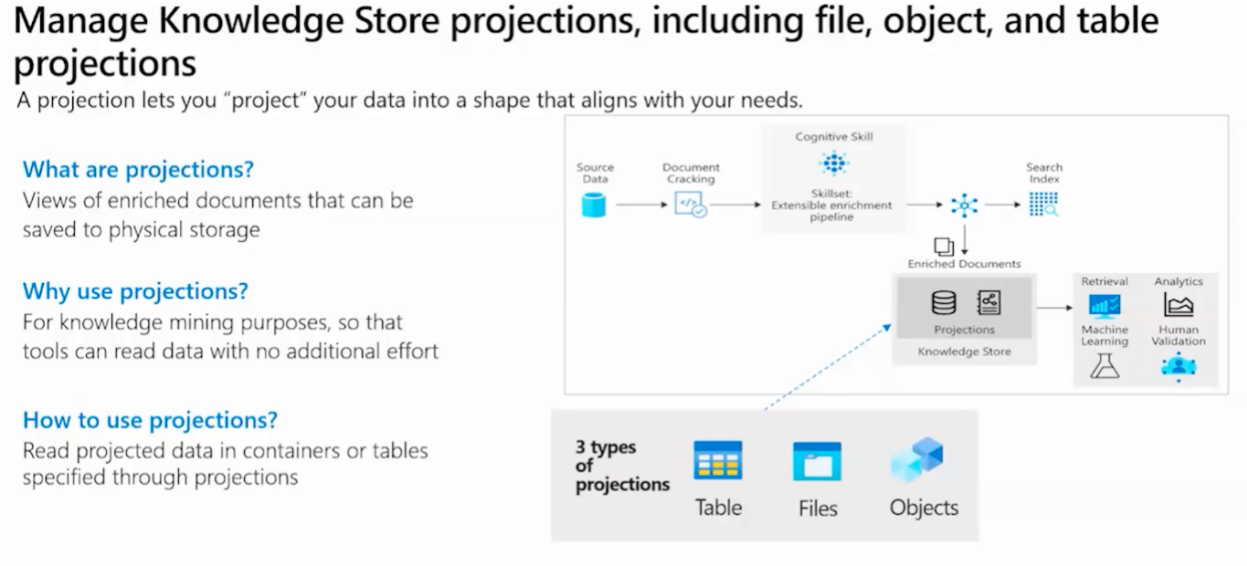
- When an **indexer runs** the pipeline to create or update an index, the **projections are generated** and persisted in the knowledge store.

- The **projections** of data to be stored in your knowledge store are **based on the document structures** generated by the enrichment pipeline in your indexing process.

- Each **skill** in your skillset iteratively **builds** a **JSON representation** of the enriched data for the documents being indexed, and you can persist some or all of the fields in the document as projections.

- To **simplify the mapping** of the field values to projections in a knowledge store, it's common to use the **Shaper** skill to create a new, field containing a simpler structure for the fields you want to map to projections.

- To define the knowledge store and the projections you want to create in it, you must create a **knowledgeStore** object in the skillset that specifies the Azure Storage connection string for the storage account where you want to create projections, and the definitions of the projections themselves.
- You can define **object projections, table projections, and file projections** depending on what you want to store;
-
- Must define a **separate *projection* for each type of projection**,
- Projection types are mutually exclusive in a projection definition, so only one of the projection type lists can be populated.



Implement an Azure AI Document Intelligence solution

- Azure AI Document Intelligence uses Azure AI Services to analyze the content of **scanned forms and convert them into data**.
- It can recognize text values in both common forms and forms that are unique to your business.
- In Azure AI Document Intelligence, three of the **prebuilt models** are for general document analysis:
    - Read
    - General document
    - Layout
- The other prebuilt models expect a common type of form or document:
    - Invoice
    - Receipt
    - W-2 US tax declaration
    - ID Document
    - Business card
    - Health insurance card

Azure AI Document Intelligence and Azure AI Vision

- If you want to **extract simple words and text** from a picture of a form or document, **without contextual** information, Azure AI **Vision OCR** is an appropriate service to consider
- Azure **AI Document Intelligence** includes **a more sophisticated analysis** of documents.
- For example, it can identify key/value pairs, tables, and context-specific fields.
- If you want to deploy a complete document analysis solution that enables users to both extract and understand text, consider Azure AI Document Intelligence.

Azure AI Document Intelligence tools

- If you want to try many features of Azure AI Document Intelligence without writing any code, you can use **Azure AI Document Intelligence Studio**
- To integrate Azure AI Document Intelligence into your own applications you'll need to **write code**.
- Azure AI Document Intelligence includes APIs using one of the languages. You can also use REST

Prebuilt models

- Microsoft has provided a set of prebuilt models with Azure AI Document Intelligence to handle the most common types of documents.
- You don't have to train these models and you can create solutions using them very quickly.
- Three of the prebuilt models:
    - **Read**: Use this model to **extract words and lines** from both printed and hand-written documents. It also **detects the language** used in the document.
    - **General document:** extends the functionality of the read model by adding the detection of **key-value pairs**, **entities**, **selection marks**, and **tables**. The model can extract these values from structured, semi-structured, and unstructured documents.
    - **Layout:** Use this model to **extract text, tables, and structure information from forms**. It can also recognize selection marks such as **check boxes and radio buttons**. Selection marks are extracted with their **bounding box**, a **confidence indicator**, and whether they're **selected or not**.

- The other prebuilt models are each designed to handle, and trained on, a specific and commonly used type of document. Some examples include:
    - **Invoice.** Use this model to extract key information from sales invoices in English and Spanish.
    - **Receipt.** Use this model to extract data from printed and handwritten receipts.
    - **W-2.** Use this model to extract data from United States government's W-2 tax declaration form.
    - **ID document.** Use this model to extract data from United States driver's licenses and international passports.
    - **Business card.** Use this model to extract names and contact details from business cards.

Custom models

- You can create a custom model and train it to analyze the specific type of document
- By using a custom model, trained on forms with similar structures and key-value pairs, you will obtain more predictable and standardized results from your unusual form types
- To **train** a custom model, you must supply **at least five examples of the completed form**
- There are two kinds of custom model:
    - **Custom template models.** A custom template model is most appropriate when the forms you want to analyze **have a consistent visual template**. Custom template models support **9 different languages for handwritten text** and a wide range of languages for printed text

- **Custom neural models.** A custom neural model can work **across the spectrum of structured to unstructured documents**. Documents like contracts with no defined structure or highly structured forms can be analyzed with a neural model. Neural models **work on English with the highest accuracy** and a marginal drop in accuracy for other languages like German, French, Italian, Spanish, and Dutch.

Composed models

- A composed model is one that consists of multiple custom models.
- Typical scenarios where composed models help are **when you don't know the submitted document type** and want to classify and then analyze it.
- They are also useful if you have **multiple variations of a form**, each with a trained individual model.
- The results from a composed model include the **docType** property, which indicates the custom model that was chosen to analyze each form.

Using the API

- To **extract** form data using a custom model, use the **analyze document** function of either a supported SDK, or the REST API, while supplying model ID (generated during model training).
- A successful JSON response contains **analyzeResult** that contains the content extracted and an **array of pages** containing information about the document content.

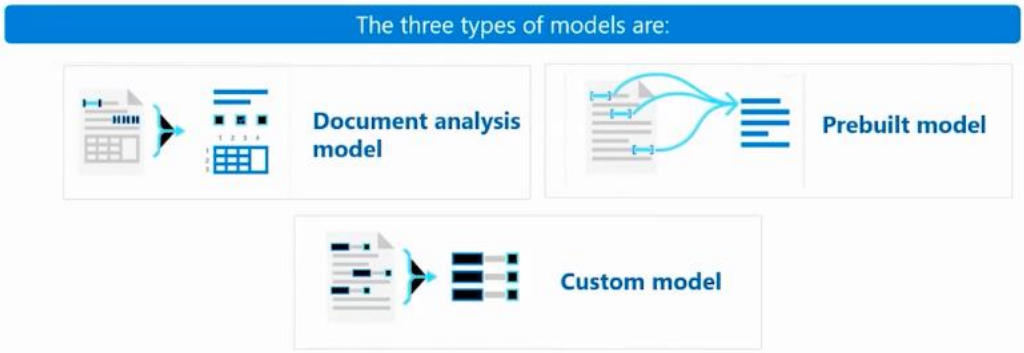
Use the Azure Document Intelligence Studio

- In addition to SDKs and the REST API, Azure Document Intelligence services can be accessed through a user interface called the Azure Document Intelligence Studio (preview).

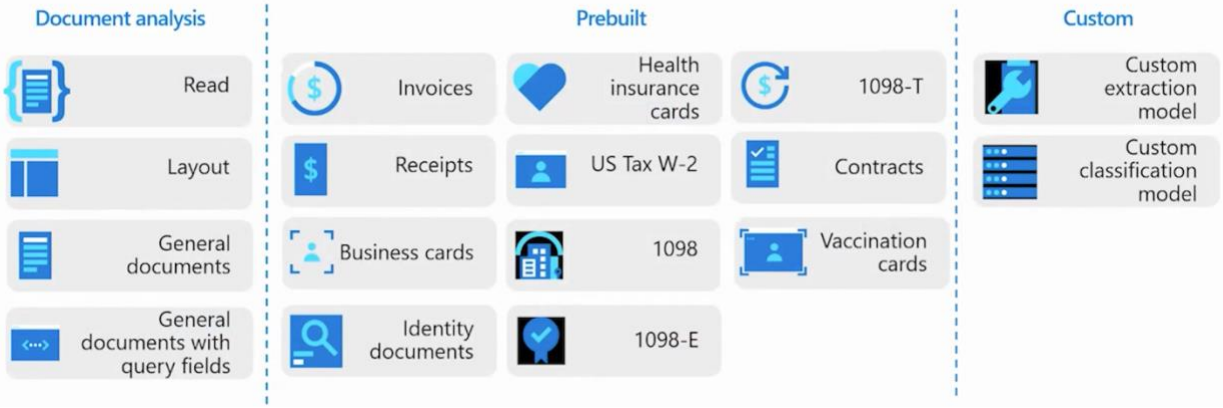
- The Studio can be used to analyze form layouts, extract data from prebuilt models, and train custom models.

Select the appropriate service for a document intelligence solution

The AI Document Intelligence Studio helps in extracting texts, key-value pairs, tables, and structures automatically from documents.

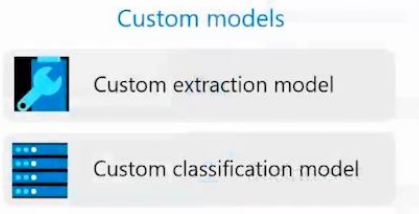


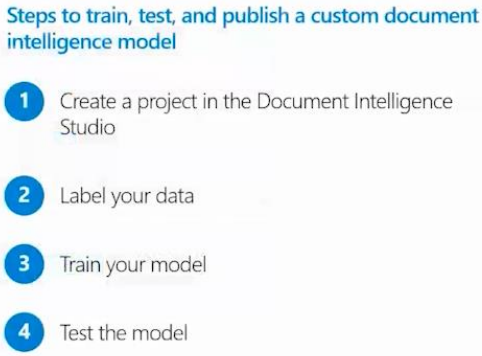
Provision a Document Intelligence resource



Implement a custom document intelligence model and train, test and publish a custom document intelligence model

Train custom models to classify documents and extract text, structure, and fields from your forms or documents.





Implement generative AI solutions (10–15%)

Use Azure OpenAI Service to generate content

- Provision an Azure OpenAI Service resource
- Select and deploy an Azure OpenAI model
- Submit prompts to generate natural language
- Submit prompts to generate code
- Use the DALL-E model to generate images
- Use Azure OpenAI APIs to submit prompts and receive responses
- Use large multimodal models in Azure OpenAI

Optimize generative AI

- Configure parameters to control generative behavior
- Apply prompt engineering techniques to improve responses
- Use your own data with an Azure OpenAI model
- Fine-tune an Azure OpenAI model

Use Azure OpenAI Service to generate content

- Generative AI models power ChatGPT's ability to produce new content, such as text, code, and images, based on a natural language prompts.
- Many generative AI models are a subset of deep learning algorithms.
- These algorithms support various workloads across vision, speech, language, decision, search, and more.

Access Azure OpenAI Service

- Provision an Azure OpenAI resource in your Azure subscription.
- Create an Azure OpenAI Service resource in Azure CLI

Use Azure AI Studio

- Azure AI Studio provides access to model management, deployment, experimentation, customization, and learning resources.

Explore types of generative AI models

Azure OpenAI includes several types of model:

- **GPT-4 models** are the latest generation of *generative pretrained* (GPT) models that can generate natural language and code completions based on natural language prompts.
- **GPT 3.5 models** can generate natural language and code completions based on natural language prompts. In particular, **GPT-35-turbo** models are optimized for chat-based interactions and work well in most generative AI scenarios.
- **Embeddings models** convert text into numeric vectors and are useful in language analytics scenarios such as comparing text sources for similarities.

- **DALL-E models** are used to generate images based on natural language prompts. Currently, DALL-E models are in preview. DALL-E models aren't listed in the Azure AI Studio interface and don't need to be explicitly deployed.

Deploy generative AI models

- You first need to deploy a model to chat with or make API calls to receive responses to prompts.
- When you create a new deployment, you need to indicate which base model to deploy.
- You can deploy any number of deployments in one or multiple Azure OpenAI resources as long as their Tokens Per Minute (TPM) stays within the deployment quota.
- You can deploy using one of the following approaches:
    - Deploy using Azure AI Studio
    - Deploy using Azure CLI
    - Deploy using the REST API

Use prompts to get completions (responses) from models

- Once the model is deployed, you can test how it completes prompts.
- A prompt is the text portion of a request that is sent to the deployed model's completions endpoint.
- **Responses** are referred to as ***completions***, which can come in form of text, code, or other formats.

Test models in Azure AI Studio's playground

- **Playgrounds** are useful interfaces in **Azure AI Studio** that you can use to **experiment** with your deployed models **without** needing to develop your own **client** application.
- Azure AI Studio offers multiple playgrounds with different parameter tuning options.
    - **Completions** playground:
        - Allows to make calls to the deployed models through a text-in, text-out interface and to adjust parameters.
        - Need to select the deployment name of your model under Deployments.
    - **Chat** playground:
        - The Chat playground is based on a conversation-in, message-out interface. You can initialize the session with a system message to set up the chat context.

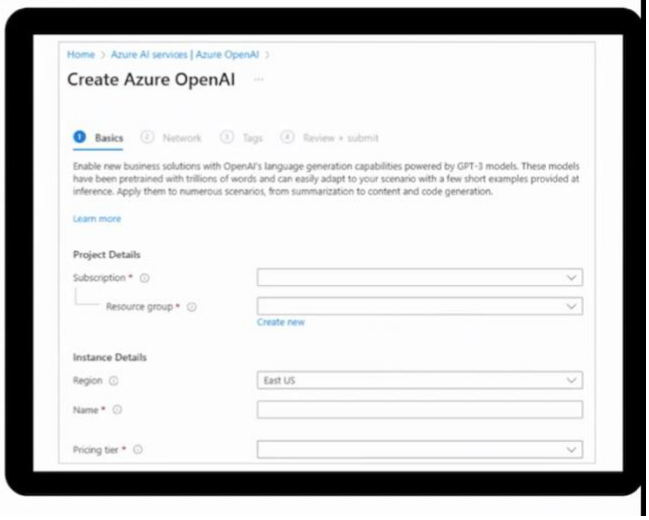
Provision an Azure OpenAI Service resource

Deploy a model in Azure OpenAI Studio

1. Apply for Azure OpenAI service access: https://aka.ms/oaiapply
2. Create an **Azure OpenAI** resource in the Azure portal.
   i. Identify resource
   ii. Configure network security
   iii. Confirm configuration and create resource

You can also use Azure CLI:

```
az cognitiveservices account create \
-n MyOpenAIResource \
-g MyResourceGroup \
-l eastus \
--kind OpenAI \
--sku s0 \
--subscription subscriptionID
```



Select and deploy an Azure OpenAI model

Model Family	Description
GPT-4	Newest, most capable chat-based models for language and code generation (*restricted*)
GPT-3.5	Natural language and code-generation models
Embeddings	Models that use embeddings for specific tasks (similarity, text search, and code search)
DALL-E	Image-generation model (*restricted preview*)

Deploy a model in Azure OpenAI Studio

1. Select the subscription and Open AI resource to use.
2. Create a new deployment:
   i. Select a model
   ii. Add a deployment name
   iii. Setting advanced features such as content filters and/or tokens per min rate limit

You can also use Azure CLI:

```
az cognitiveservices account deployment create \
   -g myResourceGroupName \
   -n MyOpenAIResource \
   --deployment-name my-gpt-model \
   --model-name gpt-35-turbo \
   --model-version "0301"  \
   --model-format OpenAI \
   --scale-settings-scale-type "Standard"
```

Submit prompts to generate natural language

Task	Prompt	Completion
Classifying content	Tweet: I enjoyed the training course. Sentiment:	Positive
Generating new content	Write a poem about databases	Databases, oh databases, You keep our information safe, From the small to the large, You store our data in a place.
Transformation/Translation	English: Hello French:	Bonjour
Summarization	Scotland is [*long description of Scotland...*] Summarize the previous text	Scotland is [*summarized description...*]
Continuation	One way to grow tomatoes is to	start with seeds...
Question answering	How many moons does Earth have?	Earth has one moon.
Chat	*Setup, followed by messages...*	*A sequence of relevant responses*

Submit prompts to generate code

Use natural language prompts to generate code

Use prompt engineering to describe what you want clearly

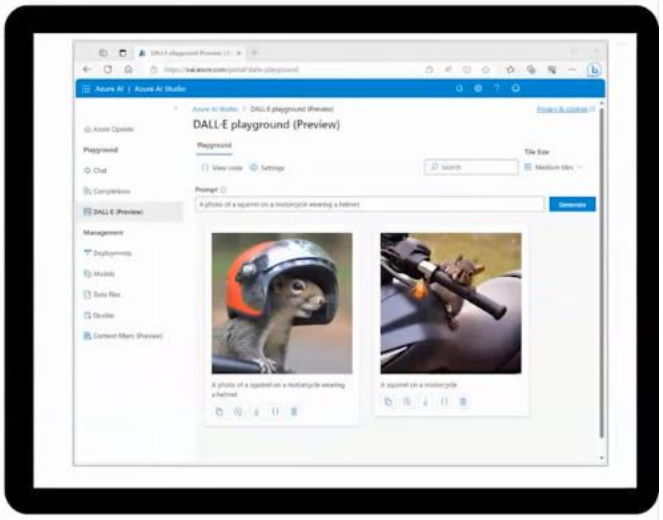
Break down complex tasks into smaller pieces

```
def binary_search(array, target):
    low = 0
    high = len(array) - 1

    while low <= high:
        mid = (low + high) // 2
        if array[mid] == target:
            return mid
        elif array[mid] < target:
            low = mid + 1
        else:
            high = mid - 1

    return -1
```

Use the DALL-E model to generate images

Generate images with a description

- Uses Neural network-based model for generating images
- Uses natural language to describe what the image should be
- Specifies content and style for better accuracy



Optimize generative AI

Apply prompt engineering techniques to improve responses

- Providing clear instructions
- Using a system message
- Primary, supporting, and grounding content
- Conversation history and few-shot learning
- Providing cues
- Chain of thought
- Requesting output composition

Use prompt engineering to improve prompts and customize responses

Prompt

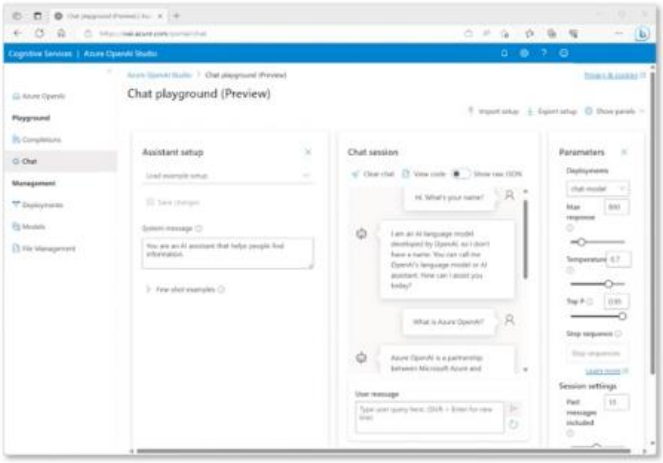
Write a table in markdown with 6 animals in it, with their genus and species

Response

Animal	Genus	Species
Red Panda	Ailurus	fulgens
African Elephant	Loxodonta	africana
Snow Leopard	Panthera	uncia
Arctic Fox	Vulpes	lagopus
Chimpanzee	Pan	troglodytes
Siberian Tiger	Panthera	tigris altaica

Configure parameters to control generative behavior

Chat playground



Use your own data with an Azure OpenAI model

Set up your data source

- Use an existing data source
- Leverage the data already in your account (such as blob storage)

Configure the studio or your app to connect to that data source

- Point the studio to the data source and set up the connection
- Specify the data source in the prompt parameters

Use the Azure OpenAI model (with your data for grounding)

- Chat with the AI model per usual (the model will use your data source if it finds relevant information)
- Limit the AI model to use only your data source.

Fine-tune an Azure OpenAI model

- Fine-tuning is a method to customize a model, like "gpt-3.5-turbo," by training it with extra data.
- Fine-tuning can improve request quality beyond just using prompts, adapt the model with larger example semantic model, and reduce the need for many examples to achieve high-quality responses.
- Fine-tuning is expensive and time-consuming, so it should be reserved for cases where it's essential.