 **General Definition of AI**: AI is generally defined as software exhibiting human-like capabilities.

 **Human-like Capabilities of AI**:

* **Visual Perception**: Uses computer vision to process images, videos, and live camera input.
* **Text Analysis and Conversation**: Uses natural language processing (NLP) to read, generate responses, and extract meaning from text.
* **Speech**: Recognizes and synthesizes speech for conversational AI, allowing interaction with AI agents similarly to human interaction.
* **Decision Making**: Uses past experience and learned correlations to assess situations and take actions, such as anomaly detection and automated responses.
* **Related Terms in AI**:
  + **Data Science**:
    - Focuses on processing and analyzing data.
    - Applies statistical techniques to uncover and visualize relationships and patterns.
    - Defines experimental models to explore these patterns.
    - Example: Analyzing data on endangered species and human activity to determine optimal conservation policies.
  + **Machine Learning**:
    - Subset of data science.
    - Involves training and validating predictive models.
    - Uses algorithms to predict values based on data features.
    - Example: Training a model to predict wildlife population changes based on environmental factors.
  + **Artificial Intelligence**:
    - Often builds on machine learning.
    - Creates software that emulates human intelligence characteristics.
    - Example: Using predictive models to analyze images from motion-activated cameras to monitor and track endangered species sightings for conservation efforts.

 **Integration of AI in Software Solutions**:

* Software engineers need to integrate AI capabilities into applications and services.
* Advances in machine learning, availability of large data volumes, and powerful compute have led to prepackaged AI services.
* Engineers can use these services as building blocks to create intelligent solutions without being experts in data science or machine learning.

 **Skills for Software Engineers**:

* Existing programming, testing, source control, and deployment skills are applicable.
* A conceptual understanding of AI and machine learning principles is necessary.

 **Model Training and Inferencing**:

* AI systems rely on predictive models trained with sample data.
* Training analyzes data to find relationships between features and labels.
* Inferencing uses the trained model to predict labels from new data.
* Development involves training a model from existing data before using it for inferencing.

 **Probability and Confidence Scores**:

* Predictions are based on probability, not absolute certainty.
* Confidence scores reflect the likelihood of predictions.
* Engineers should use confidence scores to evaluate predictions and set appropriate thresholds to ensure reliability.

 **Responsible AI and Ethics**:

* Consider the societal and ethical impact of AI applications.
* AI predictions are based on probabilistic models and training data.
* Users might trust AI too much, leading to potential harm from incorrect predictions.
* Ensure fairness, reliability, and protection from harm or discrimination in AI-enabled solutions.

 **Fairness**:

* AI systems should treat all people fairly.
* Avoid biases based on gender, ethnicity, or other factors.
* Fairness requires careful review of training data and evaluation of predictive performance for diverse user groups.
* Tools exist for evaluating and mitigating unfairness, but fairness must be considered from the start of development.

 **Reliability and Safety**:

* AI systems should perform reliably and safely.
* Examples include autonomous vehicles and medical diagnosis models.
* Rigorous testing and deployment management are essential.
* Consider the probabilistic nature of machine learning models and use appropriate confidence score thresholds.

 **Privacy and Security**:

* AI systems should be secure and respect privacy.
* Machine learning models use large volumes of data, often containing personal details.
* Implement safeguards to protect data and customer content, both during training and in production.

 **Inclusiveness**:

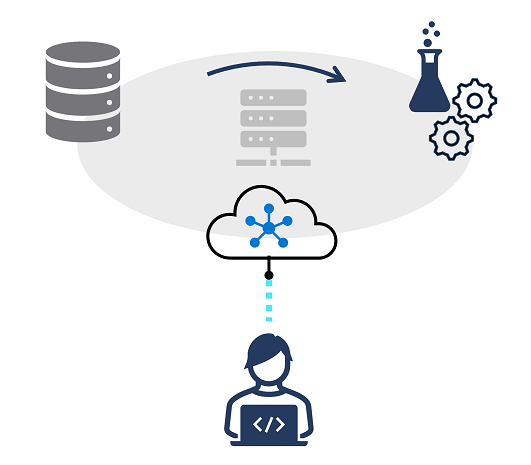
* AI systems should empower and engage everyone.
* AI should benefit all parts of society, regardless of physical ability, gender, sexual orientation, ethnicity, or other factors.
* Ensure diverse input in design, development, and testing processes.

 **Transparency**:

* AI systems should be understandable.
* Users should be aware of the system's purpose, operation, and limitations.
* Inform users about factors affecting prediction accuracy and confidence scores.
* Clearly explain the use, retention, and access of personal data in AI applications.

 **Accountability**:

* People should be accountable for AI systems.
* Developers and designers are responsible for ensuring the system meets ethical and legal standards.
* Work within a governance framework to ensure responsibility requirements are met.

**Azure Machine Learning Service Overview**

**Features and Capabilities**

* **Automated Machine Learning**:
  + Enables non-experts to quickly create effective machine learning models from data.
* **Azure Machine Learning Designer**:
  + Provides a graphical, no-code interface for developing machine learning solutions.
* **Data and Compute Management**:
  + Offers cloud-based data storage and compute resources for running data experiment code at scale.
* **Pipelines**:
  + Allows the definition of pipelines to orchestrate model training, deployment, and management tasks.

**Use Cases for Data Scientists**

* **Data Ingestion and Preparation**:
  + Collect and prepare data for analysis and model training.
* **Experimentation**:
  + Run experiments to explore data and train predictive models.
* **Model Deployment and Management**:
  + Deploy trained models as web services and manage them in production.

**Use Cases for Software Engineers**

* **Model Training and Deployment**:
  + Use Automated Machine Learning or the Azure Machine Learning designer to train and deploy models as services for integration into AI-enabled applications.
* **Collaboration with Data Scientists**:
  + Work with data scientists to deploy models using frameworks like Scikit-Learn, PyTorch, and TensorFlow as web services and integrate them into applications.
* **DevOps Orchestration**:
  + Utilize Azure Machine Learning SDKs or CLI scripts to manage versioning, deployment, and testing of machine learning models as part of the application delivery process.

Azure AI Services are cloud-based services that encapsulate AI capabilities.AI Services as a set of individual services that you can use as building blocks to compose sophisticated, intelligent applications.

Azure AI services offer a wide range of prebuilt AI capabilities across multiple categories, with examples shown in the following table.

A screenshot of a black screen

Description automatically generated

**Azure OpenAI Service**

* **Generative AI**:
  + Focuses on AI models that generate content (text, images, code, etc.).
  + Interacts in a way that feels like a real conversation with a person.
  + Based on large language models (LLMs) using transformer architecture.
* **Azure OpenAI Service**:
  + Deploy, utilize, and fine-tune models developed by OpenAI.
  + Provides access to models behind applications like ChatGPT.
  + Enables development of applications using powerful generative AI models.
* **Development Tools**:
  + REST APIs and language-specific SDKs available for application development.

**Azure Cognitive Search**

* **General Features**:
  + Allows ingestion and indexing of data from various sources.
  + Enables searching the index to find, filter, and sort information extracted from source data.
* **AI Enrichment Pipeline**:
  + Enhances the index with insights using AI skills.
  + Examples include generating descriptions of images, extracting text from scanned documents, and identifying key phrases in large documents.
* **Advanced Search Experience**:
  + AI enrichment produces a more useful and relevant search experience.
* **Knowledge Store**:
  + Persist extracted insights for further analysis.
  + Integrate insights into data pipelines for business intelligence solutions.

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 includes a wide range of individual services across multiple categories, as shown in the following table..

A screenshot of a black screen

Description automatically generated

You can use AI services to build your own AI solutions to provide out-of-the-box solutions for common AI scenarios.

Azure AI services include:

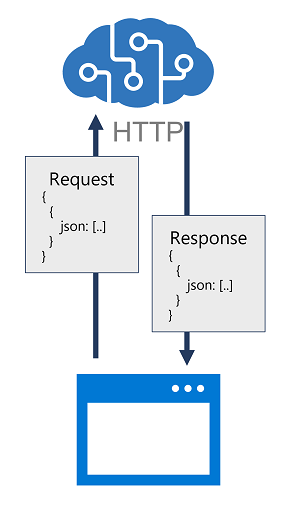
* Azure AI Document Intelligence - An optical character recognition (OCR) solution that can extract semantic meaning from forms, such as invoices, receipts, and others.
* Azure AI Immersive Reader - A reading solution that supports people of all ages and abilities.
* Azure Cognitive Search - A cloud-scale search solution that uses AI services to extract insights from data and documents.
* Azure OpenAI - An Azure Cognitive Service that provides access to the capabilities of OpenAI GPT-4.

**Provisioning Azure AI Services Resources**

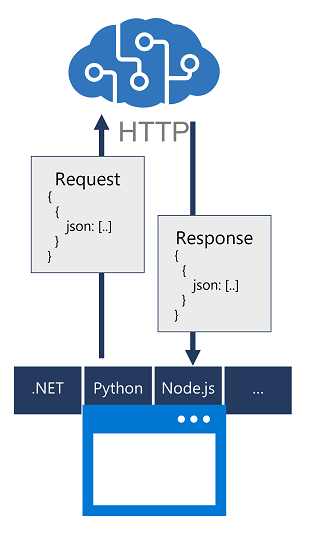
* **Overview**:
  + To use Azure AI services, create resources in an Azure subscription.
  + Resources define endpoints for service consumption, provide access keys for authentication, and manage billing.
* **Provisioning Options**:
  + **Multi-service Resource**:
    - Supports multiple AI services (e.g., Azure AI Language, Vision, Speech).
    - Benefits:
      * Single set of access credentials.
      * Single endpoint for multiple services.
      * Unified billing for all services.
  + **Single-service Resource**:
    - Individual provisioning for each AI service (e.g., separate AI Language and AI Vision resources).
    - Benefits:
      * Separate endpoints for each service, allowing geographical distribution.
      * Independent management of access credentials.
      * Separate billing for each service.
    - Often includes a free tier with usage restrictions for trial purposes.
  + **Training and Prediction Resources**:
    - Some AI services require separate resources for model training and prediction.
    - Benefits:
      * Separate billing for model training and consumption.
      * Use a dedicated resource for training and a generic resource for inferencing.

**Endpoint and Keys for Azure AI Services**

When provisioning an Azure AI services resource, you define an endpoint and access keys necessary for application consumption:

* **Endpoint URI**:
  + HTTP address for accessing the REST interface of the service.
  + Used by most SDKs to establish a connection.
* **Subscription Key**:
  + Access to the endpoint is restricted by a subscription key.
  + Client applications must provide a valid key to consume the service.
  + Two keys are generated upon provisioning, allowing flexibility.
  + Keys can be regenerated as needed to control access.
* **Resource Location**:
  + Azure resources are assigned to a location, determining the data center where the resource resides.
  + While most SDKs use the endpoint URI, some may require the location information for connection.

**Consuming Azure AI Services via REST APIs**

* **REST APIs**:
  + Provide interfaces for client applications to consume Azure AI services.
  + Functions can typically be called by submitting data in JSON format over an HTTP request.
  + Request methods may include POST, PUT, or GET depending on the specific function being called.
  + Results are returned to the client as an HTTP response, often in JSON format encapsulating the output data.
* **Language and Tool Compatibility**:
  + Any programming language or tool capable of handling JSON over HTTP can consume AI services.
  + Common programming languages such as Microsoft C#, Python, and JavaScript are supported.
  + Utilities like Postman and cURL are useful for testing API requests and responses.
* **Flexibility**:
  + REST interfaces allow for integration with various programming languages and tools, making AI services accessible across different development environments.
  + Developers can choose the language or tool that best suits their application requirements and preferences.

**Using Language-Specific SDKs for Azure AI Services**

* **Simplified Development**:
  + Utilize native libraries specific to your programming language for easier development.
  + Abstracts away the complexities of REST interfaces, simplifying the integration process.
* **SDK Availability**:
  + SDKs are available for common programming languages such as:
    - Microsoft C# (.NET Core)
    - Python
    - JavaScript (Node.js)
    - Go
    - Java
* **Package Installation**:
  + SDKs include packages that can be installed to incorporate service-specific libraries into your code.
* **Online Documentation**:
  + Access comprehensive online documentation provided with each SDK.
  + Learn about available classes, methods, and parameters necessary for working with the service effectively.

**Authentication for Azure AI Services**

**Regenerate Keys:**

* **Regenerate Regularly**:
  + Protect against unauthorized access by regularly regenerating subscription keys.
  + Keys can be regenerated via Azure portal or Azure CLI using the az cognitiveservices account keys regenerate command.
* **Two Keys**:
  + Each AI service resource comes with two keys, allowing seamless key rotation without service interruption.
  + Transition applications from using one key to the other before regeneration to ensure continuous access.

**Azure Key Vault Integration:**

* **Secure Storage**:
  + Store subscription keys in Azure Key Vault for enhanced security.
  + Access keys securely without exposing them in application code or configuration files.

**Token-based Authentication:**

* **Token Usage**:
  + Some AI services support or require token-based authentication.
  + Initial request with subscription key obtains an authentication token valid for 10 minutes.
  + Subsequent requests must present the token for validation.

**Microsoft Identity Platform Integration (ENTRA):**

* **Service Principals**:
  + Authenticate against Azure AI services using service principals.
  + Steps include creating a custom subdomain, registering an application, and assigning roles to service principals.

**Managed Identities:**

* **System-assigned and User-assigned**:
  + Assign managed identities to resources for authentication.
  + System-assigned: Linked to a specific resource, deleted when the resource is deleted.
  + User-assigned: Usable by multiple resources, independent of any single resource.

**Implement network security**

Network security is an important measure to ensure unauthorized users can't reach the services that you are protecting. Limiting what users can see is always a great idea, since they can’t compromise what they can’t see.

**Apply network access restrictions**

By default, Azure AI services are accessible from all networks. Some individual AI services resources (such as Azure AI Face service, Azure AI Vision, and others) can be configured to restrict access to specific network addresses - either public Internet addresses or addresses on virtual networks.

With network restrictions enabled, a client trying to connect from an IP address that isn't allowed will receive an Access Denied error.

**Monitoring Cost for Azure AI Services**

**Cost Efficiency in Cloud Services:**

* **Pay-as-You-Go Model**:
  + Cloud services allow cost efficiencies by paying only for services used.
* **Free Tier and Billed Tiers**:
  + Some Azure AI services offer a free tier for development and testing purposes.
  + Billed tiers incur charges based on transactions, with specific rates depending on the resource type.

**Planning Costs:**

* **Estimation with Pricing Calculator**:
  + Utilize the Azure Pricing Calculator to estimate costs for AI services.
  + Select the desired AI service API, region, pricing tier, and expected usage metrics.
  + Add multiple AI service APIs to the estimate if needed.

**Viewing Costs:**

* **Azure Portal**:
  + Sign in to the Azure portal and select your subscription.
  + View accumulated costs for AI services resources under the Cost analysis tab.
  + Apply filters to view costs specifically for AI services by restricting data to resources with a service name of Cognitive Services.

**Create 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. These alerts will ensure that the correct team knows when a problem arises.

**Alert rules**

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. To define the alert rule, you must specify:

* 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.

**Viewing Metrics for Azure AI Services**

**Azure Monitor**

* **Purpose**:
  + Tracks indicators of resource utilization, health, and performance.
  + Collects metrics for Azure AI services, including endpoint requests, data submitted and returned, errors, etc.

**Viewing Metrics in the Azure Portal**

* **Access Metrics Page**:
  + Select the specific Azure AI services resource.
  + Navigate to the Metrics page to view collected metrics.
* **Creating and Customizing Charts**:
  + By default, an empty chart is created.
  + Add resource-specific metrics to the chart.
  + Create additional charts as required.
  + Add multiple metrics to a single chart.
  + Choose appropriate aggregations and chart types.
* **Sharing and Cloning Charts**:
  + Export charts to Excel.
  + Copy a link to the chart for sharing.
  + Clone charts to create duplicates for different views.

**Adding Metrics to a Dashboard**

* **Creating Dashboards**:
  + In the Azure portal, select Dashboard from the menu.
  + Create up to 100 named dashboards to track specific aspects of Azure services.
* **Adding Visualizations**:
  + Add various tiles and visualizations to the dashboard.
  + Add charts to a new or existing dashboard when viewing metrics for a resource.

**Managing Diagnostic Logging for Azure AI Services**

**Diagnostic Logging Overview**

* **Purpose**: Capture operational data for analyzing service usage and troubleshooting problems.

**Creating Resources for Diagnostic Log Storage**

* **Azure Event Hubs**: For forwarding data to custom telemetry solutions.
* **Azure Log Analytics**:
  + Service to query and visualize log data within the Azure portal.
* **Azure Storage**:
  + Cloud-based data store for log archives.
  + Store log data in the same region as the AI services resource for efficiency.

**Configuring Diagnostic Settings**

1. **Navigate to Diagnostic Settings**:
   * Go to the Diagnostic settings page of the blade for your AI services resource in the Azure portal.
2. **Add Diagnostic Settings**:
   * **Name**: Specify a name for your diagnostic settings.
   * **Log Event Categories**: Select the categories of log event data to capture.
   * **Destinations**: Define where to store the log data (Azure Log Analytics, Azure Storage, or both).

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.

Now 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.

**Containers for Azure AI Services**

**Definition and Benefits**

* **Container Basics**:
  + **Encapsulation**: Contains an application or service along with its runtime components.
  + **Abstraction**: Hides underlying OS and hardware specifics.
* **Advantages**:
  + **Portability**: Containers can run across different hosts regardless of the underlying OS or hardware.
  + **Isolation**: Multiple containers can run on a single host, each with its own runtime configuration.

**Components**

* **Container Image**:
  + Encapsulates the software and configuration needed to run the container.
  + Stored in registries like Docker Hub or private registries.

**Deployment**

* **Container Hosts**:
  + **Docker Server**: An open-source solution for managing containers, available for Windows and Linux.
  + **Azure Container Instance (ACI)**: A cloud-based container hosting solution.
  + **Azure Kubernetes Service (AKS)**: A managed Kubernetes container orchestration service in Azure.

**Use Cases**

* **Cloud Deployment**:
  + Use Azure services to host containers in the cloud.
* **On-Premises Deployment**:
  + Deploy containers on local infrastructure for data sensitivity and latency reduction.

**Summary**

Containers provide a flexible, portable, and isolated environment for running Azure AI services. They simplify the deployment process across various platforms and can enhance performance and security by keeping data on-premises.

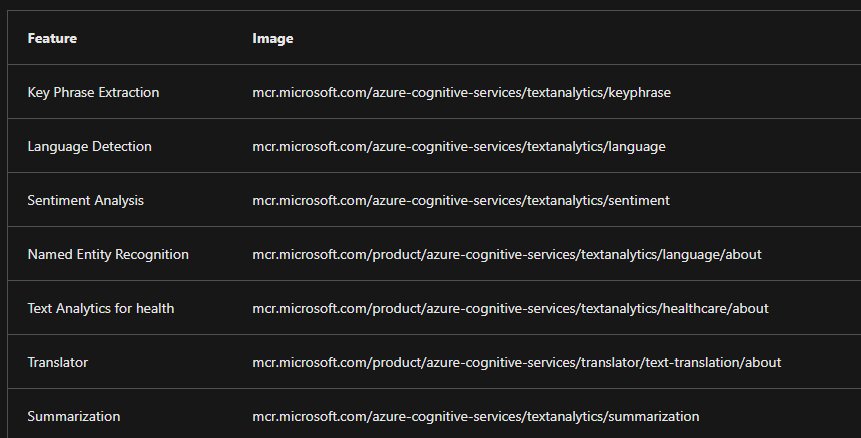
**Key Points about Using Azure AI Services Containers**

**Overview**

* **Container Images**:
  + Available in Microsoft Container Registry.
  + Encapsulate specific Azure AI services APIs.
* **Deployment Activities**:
  + **Download and Deploy**: Pull container image and deploy to a container host (Docker, ACI, AKS).
  + **Client Interaction**: Clients interact with the containerized service endpoint.
  + **Billing**: Container sends usage metrics to Azure for billing.
* **Container Hosts**:
  + Local Docker server
  + Azure Container Instance (ACI)
  + Azure Kubernetes Service (AKS)

**Azure AI Services Container Images**

Language containers

For the AI Language service, the core features map to separate images: 

Speech containersA screenshot of a computer

Description automatically generated

A screenshot of a black box

Description automatically generated