**Plan and prepare to develop AI solutions on Azure**

**Introduction**

The growth in the use of artificial intelligence (AI) in general, and *generative* AI in particular means that developers are increasingly required to create comprehensive AI solutions. These solutions need to combine machine learning models, AI services, prompt engineering solutions, and custom code.

Microsoft Azure provides multiple services that you can use to create AI solutions. However, before embarking on an AI application development project, it's useful to consider the available options for services, tools, and frameworks as well as some principles and practices that can help you succeed.

This module explores some of the key considerations for planning an AI development project, and introduces **Azure AI Foundry**; a comprehensive platform for AI development on Microsoft Azure.

**What is AI?**

The term "Artificial Intelligence" (AI) covers a wide range of software capabilities that enable applications to exhibit human-like behavior. AI has been around for many years, and its definition has varied as the technology and use cases associated with it have evolved. In today's technological landscape, AI solutions are built on machine learning *models* that encapsulate semantic relationships found in huge quantities of data; enabling applications to appear to interpret input in various formats, reason over the input data, and generate appropriate responses and predictions.

Common AI capabilities that developers can integrate into a software application include:

| **Capability** | **Description** |
| --- | --- |
| Diagram of speech bubbles. **Generative AI** | The ability to generate original responses to natural language *prompts*. For example, software for a real estate business might be used to automatically generate property descriptions and advertising copy for a property listing. |
| Diagram of a human head with a cog for a brain. **Agents** | Generative AI applications that can respond to user input or assess situations autonomously, and take appropriate actions. For example, an "executive assistant" agent could provide details about the location of a meeting on your calendar, or even attach a map or automate the booking of a taxi or rideshare service to help you get there. |
| Diagram of an eye being scanned. **Computer vision** | The ability to accept, interpret, and process visual input from images, videos, and live camera streams. For example, an automated checkout in a grocery store might use computer vision to identify which products a customer has in their shopping basket, eliminating the need to scan a barcode or manually enter the product and quantity. |
| Diagram of a speech bubble and a sound wave. **Speech** | The ability to recognize and synthesize speech. For example, a digital assistant might enable users to ask questions or provide audible instructions by speaking into a microphone, and generate spoken output to provide answers or confirmations. |
| Diagram of a text document. **Natural language processing** | The ability to process natural language in written or spoken form, analyze it, identify key points, and generate summaries or categorizations. For example, a marketing application might analyze social media messages that mention a particular company, translate them to a specific language, and categorize them as positive or negative based on sentiment analysis. |
| Diagram of a form containing information. **Information extraction** | The ability to use computer vision, speech, and natural language processing to extract key information from documents, forms, images, recordings, and other kinds of content. For example, an automated expense claims processing application might extract purchase dates, individual line item details, and total costs from a scanned receipt. |
| Diagram of a chart showing an upward trend. **Decision support** | The ability to use historic data and learned correlations to make predictions that support business decision making. For example, analyzing demographic and economic factors in a city to predict real estate market trends that inform property pricing decisions. |

Determining the specific AI capabilities you want to include in your application can help you identify the most appropriate AI services that you'll need to provision, configure, and use in your solution.

**Azure AI services**

Microsoft Azure provides a wide range of cloud services that you can use to develop, deploy, and manage an AI solution. The most obvious starting point for considering AI development on Azure is Azure AI services; a set of out-of-the-box prebuilt APIs and models that you can integrate into your applications. The following table lists some commonly used Azure AI services (for a full list of all available Azure AI services, see [Available Azure AI services](https://learn.microsoft.com/en-us/azure/ai-services/what-are-ai-services#available-azure-ai-services?azure-portal=true)).

| **Service** | **Description** |
| --- | --- |
| Azure OpenAi service icon. **Azure OpenAI** | The Azure OpenAI service provides access to OpenAI generative AI models including the GPT family of large and small language models and DALL-E image-generation models within a scalable and securable cloud service on Azure. |
| Azure AI Vision service icon. **Azure AI Vision** | The Azure AI Vision service provides a set of models and APIs that you can use to implement common computer vision functionality in an application. With the AI Vision service, you can detect common objects in images, generate captions, descriptions, and tags based on image contents, and read text in images. |
| Azure AI Speech service icon. **Azure AI Speech** | The Azure AI Speech service provides APIs that you can use to implement *text to speech* and *speech to text* transformation, as well as specialized speech-based capabilities like speaker recognition and translation. |
| Azure AI Language service icon. **Azure AI Language** | The Azure AI Language service provides models and APIs that you can use to analyze natural language text and perform tasks such as entity extraction, sentiment analysis, and summarization. The AI Language service also provides functionality to help you build conversational language models and question answering solutions. |
| Azure AI Content Safety service icon. **Azure AI Content Safety** | Azure AI Content Safety provides developers with access to advanced algorithms for processing images and text and flagging content that is potentially offensive, risky, or otherwise undesirable. |
| Azure AI Translator service icon. **Azure AI Translator** | The Azure AI Translator service uses state-of-the-art language models to translate text between a large number of languages. |
| Azure AI Face service icon. **Azure AI Face** | The Azure AI Face service is a specialist computer vision implementation that can detect, analyze, and recognize human faces. Because of the potential risks associated with personal identification and misuse of this capability, access to some features of the AI Face service are restricted to approved customers. |
| Azure AI Custom Vision service icon. **Azure AI Custom Vision** | The Azure AI Custom Vision service enables you to train and use custom computer vision models for image classification and object detection. |
| Azure AI Document Intelligence service icon. **Azure AI Document Intelligence** | With Azure AI Document Intelligence, you can use pre-built or custom models to extract fields from complex documents such as invoices, receipts, and forms. |
| Azure AI Content Understanding service icon. **Azure AI Content Understanding** | The Azure AI Content Understanding service provides multi-modal content analysis capabilities that enable you to build models to extract data from forms and documents, images, videos, and audio streams. |
| Azure AI Search service icon. **Azure AI Search** | The Azure AI Search service uses a pipeline of AI skills based on other Azure AI Services and custom code to extract information from content and create a searchable index. AI Search is commonly used to create vector indexes for data that can then be used to *ground* prompts submitted to generative AI language models, such as those provided in the Azure OpenAI service. |

**Considerations for Azure AI services resources**

To use Azure AI services, you create one or more Azure AI resources in an Azure subscription and implement code in client applications to consume them. In some cases, AI services include web-based visual interfaces that you can use to configure and test your resources - for example to train a custom image classification model using the **Custom Vision** service you can use the visual interface to upload training images, manage training jobs, and deploy the resulting model.

**Single service or multi-service resource?**

Most Azure AI services, such as **Azure AI Vision**, **Azure AI Language**, and so on, can be provisioned as standalone resources, enabling you to create only the Azure resources you specifically need. Additionally, standalone Azure AI services often include a free-tier SKU with limited functionality, enabling you to evaluate and develop with the service at no cost. Each standalone Azure AI resource provides an endpoint and authorization keys that you can use to access it securely from a client application.

Alternatively, you can provision a multi-service **Azure AI services** resource that encapsulates the following services in a single Azure resource:

* Azure OpenAI
* Azure AI Speech
* Azure AI Vision
* Azure AI Language
* Azure AI Content Safety
* Azure AI Translator
* Azure AI Document Intelligence
* Azure AI Content Understanding

Using a multi-service resource can make it easier to manage applications that use multiple AI capabilities.

**Regional availability**

Some services and models are available in only a subset of Azure regions. Consider service availability and any regional quota restrictions for your subscription when provisioning Azure AI services. Use the [product availability table](https://azure.microsoft.com/explore/global-infrastructure/products-by-region/table) to check regional availability of Azure services. Use the [model availability table](https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models#model-summary-table-and-region-availability?azure-portal=true) in the Azure OpenAI service documentation to determine regional availability for Azure OpenAI models.

**Cost**

Azure AI services are charged based on usage, with different pricing schemes available depending on the specific services being used. As you plan an AI solution on Azure, use the [Azure AI services pricing](https://azure.microsoft.com/pricing/details/cognitive-services) documentation to understand pricing for the AI services you intend to incorporate into your application. You can use the [Azure pricing calculator](https://azure.microsoft.com/pricing/calculator) to estimate the costs your expected usage will incur.

**Azure AI Foundry**

Azure AI Foundry is a platform for AI development on Microsoft Azure. While you *can* provision individual Azure AI services resources and build applications that consume them without it, the project organization, resource management, and AI development capabilities of Azure AI Foundry makes it the recommended way to build all but the most simple solutions.

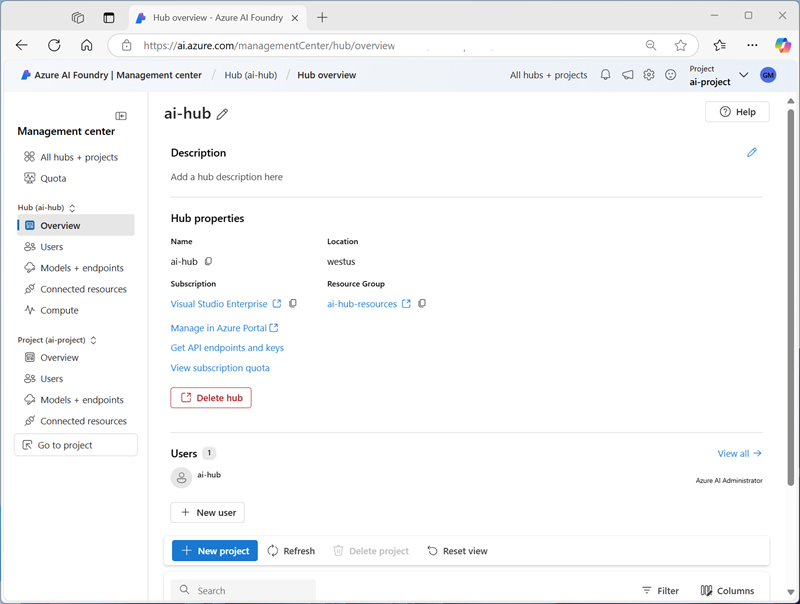
Azure AI Foundry provides the *Azure AI Foundry portal*, a web-based visual interface for working with AI projects. It also provides the *Azure AI Foundry SDK*, which you can use to build AI solutions programmatically.

**Hubs and projects**

In Azure AI Foundry, you manage the resources, assets, code, and other elements of the AI solution in hubs and projects. *Hubs* provide a top-level container for managing shared resources, data, connections and security configuration for AI application development. A hub can support multiple *projects*, in which developers collaborate on building a specific solution.

**Hubs**

A hub provides a centrally managed collection of shared resources and management configuration for AI solution development. You need at least one hub to use all of the solution development features and capabilities of AI Foundry.



In a hub, you can define shared resources to be used across multiple projects. When you create a hub using the Azure AI Foundry portal, an Azure **Azure AI Hub** resource is created in a resource group associated with the hub. Additionally, the following resources are created for the hub:

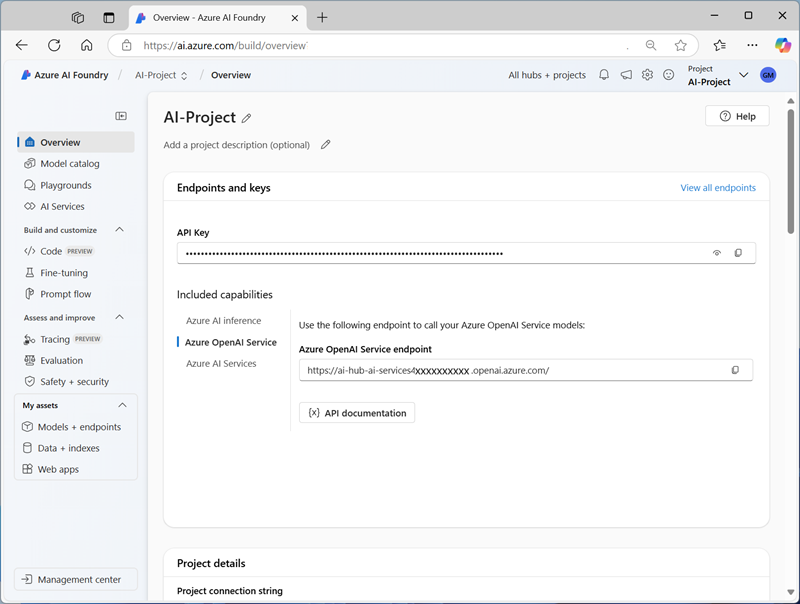
* A multi-service **Azure AI services** resource to provide access to Azure OpenAI and other Azure AI services.
* A **Key vault** in which sensitive data such as connections and credentials can be stored securely.
* A **Storage account** for data used in the hub and its projects.
* Optionally, an **Azure AI Search** resource that can be used to index data and support grounding for generative AI prompts.

You can create more resources as required (for example, an **Azure AI Face** resource) and add it to the hub (or an individual project) by defining a **connected resource**. As you create more items in your hub, such as compute instances or endpoints, more resources will be created for them in the Azure resource group.

Access to the resources in a hub is governed by creating *users* and assigning them to *roles*. An IT administrator can manage access to the resources centrally at the hub level, and projects associated with the hub inherit the resources and role assignments; enabling development teams to use the resources they need without needing to request access on a project-by-project basis.

**Projects**

A hub can support one or more projects, each of which is used to organize the resources and assets required for a particular AI development effort.



Users can collaborate in a project, sharing data in project-specific storage containers and connected resources, and using the shared resources defined in the hub associated with the project. Azure AI Foundry provides tools and functionality within a project that developers can use to build AI solutions efficiently, including:

* A *model catalog* in which you can find and deploy machine learning models from multiple sources, including Azure OpenAI and the Hugging Face model library.
* *Playgrounds* in which you can test prompts with generative AI models.
* Access to *Azure AI services*, including visual interfaces to experiment with and configure services as well as endpoints and keys that you can use to connect to them from client applications.
* *Visual Studio Code* containers that define a hosted development environment in which you can write, test, and deploy code.
* *Fine-tuning* functionality for generative AI models that you need to customize based on custom training prompts and responses.
* *Prompt Flow*, a prompt orchestration tool that you can use to define the logic for a generative AI application's interaction with a model.
* Tools to assess, evaluate, and improve your AI applications, including *tracing*, *evaluations*, and *content safety and security* management.
* Management of project *assets*, including models and endpoints, data and indexes, and deployed web apps.

**Considerations for Azure AI Foundry**

When planning an AI solution built on Azure AI Foundry, there are some additional considerations to those discussed previously in relation to Azure AI services.

**Hub and project organization**

Plan your hub and project organization for the most effective management of resources and efficiency of administration. Use Hubs to centralize management of users and shared resources that are involved in related projects, and then add project-specific resources as necessary. For example, an organization might have separate software development teams for each area of the business, so it may make sense to create separate hubs for each business area (such as Marketing, HR, and so on) in which AI application development projects for each business area can be created. The shared resources in each hub will automatically be available in projects created in those hubs.

**Connected resources**

At the hub level, an IT administrator can create shared resource connections in a hub that will be used in downstream projects. Projects access the connected resources by proxy on behalf of project users, so users in those projects don't need direct access to those resources in order to use them within the context of the project. Connections in a hub are automatically available in new projects in the hub without further requests to the IT administrator. If an individual project needs access to a specific resource that other projects in the same hub don't use, you can create more connected resources at the project level.

As you plan your Azure AI Foundry hubs and projects, identify the shared connected resources you should add to each hub so that they're inherited by projects in that hub, while allowing for project-level exceptions.

**Security and authorization**

For each hub and project, identify the users who will need access and the roles to which they should be assigned.

Hub-level roles can perform infrastructure management tasks, such as creating hub-level connected resources or new projects. The default roles in a hub are:

* **Owner**: Full access to the hub, including the ability to manage and create new hubs and assign permissions. This role is automatically assigned to the hub creator
* **Contributor**: Full access to the hub, including the ability to create new hubs, but isn't able to manage hub permissions on the existing resource.
* **Azure AI Developer**: All permissions except create new hubs and manage the hub permissions.
* **Azure AI Inference Deployment Operator**: All permissions required to create a resource deployment within a resource group.
* **Reader**: Read only access to the hub. This role is automatically assigned to all project members within the hub.

Project-level roles determine the tasks that a user can perform within an individual project. The default roles in a project are:

* **Owner**: Full access to the project, including the ability to assign permissions to project users.
* **Contributor**: Full access to the project but can't assign permissions to project users.
* **Azure AI Developer**: Permissions to perform most actions, including create deployments, but can't assign permissions to project users.
* **Azure AI Inference Deployment Operator**: Permissions to perform all actions required to create a resource deployment within a resource group.
* **Reader**: Read only access to the project.

**Regional availability**

As with all Azure services, the availability of specific Azure AI Foundry capabilities can vary by region. As you plan your solution, determine regional availability for the capabilities you require.

**Costs and quotas**

In addition to the cost of the Azure AI services your solution uses, there are costs associated with Azure AI Foundry related to the resources that support hubs and projects as well as storage and compute for assets, development, and deployed solutions. You should consider these costs when planning to use Azure AI Foundry for AI solution development.

In addition to service consumption costs, you should consider the resource quotas you need to support the AI applications you intend to build. Quotas are used to limit utilization, and play a key role in cost management and managing Azure capacity. In some cases, you may need to request additional quota to increase rate limits for AI model operations or available compute for development and solution deployment.

**Developer tools and SDKs**

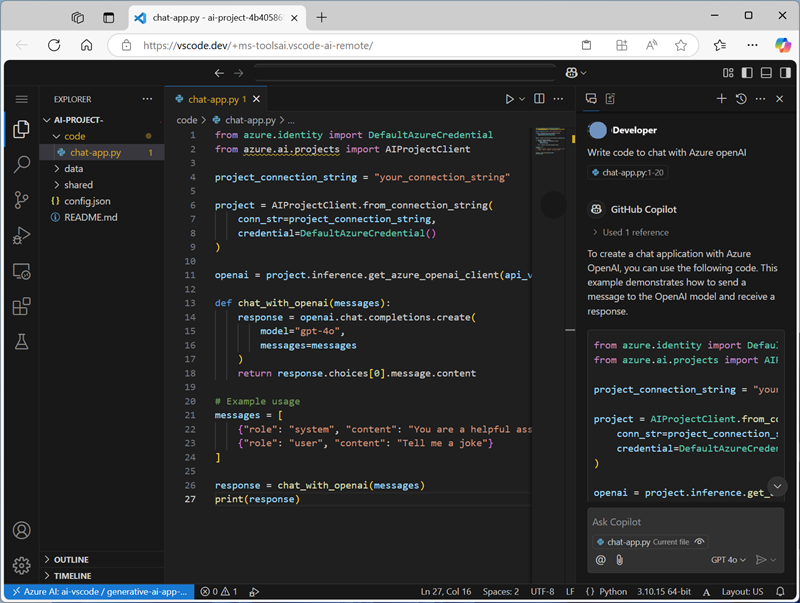
While you can perform many of the tasks needed to develop an AI solution directly in the Azure AI Foundry portal, developers also need to write, test, and deploy code.

**Development tools and environments**

There are many development tools and environments available, and developers should choose one that supports the languages, SDKs, and APIs they need to work with and with which they're most comfortable. For example, a developer who focuses strongly on building applications for Windows using the .NET Framework might prefer to work in an integrated development environment (IDE) like Microsoft Visual Studio. Conversely, a web application developer who works with a wide range of open-source languages and libraries might prefer to use a code editor like Visual Studio Code (VS Code). Both of these products are suitable for developing AI applications on Azure.

**The Azure AI Foundry VS Code container image**

As an alternative to installing and configuring your own development environment, within Azure AI Foundry portal, you can create compute and use it to host a container image for VS Code (installed locally or as a hosted web application in a browser). The benefit of using the container image is that it includes the latest versions of the SDK packages you're most likely to work with when building AI applications with Azure AI Foundry.



**Important**

When planning to use the VS Code container image in Azure AI Foundry, consider the cost of the compute required to host it and the quota you have available to support developers using it.

**GitHub and GitHub Copilot**

GitHub is the world's most popular platform for source control and DevOps management, and can be a critical element of any team development effort. Visual Studio and VS Code (including the Azure AI Foundry VS Code container image) both provide native integration with GitHub, and access to GitHub Copilot; an AI assistant that can significantly improve developer productivity and effectiveness.

**Programming languages, APIs, and SDKs**

You can develop AI applications using many common programming languages and frameworks, including Microsoft C#, Python, Node, TypeScript, Java, and others. When building AI solutions on Azure, some common SDKs you should plan to install and use include:

* The [**Azure AI Foundry SDK**](https://learn.microsoft.com/en-us/azure/ai-studio/how-to/develop/sdk-overview), which enables you to write code to connect to Azure AI Foundry projects and access resource connections, which you can then work with using service-specific SDKs.
* [**Azure AI Services SDKs**](https://learn.microsoft.com/en-us/azure/ai-services/reference/sdk-package-resources) - AI service-specific libraries for multiple programming languages and frameworks that enable you to consume Azure AI Services resources in your subscription. You can also use Azure AI Services through their [REST APIs](https://learn.microsoft.com/en-us/azure/ai-services/reference/rest-api-resources).
* The [**Azure AI Agent Service**](https://learn.microsoft.com/en-us/azure/ai-services/agents/overview), which is accessed through the Azure AI Foundry SDK and can be integrated with frameworks like [AutoGen](https://microsoft.github.io/autogen/0.2/docs/Getting-Started" \t "az-portal) and [Semantic Kernel](https://learn.microsoft.com/en-us/semantic-kernel/overview) to build comprehensive AI agent solutions.
* The [**Prompt Flow**](https://microsoft.github.io/promptflow/index.html) SDK, which you can use to implement orchestration logic to manage prompt interactions with generative AI models.

**Responsible AI**

It's important for software engineers to consider the impact of their software on users, and society in general; including considerations for its responsible use. When the application is imbued with artificial intelligence, these considerations are particularly important due to the nature of how AI systems work and inform decisions; often based on probabilistic models, which are in turn dependent on the data with which they were trained.

The human-like nature of AI solutions is a significant benefit in making applications user-friendly, but it can also lead users to place a great deal of trust in the application's ability to make correct decisions. The potential for harm to individuals or groups through incorrect predictions or misuse of AI capabilities is a major concern, and software engineers building AI-enabled solutions should apply due consideration to mitigate risks and ensure fairness, reliability, and adequate protection from harm or discrimination.

Let's discuss some core principles for responsible AI that have been adopted at Microsoft.

**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.

Fairness of machine learned systems is a highly active area of ongoing research, and some software solutions exist for evaluating, quantifying, and mitigating unfairness in machine learned models. However, tooling alone isn't sufficient to ensure fairness. Consider fairness from the beginning of the application development process; carefully reviewing training data to ensure it's representative of all potentially affected subjects, and evaluating predictive performance for subsections of your user population throughout the development lifecycle.

**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.

As with any software, AI-based software application development must be subjected to rigorous testing and deployment management processes to ensure that they work as expected before release. Additionally, software engineers need to take into account the probabilistic nature of machine learning models, and apply appropriate thresholds when evaluating confidence scores for predictions.

**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. Even after models are trained and the system is in production, they use new data to make predictions or take action that may be subject to privacy or security concerns; so appropriate safeguards to protect data and customer content must be implemented.

**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.

One way to optimize for inclusiveness is to ensure that the design, development, and testing of your application includes input from as diverse a group of people as possible.

**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.

For example, when an AI system is based on a machine learning model, you should generally make users aware of factors that may affect the accuracy of its predictions, such as the number of cases used to train the model, or the specific features that have the most influence over its predictions. You should also share information about the confidence score for predictions.

When an AI application relies on personal data, such as a facial recognition system that takes images of people to recognize them; you should make it clear to the user how their data is used and retained, and who has access to it.

**Accountability**



People should be accountable for AI systems. Although many AI systems seem to operate autonomously, ultimately it's the responsibility of the developers who trained and validated the models they use, and defined the logic that bases decisions on model predictions to ensure that the overall system meets responsibility requirements. To help meet this goal, designers and developers of AI-based solution should work within a framework of governance and organizational principles that ensure the solution meets responsible and legal standards that are clearly defined.

Knowledge Check

**1. Which Azure resource provides language and vision services from a single endpoint?**

Azure AI Language

Azure AI Vision

Azure AI Services

**Correct. Azure AI Services provides multiple services from a single endpoint.**

**2. How should you provide access to resources for developers who will work on multiple AI projects?**

Create resource connections in an Azure AI Foundry hub.

**Correct. Creating resource connections in a hub means they can be used from all projects created in that hub.**

Create resource connections in each Azure AI Foundry project.

Assign each developer direct access to all of the resources.

**3. Which SDK enables you to connect to shared resources in a hub?**

Azure AI Services SDK

Semantic Kernel SDK

Azure AI Foundry SDK

**Correct. The Azure AI Foundry SDK provides the AI Projects client library, which you can use to access connections in a project or hub.**

**Create and consume Azure AI services**

**Introduction**

**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 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 GPT-4.

While the details of each AI service can vary, the approach to provisioning and consuming them is generally the same.

**Provision an Azure AI services resource**

Azure AI services include a wide range of AI capabilities that you can use in your applications. To use any of the AI services, you need to create appropriate resources in an Azure subscription to define an endpoint where the service can be consumed, provide access keys for authenticated access, and to manage billing for your application's usage of the service.

**Options for Azure resources**

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 (for example to provision them in different geographical regions) 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.

**Training and prediction resources**

While most AI services can be used through a single Azure resource, some offer (or require) separate resources for model *training* and *prediction*. This enables you to manage billing for training custom models separately from model consumption by applications, and in most cases enables you to use a dedicated service-specific resource to train a model, but a generic **AI services** resource to make the model available to applications for inferencing.

**Identify endpoints and keys**

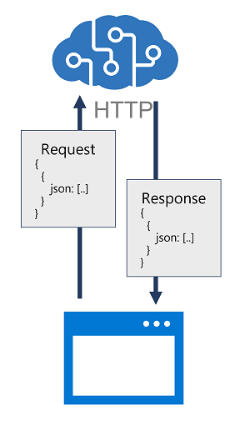
When you provision an Azure AI services service resource in your Azure subscription, you are defining an endpoint through which the service can be consumed by an application.

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.

**Use a REST API**

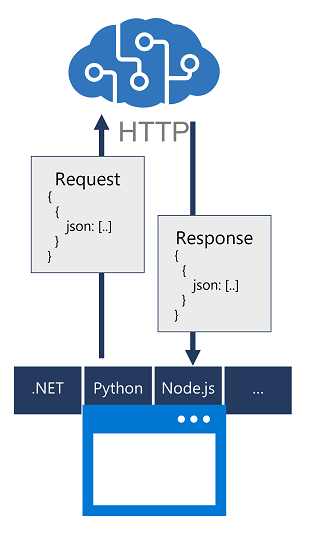
Azure AI services provide REST application programming interfaces (APIs) that client applications can use to consume services. In most cases, service functions can be called by submitting data in JSON format over an HTTP request, which may be a POST, PUT, or GET request depending on the specific function being called. The results of the function are returned to the client as an HTTP response, often with JSON contents that encapsulate the output data from the function.



The use of REST interfaces with an HTTP endpoint means that any programming language or tool capable of submitting and receiving JSON over HTTP can be used to consume AI services. You can use common programming languages such as Microsoft C#, Python, and JavaScript; as well as utilities such as Postman and cURL, which can be useful for testing.

**Use an SDK**

You can develop an application that uses Azure AI services using REST interfaces, but it's easier to build more complex solutions by using native libraries for the programming language in which you're developing the application.



Software development kits (SDKs) for common programming languages abstract the REST interfaces for most AI services. SDK availability varies by individual AI services, but for most services there's an SDK for languages such as:

* Microsoft C# (.NET Core)
* Python
* JavaScript (Node.js)
* Go
* Java

Each SDK includes packages that you can install in order to use service-specific libraries in your code, and online documentation to help you determine the appropriate classes, methods, and parameters used to work with the service.

**1. How are client applications typically granted access to an Azure AI Services endpoint?**

The application must specify a valid subscription key for the Azure resource.

**Correct. By default, access to an Azure AI Services resource is based on a subscription key.**

The user of the application must enter a user name and password associated with the Azure subscription.

Access to Azure AI Services is granted to anonymous users by default.

**2. In which format are message exchanged between a client app and an Azure AI Services resource when using a REST API?**

XML

JSON

**Correct. Client apps submit JSON requests and receive JSON responses.**

HTML

**Secure Azure AI services**

**Introduction**

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

After completing this module, you'll learn how to:

* Consider authentication for Azure AI services.
* Manage network security for Azure AI services.

**Consider authentication**

By default, access to Azure AI services resources is restricted by using subscription keys. Management of access to these keys is a primary consideration for security.

**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.

Each AI service is provided with two keys, enabling you to regenerate keys without service interruption. To accomplish this:

1. If you're using both keys in production, change your code so that only one key is in use. For example, configure all production applications to use key 1.
2. Regenerate key 2.
3. Switch all production applications to use the newly regenerated key 2.
4. Regenerate key 1
5. Finally, update your production code to use the new key 1.

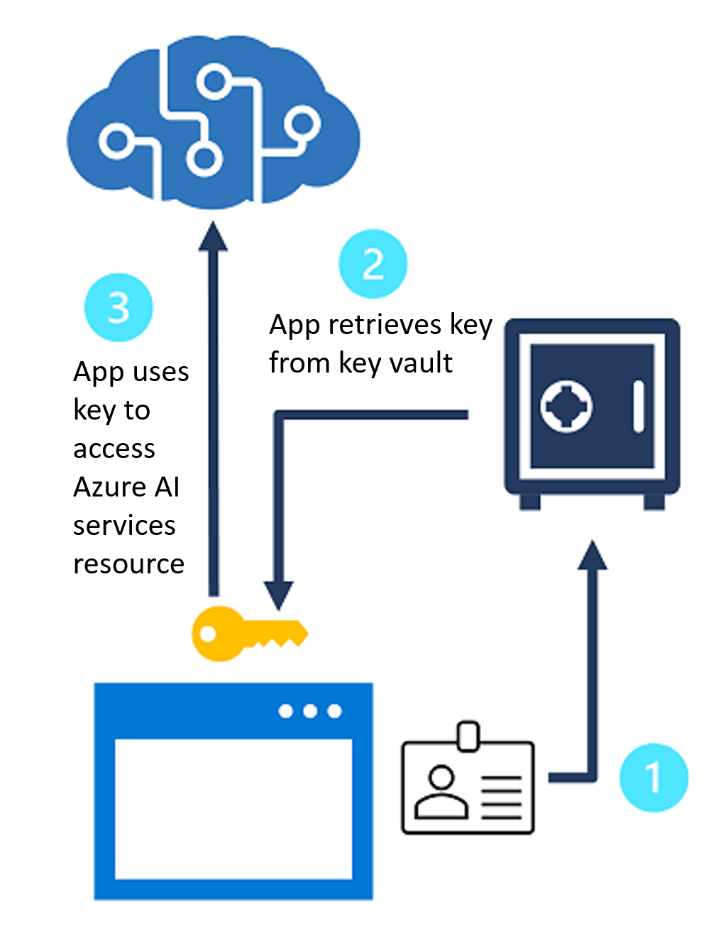
For example, to regenerate keys in the Azure portal, you can do the following:

1. In the Azure portal, go to your resource's Keys and Endpoint pane.
2. Then select **Regenerate Key1** or select **Regenerate Key2**, depending on which one you want to regenerate at the time.

**Protect keys with Azure Key Vault**

Azure Key Vault is an Azure service in which you can securely store secrets (such as passwords and keys). 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. Controlling access to the secret in this way minimizes the risk of it being compromised by being hard-coded in an application or saved in a configuration file.

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. The applications can then retrieve the key as needed from the key vault, without risk of exposing it to unauthorized users.



**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.

When using an SDK, the calls to obtain and present a token are handled for you by the SDK.

**Microsoft Entra ID authentication**

Azure AI services supports 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**

The overall process to authenticate against Azure AI services using service principals is as follows:

**Create a custom subdomain**

You can create a custom subdomain in different ways including through the Azure portal, Azure CLI, or PowerShell.

For example, you can create a subdomain using PowerShell in the Azure Cloud Shell. To do this, you select your subscription using the following command:

PowerShellCopy

Set-AzContext -SubscriptionName <Your-Subscription-Name>

Then, you create your Azure AI services resource specifying a custom subdomain by running the following:

PowerShellCopy

$account = New-AzCognitiveServicesAccount -ResourceGroupName <your-resource-group-name> -name <your-account-name> -Type <your-account-type> -SkuName <your-sku-type> -Location <your-region> -CustomSubdomainName <your-unique-subdomain-name>

Once created, your subdomain name will be returned in the response.

**Assign a role to a service principal**

You've created an Azure AI resource that is linked with a custom subdomain. Next, you assign a role to a service principal.

To start, you'll need to register an application. To do this, you run the following command:

PowerShellCopy

$SecureStringPassword = ConvertTo-SecureString -String <your-password> -AsPlainText -Force

$app = New-AzureADApplication -DisplayName <your-app-display-name> -IdentifierUris <your-app-uris> -PasswordCredentials $SecureStringPassword

This creates the application resource.

Then you use the **New-AzADServicePrincipal** command to create a service principal and provide your application's ID:

PowerShellCopy

New-AzADServicePrincipal -ApplicationId <app-id>

Finally, you assign the **Cognitive Services Users** role to your service principal by running:

PowerShellCopy

New-AzRoleAssignment -ObjectId <your-service-principal-object-id> -Scope <account-id> -RoleDefinitionName "Cognitive Services User"

**Authenticate using managed identities**

Managed identities come in two types:

* **System-assigned managed identity**: A managed identity is created and linked to a specific resource, such as a virtual machine that needs to access Azure AI services. When the resource is deleted, the identity is deleted as well.
* **User-assigned managed identity**: The managed identity is created to be useable by multiple resources instead of being tied to one. It exists independently of any single resource.

You can assign each type of managed identity to a resource either during creation of the resource, or after it has already been created.

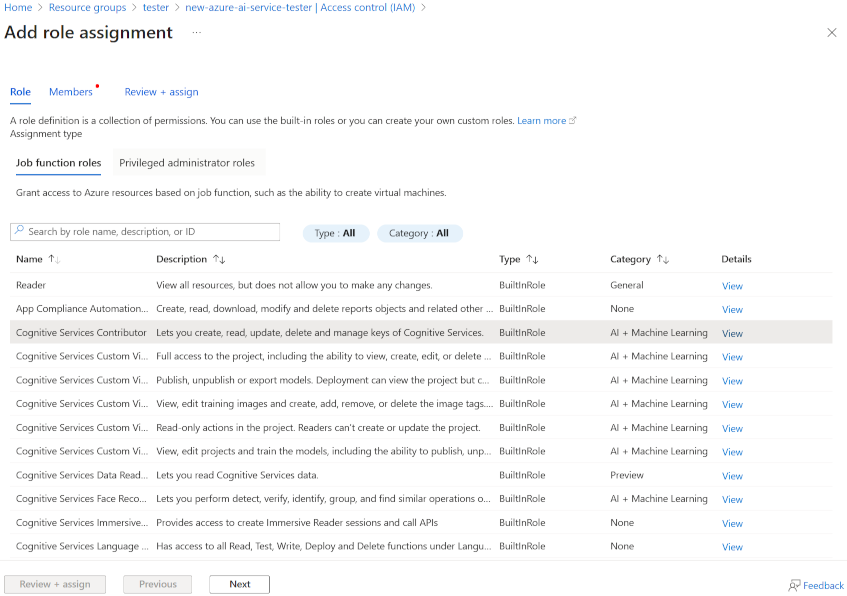
For example, suppose you have a virtual machine in Azure that you intend to use for daily access to Azure AI services. To enable a system-assigned identity for this virtual machine, first you make sure your Azure account has the [Virtual Machine Contributor role](https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles). Then you can run the following command using Azure CLI in the Azure Cloud Shell terminal:

Azure CLICopy

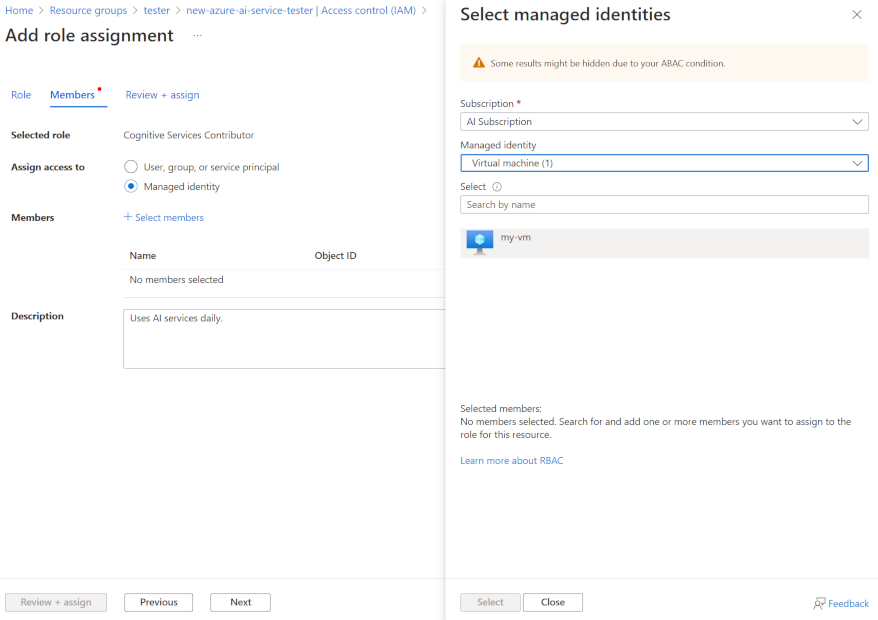
az vm identity assign -g <my-resource-group> -n <my-vm>

Then you can grant access to Azure AI services in the Azure portal using the following:

1. Go to the Azure AI services resource you want to grant the virtual machine's managed identity access.
2. In the overview panel, select **Access control (IAM)**.
3. Select **Add**, and then select **Add role assignment**.
4. In the Role tab, select **Cognitive Services Contributor**.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/secure-ai-services/media/select-contributor-role.png#lightbox)

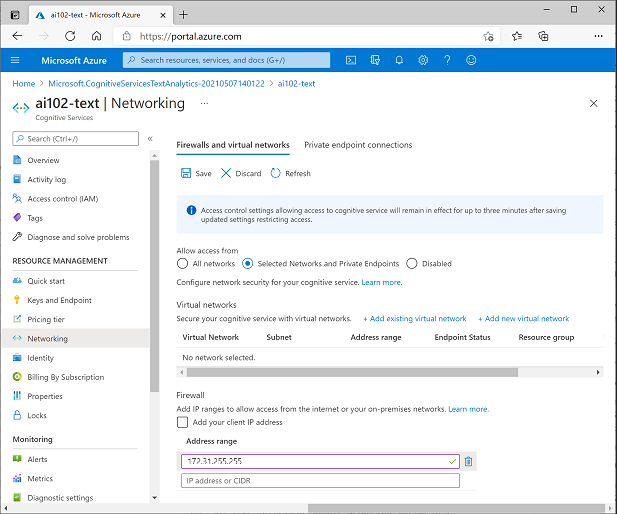
1. In the Members tab, for the Assign access to, select **Managed identity**. Then, select **+ Select members**.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/secure-ai-services/media/select-managed-identity.png#lightbox)

1. Ensure that your subscription is selected in the Subscription dropdown. And for Managed identity, select **Virtual machine**.
2. Select your virtual machine in the list, and select **Select**.
3. Finally, select **Review + assign** to review, and then **Review + assign** again to finish.

**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.

**Knowledge Check**

**1. You need to regenerate the primary subscription key for an Azure AI Services resource that an app uses. What should you do first to minimize service interruption for the app?**

Switch the app to use the secondary key

**Correct. The app can use the secondary key to access the resource.**

Change the resource endpoint

Enable a firewall

**2. You want to store the subscription keys for an Azure AI Services resource securely, so that authorized apps can retrieve them when needed. What kind of Azure resource should you provision?**

Azure Storage

Azure Key Vault

**Correct. Use Azure Key Vault to store credentials securely.**

Azure App Service

**3.When running code on your computer that connects to Azure AI Services, you receive an error that access is denied due to Virtual Network/Firewall rules. What configuration do you need to set in the Azure AI Services instance?**

In the Networking properties, configure Selected Networks and Private Endpoints.

In Networking properties, add your client IP address to the Firewall allowed list.

**Correct. The public IP address for your computer can be allowed access in the Firewall settings.**

In Access control, add your Microsoft Entra ID user account to a role.

**Monitor Azure AI services**

**Introduction**

Azure AI services provides a cloud-based platform for building artificial intelligence capabilities into your applications. Like any software service, you should monitor AI services to track costs, identify utilization trends, and detect potential issues.

After completing this module, you'll be able to:

* Monitor Azure AI services costs.
* Create alerts and view metrics for Azure AI services.
* Manage Azure AI services diagnostic logging.

**Monitor cost**

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. The specific billing rate depends on the resource type.

**Plan costs for AI services**

Before deploying a solution that depends on AI services, you can estimate costs by using the [Azure Pricing Calculator](https://azure.microsoft.com/pricing/calculator/).

To use the pricing calculator to estimate AI services costs, create a new estimate and select **Azure AI services** in the **AI + Machine Learning** category. Select the specific AI service API you plan to use. For example, *Azure AI Language*. For the API, select the region where you plan to provision it and the pricing tier of the instance you plan to use. Then, fill in the expected usage metrics and support option. To create an estimate that includes multiple AI services APIs, add more **Azure AI services** products to the estimate.

After you create an estimate, you can save it. You can also export it in Microsoft Excel format.

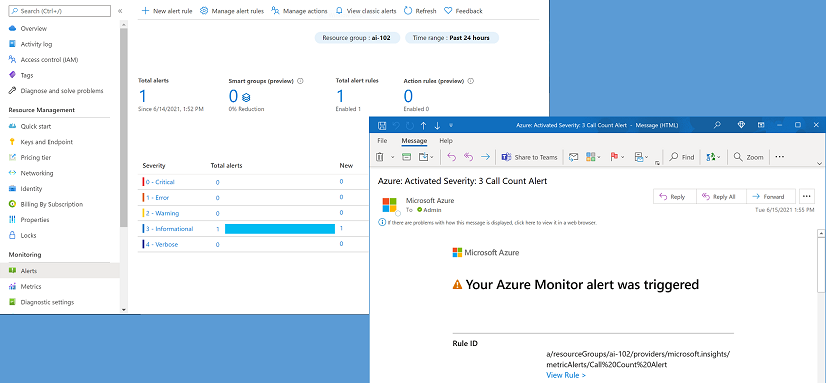
**View costs for AI services**

In common with other Azure resources, you can view details of accumulated costs for AI services resources in the Azure portal.

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 **Azure AI 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.

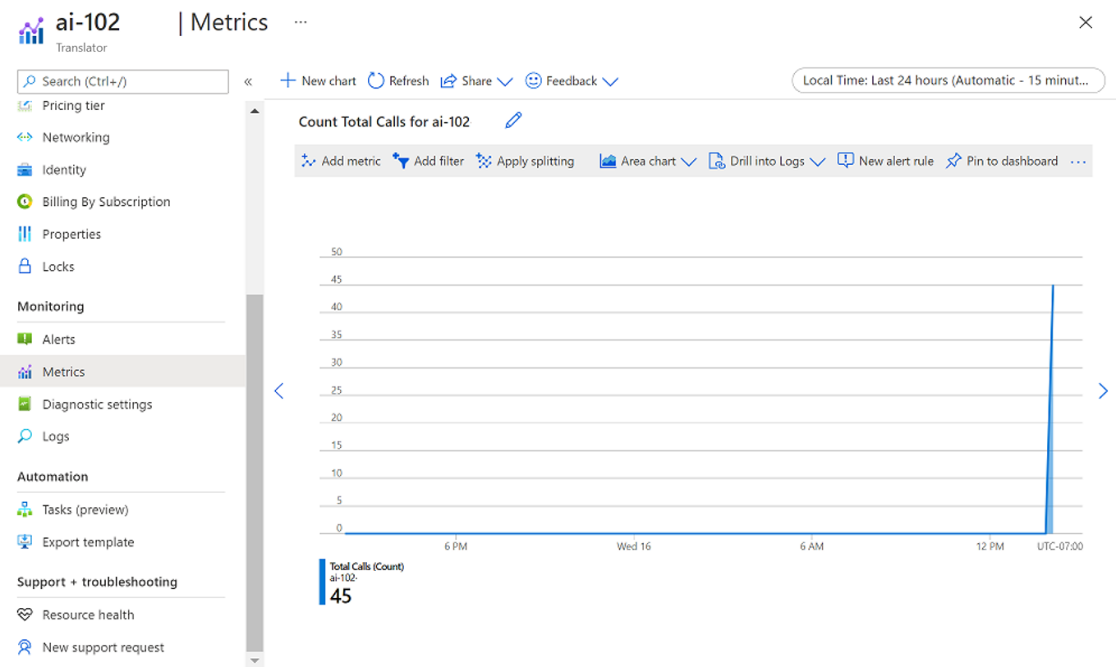
**View metrics**

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.

**View metrics in the Azure portal**

You can view metrics for an individual resource in the Azure portal by selecting the resource and viewing its **Metrics** page. On this page, you can add resource-specific metrics to charts. By default an empty chart is created for you, and you can add more charts as required.

For example, the following image shows the **Metrics** page for an AI services resource, showing the count of total calls to the service over a period of time.



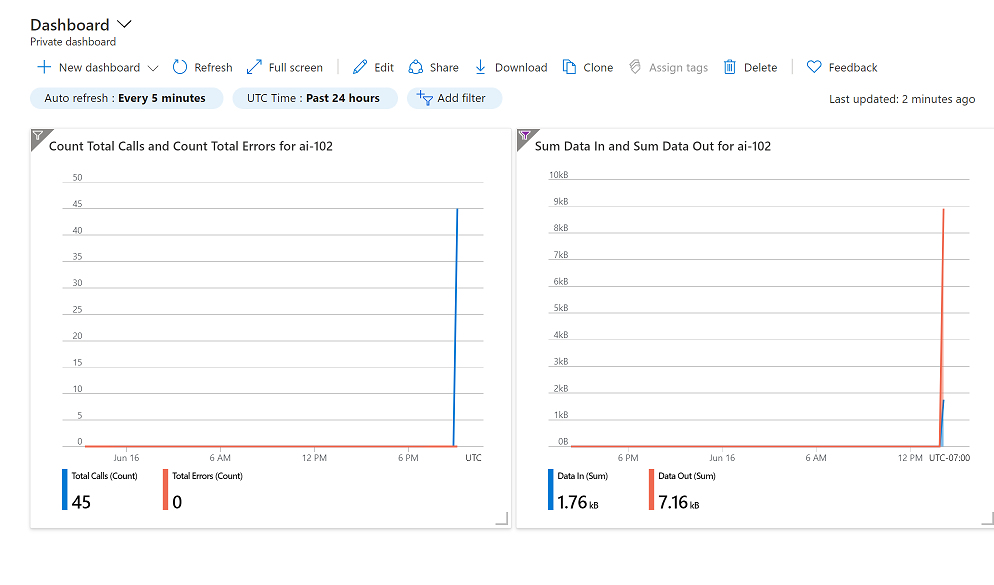
You can add multiple metrics to a chart and choose appropriate aggregations and chart types. When you're satisfied with chart, you can *share* it by exporting it to Excel or copying a link to it, and you can *clone* it to create a duplicate chart in the **Metrics** page - potentially as a starting point for a new chart that shows the same metrics in a different way.

**Add metrics to a dashboard**

In the Azure portal, you can create *dashboards* that consist of multiple visualizations from different resources in your Azure environment to help you gain an overall view of the health and performance of your Azure resources.

To create a dashboard, select **Dashboard** in the Azure portal menu (your default view may already be set to a dashboard rather than the portal home page). From here, you can add up to 100 named dashboards to encapsulate views for specific aspects of your Azure services that you want to track.

You can add a range of tiles and other visualizations to a dashboard, and when viewing metrics for a specific resource in a chart, as described previously, you can add the chart to a new or existing dashboard. In the following image, two charts showing metrics for an AI services resource have been added to a dashboard.



**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.

**Create resources for diagnostic log storage**

To capture diagnostic logs for an AI services resource, you need a destination for the log data. In certain cases, you can use Azure Event Hubs as a destination for the log data. Azure Event Hubs allows you to forward the data on to a custom telemetry solution and connect directly to some third-party solutions. However, in most cases you'll use one (or both) of the following kinds of resource within your Azure subscription:

* **Azure Log Analytics** - a service that enables you to query and visualize log data within the Azure portal.
* **Azure Storage** - a cloud-based data store that you can use to store log archives (which can be exported for analysis in other tools as needed).

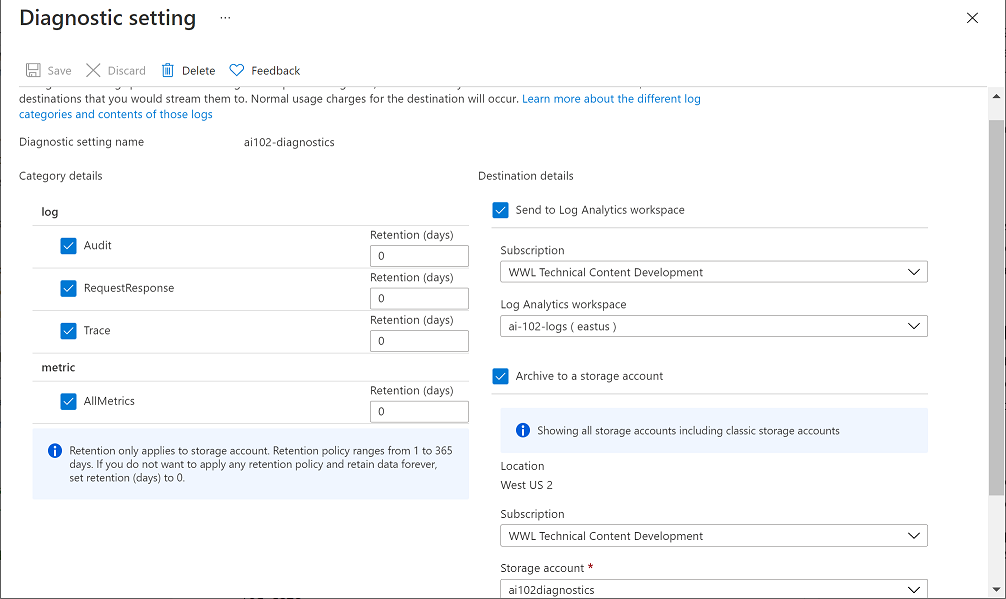
You should create these resources before configuring diagnostic logging for your AI services resource. If you intend to archive log data to Azure Storage, create the Azure Storage account in the same region as your AI services resource.

**Configure diagnostic settings**

With your log destinations in place, you can configure diagnostic settings for your AI services resource. You define diagnostic settings on the **Diagnostic settings** page of the blade for your AI services resource in the Azure portal. When you add diagnostic settings, you must specify:

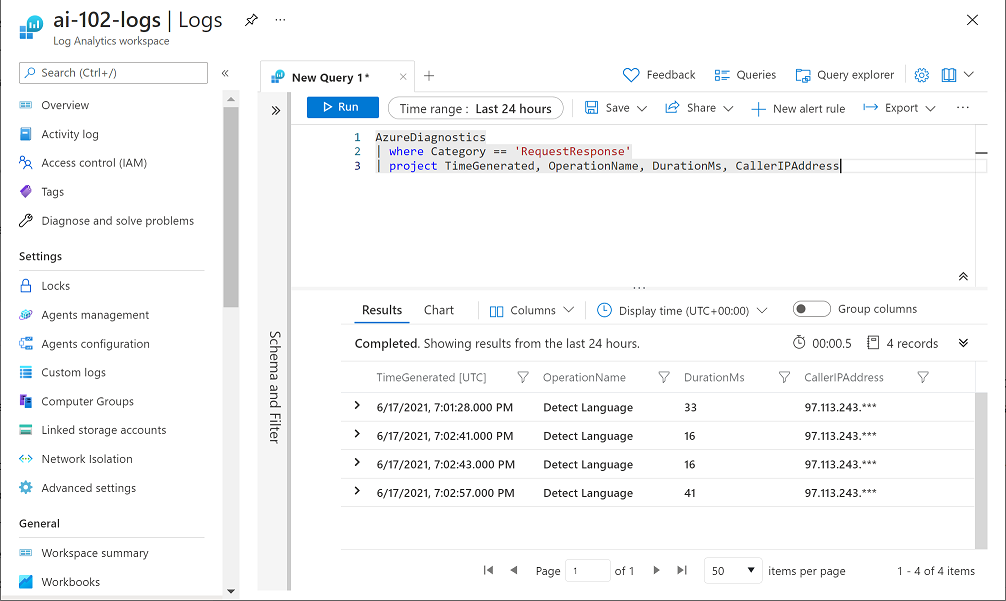
* A name for your diagnostic settings.
* The categories of log event data that you want to capture.
* Details of the destinations in which you want to store the log data.

In the following example, the diagnostic settings store all available log data and metrics in Azure Log Analytics and Azure Storage.



**View log data in Azure Log Analytics**

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, as shown in this example.



**Knowledge check**

Top of Form

**1. How should you collect telemetry for your Azure AI Services resource for later analysis?**

Create an alert.

Configure diagnostic settings.

**Correct. Diagnostic settings enable you to capture data for subsequent analysis.**

Create a dashboard.

**2. You are defining an alert that notifies you when a key regeneration event is recorded in the activity log for your Azure AI Services resource. What should you do?**

Specify a Scope of Activity Log.

Specify an Action that uses an Azure Logic App to read the activity log.

Specify a Condition with a Signal Type of Activity Log.

**Correct. The Regenerate Key event is an Activity Log signal.**

**3. You are viewing a metric for your Azure AI Services resource in a chart. You want to combine the chart with visualizations of other resources and data. What should you do?**

Add the chart to a dashboard.

**Correct. A dashboard enables you to combine visualizations from multiple resources.**

Share the chart.

Clone the chart.

**Deploy Azure AI services in containers**

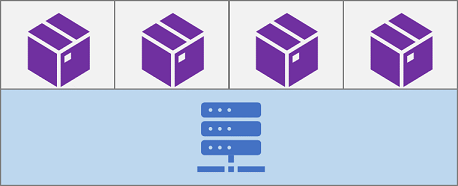
**Introduction**

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.

**Understand containers**

When you deploy a software service, it must be hosted in an environment that provides the hardware, operating system, and supporting runtime components on which the service depends.

Azure AI services is provided as a cloud service, in which the service software is hosted in an Azure data center that provides the underlying runtime services, operating system, and hardware. However, you can also deploy some Azure AI services in a *container*, which encapsulates the necessary runtime components, and which is in turn deployed in a container host that provides the underlying operating system and hardware.



**What is a container?**

A container comprises an application or service and the runtime components needed to run it, while abstracting the underlying operating system and hardware. In practice, this abstraction results in two significant benefits:

* Containers are portable across hosts, which may be running different operating systems or use different hardware - making it easier to move an application and all its dependencies.
* A single container host can support multiple isolated containers, each with its own specific runtime configuration - making it easier to consolidate multiple applications that have different configuration requirement.

A container is encapsulated in a *container image* that defines the software and configuration it must support. Images can be stored in a central registry, such as *Docker Hub*, or you can maintain a set of images in your own registry.

**Container deployment**

To use a container, you typically pull the container image from a registry and deploy it to a container host, specifying any required configuration settings. The container host can be in the cloud, in a private network, or on your local computer. For example:

* A *Docker*\* server.
* An Azure Container Instance (ACI).
* An Azure Kubernetes Service (AKS) cluster.

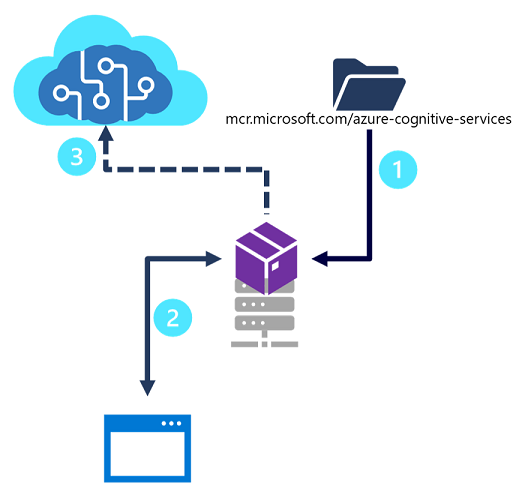
\**Docker is an open source solution for container development and management that includes a server engine that you can use to host containers. There are versions of the Docker server for common operating systems, including Microsoft Windows and Linux.*

**Use Azure AI services containers**

There are container images for Azure AI services in the Microsoft Container Registry that you can use to deploy a containerized service that encapsulates an individual Azure AI services service API.

To deploy and use an Azure AI services container, the following three activities must occur:

1. The container image for the specific Azure AI services API you want to use is downloaded and deployed to a container host, such as a local Docker server, an Azure Container Instance (ACI), or Azure Kubernetes Service (AKS).
2. Client applications submit data to the endpoint provided by the containerized service, and retrieve results just as they would from an Azure AI services cloud resource in Azure.
3. Periodically, usage metrics for the containerized service are sent to an Azure AI services resource in Azure in order to calculate billing for the service.



Even when using a container, you must provision an Azure AI services resource in Azure for billing purposes. Client applications send their requests to the containerized service, meaning that potentially sensitive data is not sent to the Azure AI services endpoint in Azure; but the container must be able to connect to the Azure AI services resource in Azure periodically to send usage metrics for billing.

**Azure AI services container images**

Each container provides a subset of Azure AI services functionality. For example, not all features of the Azure AI Language service are in a single container. Language detection, translation, and sentiment analysis are each separate container images. However, the setup steps are similar for each container.

**Language containers**

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

Expand table

| **Feature** | **Image** |
| --- | --- |
| Key Phrase Extraction | mcr.microsoft.com/azure-cognitive-services/textanalytics/keyphrase |
| Language Detection | mcr.microsoft.com/azure-cognitive-services/textanalytics/language |
| Sentiment Analysis | mcr.microsoft.com/azure-cognitive-services/textanalytics/sentiment |
| Named Entity Recognition | mcr.microsoft.com/product/azure-cognitive-services/textanalytics/language/about |
| Text Analytics for health | mcr.microsoft.com/product/azure-cognitive-services/textanalytics/healthcare/about |
| Translator | mcr.microsoft.com/product/azure-cognitive-services/translator/text-translation/about |
| Summarization | mcr.microsoft.com/azure-cognitive-services/textanalytics/summarization |

**Note**

Sentiment Analysis supports other languages by replacing the *en* in the image with the correct language code

**Speech containers**

| **Feature** | **Image** |
| --- | --- |
| Speech to text | mcr.microsoft.com/product/azure-cognitive-services/speechservices/speech-to-text/about |
| Custom Speech to text | mcr.microsoft.com/product/azure-cognitive-services/speechservices/custom-speech-to-text/about |
| Neural Text to speech | mcr.microsoft.com/product/azure-cognitive-services/speechservices/neural-text-to-speech/about |
| Speech language detection | mcr.microsoft.com/product/azure-cognitive-services/speechservices/language-detection/about |

**Vision containers**

| **Feature** | **Image** |
| --- | --- |
| Read OCR | mcr.microsoft.com/product/azure-cognitive-services/vision/read/about |
| Spatial analysis | mcr.microsoft.com/product/azure-cognitive-services/vision/spatial-analysis/about |

You can use the Docker *pull* command to download container images to work with them directly from your machine. Some of the containers are in a "Gated" public preview state, and you need to explicitly request access to use them. Otherwise the containers are available for anyone to use with their Azure AI services deployment.

For a full list of currently available Azure AI services container images, and specific notes for each one, see [Azure AI services container image tags and release notes](https://learn.microsoft.com/en-us/azure/ai-services/cognitive-services-container-support#containers-in-azure-ai-services).

**Azure AI services container configuration**

When you deploy an Azure AI services container image to a host, you must specify three settings.

Expand table

| **Setting** | **Description** |
| --- | --- |
| 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. |

**Consuming Azure AI services from a Container**

After your Azure AI services container is deployed, applications consume the containerized Azure AI services endpoint rather than the default Azure endpoint. The client application must be configured with the appropriate endpoint for your container, but does not need to provide a subscription key to be authenticated. You can implement your own authentication solution and apply network security restrictions as appropriate for your specific application scenario.

**Knowledge Check**

**1. You plan to use an Azure AI services container in a local Docker host. Which of the following statements is true?**

Client applications must pass a subscription key to the Azure resource endpoint before using the container.

The container must be able to connect to the Azure resource endpoint to send usage data for billing.

**Correct. container usage metrics are sent to the Azure AI services resource in Azure to calculate billing.**

All data passed from the client application to the container is forwarded to the Azure resource endpoint.

**2. Which of the following parameters must you specify when deploying an Azure AI services container image?**

EULA

**Correct. You must specify a EULA parameter with the value "yes" to explicitly accept the license agreement.**

ResourceGroup

SubscriptionName

**3. You plan to use the language detection functionality of Azure AI Language in a container. Which container image should you deploy?**

mcr.microsoft.com/azure-ai-services/textanalytics

mcr.microsoft.com/azure-ai-services

mcr.microsoft.com/azure-ai-services/textanalytics/language

**Correct. You must deploy the image that is specific to language detection**

**Use AI responsibly with Azure AI Content Safety**

**Introduction**

The amount of user-generated content being posted online is growing rapidly. We are also increasingly aware of the need to protect everyone from inappropriate or harmful content.

Azure AI Content Safety is an AI service designed to help developers include advanced content safety into their applications and services.

The challenges in maintaining safe and respectful online spaces are growing for developer teams responsible for hosting online discussions. Azure AI Content Safety identifies potentially unsafe content and helps organizations to comply with regulations and meet their own quality standards.

The need for improving online content safety has four main drivers:

* **Increase in harmful content**: There's been a huge growth in user-generated online content, including harmful and inappropriate content.
* **Regulatory pressures**: Government pressure to regulate online content.
* **Transparency**: Users need transparency in content moderation standards and enforcement.
* **Complex content**: Advances in technology are making it easier for users to post multimodal content and videos.

**What is Content Safety**

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.

Online content safeguarding is needed in a growing number of situations. Not only are we concerned with moderating content generated by people, but must also guard against the malicious use of AI.

**Trusting user-generated content**

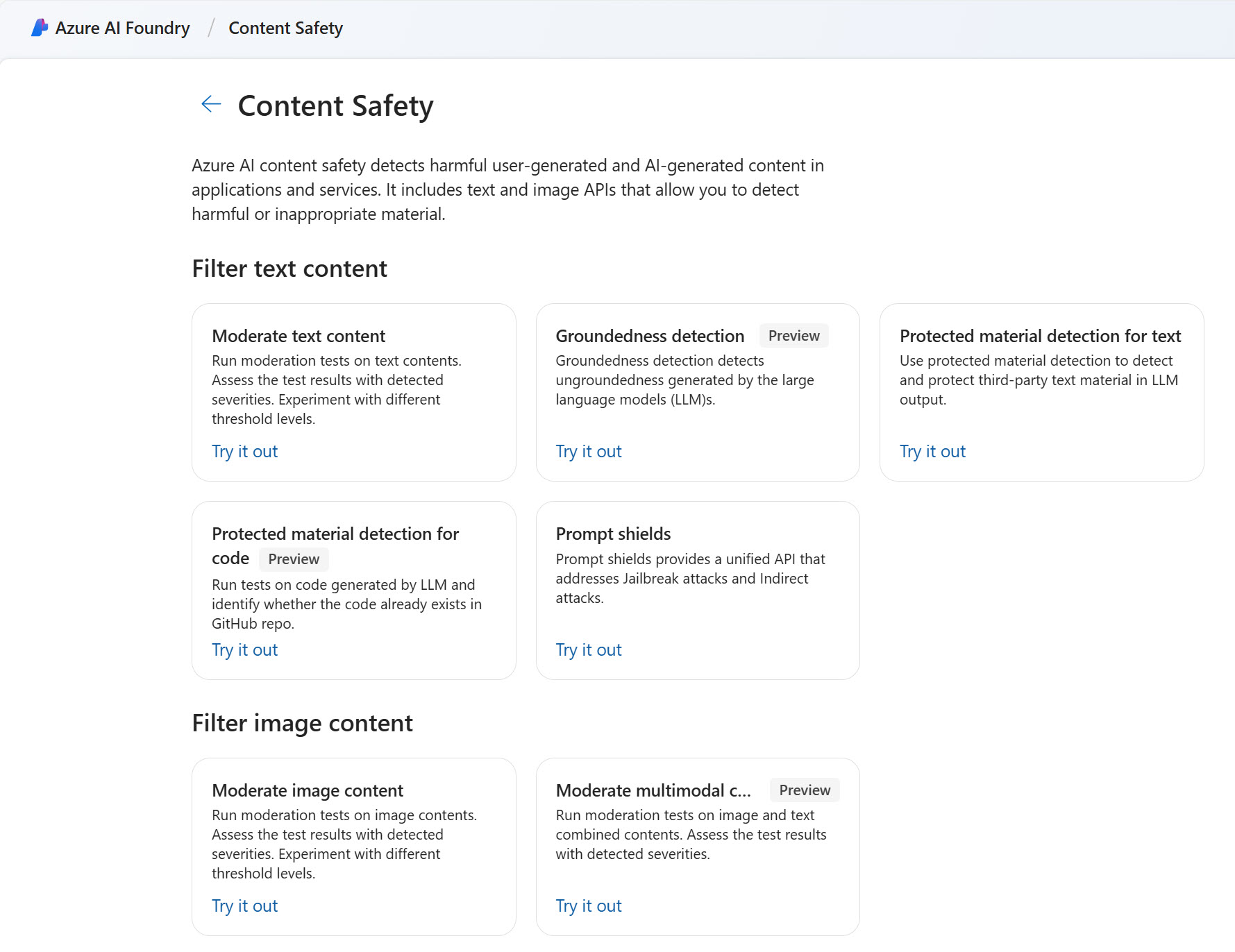
Social interaction is increasingly a part of many digital spaces. Genuine user-generated content is seen as independent and trustworthy, and used alongside advertising and marketing. Different industries are encouraging their customers to connect with each other and their brand.

Harmful content has many negative effects. It damages trusted brands, discourages users from participating in online forums, and can have a devastating impact on individuals.

Azure AI Content Safety is designed to be used in applications and services to protect against harmful user-generated and AI-generated content.

**Content Safety in Azure AI Foundry**

Azure AI Content Safety is available as part of [Azure AI Foundry](https://ai.azure.com/explore/contentsafety), a unified platform that enables you to explore many different Azure AI services, including Content Safety.



From the [Azure AI Foundry](https://ai.azure.com/) home page, scroll down and select *Explore Azure AI Services*. From here, you can explore Content Safety by selecting **View all Content Safety capabilities**.

Azure AI Content Safety Studio enables you to explore and test Content Safety features for yourself. Select the feature you want to try, and then select *Try it out*. You can then use the user interface to test samples or your own material. Select *View code* to generate sample code in C#, Java, or Python. You can then copy and paste the sample code and amend the variables to use your own data.

**Note**

You can access Azure AI Content Studio either through [**Azure AI Foundry**](https://ai.azure.com/) or through [**Content Safety Studio**](https://contentsafety.cognitive.azure.com/).

For guided practice using Azure AI Content Studio, see [Moderate content and detect harm with Azure AI Content Safety Studio](https://learn.microsoft.com/en-us/training/modules/moderate-content-detect-harm-azure-ai-content-safety-studio/).

**How does Azure AI Content Safety work?**

Azure AI Content Safety works with text and images, and AI-generated content.

Content Safety vision capabilities are powered by Microsoft's Florence foundation model, which has been trained with billions of text-image pairs. Text analysis uses natural language processing techniques, giving a better understanding of nuance and context. Azure AI Content Safety is multilingual and can detect harmful content in both short form and long form. It's currently available in English, German, Spanish, French, Portuguese, Italian, and Chinese.

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. This level helps to prioritize what needs immediate attention by people, and how urgently. You can also create a blocklist to scan for terms specific to your situation.
* **Prompt shields** is a unified API to identify and block jailbreak attacks from inputs to LLMs. It includes both user input and documents. These attacks are prompts to LLMs that attempt to bypass the model's in-built safety features. User prompts are tested to ensure the input to the LLM is safe. Documents are tested to ensure they don't contain unsafe instructions embedded within the text.
* **Protected material detection** checks AI-generated text for protected text such as recipes, copyrighted song lyrics, or other original material.
* **Groundedness detection** protects against inaccurate responses in AI-generated text by LLMs. Public LLMs use data available at the time they were trained. However, data can be introduced after the original training of the model or be built on private data. A grounded response is one where the model’s output is based on the source information. An ungrounded response is one where the model's output varies from the source information. Groundedness detection includes a *reasoning* option in the API response. This adds a *reasoning* field that explains any ungroundedness detection. However, reasoning increases processing time and costs.

**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. You then set a threshold level of low, medium, or high. The combination of the severity and threshold level determines whether the image is allowed or blocked for each category.
* **Moderate multimodal content** scans both images and text, including text extracted from an image using optical character recognition (OCR). Content is analyzed across four categories: violence, hate speech, sexual content, and self-harm.

**Custom safety solutions**

* **Custom categories** enables you to create your own categories by providing positive and negative examples, and training the model. Content can then be scanned according to your own category definitions.
* **Safety system message** helps you to write effective prompts to guide an AI system's behavior.

**Limitations**

Azure AI Content Safety uses AI algorithms, and so may not always detect inappropriate language. And on occasions it might block acceptable language because it relies on algorithms and machine learning to detect problematic language.

Azure AI Content Safety should be tested and evaluated on real data before being deployed. And once deployed, you should continue to monitor the system to see how accurately it's performing.

**Evaluating accuracy**

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 works best to support human moderators who can resolve cases of incorrect identification. When people add content to a site, they don't expect posts to be removed without reason. Communicating with users about why content is removed or flagged as inappropriate helps everyone to understand what is permissible and what isn't.

**When to use Azure AI Content Safety**

Many online sites encourage users to share their views. People trust other people's feedback about products, services, brands, and more. These comments are often frank, insightful, and seen to be free of marketing bias. But not all content is well intended.

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:

**Education**

The number of learning platforms and online educational sites is growing rapidly, with more and more information being added all the time. Educators need to be sure that students aren't being exposed to inappropriate content, or inputting harmful requests to LLMs. In addition, both educators and students want to know that the content they're consuming is correct and close to the source material.

**Social**

Social media platforms are dynamic and fast moving, requiring real-time moderation. Moderation of user-generated content includes posts, comments, and images. Azure AI Content Safety helps moderate content that is nuanced and multi-lingual to identify harmful material.

**Brands**

Brands are making more use of chat rooms and message forums to encourage loyal customers to share their views. However offensive material can damage a brand, and discourage customers from contributing. They want to be assured that inappropriate material can be quickly identified and removed. Brands are also adding generative AI services to help people to communicate with them, and therefore need to guard against bad actors attempting to exploit large language models (LLMs).

**E-Commerce**

User content is generated by reviewing products and discussing products with other people. This material is powerful marketing, but when inappropriate content is posted it damages consumer confidence. In addition, regulatory and compliance issues are increasingly important. Azure AI Content Safety helps screen product listings for fake reviews and other unwanted content.

**Gaming**

Gaming is a challenging area to moderate due to its highly visual and often violent graphics. Gaming has strong communities where people are enthusiastic about sharing progress and their experiences. Supporting human moderators to keep gaming safe includes monitoring avatars, usernames, images, and text-based materials. Azure AI Content Safety has advanced AI vision tools to help moderate gaming platforms to detect misconduct.

**Generative AI services**

Organizations are increasingly using generative AI services to enable internal data to be accessed more easily. To maintain the integrity and safety of internal data, both user prompts and AI-generated outputs need to be checked to prevent malicious use of these systems.

**News**

News websites need to moderate user comments to prevent the spread of misinformation. Azure AI Content Safety can identify language that includes hate speech and other harmful content.

**Other situations**

There are many other situations where content needs to be moderated. Azure AI Content Safety can be customized to identify problematic language for specific cases.

**Knowledge Check**

**1. Which feature of Azure AI Content Safety helps protect large language models from document injection attacks?**

Prompt Shields

**Correct. Prompt shields block jailbreak attacks from inputs to LLMs including user input prompts and document injection attacks.**

Groundedness detection

Custom categories

**2. What is the purpose of the Groundedness detection feature in Azure AI Content Safety?**

To verify AI-generated text is based on provided source materials.

**Correct. Groundedness detection protects against inaccurate responses in AI-generated text by LLMs.**

To detect harmful images.

To moderate multimodal content

**3. Which social media issues does Azure AI Content Safety address?**

The growth of inappropriate online content including bullying and hate speech.

**Correct. Azure AI Content Safety helps moderate inappropriate content including bullying and hate speech.**

The popularity of online gaming.

The need to train AI models with more accurate content.

**4. How does Azure AI Content Safety help businesses to protect their brand image?**

By moderating comments and messages from customers.

**Correct. Brands often encourage customers to leave comments and reviews, which need to be moderated to prevent inappropriate content.**

By detecting a brand's violent graphics.

To ensure students are consuming accurate content.

**5. What is a benefit of Azure AI Content Safety?**

Reducing the amount of psychologically damaging material that human moderators are exposed to.

**Correct. Human moderators can reduce the amount of harmful material they have to read by automatically assigning a severity score to content.**

Providing an easy-to-implement service for users to leave comments on e-commerce sites.

Allowing e-commerce companies to create custom categories for their products and services.

Bottom of Form