**Detect the language used in text**

Language detection is one of the features offered by [Azure AI Language](https://learn.microsoft.com/en-us/azure/ai-services/language-service/overview), a collection of machine learning and AI algorithms in the cloud for developing intelligent applications that involve written language. Language detection is able to detect more than 100 languages in their primary script. In addition, it offers [script detection](https://learn.microsoft.com/en-us/azure/ai-services/language-service/language-detection/how-to/call-api#script-name-and-script-code) to detect supported scripts for each detected language according to the [ISO 15924 standard](https://wikipedia.org/wiki/ISO_15924) for a [select number of languages](https://learn.microsoft.com/en-us/azure/ai-services/language-service/language-detection/language-support#script-detection) supported by Azure AI Language Service.

**Language detection features**

* Language detection: Returns one predominant language for each document you submit, along with its ISO 639-1 name, a human-readable name, confidence score, script name and script code according to ISO 15924 standard.
* Script detection: To distinguish between multiple scripts used to write certain languages, such as Kazakh, language detection returns a script name and script code according to the ISO 15924 standard.
* Ambiguous content handling: To help disambiguate language based on the input, you can specify an ISO 3166-1 alpha-2 country/region code. For example, the word "communication" is common to both English and French. Specifying the origin of the text as France can help the language detection model determine the correct language.

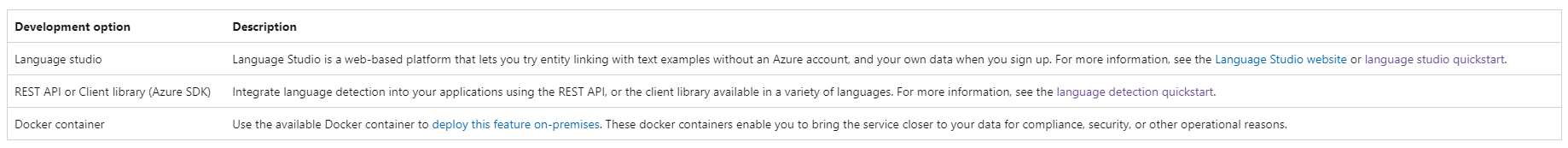
**Typical workflow**

To use this feature, you submit data for analysis and handle the API output in your application. Analysis is performed as-is, with no added customization to the model used on your data.

1. Create an Azure AI Language resource, which grants you access to the features offered by Azure AI Language. It generates a password (called a key) and an endpoint URL that you use to authenticate API requests.
2. Create a request using either the REST API or the client library for C#, Java, JavaScript, and Python. You can also send asynchronous calls with a batch request to combine API requests for multiple features into a single call.
3. Send the request containing your text data. Your key and endpoint are used for authentication.
4. Stream or store the response locally.

**Get started with language detection**

To use language detection, you submit raw unstructured text for analysis and handle the API output in your application. Analysis is performed as-is, with no additional customization to the model used on your data. There are three ways to use language detection:

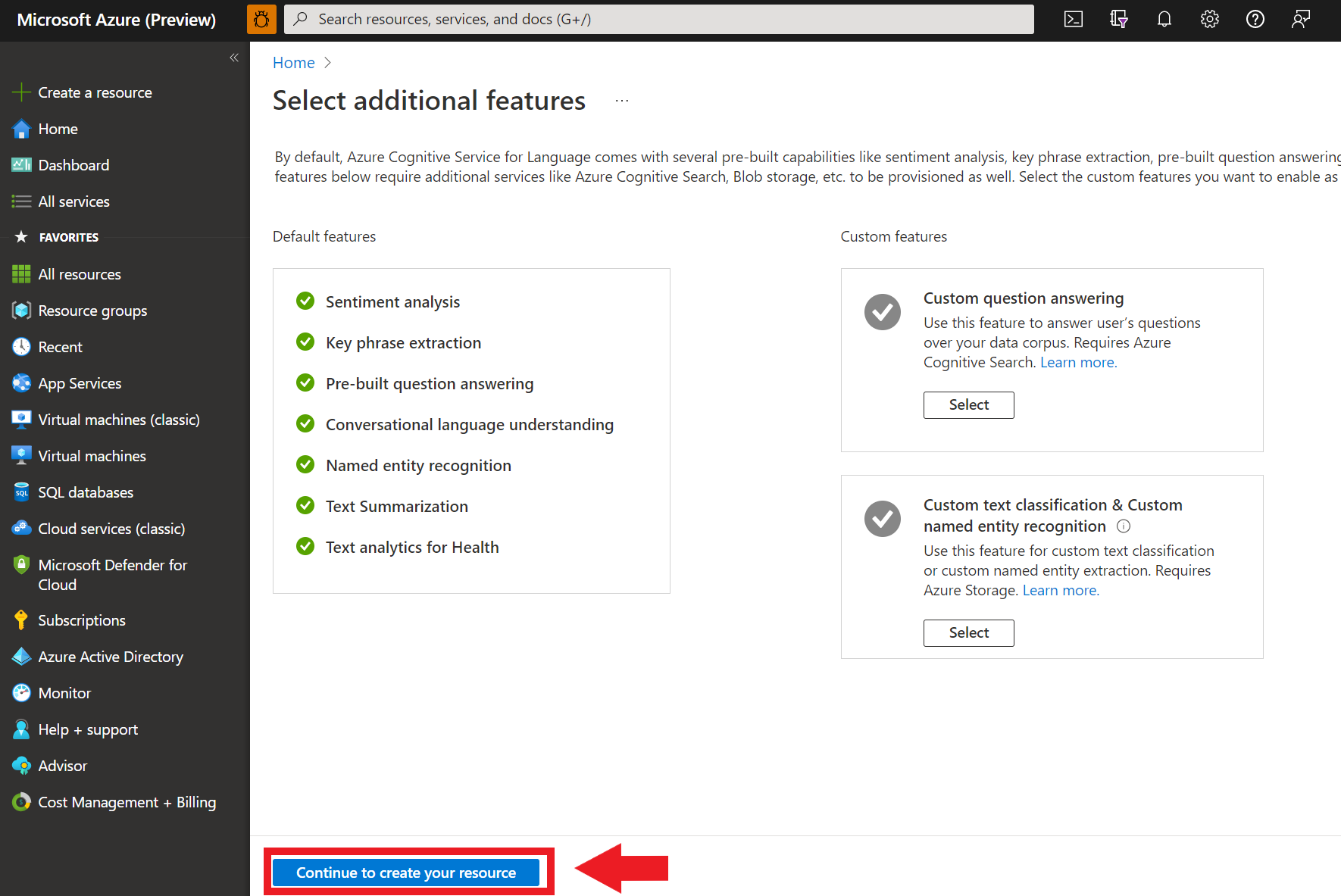


**using the client library and REST API**

**Create an Azure resource**

To use the code sample below, you'll need to deploy an Azure resource. This resource will contain a key and endpoint you'll use to authenticate the API calls you send to the Language service.

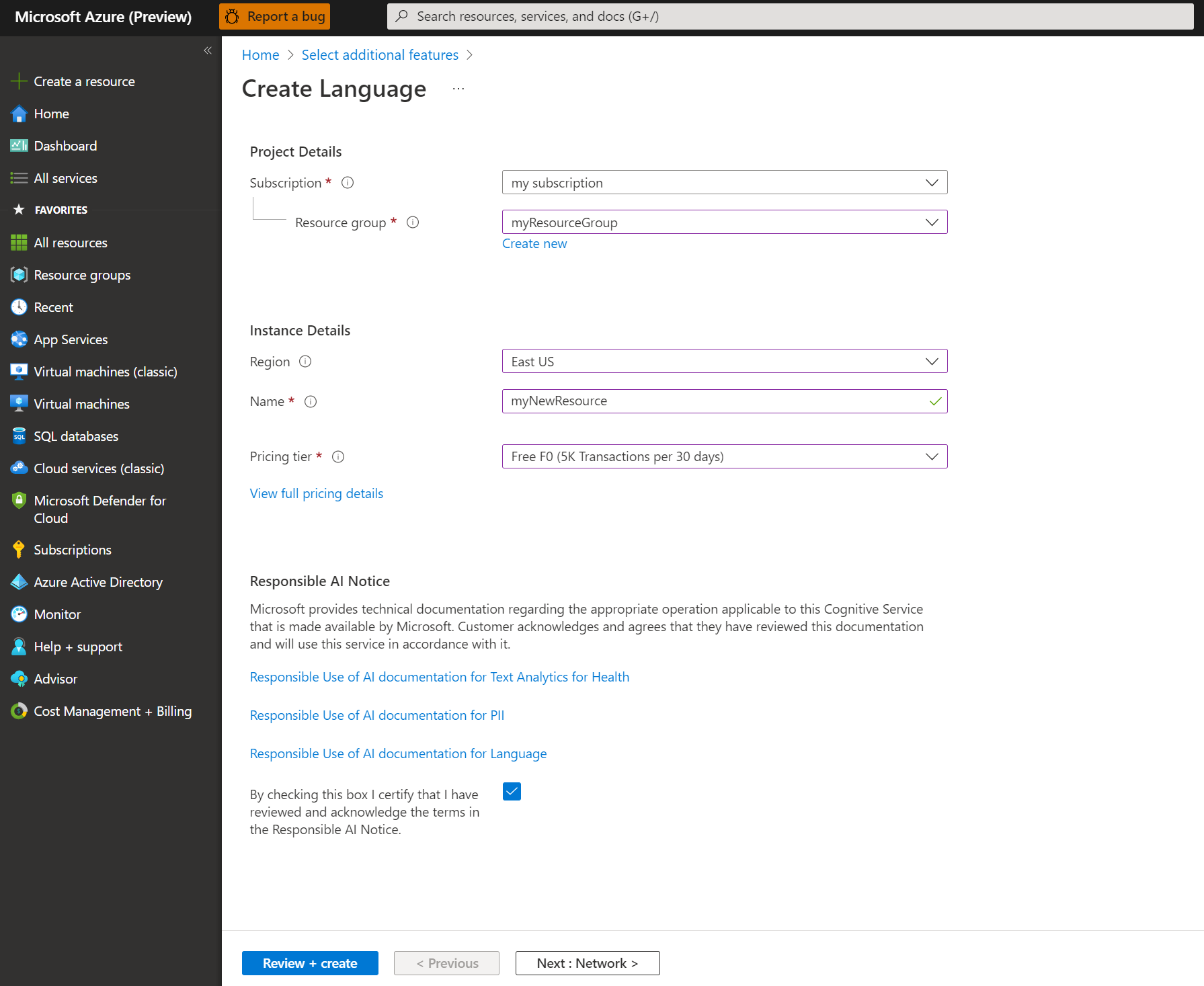
1. Use the following link to [create a language resource](https://portal.azure.com/#create/Microsoft.CognitiveServicesTextAnalytics) using the Azure portal. You will need to sign in using your Azure subscription.
2. On the **Select additional features** screen that appears, select **Continue to create your resource**.

[](https://learn.microsoft.com/en-us/azure/ai-services/language-service/media/portal-resource-additional-features.png#lightbox)

1. In the **Create language** screen, provide the following information:

A close up of a white card

Description automatically generated

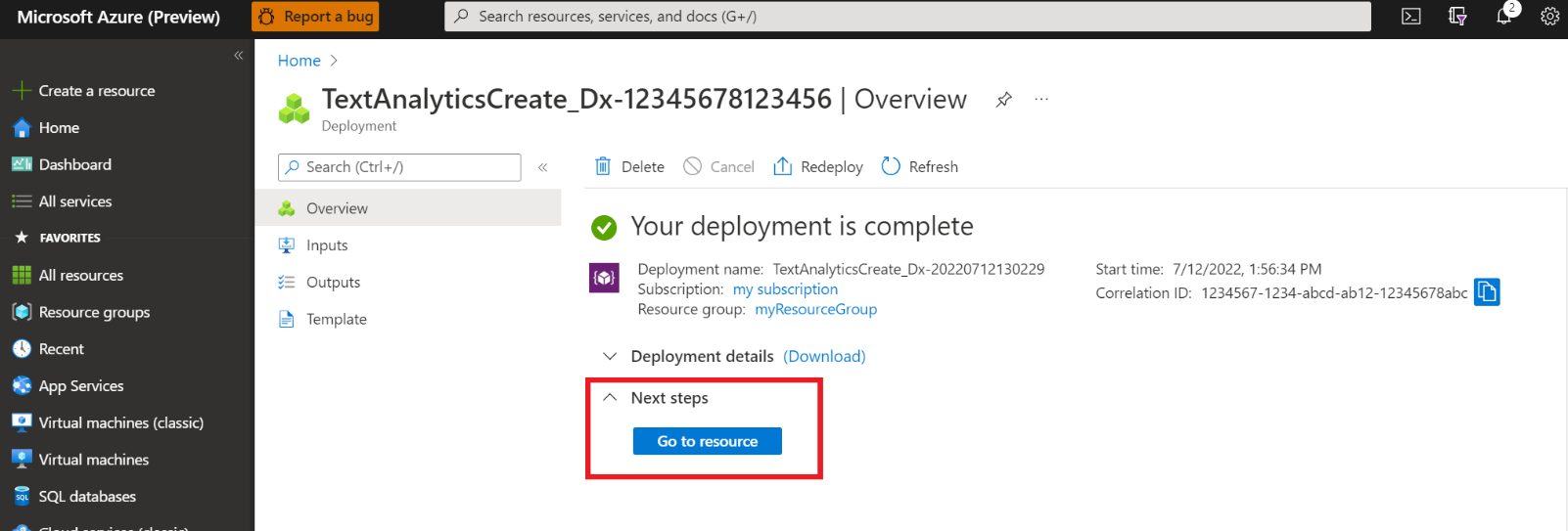
[](https://learn.microsoft.com/en-us/azure/ai-services/language-service/media/portal-resource-creation-details.png#lightbox)

1. Make sure the **Responsible AI Notice** checkbox is checked.
2. Select **Review + Create** at the bottom of the page.
3. In the screen that appears, make sure the validation has passed, and that you entered your information correctly. Then select **Create**.

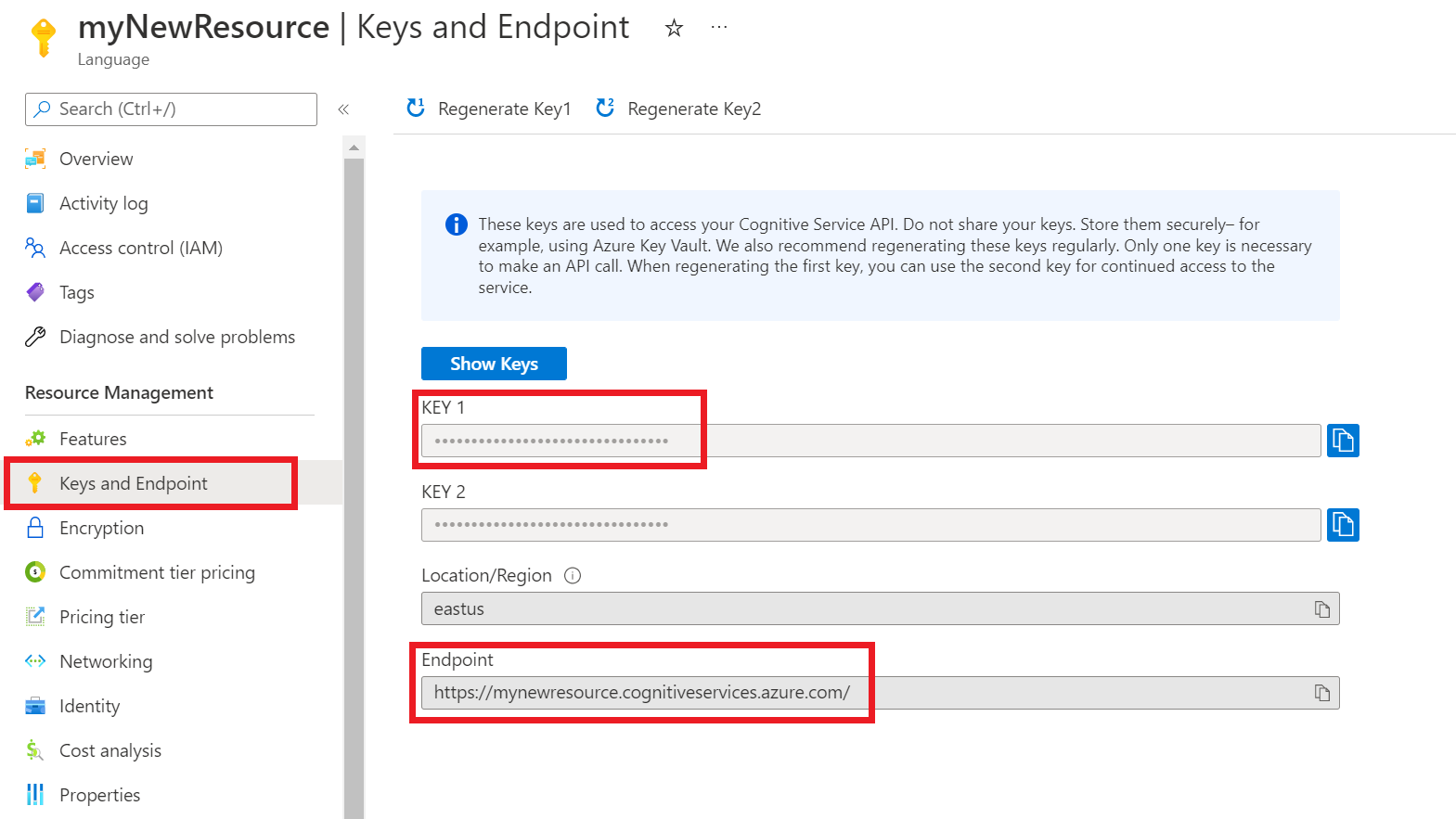
**Get your key and endpoint**

Next you will need the key and endpoint from the resource to connect your application to the API. You'll paste your key and endpoint into the code later in the quickstart.

1. After the Language resource deploys successfully, click the **Go to Resource** button under **Next Steps**.

[](https://learn.microsoft.com/en-us/azure/ai-services/language-service/media/portal-resource-next-steps.png#lightbox)

1. On the screen for your resource, select **Keys and endpoint** on the left navigation menu. You will use one of your keys and your endpoint in the steps below.

[](https://learn.microsoft.com/en-us/azure/ai-services/language-service/media/azure-portal-resource-credentials.png#lightbox)

**Create a new .NET Core application**

Using the Visual Studio IDE, create a new .NET Core console app. This creates a "Hello World" project with a single C# source file: *program.cs*.

Install the client library by right-clicking on the solution in the Solution Explorer and selecting Manage NuGet Packages. In the package manager that opens select Browse and search for Azure.AI.TextAnalytics. Select version 5.2.0, and then Install. You can also use the [Package Manager Console](https://learn.microsoft.com/en-us/nuget/consume-packages/install-use-packages-powershell#find-and-install-a-package).

**Code example**

**A screenshot of a computer program

Description automatically generated**

**Output**

**A white rectangular object with a black border

Description automatically generated**

**Use Language Studio with your own text**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated