

Contents

[Azure Cognitive Services Documentation](#)

[Overview](#)

[What are Cognitive Services?](#)

[What's new in Cognitive Services docs?](#)

[Language support](#)

[Pricing](#)

[Quickstarts](#)

[Create a new resource using the Azure portal](#)

[Create a new resource using the Azure CLI](#)

[Create a new resource using the client library](#)

[Create a new resource using an ARM template](#)

[How-To](#)

[Plan and manage costs](#)

[Diagnostic logging](#)

[Manage resources](#)

[Deploy](#)

[Containers](#)

[Cognitive Services containers](#)

[Container image tags and releases notes](#)

[Container FAQ](#)

[Container recipes](#)

[Container Reuse](#)

[Deploy to Azure Container Instance](#)

[Deploy to Azure Kubernetes](#)

[Deploy with Docker Compose](#)

[Other Container Services](#)

[Azure Container Instances](#)

[Azure Container Registry](#)

[Azure Kubernetes Service \(AKS\)](#)

[Cognitive Services containers documentation](#)

[Security](#)

[Cognitive Services security](#)

[Virtual networks for Cognitive Services](#)

[Configure customer-managed keys](#)

[Authenticate requests to Cognitive Services](#)

[Configure data loss prevention](#)

[Security baseline](#)

[Security controls by Azure Policy](#)

[Use with big data](#)

[Cognitive Services for Big Data](#)

[Getting started](#)

[Samples](#)

[Python](#)

[Scala](#)

[Recipes](#)

[Recipe: Anomaly Detector](#)

[Recipe: Art Explorer](#)

[Concepts](#)

[Cognitive Services development options](#)

[Cognitive Services and Machine learning](#)

[Custom subdomains for Cognitive Services](#)

[Reference](#)

[REST API](#)

[.NET](#)

[Node.js](#)

[Python](#)

[Java](#)

[Go](#)

[Azure CLI](#)

[Azure PowerShell](#)

[Azure Policy built-ins](#)

Resources

[Support & help options](#)

[Region support](#)

[Azure updates](#)

[Responsible use of AI](#)

[Pricing calculator](#)

[Compliance](#)

[Service Level Agreement \(SLA\)](#)

[Azure Government](#)

[Videos](#)

[Azure Blog](#)

[Azure AI Blog](#)

What are Azure Cognitive Services?

7/19/2021 • 5 minutes to read • [Edit Online](#)

Azure Cognitive Services are cloud-based services with REST APIs and client library SDKs available to help you build cognitive intelligence into your applications. You can add cognitive features to your applications without having artificial intelligence (AI) or data science skills. Azure Cognitive Services comprise various AI services that enable you to build cognitive solutions that can see, hear, speak, understand, and even make decisions.

Categories of Cognitive Services

The catalog of cognitive services that provide cognitive understanding is categorized into five main pillars:

- Vision
- Speech
- Language
- Decision
- Search

The following sections in this article provide a list of services that are part of these five pillars.

Vision APIs

SERVICE NAME	SERVICE DESCRIPTION
Computer Vision	The Computer Vision service provides you with access to advanced cognitive algorithms for processing images and returning information. See Computer Vision quickstart to get started with the service.
Custom Vision Service	The Custom Vision Service lets you build, deploy, and improve your own image classifiers. An image classifier is an AI service that applies labels to images, based on their visual characteristics.
Face	The Face service provides access to advanced face algorithms, enabling face attribute detection and recognition. See Face quickstart to get started with the service.

Speech APIs

SERVICE NAME	SERVICE DESCRIPTION
Speech service	Speech service adds speech-enabled features to applications. Speech service includes various capabilities like speech-to-text, text-to-speech, speech translation, and many more.

Language APIs

SERVICE NAME	SERVICE DESCRIPTION
Language Understanding LUIS	Language Understanding (LUIS) is a cloud-based conversational AI service that applies custom machine-learning intelligence to a user's conversational, natural language text to predict overall meaning, and pull out relevant, detailed information. See LUIS quickstart to get started with the service.
QnA Maker	QnA Maker allows you to build a question and answer service from your semi-structured content. See QnA Maker quickstart to get started with the service.
Text Analytics	Text Analytics provides natural language processing over raw text for sentiment analysis, key phrase extraction, and language detection. See Text Analytics quickstart to get started with the service.
Translator	Translator provides machine-based text translation in near real-time.

Decision APIs

SERVICE NAME	SERVICE DESCRIPTION
Anomaly Detector	Anomaly Detector allows you to monitor and detect abnormalities in your time series data. See Anomaly Detector quickstart to get started with the service.
Content Moderator	Content Moderator provides monitoring for possible offensive, undesirable, and risky content. See Content Moderator quickstart to get started with the service.
Personalizer	Personalizer allows you to choose the best experience to show to your users, learning from their real-time behavior. See Personalizer quickstart to get started with the service.

Search APIs

NOTE

Looking for [Azure Cognitive Search](#)? Although it uses Cognitive Services for some tasks, it's a different search technology that supports other scenarios.

SERVICE NAME	SERVICE DESCRIPTION
Bing News Search	Bing News Search returns a list of news articles determined to be relevant to the user's query.
Bing Video Search	Bing Video Search returns a list of videos determined to be relevant to the user's query.
Bing Web Search	Bing Web Search returns a list of search results determined to be relevant to the user's query.

SERVICE NAME	SERVICE DESCRIPTION
Bing Autosuggest	Bing Autosuggest allows you to send a partial search query term to Bing and get back a list of suggested queries.
Bing Custom Search	Bing Custom Search allows you to create tailored search experiences for topics that you care about.
Bing Entity Search	Bing Entity Search returns information about entities that Bing determines are relevant to a user's query.
Bing Image Search	Bing Image Search returns a display of images determined to be relevant to the user's query.
Bing Visual Search	Bing Visual Search returns insights about an image such as visually similar images, shopping sources for products found in the image, and related searches.
Bing Local Business Search	Bing Local Business Search API enables your applications to find contact and location information about local businesses based on search queries.
Bing Spell Check	Bing Spell Check allows you to perform contextual grammar and spell checking.

Get started with Cognitive Services

Start by creating a Cognitive Services resource with hands-on quickstarts using the following methods:

- [Azure portal](#)
- [Azure CLI](#)
- [Azure SDK client libraries](#)
- [Azure Resource Manager \(ARM\) templates](#)

Using Cognitive Services in different development environments

With Azure and Cognitive Services, you have access to several development options, such as:

- Automation and integration tools like Logic Apps and Power Automate.
- Deployment options such as Azure Functions and the App Service.
- Cognitive Services Docker containers for secure access.
- Tools like Apache Spark, Azure Databricks, Azure Synapse Analytics, and Azure Kubernetes Service for Big Data scenarios.

To learn more, see [Cognitive Services development options](#).

Using Cognitive Services securely

Azure Cognitive Services provides a layered security model, including [authentication](#) via Azure Active Directory credentials, a valid resource key, and [Azure Virtual Networks](#).

Containers for Cognitive Services

Azure Cognitive Services provides several Docker containers that let you use the same APIs that are available in Azure, on-premises. Using these containers gives you the flexibility to bring Cognitive Services closer to your

data for compliance, security or other operational reasons. Learn more about [Cognitive Services Containers](#).

Regional availability

The APIs in Cognitive Services are hosted on a growing network of Microsoft-managed data centers. You can find the regional availability for each API in [Azure region list](#).

Looking for a region we don't support yet? Let us know by filing a feature request on our [UserVoice forum](#).

Supported cultural languages

Cognitive Services supports a wide range of cultural languages at the service level. You can find the language availability for each API in the [supported languages list](#).

Certifications and compliance

Cognitive Services has been awarded certifications such as CSA STAR Certification, FedRAMP Moderate, and HIPAA BAA. You can [download](#) certifications for your own audits and security reviews.

To understand privacy and data management, go to the [Trust Center](#).

Support

Cognitive Services provides several support options to help you move forward with creating intelligent applications. Cognitive Services also has a strong community of developers that can help answer your specific questions. For a full list of options available to you, see [Cognitive Services support and help options](#).

Next steps

- [Create a Cognitive Services account](#)
- [What's new in Cognitive Services docs](#)
- [Plan and manage costs for Cognitive Services](#)

Cognitive Services docs: What's new for May 1, 2021 - May 31, 2021

9/10/2021 • 2 minutes to read • [Edit Online](#)

Welcome to what's new in the Cognitive Services docs from May 1, 2021 through May 31, 2021. This article lists some of the major changes to docs during this period.

Containers

New articles

- [Install and run Translator containers](#)
- [Configure Translator Docker containers](#)
- [Container: Translator translate method](#)

Updated articles

- [Azure Cognitive Services container image tags and release notes](#)

Form Recognizer

New articles

- [Reference: Azure Form Recognizer client library v3.0.0 and REST API v2.0](#)

Updated articles

- [Form Recognizer prebuilt business cards model](#)
- [Quickstart: Get started with the client library SDKs or REST API](#)
- [What's new in Form Recognizer](#)
- [Form Recognizer landing page](#)

Translator

New articles

- [Frequently asked questions—Translator API](#)

Updated articles

Document Translation is now GA

- [Get started with Document Translation](#)
- All nine reference pages have been updated to remove the preview parameter from the REST API endpoint.

Personalizer

Updated articles

- [What's new in Personalizer](#)

Text Analytics

Updated articles

- [Tutorial: Integrate Power BI with the Text Analytics Cognitive Service](#)

- [Extract information in Excel using Text Analytics and Power Automate](#)
- [How to call the Text Analytics REST API](#)
- [How to use Named Entity Recognition in Text Analytics](#)
- [What's new in the Text Analytics API?](#)

Community contributors

The following people contributed to the Cognitive Services docs during this period. Thank you! Learn how to contribute by following the links under "Get involved" in the [what's new landing page](#).

- [enzocanoo](#) - Enzo Cano (2)
- [hyoshioka0128](#) - Hiroshi Yoshioka (2)
- [sassdawe](#) - DavidSass (1)
- [SzymonSel](#) - Szymon Seliga (1)
- [thomash0815](#) (1)

Service-specific updates

- [Computer Vision](#)
- [Custom Vision](#)
- [Language Understanding \(LUIS\)](#)
- [Personalizer](#)
- [QnA Maker](#)
- [Text Analytics](#)

Natural language support for Azure Cognitive Services

9/3/2021 • 2 minutes to read • [Edit Online](#)

Azure Cognitive Services enable you to build applications that see, hear, speak with, and understand your users. Between these services, more than three dozen languages are supported, allowing users to communicate with your application in natural ways. Use the links below to view language availability by service.

These Cognitive Services are language agnostic and don't have limitations based on human language.

- [Anomaly Detector \(Preview\)](#)
- [Custom Vision](#)
- [Face](#)
- [Personalizer](#)

Vision

- [Computer Vision](#)
- [Ink Recognizer \(Preview\)](#)
- [Video Indexer](#)

Language

- [Language Understanding \(LUIS\)](#)
- [QnA Maker](#)
- [Text Analytics](#)
- [Translator](#)

Speech

- [Speech Service: Speech-to-Text](#)
- [Speech Service:Text-to-Speech](#)
- [Speech Service: Speech Translation](#)

Decision

- [Content Moderator](#)

See also

- [What are the Cognitive Services?](#)
- [Create an account](#)

Quickstart: Create a Cognitive Services resource using the Azure portal

9/3/2021 • 5 minutes to read • [Edit Online](#)

Use this quickstart to start using Azure Cognitive Services. After creating a Cognitive Service resource in the Azure portal, you'll get an endpoint and a key for authenticating your applications.

Azure Cognitive Services are cloud-based services with REST APIs, and client library SDKs available to help developers build cognitive intelligence into applications without having direct artificial intelligence (AI) or data science skills or knowledge. Azure Cognitive Services enables developers to easily add cognitive features into their applications with cognitive solutions that can see, hear, speak, understand, and even begin to reason.

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

Prerequisites

- A valid Azure subscription - [Create one for free](#).
- Your Azure account must have a Cognitive Services Contributor role assigned in order for you to agree to the responsible AI terms and create a resource. Contact your administrator to get this role assigned to your account.

Create a new Azure Cognitive Services resource

1. Create a resource.

- [Multi-service resource](#)
- [Single-service resource](#)

The multi-service resource is named **Cognitive Services** in the portal. [Create a Cognitive Services resource](#).

At this time, the multi-service resource enables access to the following Cognitive Services:

- **Vision** - Computer Vision, Custom Vision, Face
- **Speech** - Speech
- **Language** - Language Understanding (LUIS), Text Analytics, Translator
- **Decision** - Content Moderator

2. On the **Create** page, provide the following information:

- [Multi-service resource](#)
- [Single-service resource](#)

PROJECT DETAILS	DESCRIPTION
Subscription	Select one of your available Azure subscriptions.
Resource group	The Azure resource group that will contain your Cognitive Services resource. You can create a new group or add it to a pre-existing group.
Region	The location of your cognitive service instance. Different locations may introduce latency, but have no impact on the runtime availability of your resource.
Name	A descriptive name for your cognitive services resource. For example, <i>MyCognitiveServicesResource</i> .
Pricing tier	The cost of your Cognitive Services account depends on the options you choose and your usage. For more information, see the API pricing details .

Create Cognitive Services



Basics Tags Review + create

Get access to Vision, Language, Search, and Speech Cognitive Services with a single API key. Quickly connect services together to achieve more insights into your content and easily integrate with other services like Azure Search. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource group * ⓘ

[Create new](#)

Instance details

Region * ⓘ



Location specifies the region only for included regional services. This does not specify a region for included non-regional services. [Click here for more details.](#)

Name * ⓘ

Pricing tier * ⓘ

[View full pricing details](#)

By checking this box, I certify that use of this service is not by or for a police department in the United States. ☐

I confirm I have read and understood the notice below. ☐

Review + create

< Previous

Next : Tags >

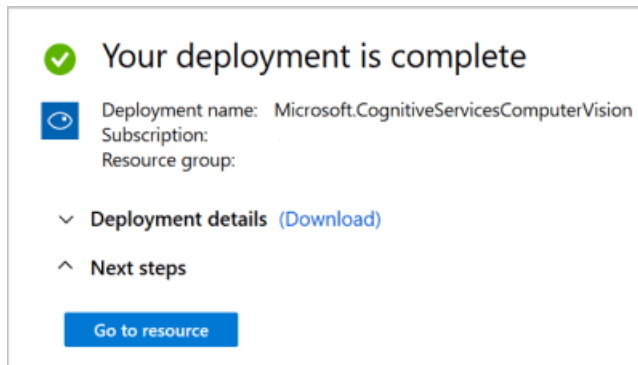
Read and accept the conditions (as applicable for you) and then select **Review + create**.

TIP

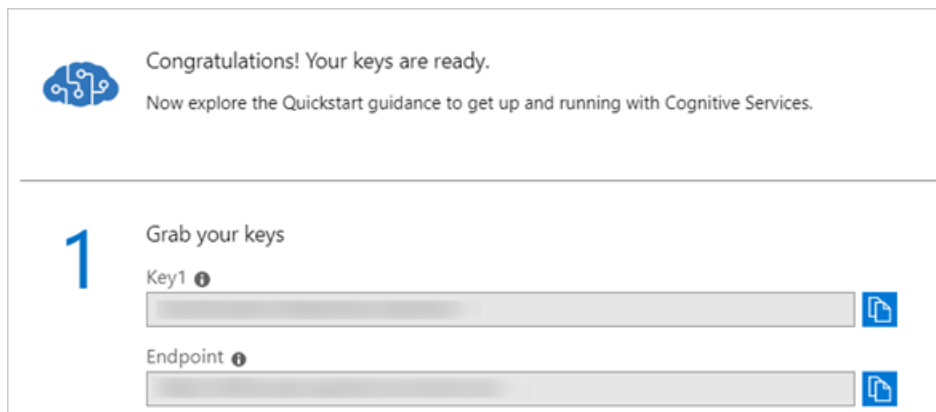
If your subscription doesn't allow you to create a Cognitive Service resource, you may need to enable that ability of the [Azure resource provider](#) with the [Azure portal](#), [PowerShell command](#) or an [Azure CLI command](#). If you are not the subscription owner, ask the *Subscription Owner* or someone with a role of *admin* to complete the registration for you or ask for the `/register/action` privileges granted to your account.

Get the keys for your resource

1. After your resource is successfully deployed, click on **Go to resource** under **Next Steps**.



2. From the quickstart pane that opens, you can access your key and endpoint.



Configure an environment variable for authentication

Applications need to authenticate access to the Cognitive Services they use. To authenticate, we recommend creating an environment variable to store the keys for your Azure Resources.

After you have your key, write it to a new environment variable on the local machine running the application. To set the environment variable, open a console window, and follow the instructions for your operating system.

Replace `your-key` with one of the keys for your resource.

- [Windows](#)
- [Linux](#)
- [macOS](#)

```
setx COGNITIVE_SERVICE_KEY "your-key"
```

After you add the environment variable, you may need to restart any running programs that will need to read the environment variable, including the console window. For example, if you are using Visual Studio as your editor, restart Visual Studio before running the example.

Clean up resources

If you want to clean up and remove a Cognitive Services subscription, you can delete the resource or resource group. Deleting the resource group also deletes any other resources contained in the group.

1. In the Azure portal, expand the menu on the left side to open the menu of services, and choose **Resource Groups** to display the list of your resource groups.
2. Locate the resource group containing the resource to be deleted
3. Right-click on the resource group listing. Select **Delete resource group**, and confirm.

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.
- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.

Quickstart: Create a Cognitive Services resource using the Azure Command-Line Interface(CLI)

8/18/2021 • 7 minutes to read • [Edit Online](#)

Use this quickstart to get started with Azure Cognitive Services using the [Azure Command Line Interface\(CLI\)](#).

Azure Cognitive Services are cloud-base services with REST APIs, and client library SDKs available to help developers build cognitive intelligence into applications without having direct artificial intelligence (AI) or data science skills or knowledge. Azure Cognitive Services enables developers to easily add cognitive features into their applications with cognitive solutions that can see, hear, speak, understand, and even begin to reason.

Cognitive Services are represented by Azure [resources](#) that you create in your Azure subscription. After creating the resource, Use the keys and endpoint generated for you to authenticate your applications.

In this quickstart, you'll learn how to sign up for Azure Cognitive Services and create an account that has a single-service or multi-service subscription, Using the [Azure Command Line Interface\(CLI\)](#). These services are represented by Azure [resources](#), which enable you to connect to one or more of the Azure Cognitive Services APIs.

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

Prerequisites

- A valid Azure subscription - [Create one](#) for free.
- The [Azure Command Line Interface\(CLI\)](#)
- Your Azure account must have a Cognitive Services Contributor role assigned in order for you to agree to the responsible AI terms and create a resource. Contact your administrator to get this role assigned to your account.
- You must create your first Face, Text Analytics, or Computer Vision resources from the Azure portal to review and acknowledge the terms and conditions. You can do so here: [Face](#), [Text Analytics](#), [Computer Vision](#). After that, you can create subsequent resources using any deployment tool (SDK, CLI, or ARM template, etc) under the same Azure subscription.

Install the Azure CLI and sign in

Install the [Azure CLI](#). To sign into your local installation of the CLI, run the [az login](#) command:

```
az login
```

You can also use the green **Try It** button to run these commands in your browser.

Create a new Azure Cognitive Services resource group

Before creating a Cognitive Services resource, you must have an Azure resource group to contain the resource. When you create a new resource, you have the option to either create a new resource group, or use an existing one. This article shows how to create a new resource group.

Choose your resource group location

To create a resource, you'll need one of the Azure locations available for your subscription. You can retrieve a list of available locations with the [az account list-locations](#) command. Most Cognitive Services can be accessed from several locations. Choose the one closest to you, or see which locations are available for the service.

IMPORTANT

- Remember your Azure location, as you will need it when calling the Azure Cognitive Services.
- The availability of some Cognitive Services can vary by region. For more information, see [Azure products by region](#).

```
az account list-locations \  
  --query "[].{Region:name}" \  
  --out table
```

After you have your Azure location, create a new resource group in the Azure CLI using the [az group create](#) command.

In the example below, replace the Azure location `westus2` with one of the Azure locations available for your subscription.

```
az group create \  
  --name cognitive-services-resource-group \  
  --location westus2
```

Create a Cognitive Services resource

Choose a cognitive service and pricing tier

When creating a new resource, you will need to know the "kind" of service you want to use, along with the [pricing tier](#) (or sku) you want. You will use this and other information as parameters when creating the resource.

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

See the list of SKUs and pricing information below.

Multi-service

SERVICE	KIND
Multiple services. For more information, see the pricing page.	<code>CognitiveServices</code>

Vision

SERVICE	KIND
Computer Vision	<code>ComputerVision</code>
Custom Vision - Prediction	<code>CustomVision.Prediction</code>
Custom Vision - Training	<code>CustomVision.Training</code>
Face	<code>Face</code>
Form Recognizer	<code>FormRecognizer</code>

Speech

SERVICE	KIND
Speech Services	<code>SpeechServices</code>
Speech Recognition	<code>SpeakerRecognition</code>

Language

SERVICE	KIND
LUIS	<code>LUIS</code>
QnA Maker	<code>QnAMaker</code>
Text Analytics	<code>TextAnalytics</code>
Text Translation	<code>TextTranslation</code>

Decision

SERVICE	KIND
Anomaly Detector	<code>AnomalyDetector</code>
Content Moderator	<code>ContentModerator</code>
Personalizer	<code>Personalizer</code>

Pricing tiers and billing

Pricing tiers (and the amount you get billed) are based on the number of transactions you send using your authentication information. Each pricing tier specifies the:

- maximum number of allowed transactions per second (TPS).

- service features enabled within the pricing tier.
- cost for a predefined number of transactions. Going above this number will cause an extra charge as specified in the [pricing details](#) for your service.

NOTE

Many of the Cognitive Services have a free tier you can use to try the service. To use the free tier, use `F0` as the SKU for your resource.

You can find a list of available Cognitive Service "kinds" with the [az cognitiveservices account list-kinds](#) command:

```
az cognitiveservices account list-kinds
```

Add a new resource to your resource group

To create and subscribe to a new Cognitive Services resource, use the [az cognitiveservices account create](#) command. This command adds a new billable resource to the resource group created earlier. When creating your new resource, you will need to know the "kind" of service you want to use, along with its pricing tier (or sku) and an Azure location:

You can create an F0 (free) resource for Anomaly Detector, named `anomaly-detector-resource` with the command below.

```
az cognitiveservices account create \  
  --name anomaly-detector-resource \  
  --resource-group cognitive-services-resource-group \  
  --kind AnomalyDetector \  
  --sku F0 \  
  --location westus2 \  
  --yes
```

TIP

If your subscription doesn't allow you to create a Cognitive Service resource, you may need to enable that ability of the [Azure resource provider](#) with the [Azure portal](#), [PowerShell command](#) or an [Azure CLI command](#). If you are not the subscription owner, ask the *Subscription Owner* or someone with a role of *admin* to complete the registration for you or ask for the `/register/action` privileges granted to your account.

Get the keys for your resource

To log into your local installation of the Command-Line Interface(CLI), use the [az login](#) command.

```
az login
```

Use the [az cognitiveservices account keys list](#) command to get the keys for your Cognitive Service resource.

```
az cognitiveservices account keys list \  
  --name anomaly-detector-resource \  
  --resource-group cognitive-services-resource-group
```

Configure an environment variable for authentication

Applications need to authenticate access to the Cognitive Services they use. To authenticate, we recommend creating an environment variable to store the keys for your Azure Resources.

After you have your key, write it to a new environment variable on the local machine running the application. To set the environment variable, open a console window, and follow the instructions for your operating system. Replace `your-key` with one of the keys for your resource.

- [Windows](#)
- [Linux](#)
- [macOS](#)

```
setx COGNITIVE_SERVICE_KEY "your-key"
```

After you add the environment variable, you may need to restart any running programs that will need to read the environment variable, including the console window. For example, if you are using Visual Studio as your editor, restart Visual Studio before running the example.

Pricing tiers and billing

Pricing tiers (and the amount you get billed) are based on the number of transactions you send using your authentication information. Each pricing tier specifies the:

- maximum number of allowed transactions per second (TPS).
- service features enabled within the pricing tier.
- The cost for a predefined number of transactions. Going above this amount will cause an extra charge as specified in the [pricing details](#) for your service.

Get current quota usage for your resource

Use the [az cognitiveservices account list-usage](#) command to get the usage for your Cognitive Service resource.

```
az cognitiveservices account list-usage \  
  --name anomaly-detector-resource \  
  --resource-group cognitive-services-resource-group \  
  --subscription subscription-name
```

Clean up resources

If you want to clean up and remove a Cognitive Services resource, you can delete it or the resource group. Deleting the resource group also deletes any other resources contained in the group.

To remove the resource group and its associated resources, use the `az group delete` command.

```
az group delete --name cognitive-services-resource-group
```

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.

- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.

Quickstart: Create a Cognitive Services resource using the Azure Management client library

6/8/2021 • 29 minutes to read • [Edit Online](#)

Use this quickstart to create and manage Azure Cognitive Services resources using the Azure Management client library.

Azure Cognitive Services is a family of cloud-base services with REST APIs and client libraries available to help developers build cognitive intelligence into their applications. Developers do not need direct artificial intelligence (AI) or data science skills or knowledge to achieve success. Azure Cognitive Services enables developers to easily add cognitive features into their applications with cognitive solutions that can see, hear, speak, understand, and even begin to reason.

Individual AI services are represented by Azure [resources](#) that you create under your Azure subscription. After you create a resource, you can use the keys and endpoint generated to authenticate your applications.

[Reference documentation](#) | [Library source code](#) | [Package \(NuGet\)](#) | [Samples](#)

C# prerequisites

- A valid Azure subscription - [Create one for free](#).
- The current version of [.NET Core](#).
- Your Azure account must have a Cognitive Services Contributor role assigned in order for you to agree to the responsible AI terms and create a resource. Contact your administrator to get this role assigned to your account.
- You must create your first Face, Text Analytics, or Computer Vision resources from the Azure portal to review and acknowledge the terms and conditions. You can do so here: [Face](#), [Text Analytics](#), [Computer Vision](#). After that, you can create subsequent resources using any deployment tool (SDK, CLI, or ARM template, etc) under the same Azure subscription.

Create an Azure Service Principal

To have your application interact with your Azure account, you need an Azure service principal to manage permissions. Follow the instructions in [Create an Azure service principal](#).

When you create a service principal, you'll see it has a secret value, an ID, and an application ID. Save the application ID and secret to a temporary location for later steps.

Create a resource group

Before you create a Cognitive Services resource, your account must have an Azure resource group to contain the resource. If you don't already have a resource group, create one in the [Azure portal](#) before continuing.

Create a new C# application

Create a new .NET Core application. In a console window (such as cmd, PowerShell, or Bash), use the `dotnet new` command to create a new console app with the name `azure-management-quickstart`. This command creates a simple "Hello World" C# project with a single source file: `program.cs`.

```
dotnet new console -n azure-management-quickstart
```

Change your directory to the newly created app folder. You can build the application with:

```
dotnet build
```

The build output should contain no warnings or errors.

```
...  
Build succeeded.  
  0 Warning(s)  
  0 Error(s)  
...
```

Install the client library

Within the application directory, install the Azure Management client library for .NET with the following command:

```
dotnet add package Microsoft.Azure.Management.CognitiveServices  
dotnet add package Microsoft.Azure.Management.Fluent  
dotnet add package Microsoft.Azure.Management.ResourceManager.Fluent
```

If you're using the Visual Studio IDE, the client library is available as a downloadable NuGet package.

Import libraries

Open *program.cs* and add the following `using` statements to the top of the file:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using Microsoft.Azure.Management.Fluent;  
using Microsoft.Azure.Management.ResourceManager.Fluent;  
using Microsoft.Azure.Management.ResourceManager.Fluent.Authentication;  
using Microsoft.Azure.Management.CognitiveServices;  
using Microsoft.Azure.Management.CognitiveServices.Models;
```

Authenticate the client

Add the following fields to the root of *program.cs* and populate their values, using the service principal you created and your Azure account information.

```

const string service_principal_application_id = "PASTE_YOUR_SERVICE_PRINCIPAL_APPLICATION_ID_HERE";
const string service_principal_secret = "PASTE_YOUR_SERVICE_PRINCIPAL_SECRET_HERE";

/* The ID of your Azure subscription. You can find this in the Azure Dashboard under Home > Subscriptions.
*/
const string subscription_id = "PASTE_YOUR_SUBSCRIPTION_ID_HERE";

/* The Active Directory tenant ID. You can find this in the Azure Dashboard under Home > Azure Active
Directory. */
const string tenant_id = "PASTE_YOUR_TENANT_ID_HERE";

/* The name of the Azure resource group in which you want to create the resource.
You can find resource groups in the Azure Dashboard under Home > Resource groups. */
const string resource_group_name = "PASTE_YOUR_RESOURCE_GROUP_NAME_HERE";

/* The name of the custom subdomain to use when you create the resource. This is optional.
For example, if you create a Bing Search v7 resource with the custom subdomain name 'my-search-resource',
your resource would have the endpoint https://my-search-resource.cognitiveservices.azure.com/.
Note not all Cognitive Services allow custom subdomain names. */
const string subdomain_name = "PASTE_YOUR_SUBDOMAIN_NAME_HERE";

```

Then, in your **Main** method, use these values to construct a **CognitiveServicesManagementClient** object. This object is needed for all of your Azure management operations.

```

var service_principal_credentials = new ServicePrincipalLoginInformation ();
service_principal_credentials.ClientId = service_principal_application_id;
service_principal_credentials.ClientSecret = service_principal_secret;

var credentials = SdkContext.AzureCredentialsFactory.FromServicePrincipal(service_principal_application_id,
service_principal_secret, tenant_id, AzureEnvironment.AzureGlobalCloud);
var client = new CognitiveServicesManagementClient(credentials);
client.SubscriptionId = subscription_id;

```

Call management methods

Add the following code to your **Main** method to list available resources, create a sample resource, list your owned resources, and then delete the sample resource. You'll define these methods in the next steps.

```

// Uncomment to list all available resource kinds, SKUs, and locations for your Azure account:
//list_available_kinds_skus_locations(client);

// Create a resource with kind TextTranslation, F0 (free tier), location global.
create_resource(client, "test_resource", "TextTranslation", "F0", "Global");

// List all resources for your Azure account and resource group:
list_resources(client);

// Delete the resource.
delete_resource(client, "test_resource");

Console.WriteLine("Press any key to exit.");
Console.ReadKey();

```

Create a Cognitive Services resource (C#)

To create and subscribe to a new Cognitive Services resource, use the **Create** method. This method adds a new billable resource to the resource group you pass in. When creating your new resource, you'll need to know the "kind" of service you want to use, along with its pricing tier (or SKU) and an Azure location. The following method takes all of these as arguments and creates a resource.

```
static void create_resource(CognitiveServicesManagementClient client, string resource_name, string kind,
string account_tier, string location)
{
    Console.WriteLine("Creating resource: " + resource_name + "...");
    /* NOTE If you do not want to use a custom subdomain name, remove the customSubDomainName
    property from CognitiveServicesAccountProperties. */
    CognitiveServicesAccount parameters =
        new CognitiveServicesAccount(null, null, kind, location, resource_name, new
CognitiveServicesAccountProperties(customSubDomainName : subdomain_name), new Sku(account_tier));
    var result = client.Accounts.Create(resource_group_name, resource_name, parameters);
    Console.WriteLine("Resource created.");
    Console.WriteLine("ID: " + result.Id);
    Console.WriteLine("Kind: " + result.Kind);
    Console.WriteLine();
}
```

Choose a service and pricing tier

When you create a new resource, you'll need to know the "kind" of service you want to use, along with the [pricing tier](#) (or SKU) you want. You'll use this and other information as parameters when creating the resource. You can find a list of available Cognitive Service "kinds" by calling the following method in your script:

```
static void list_available_kinds_skus_locations(CognitiveServicesManagementClient client)
{
    Console.WriteLine("Available SKUs:");
    var result = client.ResourceSkus.List();
    Console.WriteLine("Kind\tSKU Name\tSKU Tier\tLocations");
    foreach (var x in result) {
        var locations = "";
        foreach (var region in x.Locations)
        {
            locations += region;
        }
        Console.WriteLine(x.Kind + "\t" + x.Name + "\t" + x.Tier + "\t" + locations);
    };
}
```

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

See the list of SKUs and pricing information below.

Multi-service

SERVICE	KIND
Multiple services. For more information, see the pricing page.	CognitiveServices

Vision

SERVICE	KIND
Computer Vision	ComputerVision
Custom Vision - Prediction	CustomVision.Prediction
Custom Vision - Training	CustomVision.Training
Face	Face
Form Recognizer	FormRecognizer

Speech

SERVICE	KIND
Speech Services	SpeechServices
Speech Recognition	SpeakerRecognition

Language

SERVICE	KIND
LUIS	LUIS
QnA Maker	QnAMaker
Text Analytics	TextAnalytics
Text Translation	TextTranslation

Decision

SERVICE	KIND
Anomaly Detector	AnomalyDetector
Content Moderator	ContentModerator
Personalizer	Personalizer

Pricing tiers and billing

Pricing tiers (and the amount you get billed) are based on the number of transactions you send using your authentication information. Each pricing tier specifies the:

- maximum number of allowed transactions per second (TPS).
- service features enabled within the pricing tier.
- cost for a predefined number of transactions. Going above this number will cause an extra charge as specified in the [pricing details](#) for your service.

NOTE

Many of the Cognitive Services have a free tier you can use to try the service. To use the free tier, use `F0` as the SKU for your resource.

View your resources

To view all of the resources under your Azure account (across all resource groups), use the following method:

```
static void list_resources(CognitiveServicesManagementClient client)
{
    Console.WriteLine("Resources in resource group: " + resource_group_name);
    var result = client.Accounts.ListByResourceGroup(resource_group_name);
    foreach (var x in result)
    {
        Console.WriteLine("ID: " + x.Id);
        Console.WriteLine("Name: " + x.Name);
        Console.WriteLine("Type: " + x.Type);
        Console.WriteLine("Kind: " + x.Kind);
        Console.WriteLine();
    }
}
```

Delete a resource

The following method deletes the specified resource from the given resource group.

```
static void delete_resource(CognitiveServicesManagementClient client, string resource_name)
{
    Console.WriteLine("Deleting resource: " + resource_name + "...");
    client.Accounts.Delete (resource_group_name, resource_name);

    Console.WriteLine("Resource deleted.");
    Console.WriteLine();
}
```

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

Run the application

Run the application from your application directory with the `dotnet run` command.

```
dotnet run
```

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.
- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.
- See [Cognitive Services Management SDK reference documentation](#) for more details on the management SDK.

Java prerequisites

- A valid Azure subscription - [Create one for free](#).
- The current version of the [Java Development Kit\(JDK\)](#)
- The [Gradle build tool](#), or another dependency manager.
- Your Azure account must have a Cognitive Services Contributor role assigned in order for you to agree to the responsible AI terms and create a resource. Contact your administrator to get this role assigned to your account.
- You must create your first Face, Text Analytics, or Computer Vision resources from the Azure portal to review and acknowledge the terms and conditions. You can do so here: [Face](#), [Text Analytics](#), [Computer Vision](#). After that, you can create subsequent resources using any deployment tool (SDK, CLI, or ARM template, etc) under the same Azure subscription.

Create an Azure Service Principal

To have your application interact with your Azure account, you need an Azure service principal to manage permissions. Follow the instructions in [Create an Azure service principal](#).

When you create a service principal, you'll see it has a secret value, an ID, and an application ID. Save the application ID and secret to a temporary location for later steps.

Create a resource group

Before you create a Cognitive Services resource, your account must have an Azure resource group to contain the resource. If you don't already have a resource group, create one in the [Azure portal](#) before continuing.

Create a new Java application

In a console window (such as cmd, PowerShell, or Bash), create a new directory for your app, and navigate to it.

```
mkdir myapp && cd myapp
```

Run the `gradle init` command from your working directory. This command will create essential build files for Gradle, including *build.gradle.kts* which is used at runtime to create and configure your application.

```
gradle init --type basic
```

When prompted to choose a DSL, select **Kotlin**.

From your working directory, run the following command:

```
mkdir -p src/main/java
```

Install the client library

This quickstart uses the Gradle dependency manager. You can find the client library and information for other dependency managers on the [Maven Central Repository](#).

In your project's *build.gradle.kts* file, include the client library as an `implementation` statement, along with the required plugins and settings.

```

plugins {
    java
    application
}
application {
    mainClass.set("FormRecognizer")
}
repositories {
    mavenCentral()
}
dependencies {
    implementation(group = "com.microsoft.azure", name = "azure-mgmt-cognitiveservices", version = "1.10.0-beta")
}

```

Import libraries

Navigate to the new `src/main/java` folder and create a file called *Management.java*. Open it in your preferred editor or IDE and add the following `import` statements:

```

import com.azure.core.management.*;
import com.azure.core.management.profile.*;
import com.azure.identity.*;
import com.azure.resourcemanager.cognitiveservices.*;
import com.azure.resourcemanager.cognitiveservices.implementation.*;
import com.azure.resourcemanager.cognitiveservices.models.*;

import java.io.*;
import java.lang.Object.*;
import java.util.*;
import java.net.*;

```

Authenticate the client

Add a class in *Management.java*, and then add the following fields and their values inside of it. Populate their values, using the service principal you created and your other Azure account information.

```

/*
Be sure to use the service principal application ID, not simply the ID.
*/

private static String applicationId = "PASTE_YOUR_SERVICE_PRINCIPAL_APPLICATION_ID_HERE";
private static String applicationSecret = "PASTE_YOUR_SERVICE_PRINCIPAL_SECRET_HERE";

/* The ID of your Azure subscription. You can find this in the Azure Dashboard under Home > Subscriptions.
*/
private static String subscriptionId = "PASTE_YOUR_SUBSCRIPTION_ID_HERE";

/* The Active Directory tenant ID. You can find this in the Azure Dashboard under Home > Azure Active Directory. */
private static String tenantId = "PASTE_YOUR_TENANT_ID_HERE";

/* The name of the Azure resource group in which you want to create the resource.
You can find resource groups in the Azure Dashboard under Home > Resource groups. */
private static String resourceGroupName = "PASTE_YOUR_RESOURCE_GROUP_NAME_HERE";

/* The name of the custom subdomain to use when you create the resource. This is optional.
For example, if you create a Bing Search v7 resource with the custom subdomain name 'my-search-resource',
your resource would have the endpoint https://my-search-resource.cognitiveservices.azure.com/.
Note not all Cognitive Services allow custom subdomain names. */
private static String subDomainName = "PASTE_YOUR_SUBDOMAIN_NAME_HERE";

```

Then, in your **main** method, use these values to construct a **CognitiveServicesManager** object. This object is needed for all of your Azure management operations.

```
/* For more information see:
https://github.com/Azure/azure-sdk-for-java/blob/main/sdk/resourcemanager/docs/AUTH.md
*/

ClientSecretCredential credential = new ClientSecretCredentialBuilder()
    .clientId(applicationId)
    .clientSecret(applicationSecret)
    .tenantId(tenantId)
    .build();
AzureProfile profile = new AzureProfile(tenantId, subscriptionId, AzureEnvironment.AZURE);

CognitiveServicesManager client = CognitiveServicesManager.authenticate(credential, profile);
```

Call management methods

Add the following code to your **Main** method to list available resources, create a sample resource, list your owned resources, and then delete the sample resource. You'll define these methods in the next steps.

```
String resourceName = "test_resource";
String resourceKind = "TextTranslation";
String resourceSku = "F0";
Region resourceRegion = Region.US_WEST;

// Uncomment to list all available resource kinds, SKUs, and locations for your Azure account.
// list_available_kinds_skus_locations (client);

// Create a resource with kind Text Translation, SKU F0 (free tier), location US West.
String resourceId = create_resource (client, resourceName, resourceGroupName, resourceKind, resourceSku,
resourceRegion);

// Uncomment this to list all resources for your Azure account.
// list_resources (client, resourceGroupName);

// Delete the resource.
delete_resource (client, resourceId);

/* NOTE: When you delete a resource, it is only soft-deleted. You must also purge it. Otherwise, if you try
to create another
resource with the same name or custom subdomain, you will receive an error stating that such a resource
already exists. */
purge_resource (client, resourceName, resourceGroupName, resourceRegion);
```

Create a Cognitive Services resource (Java)

To create and subscribe to a new Cognitive Services resource, use the **create** method. This method adds a new billable resource to the resource group you pass in. When creating your new resource, you'll need to know the "kind" of service you want to use, along with its pricing tier (or SKU) and an Azure location. The following method takes all of these as arguments and creates a resource.

```

public static String create_resource (CognitiveServicesManager client, String resourceName, String
resourceGroupName, String resourceKind, String resourceSku, Region resourceRegion) {
    System.out.println ("Creating resource: " + resourceName + "...");

    /* NOTE: If you do not want to use a custom subdomain name, remove the withCustomSubDomainName
    setter from the AccountProperties object. */
    Account result = client.accounts().define(resourceName)
        .withExistingResourceGroup(resourceGroupName)
        // Note: Do not call withRegion() first, as it does not exist on the Blank interface returned by
    define().
        .withRegion(resourceRegion)
        .withKind(resourceKind)
        .withSku(new Sku().withName(resourceSku))
        .withProperties(new AccountProperties().withCustomSubDomainName(subDomainName))
        .create();

    System.out.println ("Resource created.");
    System.out.println ("ID: " + result.id());
    System.out.println ("Provisioning state: " + result.properties().provisioningState().toString());
    System.out.println ();

    return result.id();
}

```

Choose a service and pricing tier

When you create a new resource, you'll need to know the "kind" of service you want to use, along with the [pricing tier](#) (or SKU) you want. You'll use this and other information as parameters when creating the resource. You can find a list of available Cognitive Service "kinds" by calling the following method:

```

public static void list_available_kinds_skus_locations (CognitiveServicesManager client) {
    System.out.println ("Available SKUs:");
    System.out.println("Kind\tSKU Name\tSKU Tier\tLocations");
    ResourceSkus skus = client.resourceSkus();
    for (ResourceSku sku : skus.list()) {
        String locations = String.join (" ", sku.locations());
        System.out.println (sku.kind() + "\t" + sku.name() + "\t" + sku.tier() + "\t" + locations);
    }
}

```

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

See the list of SKUs and pricing information below.

Multi-service

SERVICE	KIND
Multiple services. For more information, see the pricing page.	CognitiveServices

Vision

SERVICE	KIND
Computer Vision	ComputerVision
Custom Vision - Prediction	CustomVision.Prediction
Custom Vision - Training	CustomVision.Training
Face	Face
Form Recognizer	FormRecognizer

Speech

SERVICE	KIND
Speech Services	SpeechServices
Speech Recognition	SpeakerRecognition

Language

SERVICE	KIND
LUIS	LUIS
QnA Maker	QnAMaker
Text Analytics	TextAnalytics
Text Translation	TextTranslation

Decision

SERVICE	KIND
Anomaly Detector	AnomalyDetector
Content Moderator	ContentModerator
Personalizer	Personalizer

Pricing tiers and billing

Pricing tiers (and the amount you get billed) are based on the number of transactions you send using your authentication information. Each pricing tier specifies the:

- maximum number of allowed transactions per second (TPS).
- service features enabled within the pricing tier.
- cost for a predefined number of transactions. Going above this number will cause an extra charge as specified in the [pricing details](#) for your service.

NOTE

Many of the Cognitive Services have a free tier you can use to try the service. To use the free tier, use `F0` as the SKU for your resource.

View your resources

To view all of the resources under your Azure account (across all resource groups), use the following method:

```
public static void list_resources (CognitiveServicesManager client, String resourceGroupName) {
    System.out.println ("Resources in resource group: " + resourceGroupName);
    // Note Azure resources are also sometimes referred to as accounts.
    Accounts accounts = client.accounts();
    for (Account account : accounts.listByResourceGroup(resourceGroupName)) {
        System.out.println ("ID: " + account.id());
        System.out.println ("Kind: " + account.kind ());
        System.out.println ("SKU Name: " + account.sku().name());
        System.out.println ("Custom subdomain name: " + account.properties().customSubDomainName());
        System.out.println ();
    }
}
```

Delete a resource

The following method deletes the specified resource from the given resource group.

```
public static void delete_resource (CognitiveServicesManager client, String resourceId) {
    System.out.println ("Deleting resource: " + resourceId + "...");
    client.accounts().deleteById (resourceId);
    System.out.println ("Resource deleted.");
    System.out.println ();
}
```

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.
- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.
- See [Azure Management SDK reference documentation](#) for more details on the management SDK.

[Reference documentation](#) | [Library source code](#) | [Package \(NPM\)](#) | [Samples](#)

JavaScript prerequisites

- A valid Azure subscription - [Create one for free](#).
- The current version of [Node.js](#)
- Your Azure account must have a Cognitive Services Contributor role assigned in order for you to agree to the responsible AI terms and create a resource. Contact your administrator to get this role assigned to your account.

- You must create your first Face, Text Analytics, or Computer Vision resources from the Azure portal to review and acknowledge the terms and conditions. You can do so here: [Face](#), [Text Analytics](#), [Computer Vision](#). After that, you can create subsequent resources using any deployment tool (SDK, CLI, or ARM template, etc) under the same Azure subscription.

Create an Azure Service Principal

To have your application interact with your Azure account, you need an Azure service principal to manage permissions. Follow the instructions in [Create an Azure service principal](#).

When you create a service principal, you'll see it has a secret value, an ID, and an application ID. Save the application ID and secret to a temporary location for later steps.

Create a resource group

Before you create a Cognitive Services resource, your account must have an Azure resource group to contain the resource. If you don't already have a resource group, create one in the [Azure portal](#) before continuing.

Create a new Node.js application

In a console window (such as cmd, PowerShell, or Bash), create a new directory for your app, and navigate to it.

```
mkdir myapp && cd myapp
```

Run the `npm init` command to create a node application with a `package.json` file.

```
npm init
```

Create a file named `index.js` before going on.

Install the client library

Install the following NPM packages:

```
npm install @azure/arm-cognitiveservices
npm install @azure/ms-rest-js
npm install @azure/ms-rest-nodeauth
```

Your app's `package.json` file will be updated with the dependencies.

Import libraries

Open your `index.js` script and import the following libraries.

```
"use strict";

/* To run this sample, install the following modules.
 * npm install @azure/arm-cognitiveservices @azure/identity
 */
var Arm = require("@azure/arm-cognitiveservices");
var Identity = require("@azure/identity");
```

Authenticate the client

Add the following fields to the root of your script and fill in their values, using the service principal you created

and your Azure account information.

```
const service_principal_application_id = "PASTE_YOUR_SERVICE_PRINCIPAL_APPLICATION_ID_HERE";
const service_principal_secret = "PASTE_YOUR_SERVICE_PRINCIPAL_SECRET_HERE";

/* The ID of your Azure subscription. You can find this in the Azure Dashboard under Home > Subscriptions.
*/
const subscription_id = "PASTE_YOUR_SUBSCRIPTION_ID_HERE";

/* The Active Directory tenant ID. You can find this in the Azure Dashboard under Home > Azure Active
Directory. */
const tenant_id = "PASTE_YOUR_TENANT_ID_HERE";

/* The name of the Azure resource group in which you want to create the resource.
You can find resource groups in the Azure Dashboard under Home > Resource groups. */
const resource_group_name = "PASTE_YOUR_RESOURCE_GROUP_NAME_HERE";

/* The name of the custom subdomain to use when you create the resource. This is optional.
For example, if you create a Bing Search v7 resource with the custom subdomain name 'my-search-resource',
your resource would have the endpoint https://my-search-resource.cognitiveservices.azure.com/.
Note not all Cognitive Services allow custom subdomain names.
*/
const subdomain_name = "PASTE_YOUR_SUBDOMAIN_NAME_HERE";
```

Next, add the following `quickstart` function to handle the main work of your program. The first block of code constructs a **CognitiveServicesManagementClient** object using the credential variables you entered above. This object is needed for all of your Azure management operations.

```
async function quickstart() {
  /* For more information see:
  https://www.npmjs.com/package/@azure/arm-cognitiveservices/v/6.0.0
  https://github.com/Azure/azure-sdk-for-
  js/blob/main/sdk/identity/identity/samples/AzureIdentityExamples.md#authenticating-a-service-principal-with-
  a-client-secret
  */
  const credentials = new Identity.ClientSecretCredential(tenant_id, service_principal_application_id,
  service_principal_secret);
  const client = new Arm.CognitiveServicesManagementClient (credentials, subscription_id);
  // Note Azure resources are also sometimes referred to as accounts.
  const accounts_client = new Arm.Accounts (client);
  const resource_skus_client = new Arm.ResourceSkus (client);
  const deleted_accounts_client = new Arm.DeletedAccounts (client);
```

Call management functions

Add the following code to the end of your `quickstart` function to list available resources, create a sample resource, list your owned resources, and then delete the sample resource. You'll define these functions in the next steps.

Create a Cognitive Services resource (Node.js)

To create and subscribe to a new Cognitive Services resource, use the **Create** function. This function adds a new billable resource to the resource group you pass in. When you create your new resource, you'll need to know the "kind" of service you want to use, along with its pricing tier (or SKU) and an Azure location. The following function takes all of these arguments and creates a resource.

```

async function create_resource (client, resource_name, resource_kind, resource_sku, resource_region) {
    console.log ("Creating resource: " + resource_name + "...");
    /* NOTE If you do not want to use a custom subdomain name, remove the customSubDomainName
    property from the properties object. */
    var parameters = { sku : { name: resource_sku }, kind : resource_kind, location : resource_region,
    properties : { customSubDomainName : subdomain_name } };
    return client.create(resource_group_name, resource_name, parameters)
        .then((result) => {
            console.log("Resource created.");
            console.log();
            console.log("ID: " + result.id);
            console.log("Kind: " + result.kind);
            console.log();
        })
        .catch((err) =>{
            console.log(err)
        })
    }

```

Choose a service and pricing tier

When you create a new resource, you'll need to know the "kind" of service you want to use, along with the [pricing tier](#) (or SKU) you want. You'll use this and other information as parameters when creating the resource. The following function lists the available Cognitive Service "kinds."

```

async function list_available_kinds_skus_locations (client) {
    console.log ("Available SKUs:");
    var result = await client.list ();
    console.log("Kind\tSKU Name\tSKU Tier\tLocations");
    result.forEach (function (x) {
        var locations = x.locations.join(",");
        console.log(x.kind + "\t" + x.name + "\t" + x.tier + "\t" + locations);
    });
}

```

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

See the list of SKUs and pricing information below.

Multi-service

SERVICE	KIND
Multiple services. For more information, see the pricing page.	CognitiveServices

Vision

SERVICE	KIND
Computer Vision	ComputerVision

SERVICE	KIND
Custom Vision - Prediction	CustomVision.Prediction
Custom Vision - Training	CustomVision.Training
Face	Face
Form Recognizer	FormRecognizer

Speech

SERVICE	KIND
Speech Services	SpeechServices
Speech Recognition	SpeakerRecognition

Language

SERVICE	KIND
LUIS	LUIS
QnA Maker	QnAMaker
Text Analytics	TextAnalytics
Text Translation	TextTranslation

Decision

SERVICE	KIND
Anomaly Detector	AnomalyDetector
Content Moderator	ContentModerator
Personalizer	Personalizer

Pricing tiers and billing

Pricing tiers (and the amount you get billed) are based on the number of transactions you send using your authentication information. Each pricing tier specifies the:

- maximum number of allowed transactions per second (TPS).
- service features enabled within the pricing tier.
- cost for a predefined number of transactions. Going above this number will cause an extra charge as specified in the [pricing details](#) for your service.

NOTE

Many of the Cognitive Services have a free tier you can use to try the service. To use the free tier, use `F0` as the SKU for your resource.

View your resources

To view all of the resources under your Azure account (across all resource groups), use the following function:

```
async function list_resources (client) {
  console.log ("Resources in resource group: " + resource_group_name);
  var result = await client.listByResourceGroup (resource_group_name);
  result.forEach (function (x) {
    console.log(x);
    console.log();
  });
}
```

Delete a resource

The following function deletes the specified resource from the given resource group.

```
async function delete_resource (client, resource_name) {
  console.log ("Deleting resource: " + resource_name + "...");
  await client.deleteMethod (resource_group_name, resource_name)
  console.log ("Resource deleted.");
  console.log ();
}
```

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

Run the application

Add the following code to the bottom of your script to call your main `quickstart` function with error handling.

```
try {
  quickstart();
}
catch (error) {
  console.log(error);
}
```

Then, in your console window, run the application with the `node` command.

```
node index.js
```

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.
- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.
- See [Azure Management SDK reference documentation](#) for more details on the management SDK.

[Reference documentation](#) | [Library source code](#) | [Package \(PyPi\)](#) | [Samples](#)

Python prerequisites

- A valid Azure subscription - [Create one for free](#).
- [Python 3.x](#)
- Your Azure account must have a Cognitive Services Contributor role assigned in order for you to agree to the responsible AI terms and create a resource. Contact your administrator to get this role assigned to your account.
- You must create your first Face, Text Analytics, or Computer Vision resources from the Azure portal to review and acknowledge the terms and conditions. You can do so here: [Face](#), [Text Analytics](#), [Computer Vision](#). After that, you can create subsequent resources using any deployment tool (SDK, CLI, or ARM template, etc) under the same Azure subscription.

Create an Azure Service Principal

To have your application interact with your Azure account, you need an Azure service principal to manage permissions. Follow the instructions in [Create an Azure service principal](#).

When you create a service principal, you'll see it has a secret value, an ID, and an application ID. Save the application ID and secret to a temporary location for later steps.

Create a resource group

Before you create a Cognitive Services resource, your account must have an Azure resource group to contain the resource. If you don't already have a resource group, create one in the [Azure portal](#) before continuing.

Create a new Python application

Create a new Python application in your preferred editor or IDE and navigate to your project in a console window.

Install the client library

You can install the client library with:

```
pip install azure-mgmt-cognitiveservices
```

If you're using the Visual Studio IDE, the client library is available as a downloadable NuGet package.

Import libraries

Open your Python script and import the following libraries.

```
import time
from azure.identity import ClientSecretCredential
from azure.mgmt.cognitiveservices import CognitiveServicesManagementClient
from azure.mgmt.cognitiveservices.models import Account, Sku
```

Authenticate the client

Add the following fields to the root of your script and fill in their values, using the service principal you created and your Azure account information.

```
# Be sure to use the service principal application ID, not simply the ID.
service_principal_application_id = "PASTE_YOUR_SERVICE_PRINCIPAL_APPLICATION_ID_HERE"
service_principal_secret = "PASTE_YOUR_SERVICE_PRINCIPAL_SECRET_HERE"

# The ID of your Azure subscription. You can find this in the Azure Dashboard under Home > Subscriptions.
subscription_id = "PASTE_YOUR_SUBSCRIPTION_ID_HERE"

# The Active Directory tenant ID. You can find this in the Azure Dashboard under Home > Azure Active Directory.
tenant_id = "PASTE_YOUR_TENANT_ID_HERE"

# The name of the Azure resource group in which you want to create the resource.
# You can find resource groups in the Azure Dashboard under Home > Resource groups.
resource_group_name = "PASTE_YOUR_RESOURCE_GROUP_NAME_HERE"

# The name of the custom subdomain to use when you create the resource. This is optional.
# For example, if you create a Bing Search v7 resource with the custom subdomain name 'my-search-resource',
# your resource would have the endpoint https://my-search-resource.cognitiveservices.azure.com/.
# Note not all Cognitive Services allow custom subdomain names.
subdomain_name = "PASTE_YOUR_SUBDOMAIN_NAME_HERE"

# How many seconds to wait between checking the status of an async operation.
wait_time = 10
```

Then add the following code to construct a **CognitiveServicesManagementClient** object. This object is needed for all of your Azure management operations.

```
credential = ClientSecretCredential(tenant_id, service_principal_application_id, service_principal_secret)
client = CognitiveServicesManagementClient(credential, subscription_id)
```

Create a Cognitive Services resource (Python)

To create and subscribe to a new Cognitive Services resource, use the **Create** function. This function adds a new billable resource to the resource group you pass in. When you create your new resource, you'll need to know the "kind" of service you want to use, along with its pricing tier (or SKU) and an Azure location. The following function takes all of these arguments and creates a resource.

```
def create_resource (resource_name, kind, sku_name, location) :
    print("Creating resource: " + resource_name + "...")

    # NOTE If you do not want to use a custom subdomain name, remove the customSubDomainName
    # property from the properties object.
    parameters = Account(sku=sku_name, kind=kind, location=location, properties={
        'custom_sub_domain_name' : subdomain_name })

    poller = client.accounts.begin_create(resource_group_name, resource_name, parameters)
    while (False == poller.done ()) :
        print ("Waiting {wait_time} seconds for operation to finish.".format (wait_time = wait_time))
        time.sleep (wait_time)
    # This will raise an exception if the server responded with an error.
    result = poller.result ()

    print("Resource created.")
    print()
    print("ID: " + result.id)
    print("Name: " + result.name)
    print("Type: " + result.type)
    print()
```

Choose a service and pricing tier

When you create a new resource, you'll need to know the "kind" of service you want to use, along with the [pricing tier](#) (or SKU) you want. You'll use this and other information as parameters when creating the resource. The following function lists the available Cognitive Service "kinds."

```
def list_available_kinds_skus_locations():
    print("Available SKUs:")
    result = client.resource_skus.list()
    print("Kind\tSKU Name\tSKU Tier\tLocations")
    for x in result:
        locations = ",".join(x.locations)
        print(x.kind + "\t" + x.name + "\t" + x.tier + "\t" + locations)
```

You can access Azure Cognitive Services through two different resources: A multi-service resource, or a single-service one.

- Multi-service resource:
 - Access multiple Azure Cognitive Services with a single key and endpoint.
 - Consolidates billing from the services you use.
- Single-service resource:
 - Access a single Azure Cognitive Service with a unique key and endpoint for each service created.
 - Use the free tier to try out the service.

See the list of SKUs and pricing information below.

Multi-service

SERVICE	KIND
Multiple services. For more information, see the pricing page.	CognitiveServices

Vision

SERVICE	KIND
Computer Vision	ComputerVision
Custom Vision - Prediction	CustomVision.Prediction
Custom Vision - Training	CustomVision.Training
Face	Face
Form Recognizer	FormRecognizer

Speech

SERVICE	KIND
Speech Services	SpeechServices
Speech Recognition	SpeakerRecognition

Language

SERVICE	KIND
LUIS	LUIS
QnA Maker	QnAMaker
Text Analytics	TextAnalytics
Text Translation	TextTranslation

Decision

SERVICE	KIND
Anomaly Detector	AnomalyDetector
Content Moderator	ContentModerator
Personalizer	Personalizer

Pricing tiers and billing

Pricing tiers (and the amount you get billed) are based on the number of transactions you send using your authentication information. Each pricing tier specifies the:

- maximum number of allowed transactions per second (TPS).
- service features enabled within the pricing tier.
- cost for a predefined number of transactions. Going above this number will cause an extra charge as specified in the [pricing details](#) for your service.

NOTE

Many of the Cognitive Services have a free tier you can use to try the service. To use the free tier, use `F0` as the SKU for your resource.

View your resources

To view all of the resources under your Azure account (across all resource groups), use the following function:

```
def list_resources():
    print("Resources in resource group: " + resource_group_name)
    result = client.accounts.list_by_resource_group(resource_group_name)
    for x in result:
        print(x.name)
        print(x)
        print()
```

Delete a resource

The following function deletes the specified resource from the given resource group.

```
def delete_resource(resource_name) :
    print("Deleting resource: " + resource_name + "...")

    poller = client.accounts.begin_delete(resource_group_name, resource_name)
    while (False == poller.done ()) :
        print ("Waiting {wait_time} seconds for operation to finish.".format (wait_time = wait_time))
        time.sleep (wait_time)
    # This will raise an exception if the server responded with an error.
    result = poller.result ()

    print("Resource deleted.")
```

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

Call management functions

Add the following code to the bottom of your script to call the above functions. This code lists available resources, creates a sample resource, lists your owned resources, and then deletes the sample resource.

```
resource_name = "test_resource"
resource_kind = "TextTranslation"
resource_sku = "F0"
resource_location = "Global"

# Uncomment this to list all available resource kinds, SKUs, and locations for your Azure account.
#list_available_kinds_skus_locations ()

# Create a resource with kind Text Translation, SKU F0 (free tier), location global.
create_resource(resource_name, resource_kind, resource_sku, resource_location)

# Uncomment this to list all resources for your Azure account.
#list_resources()

# Delete the resource.
delete_resource(resource_name)

# NOTE: Deleting a resource only soft-deletes it. To delete it permanently, you must purge it.
# Otherwise, if you later try to create a resource with the same name, you will receive the following error:
# azure.core.exceptions.ResourceExistsError: (FlagMustBeSetForRestore) An existing resource with ID '<your
resource ID>' has been soft-deleted. To restore the resource, you must specify 'restore' to be 'true' in the
property. If you don't want to restore existing resource, please purge it first.
# Code: FlagMustBeSetForRestore

# Purge the resource.
purge_resource(resource_name, resource_location)
```

Run the application

Run your application from the command line with the `python` command.

```
python <your-script-name>.py
```

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.

- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.
- See [Azure Management SDK reference documentation](#) for more details on the management SDK.

Quickstart: Create a Cognitive Services resource using an ARM template

6/11/2021 • 4 minutes to read • [Edit Online](#)

This quickstart describes how to use an Azure Resource Manager template (ARM template) to create Cognitive Services.

Azure Cognitive Services are cloud-based services with REST APIs, and client library SDKs available to help developers build cognitive intelligence into applications without having direct artificial intelligence (AI) or data science skills or knowledge. Azure Cognitive Services enables developers to easily add cognitive features into their applications with cognitive solutions that can see, hear, speak, understand, and even begin to reason.

Create a resource using an Azure Resource Manager template (ARM template). This multi-service resource lets you:

- Access multiple Azure Cognitive Services with a single key and endpoint.
- Consolidate billing from the services you use.
- You must create your first Face, Text Analytics, or Computer Vision resources from the Azure portal to review and acknowledge the terms and conditions. You can do so here: [Face](#), [Text Analytics](#), [Computer Vision](#). After that, you can create subsequent resources using any deployment tool (SDK, CLI, or ARM template, etc) under the same Azure subscription.

An [ARM template](#) is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. In declarative syntax, you describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template will open in the Azure portal.



Prerequisites

- If you don't have an Azure subscription, [create one for free](#).

Review the template

The template used in this quickstart is from [Azure Quickstart Templates](#).

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "cognitiveServiceName": {
      "type": "string",
      "defaultValue": "[concat('CognitiveService-', uniqueString(resourceGroup().id))]",
      "metadata": {
        "description": "That name is the name of our application. It has to be unique.Type a name followed
by your resource group name. (<name>-<resourceGroupName>)"
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Location for all resources."
      }
    },
    "sku": {
      "type": "string",
      "defaultValue": "S0",
      "allowedValues": [
        "S0"
      ]
    }
  },
  "resources": [
    {
      "type": "Microsoft.CognitiveServices/accounts",
      "apiVersion": "2017-04-18",
      "name": "[parameters('cognitiveServiceName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "[parameters('sku')]"
      },
      "kind": "CognitiveServices",
      "properties": {
        "statisticsEnabled": false
      }
    }
  ]
}
```

One Azure resource is defined in the template:

- [Microsoft.CognitiveServices/accounts](#): creates a Cognitive Services resource.

Deploy the template

- [Azure portal](#)
- [Azure CLI](#)

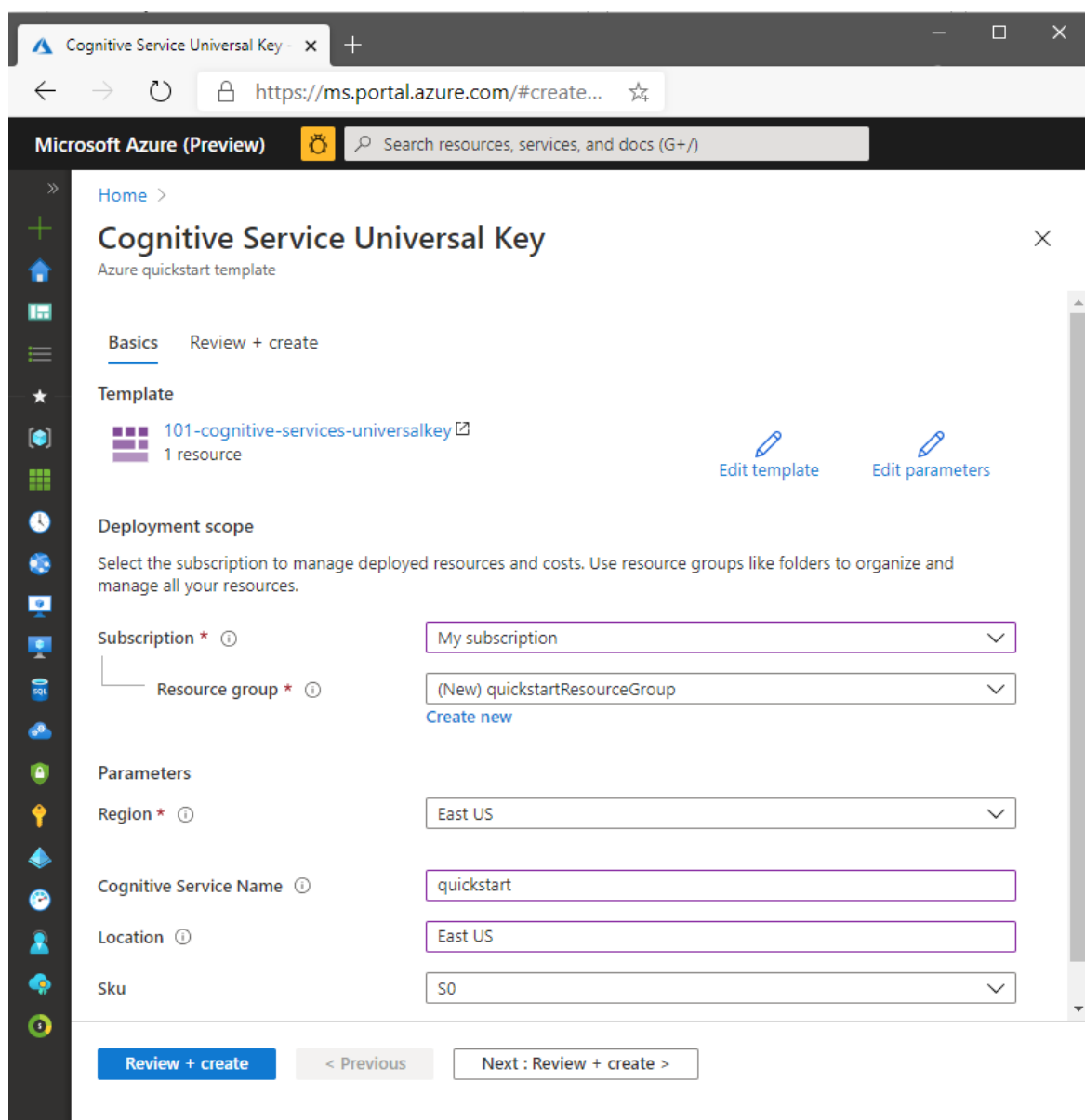
1. Click the **Deploy to Azure** button.



2. Enter the following values.

VALUE	DESCRIPTION
Subscription	Select an Azure subscription.

VALUE	DESCRIPTION
Resource group	Select Create new , enter a unique name for the resource group, and then click OK .
Region	Select a region. For example, East US
Cognitive Service Name	Replace with a unique name for your resource. You will need the name in the next section when you validate the deployment.
Location	Replace with the region used above.
Sku	The pricing tier for your resource.



The screenshot shows the Azure portal interface for deploying a 'Cognitive Service Universal Key'. The 'Review + create' tab is active. The deployment scope is set to 'My subscription' and '(New) quickstartResourceGroup'. The parameters are: Region 'East US', Cognitive Service Name 'quickstart', Location 'East US', and Sku 'S0'. The 'Review + create' button is highlighted in blue.

3. Select **Review + Create**, then **Create**. After the resource has successfully finished deploying, the **Go to resource** button will be highlighted.

TIP

If your subscription doesn't allow you to create a Cognitive Service resource, you may need to enable that ability of the [Azure resource provider](#) with the [Azure portal](#), [PowerShell command](#) or an [Azure CLI command](#). If you are not the subscription owner, ask the *Subscription Owner* or someone with a role of *admin* to complete the registration for you or ask for the `/register/action` privileges granted to your account.

Review deployed resources

- [Portal](#)
- [Azure CLI](#)

When your deployment finishes, you will be able to click the **Go to resource** button to see your new resource. You can also find the resource group by:

1. Selecting **Resource groups** from the left navigation menu.
2. Selecting the resource group name.

Clean up resources

If you want to clean up and remove a Cognitive Services subscription, you can delete the resource or resource group. Deleting the resource group also deletes any other resources contained in the group.

- [Azure portal](#)
- [Azure CLI](#)

1. In the Azure portal, expand the menu on the left side to open the menu of services, and choose **Resource Groups** to display the list of your resource groups.
2. Locate the resource group containing the resource to be deleted
3. Right-click on the resource group listing. Select **Delete resource group**, and confirm.

If you need to recover a deleted resource, see [Recover deleted Cognitive Services resources](#).

See also

- See [Authenticate requests to Azure Cognitive Services](#) on how to securely work with Cognitive Services.
- See [What are Azure Cognitive Services?](#) to get a list of different categories within Cognitive Services.
- See [Natural language support](#) to see the list of natural languages that Cognitive Services supports.
- See [Use Cognitive Services as containers](#) to understand how to use Cognitive Services on-prem.
- See [Plan and manage costs for Cognitive Services](#) to estimate cost of using Cognitive Services.

Plan and manage costs for Azure Cognitive Services

6/14/2021 • 6 minutes to read • [Edit Online](#)

This article describes how you plan for and manage costs for Azure Cognitive Services. First, you use the Azure pricing calculator to help plan for Cognitive Services costs before you add any resources for the service to estimate costs. Next, as you add Azure resources, review the estimated costs. After you've started using Cognitive Services resources (for example Speech, Computer Vision, LUIS, Text Analytics, Translator, etc.), use Cost Management features to set budgets and monitor costs. You can also review forecasted costs and identify spending trends to identify areas where you might want to act. Costs for Cognitive Services are only a portion of the monthly costs in your Azure bill. Although this article explains how to plan for and manage costs for Cognitive Services, you're billed for all Azure services and resources used in your Azure subscription, including the third-party services.

Prerequisites

Cost analysis in Cost Management supports most Azure account types, but not all of them. To view the full list of supported account types, see [Understand Cost Management data](#). To view cost data, you need at least read access for an Azure account. For information about assigning access to Azure Cost Management data, see [Assign access to data](#).

Estimate costs before using Cognitive Services

Use the [Azure pricing calculator](#) to estimate costs before you add Cognitive Services.

Cognitive Services

API:	REGION:	INSTANCE:
Computer Vision ▼	West US ▼	S1 ▼

 Up to 10 transactions per second.

Features

Tag, Face, GetThumbnail Color, Image Type

20000	=	\$20.00
Transactions		

OCR, Adult, Celebrity, and Landmark

10000	=	\$15.00
Transactions		

Describe and Recognize Text

100000	×	\$2.50	=	\$250.00
Transactions		Per 1,000 transactions		

Upfront cost	\$0.00
Monthly cost	\$285.00

Support

SUPPORT:		\$29.00
Developer ▼		

Programs and Offers

LICENSING PROGRAM:	
Microsoft Online Services Agreement ▼	

☒ SHOW DEV/TEST PRICING 

Estimated upfront cost	\$0.00
Estimated monthly cost	\$314.00

As you add new resources to your workspace, return to this calculator and add the same resource here to update your cost estimates.

For more information, see [Azure Cognitive Services pricing](#).

Understand the full billing model for Cognitive Services

Cognitive Services runs on Azure infrastructure that [accrues costs](#) when you deploy the new resource. It's important to understand that more infrastructure might accrue costs. You need to manage that cost when you make changes to deployed resources.

How you're charged for Cognitive Services

When you create or use Cognitive Services resources, you might get charged for the following meters based on the services that you use:

SERVICE	METER(S)	BILLING INFORMATION
Vision		
Computer Vision	Free, Standard (S1)	Billed by the number of transactions. Price per transaction varies per feature (Read, OCR, Spatial Analysis). For full details, see Pricing .

SERVICE	METER(S)	BILLING INFORMATION
Custom Vision	Free, Standard	<ul style="list-style-type: none"> • Predictions are billed by the number of transactions. • Training is billed by compute hour(s). • Image storage is billed by number of images (up to 6 MB per image).
Face	Free, Standard	Billed by the number of transactions.
Speech		
Speech service	Free, Standard	Billing varies by feature (speech-to-text, text-to-speech, speech translation, speaker recognition). Primarily, billing is by transaction count or character count. For full details, see Pricing .
Language		
Language Understanding (LUIS)	Free Authoring, Free Prediction, Standard	Billed by number of transactions. Price per transaction varies by feature (speech requests, text requests). For full details, see Pricing .
QnA Maker	Free, Standard	Subscription fee billed monthly. For full details, see Pricing .
Text Analytics	Free, Standard	Billed by number of text records.
Translator	Free, Pay-as-you-go (S1), Volume discount (S2, S3, S4, C2, C3, C4, D3)	Pricing varies by meter and feature. For full details, see Pricing . <ul style="list-style-type: none"> • Text translation is billed by number of characters translated. • Document translation is billed by characters translated. • Custom translation is billed by characters of source and target training data.
Decision		
Anomaly Detector	Free, Standard	Billed by the number of transactions.
Content Moderator	Free, Standard	Billed by the number of transactions.
Personalizer	Free, Standard (S0)	Billed by transactions per month. There are storage and transaction quotas. For full details, see Pricing .

Costs that typically accrue with Cognitive Services

Typically, after you deploy an Azure resource, costs are determined by your pricing tier and the API calls you make to your endpoint. If the service you're using has a commitment tier, going over the allotted calls in your tier may incur an overage charge.

Extra costs may accrue when using these services:

QnA Maker

When you create resources for QnA Maker, resources for other Azure services may also be created. They include:

- [Azure App Service \(for the runtime\)](#)
- [Azure Cognitive Search \(for the data\)](#)

Costs that might accrue after resource deletion

QnA Maker

After you delete QnA Maker resources, the following resources might continue to exist. They continue to accrue costs until you delete them.

- [Azure App Service \(for the runtime\)](#)
- [Azure Cognitive Search \(for the data\)](#)

Using Azure Prepayment credit with Cognitive Services

You can pay for Cognitive Services charges with your Azure Prepayment (previously called monetary commitment) credit. However, you can't use Azure Prepayment credit to pay for charges for third-party products and services including those from the Azure Marketplace.

Monitor costs

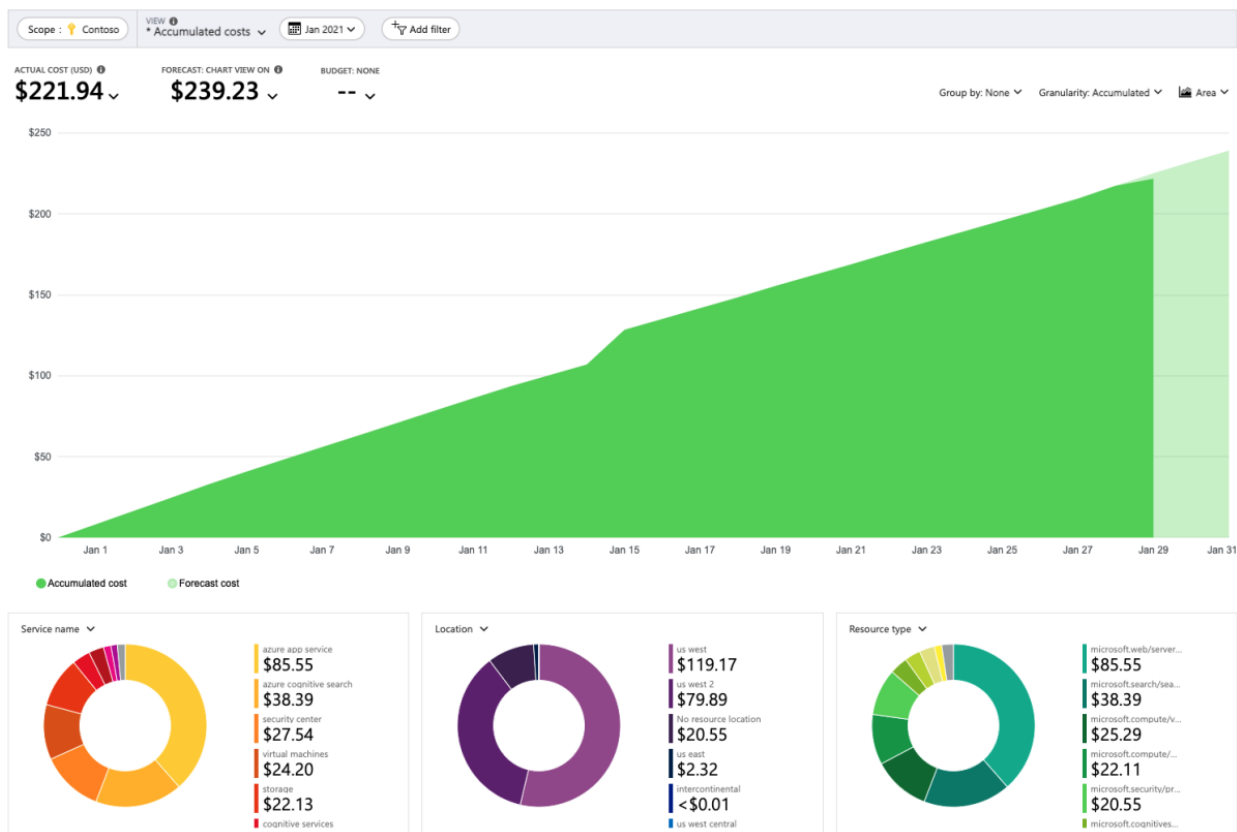
As you use Azure resources with Cognitive Services, you incur costs. Azure resource usage unit costs vary by time intervals (seconds, minutes, hours, and days) or by unit usage (bytes, megabytes, and so on). As soon as use of a Cognitive Service (or Cognitive Services) starts, costs are incurred and you can see the costs in [cost analysis](#).

When you use cost analysis, you view Cognitive Services costs in graphs and tables for different time intervals. Some examples are by day, current and prior month, and year. You also view costs against budgets and forecasted costs. Switching to longer views over time can help you identify spending trends. And you see where overspending might have occurred. If you've created budgets, you can also easily see where they're exceeded.

To view Cognitive Services costs in cost analysis:

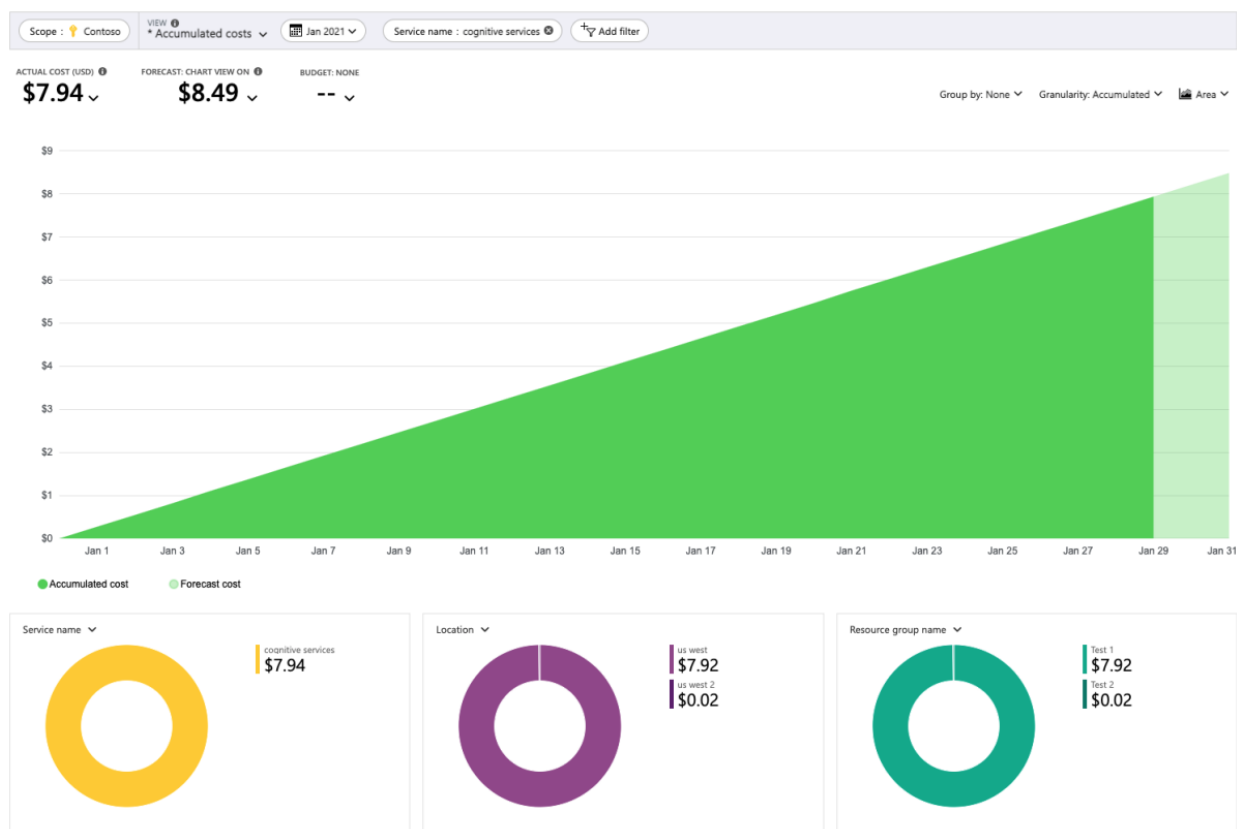
1. Sign in to the Azure portal.
2. Open the scope in the Azure portal and select **Cost analysis** in the menu. For example, go to **Subscriptions**, select a subscription from the list, and then select **Cost analysis** in the menu. Select **Scope** to switch to a different scope in cost analysis.
3. By default, cost for services are shown in the first donut chart. Select the area in the chart labeled Cognitive Services.

Actual monthly costs are shown when you initially open cost analysis. Here's an example showing all monthly usage costs.



- To narrow costs for a single service, like Cognitive Services, select **Add filter** and then select **Service name**. Then, select **Cognitive Services**.

Here's an example showing costs for just Cognitive Services.



In the preceding example, you see the current cost for the service. Costs by Azure regions (locations) and Cognitive Services costs by resource group are also shown. From here, you can explore costs on your own.

Create budgets

You can create [budgets](#) to manage costs and create [alerts](#) that automatically notify stakeholders of spending anomalies and overspending risks. Alerts are based on spending compared to budget and cost thresholds. Budgets and alerts are created for Azure subscriptions and resource groups, so they're useful as part of an overall cost monitoring strategy.

Budgets can be created with filters for specific resources or services in Azure if you want more granularity present in your monitoring. Filters help ensure that you don't accidentally create new resources that cost you more money. For more about the filter options when you create a budget, see [Group and filter options](#).

Export cost data

You can also [export your cost data](#) to a storage account. This is helpful when you or others need to do more data analysis for costs. For example, finance teams can analyze the data using Excel or Power BI. You can export your costs on a daily, weekly, or monthly schedule and set a custom date range. Exporting cost data is the recommended way to retrieve cost datasets.

Next steps

- Learn [how to optimize your cloud investment with Azure Cost Management](#).
- Learn more about managing costs with [cost analysis](#).
- Learn about how to [prevent unexpected costs](#).
- Take the [Cost Management](#) guided learning course.

Enable diagnostic logging for Azure Cognitive Services

8/9/2021 • 4 minutes to read • [Edit Online](#)

This guide provides step-by-step instructions to enable diagnostic logging for an Azure Cognitive Service. These logs provide rich, frequent data about the operation of a resource that are used for issue identification and debugging. Before you continue, you must have an Azure account with a subscription to at least one Cognitive Service, such as [Speech Services](#), or [LUIS](#).

Prerequisites

To enable diagnostic logging, you'll need somewhere to store your log data. This tutorial uses Azure Storage and Log Analytics.

- [Azure storage](#) - Retains diagnostic logs for policy audit, static analysis, or backup. The storage account does not have to be in the same subscription as the resource emitting logs as long as the user who configures the setting has appropriate Azure RBAC access to both subscriptions.
- [Log Analytics](#) - A flexible log search and analytics tool that allows for analysis of raw logs generated by an Azure resource.

NOTE

- Additional configuration options are available. To learn more, see [Collect and consume log data from your Azure resources](#).
- "Trace" in diagnostic logging is only available for [Custom question answering](#).

Enable diagnostic log collection

Let's start by enabling diagnostic logging using the Azure portal.

NOTE

To enable this feature using PowerShell or the Azure CLI, use the instructions provided in [Collect and consume log data from your Azure resources](#).

1. Navigate to the Azure portal. Then locate and select a Cognitive Services resource. For example, your subscription to Speech Services.
2. Next, from the left-hand navigation menu, locate **Monitoring** and select **Diagnostic settings**. This screen contains all previously created diagnostic settings for this resource.
3. If there is a previously created resource that you'd like to use, you can select it now. Otherwise, select + **Add diagnostic setting**.
4. Enter a name for the setting. Then select **Archive to a storage account** and **Send to log Analytics**.
5. When prompted to configure, select the storage account and OMS workspace that you'd like to use to store your diagnostic logs. **Note:** If you don't have a storage account or OMS workspace, follow the prompts to create one.
6. Select **Audit**, **RequestResponse**, and **AllMetrics**. Then set the retention period for your diagnostic log data. If a retention policy is set to zero, events for that log category are stored indefinitely.

7. Click **Save**.

It can take up to two hours before logging data is available to query and analyze. So don't worry if you don't see anything right away.

View and export diagnostic data from Azure Storage

Azure Storage is a robust object storage solution that is optimized for storing large amounts of unstructured data. In this section, you'll learn to query your storage account for total transactions over a 30-day timeframe and export the data to excel.

1. From the Azure portal, locate the Azure Storage resource that you created in the last section.
2. From the left-hand navigation menu, locate **Monitoring** and select **Metrics**.
3. Use the available drop-downs to configure your query. For this example, let's set the time range to **Last 30 days** and the metric to **Transaction**.
4. When the query is complete, you'll see a visualization of transaction over the last 30 days. To export this data, use the **Export to Excel** button located at the top of the page.

Learn more about what you can do with diagnostic data in [Azure Storage](#).

View logs in Log Analytics

Follow these instructions to explore log analytics data for your resource.

1. From the Azure portal, locate and select **Log Analytics** from the left-hand navigation menu.
2. Locate and select the resource you created when enabling diagnostics.
3. Under **General**, locate and select **Logs**. From this page, you can run queries against your logs.

Sample queries

Here are a few basic Kusto queries you can use to explore your log data.

Run this query for all diagnostic logs from Azure Cognitive Services for a specified time period:

```
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.COGNITIVESERVICES"
```

Run this query to see the 10 most recent logs:

```
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.COGNITIVESERVICES"
| take 10
```

Run this query to group operations by **Resource**:

```
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.COGNITIVESERVICES" |
summarize count() by Resource
```

Run this query to find the average time it takes to perform an operation:

```
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.COGNITIVESERVICES"
| summarize avg(DurationMs)
by OperationName
```

Run this query to view the volume of operations over time split by OperationName with counts binned for every 10s.

```
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.COGNITIVESERVICES"
| summarize count()
by bin(TimeGenerated, 10s), OperationName
| render areachart kind=unstacked
```

Next steps

- To understand how to enable logging, and also the metrics and log categories that are supported by the various Azure services, read both the [Overview of metrics](#) in Microsoft Azure and [Overview of Azure Diagnostic Logs](#) articles.
- Read these articles to learn about event hubs:
 - [What is Azure Event Hubs?](#)
 - [Get started with Event Hubs](#)
- Read [Download metrics and diagnostic logs from Azure Storage](#).
- Read [Understand log searches in Azure Monitor logs](#).

Recover deleted Cognitive Services resources

7/15/2021 • 2 minutes to read • [Edit Online](#)

This article provides instructions on how to recover a Cognitive Services resource that is already deleted. The article also provides instructions on how to purge a deleted resource.

NOTE

The instructions in this article are applicable to both a multi-service resource and a single-service resource. A multi-service resource enables access to multiple cognitive services using a single key and endpoint. On the other hand, a single-service resource enables access to just that specific cognitive service for which the resource was created.

Prerequisites

- The resource to be recovered must have been deleted within the past 48 hours.
- The resource to be recovered must not have been purged already. A purged resource cannot be recovered.
- Before you attempt to recover a deleted resource, make sure that the resource group for that account exists. If the resource group was deleted, you must recreate it. Recovering a resource group is not possible. For more information, see [Manage resource groups](#).
- If the deleted resource used customer-managed keys with Azure Key Vault and the key vault has also been deleted, then you must restore the key vault before you restore the Cognitive Services resource. For more information, see [Azure Key Vault recovery management](#).
- If the deleted resource used a customer-managed storage and storage account has also been deleted, you must restore the storage account before you restore the Cognitive Services resource. For instructions, see [Recover a deleted storage account](#).

Your subscription must have `Microsoft.CognitiveServices/locations/resourceGroups/deletedAccounts/delete` permissions to purge resources, such as [Cognitive Services Contributor](#) or [Contributor](#).

Recover a deleted resource

To recover a deleted cognitive service resource, use the following commands. Where applicable, replace:

- `{subscriptionID}` with your Azure subscription ID
- `{resourceGroup}` with your resource group
- `{resourceName}` with your resource name
- `{location}` with the location of your resource

Using the REST API

Use the following `PUT` command:

```
https://management.azure.com/subscriptions/{subscriptionID}/resourceGroups/{resourceGroup}/providers/Microsoft.CognitiveServices/accounts/{resourceName}?Api-Version=2021-04-30
```

In the request body, use the following JSON format:

```
{
  "location": "{location}",
  "properties": {
    "restore": true
  }
}
```

Using PowerShell

Use the following command to restore the resource:

```
New-AzResource -Location {location}-Properties @{restore=$true} -ResourceId
/subscriptions/{subscriptionID}/resourceGroups/{resourceGroup}/providers/Microsoft.CognitiveServices/account
s/{resourceName} -ApiVersion 2021-04-30
```

If you need to find the name of your deleted resources, you can get a list of deleted resource names with the following command:

```
Get-AzResource -ResourceId
/subscriptions/{subscriptionID}/providers/Microsoft.CognitiveServices/deletedAccounts -ApiVersion 2021-04-30
```

Using the Azure CLI

```
az resource create --subscription {subscriptionID} -g {resourceGroup} -n {resourceName} --location
{location} --namespace Microsoft.CognitiveServices --resource-type accounts --properties "{\"restore\":
true}"
```

Purge a deleted resource

Once you delete a resource, you won't be able to create another one with the same name for 48 hours. To create a resource with the same name, you will need to purge the deleted resource.

To purge a deleted cognitive service resource, use the following commands. Where applicable, replace:

- {subscriptionID} with your Azure subscription ID
- {resourceGroup} with your resource group
- {resourceName} with your resource name
- {location} with the location of your resource

NOTE

Once a resource is purged, it is permanently deleted and cannot be restored. You will lose all data and keys associated with the resource.

Using the REST API

Use the following **DELETE** command:

```
https://management.azure.com/subscriptions/{subscriptionID}/providers/Microsoft.CognitiveServices/locations/
{location}/resourceGroups/{resourceGroup}/deletedAccounts/{resourceName}?Api-Version=2021-04-30`
```

Using PowerShell

```
Remove-AzResource -ResourceId  
/subscriptions/{subscriptionID}/providers/Microsoft.CognitiveServices/locations/{location}/resourceGroups/{r  
esourceGroup}/deletedAccounts/{resourceName} -ApiVersion 2021-04-30`
```

Using the Azure CLI

```
az resource delete --ids  
/subscriptions/{subscriptionId}/providers/Microsoft.CognitiveServices/locations/{location}/resourceGroups/{r  
esourceGroup}/deletedAccounts/{resourceName}
```

See also

- [Create a new resource using the Azure portal](#)
- [Create a new resource using the Azure CLI](#)
- [Create a new resource using the client library](#)
- [Create a new resource using an ARM template](#)

Azure Cognitive Services containers

9/3/2021 • 7 minutes to read • [Edit Online](#)

Azure Cognitive Services provides several [Docker containers](#) that let you use the same APIs that are available in Azure, on-premises. Using these containers gives you the flexibility to bring Cognitive Services closer to your data for compliance, security or other operational reasons. Container support is currently available for a subset of Azure Cognitive Services.

Containerization is an approach to software distribution in which an application or service, including its dependencies & configuration, is packaged together as a container image. With little or no modification, a container image can be deployed on a container host. Containers are isolated from each other and the underlying operating system, with a smaller footprint than a virtual machine. Containers can be instantiated from container images for short-term tasks, and removed when no longer needed.

Features and benefits

- **Immutable infrastructure:** Enable DevOps teams' to leverage a consistent and reliable set of known system parameters, while being able to adapt to change. Containers provide the flexibility to pivot within a predictable ecosystem and avoid configuration drift.
- **Control over data:** Choose where your data gets processed by Cognitive Services. This can be essential if you can't send data to the cloud but need access to Cognitive Services APIs. Support consistency in hybrid environments – across data, management, identity, and security.
- **Control over model updates:** Flexibility in versioning and updating of models deployed in their solutions.
- **Portable architecture:** Enables the creation of a portable application architecture that can be deployed on Azure, on-premises and the edge. Containers can be deployed directly to [Azure Kubernetes Service](#), [Azure Container Instances](#), or to a [Kubernetes](#) cluster deployed to [Azure Stack](#). For more information, see [Deploy Kubernetes to Azure Stack](#).
- **High throughput / low latency:** Provide customers the ability to scale for high throughput and low latency requirements by enabling Cognitive Services to run physically close to their application logic and data. Containers do not cap transactions per second (TPS) and can be made to scale both up and out to handle demand if you provide the necessary hardware resources.
- **Scalability:** With the ever growing popularity of containerization and container orchestration software, such as [Kubernetes](#); scalability is at the forefront of technological advancements. Building on a scalable cluster foundation, application development caters to high availability.

Containers in Azure Cognitive Services

Azure Cognitive Services containers provide the following set of Docker containers, each of which contains a subset of functionality from services in Azure Cognitive Services. You can find instructions and image locations in the tables below. A list of [container images](#) is also available.

Decision containers

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
---------	-----------	-------------	--------------

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
Anomaly detector	Anomaly Detector (image)	The Anomaly Detector API enables you to monitor and detect abnormalities in your time series data with machine learning.	Generally available

Language containers

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
LUIS	LUIS (image)	Loads a trained or published Language Understanding model, also known as a LUIS app, into a docker container and provides access to the query predictions from the container's API endpoints. You can collect query logs from the container and upload these back to the LUIS portal to improve the app's prediction accuracy.	Generally available
Text Analytics	Key Phrase Extraction (image)	Extracts key phrases to identify the main points. For example, for the input text "The food was delicious and there were wonderful staff", the API returns the main talking points: "food" and "wonderful staff".	Preview
Text Analytics	Text Language Detection (image)	For up to 120 languages, detects which language the input text is written in and report a single language code for every document submitted on the request. The language code is paired with a score indicating the strength of the score.	Generally available
Text Analytics	Sentiment Analysis v3 (image)	Analyzes raw text for clues about positive or negative sentiment. This version of sentiment analysis returns sentiment labels (for example <i>positive</i> or <i>negative</i>) for each document and sentence within it.	Generally available
Text Analytics	Text Analytics for health	Extract and label medical information from unstructured clinical text.	Preview

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
Translator	Translator	Translate text in several languages and dialects.	Gated preview. Request access .

Speech containers

NOTE

To use Speech containers, you will need to complete an [online request form](#).

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
Speech Service API	Speech-to-text (image)	Transcribes continuous real-time speech into text.	Generally available
Speech Service API	Custom Speech-to-text (image)	Transcribes continuous real-time speech into text using a custom model.	Generally available
Speech Service API	Text-to-speech (image)	Converts text to natural-sounding speech.	Generally available
Speech Service API	Custom Text-to-speech (image)	Converts text to natural-sounding speech using a custom model.	Gated preview
Speech Service API	Neural Text-to-speech (image)	Converts text to natural-sounding speech using deep neural network technology, allowing for more natural synthesized speech.	Generally available
Speech Service API	Speech language detection (image)	Determines the language of spoken audio.	Gated preview

Vision containers

WARNING

On June 11, 2020, Microsoft announced that it will not sell facial recognition technology to police departments in the United States until strong regulation, grounded in human rights, has been enacted. As such, customers may not use facial recognition features or functionality included in Azure Services, such as Face or Video Indexer, if a customer is, or is allowing use of such services by or for, a police department in the United States.

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
---------	-----------	-------------	--------------

SERVICE	CONTAINER	DESCRIPTION	AVAILABILITY
Computer Vision	Read OCR (image)	The Read OCR container allows you to extract printed and handwritten text from images and documents with support for JPEG, PNG, BMP, PDF, and TIFF file formats. For more information, see the Read API documentation .	Gated preview. Request access .
Spatial Analysis	Spatial analysis (image)	Analyzes real-time streaming video to understand spatial relationships between people, their movement, and interactions with objects in physical environments.	Preview
Face	Face	Detects human faces in images, and identifies attributes, including face landmarks (such as noses and eyes), gender, age, and other machine-predicted facial features. In addition to detection, Face can check if two faces in the same image or different images are the same by using a confidence score, or compare faces against a database to see if a similar-looking or identical face already exists. It can also organize similar faces into groups, using shared visual traits.	Unavailable

Additionally, some containers are supported in the Cognitive Services [multi-service resource](#) offering. You can create one single Cognitive Services All-In-One resource and use the same billing key across supported services for the following services:

- Computer Vision
- Face
- LUIS
- Text Analytics

Prerequisites

You must satisfy the following prerequisites before using Azure Cognitive Services containers:

Docker Engine: You must have Docker Engine installed locally. Docker provides packages that configure the Docker environment on [macOS](#), [Linux](#), and [Windows](#). On Windows, Docker must be configured to support Linux containers. Docker containers can also be deployed directly to [Azure Kubernetes Service](#) or [Azure Container Instances](#).

Docker must be configured to allow the containers to connect with and send billing data to Azure.

Familiarity with Microsoft Container Registry and Docker: You should have a basic understanding of both Microsoft Container Registry and Docker concepts, like registries, repositories, containers, and container images, as well as knowledge of basic `docker` commands.

For a primer on Docker and container basics, see the [Docker overview](#).

Individual containers can have their own requirements, as well, including server and memory allocation requirements.

Azure Cognitive Services container security

Security should be a primary focus whenever you're developing applications. The importance of security is a metric for success. When you're architecting a software solution that includes Cognitive Services containers, it's vital to understand the limitations and capabilities available to you. For more information about network security, see [Configure Azure Cognitive Services virtual networks](#).

IMPORTANT

By default there is *no security* on the Cognitive Services container API. The reason for this is that most often the container will run as part of a pod which is protected from the outside by a network bridge. However, it is possible to enable authentication which works identically to the authentication used when accessing the [cloud-based Cognitive Services](#).

The diagram below illustrates the default and **non-secure** approach:

As an alternative and *secure* approach, consumers of Cognitive Services containers could augment a container with a front-facing component, keeping the container endpoint private. Let's consider a scenario where we use [Istio](#) as an ingress gateway. Istio supports HTTPS/TLS and client-certificate authentication. In this scenario, the Istio frontend exposes the container access, presenting the client certificate that is approved beforehand with Istio.

[Nginx](#) is another popular choice in the same category. Both Istio and Nginx act as a service mesh and offer additional features including things like load-balancing, routing, and rate-control.

Container networking

The Cognitive Services containers are required to submit metering information for billing purposes. Failure to allow list various network channels that the Cognitive Services containers rely on will prevent the container from working.

Allow list Cognitive Services domains and ports

The host should allow list **port 443** and the following domains:

- `*.cognitive.microsoft.com`
- `*.cognitiveservices.azure.com`

Disable deep packet inspection

[Deep packet inspection](#) (DPI) is a type of data processing that inspects in detail the data being sent over a computer network, and usually takes action by blocking, re-routing, or logging it accordingly.

Disable DPI on the secure channels that the Cognitive Services containers create to Microsoft servers. Failure to do so will prevent the container from functioning correctly.

Developer samples

Developer samples are available at our [GitHub repository](#).

Next steps

Learn about [container recipes](#) you can use with the Cognitive Services.

Install and explore the functionality provided by containers in Azure Cognitive Services:

- [Anomaly Detector containers](#)
- [Computer Vision containers](#)
- [Face containers](#)
- [Language Understanding \(LUIS\) containers](#)
- [Speech Service API containers](#)
- [Text Analytics containers](#)
- [Translator containers](#)

Azure Cognitive Services container image tags and release notes

9/10/2021 • 81 minutes to read • [Edit Online](#)

Azure Cognitive Services offers many container images. The container registries and corresponding repositories vary between container images. Each container image name offers multiple tags. A container image tag is a mechanism of versioning the container image. This article is intended to be used as a comprehensive reference for listing all the Cognitive Services container images and their available tags.

TIP

When using `docker pull`, pay close attention to the casing of the container registry, repository, container image name and corresponding tag - as they are **case sensitive**.

Anomaly Detector

The [Anomaly Detector](#) container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/decision` repository and is named `anomaly-detector`. The fully qualified container image name is, `mcr.microsoft.com/azure-cognitive-services/decision/anomaly-detector`.

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous versions](#)

IMAGE TAGS	NOTES
<code>latest</code>	
<code>1.1.013560003-amd64-preview</code>	

Read OCR (Optical Character Recognition)

The [Computer Vision](#) Read OCR container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services` repository and is named `read`. The fully qualified container image name is, `mcr.microsoft.com/azure-cognitive-services/vision/read`.

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous versions](#)

Release notes for `3.2`:

- Read OCR container is now generally available.

IMAGE TAGS	NOTES
<code>3.2</code>	

Language Understanding (LUIS)

The [LUIS](#) container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/language` repository and is named `luis`. The fully qualified container image name is, `mcr.microsoft.com/azure-cognitive-services/language/luis`.

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous version](#)

IMAGE TAGS	NOTES
latest	
1.1.012280003-amd64-preview	

Custom Speech-to-text

The [Custom Speech-to-text](#) container image can be found on the [mcr.microsoft.com](#) container registry syndicate. It resides within the [azure-cognitive-services/speechservices/](#) repository and is named [custom-speech-to-text](#). The fully qualified container image name is, [mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text](#). You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous version](#)

Release note for [2.14.0-amd64](#):

Regular monthly release

Note that due to the phrase lists feature, the size of this container image has increased.

IMAGE TAGS	NOTES	DIGEST
latest		sha256:c83c4691f89dfcad9c92d8c73e24b23946706936e2c8a76b1c
2.14.0-amd64		sha256:c83c4691f89dfcad9c92d8c73e24b23946706936e2c8a76b1c

Custom Text-to-speech

The [Custom Text-to-speech](#) container image can be found on the [mcr.microsoft.com](#) container registry syndicate. It resides within the [azure-cognitive-services/speechservices/](#) repository and is named [custom-text-to-speech](#). The fully qualified container image name is, [mcr.microsoft.com/azure-cognitive-services/speechservices/custom-text-to-speech](#). You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous version](#)

Release note for [1.15.0-amd64](#):

Regular monthly release

IMAGE TAGS	NOTES	DIGEST
latest		sha256:06eef68482a917a5c405b61146dc159cff6aef0bd8e13cf8f
1.15.0-amd64		sha256:06eef68482a917a5c405b61146dc159cff6aef0bd8e13cf8f

Speech-to-text

The [Speech-to-text](#) container image can be found on the [mcr.microsoft.com](#) container registry syndicate. It resides within the [azure-cognitive-services/speechservices/](#) repository and is named [speech-to-text](#). The fully qualified container image name is, [mcr.microsoft.com/azure-cognitive-services/speechservices/speech-to-text](#). You can find a full list of [tags on the MCR](#).

Since Speech-to-text v2.5.0, images are supported in the *US Government Virginia* region. Please use the *US Government Virginia* billing endpoint and API keys when using this region.

- [Latest version](#)
- [Previous version](#)

Release note for [2.14.0-amd64-⟨locale⟩](#):

Regular monthly release

Note that due to the phrase lists feature, the size of this container image has increased.

IMAGE TAGS	NOTES
latest	Container image with the en-US locale.
2.14.0-amd64-<locale>	Replace <locale> with one of the available locales, listed below. For example 2.14.0-amd64-en-us .

This container has the following locales available.

LOCALE FOR V2.14.0	NOTES	DIGEST
ar-ae	Container image with the ar-AE locale.	sha256:1f9fc0564b2ba2bdbeb5a3160e7afe6d867f3ad48cc9082505
ar-bh	Container image with the ar-BH locale.	sha256:7af3ad10e6095078ee67d0426863117e2c7c861299b3f9323b
ar-eg	Container image with the ar-EG locale.	sha256:ebe36bd9689e12ed888a327de459b3ae26b261ff3371a69692
ar-iq	Container image with the ar-IQ locale.	sha256:dd46d062ba1b7ad03b59c9dd04816a139f976db04739788316
ar-jo	Container image with the ar-JO locale.	sha256:10284e45719cc5ad1f0783807e5a3b731bb8728fc09e562067
ar-kw	Container image with the ar-KW locale.	sha256:1f9fc0564b2ba2bdbeb5a3160e7afe6d867f3ad48cc9082505
ar-lb	Container image with the ar-LB locale.	sha256:1bf6456a34e1ae5f6797741039848020b1c4b7fb68f1816533
ar-om	Container image with the ar-OM locale.	sha256:f1389c71a85ea2bc16c9f990bfcd74b0c8d92e576f0dfd5e68
ar-qa	Container image with the ar-QA locale.	sha256:1f9fc0564b2ba2bdbeb5a3160e7afe6d867f3ad48cc9082505
ar-sa	Container image with the ar-SA locale.	sha256:1f9fc0564b2ba2bdbeb5a3160e7afe6d867f3ad48cc9082505
ar-sy	Container image with the ar-SY locale.	sha256:58ffbf778fa71cacfdddcb6421d9e2514356b75797a3f0f689
bg-bg	Container image with the bg-BG locale.	sha256:37baff85bfe5d78b3858c8f7bf921af4c8d73b02fa40b731a0
ca-es	Container image with the ca-ES locale.	sha256:43abb6d9c2a85fb3f7daf757accbcb67058cd5d49d268ef043
cs-cz	Container image with the cs-CZ locale.	sha256:db9192414bc542b77670f4a281f0f2b818d23a95cba2751fa4
da-dk	Container image with the da-DK locale.	sha256:88c502880609a9cd2f35faa7d6d4a527e4e4bc80477deb2197
de-de	Container image with the de-DE locale.	sha256:3567cf9cfbc72a0280cc79b561e832c1a3a26d63ddcf41fa2d
el-gr	Container image with the el-GR locale.	sha256:484935e2a676d561c94a2e2a335f5328688e0b71a9683351ef
en-au	Container image with the en-AU locale.	sha256:b1d18f984bbb86f3cbcc9401608a31c85b9af1c0c6a6cc0f73

LOCALE FOR V2.14.0	NOTES	DIGEST
en-ca	Container image with the en-CA locale.	sha256:c04c67628e49557136860cbb64ea350aee8f09ab0664ed00fa
en-gb	Container image with the en-GB locale.	sha256:dad0620af6f3c4880914b9ca25266c6436d372127a41531f63
en-hk	Container image with the en-HK locale.	sha256:43dc4ea303c9509f562d50f470b3590beb755aab295b40d9de
en-ie	Container image with the en-IE locale.	sha256:c8370e1398b7ec2b4ca88b4d2e6d62df9e4495c25644231abe
en-in	Container image with the en-IN locale.	sha256:cdf5a3f4dc32113b9fd7e667bddc36820ac359c65b860cc8b9
en-nz	Container image with the en-NZ locale.	sha256:5d4d5811f02295831b90133aa47ca370a3243ea854ae52971c
en-ph	Container image with the en-PH locale.	sha256:051497eadedd0d9de7a36ce111ea2b82b37e2c98b3e8b06b40
en-sg	Container image with the en-SG locale.	sha256:d2b89396713a1188eef7873f479f8deca9ba2a80c43e19a7d2
en-us	Container image with the en-US locale.	sha256:5f66867b47fd9fd8d1bc67c05da8ae775f937ab208c192f419
en-za	Container image with the en-ZA locale.	sha256:e08d7cc82725de9ff7aa392a08dc484407f60c950b85525f33
es-ar	Container image with the es-AR locale.	sha256:126d73f1cb82c3e2b8995afab07a9d6470ca7b236681ef7aff
es-bo	Container image with the es-BO locale.	sha256:7cfe66dc2bcc9c7b975841954735061e0b287664083f35bb75
es-cl	Container image with the es-CL locale.	sha256:27c45610f38099a50934e214b75bbb578d3ed61fb982e49427
es-co	Container image with the es-CO locale.	sha256:b06e4a35f6ad8b195870dfa9816fb81016a9cbdd8adb3c31f3
es-cr	Container image with the es-CR locale.	sha256:8c50b7e3847f095de6bd7599c7a953b82fca9f849411cc7407
es-cu	Container image with the es-CU locale.	sha256:73434492751b1ff9e2a3f141f0c5857d5cf2c5891f1084ce98
es-do	Container image with the es-DO locale.	sha256:769db62fab433e1337a8a47db155127e887de4826d035e0534
es-ec	Container image with the es-EC locale.	sha256:e765c40a9b09fc4d9f42a9ef4bd138181c4f4826e63af17874
es-es	Container image with the es-ES locale.	sha256:b589b794035513de33740d5b5b6ecdda04fd059e9efcb11105
es-gt	Container image with the es-GT locale.	sha256:b30b4a330b7e74777e5d2575b1c2bd4dbb7a920d7165dad40b
es-hn	Container image with the es-HN locale.	sha256:8d29f96322db11e99200cf14390e225d8c332b3a6848eb6d23
es-mx	Container image with the es-MX locale.	sha256:bf6edf5bd61b57095181546579df3033e26aca7261e822b932

LOCALE FOR V2.14.0	NOTES	DIGEST
es-ni	Container image with the es-NI locale.	sha256:a02537bbfd3a4231938a321dbf9a575178018122aa4c387ebc
es-pa	Container image with the es-PA locale.	sha256:82f14c7711bcc02b82b75e3be1620b528991e4c5f185915592
es-pe	Container image with the es-PE locale.	sha256:500fa361a26d3da4e1d9c2523b232ab0d6c00ab4a15141bed6
es-pr	Container image with the es-PR locale.	sha256:e364eec54c48e2bb5c14d53356ac3942f8bfdd65d7e139c86c
es-py	Container image with the es-PY locale.	sha256:165bab6f0a5a12c58c8ec04e1b5168228f97031b80ca368529
es-sv	Container image with the es-SV locale.	sha256:42855d56f39956d456c5337b022c71352bd54bfd2d7a6a9a9f
es-us	Container image with the es-US locale.	sha256:622193b64874a3169c21862762017c4b9a46590057e388330f
es-uy	Container image with the es-UY locale.	sha256:a33ca1aeb6181f6b034ce831aaf3ca1da0df8260b08c87749e
es-ve	Container image with the es-VE locale.	sha256:6cf2522c507c77ccf7fa61ef4b54e8dc4a3f3b47b7602aa830
et-ee	Container image with the et-EE locale.	sha256:e6672fc9da94245f7d75d91b7d09e4b929d60ef0157d28885e
fi-fi	Container image with the fi-FI locale.	sha256:3731f5a8baeebf0230f3d40a3ea3e3fdf343b817122eba47a
fr-ca	Container image with the fr-CA locale.	sha256:1567e85ffe2e2660585a40b43e61fa936abf4ccdd8eb89a372
fr-fr	Container image with the fr-FR locale.	sha256:bea4884eee3382741e1d02d99a530b608d30e48a446ba52a73
ga-ie	Container image with the ga-IE locale.	sha256:f40151a6519e0678969adb2d492f240e355ac7ac9b4f57f75e
gu-in	Container image with the gu-IN locale.	sha256:ab196670c90b23d8a40448431f703483da9605f92e71f6ebc7
hi-in	Container image with the hi-IN locale.	sha256:ec26ed76cde3ae36eae3a5f0b6567ac60ea341e2f95838a845
hr-hr	Container image with the hr-HR locale.	sha256:054f188fd9be04f57858053a0a6c1146c13c2781eecd732c13
hu-hu	Container image with the hu-HU locale.	sha256:ea6bf9b3b4bfa1c4a25f1890c90a18fc309ef17afd6044cb66
it-it	Container image with the it-IT locale.	sha256:6869362837c0124964ed75ab8901fcfedec3894902018b41b40
ja-jp	Container image with the ja-JP locale.	sha256:7f8557bc112fd4ffef29df308bd10c0803cce1f1f6e79e69f8
ko-kr	Container image with the ko-KR locale.	sha256:746602c288d80c0599af276acd50f1434a331f288c95a3cf9b
lt-lt	Container image with the lt-LT locale.	sha256:b6f432bc80770f13ca537a2b49a4e89e8f979a8167f2cdaaa8

LOCALE FOR V2.14.0	NOTES	DIGEST
lv-lv	Container image with the locale.	sha256:f2126bc8886218374550f2f9a941500cf48675abdb70332990
mr-in	Container image with the locale.	sha256:82c75a0c741543c2195d271dd82bfd4400901204584d9f7b83
mt-mt	Container image with the locale.	sha256:e95f2edc5bc2090e0359c63047c4c5c879522080f8bf7cbc94
nb-no	Container image with the locale.	sha256:49c5d9b0d9de260d88deda5eeaa09979f44972610b26ddfad7
nl-nl	Container image with the locale.	sha256:3c5789fbb82c62eaa68451d391ec736ae78c298248f3afba02
pl-pl	Container image with the locale.	sha256:79a2bb077362c29495fdbee7fc6c8fd0990f080718390fb469
pt-br	Container image with the locale.	sha256:ef80359958fdf6b07461c3742c3f860c22652ccf9d123693a9
pt-pt	Container image with the locale.	sha256:000345b6a1a28cb5970c471e47963f10209acee74cd46afa7d
ro-ro	Container image with the locale.	sha256:c4a996b483f91f278f42f1696ed1d89d2e4ee8c0ed409e9d21
ru-ru	Container image with the locale.	sha256:396bab6bcfe341b53b0992fc2aaf4809767d47b8637f6fd214
sk-sk	Container image with the locale.	sha256:73624839708f88c93645a6a35278d0c2ce9a944c5992e237a4
sl-si	Container image with the locale.	sha256:86fbdc4e994507b020ff27735b741407d9f7d1e01fce2b17e6
sv-se	Container image with the locale.	sha256:c3ee782b60499ef16127b5829c36fd98c1933890752fdc4af6
ta-in	Container image with the locale.	sha256:d10ced4e32336d4b411e65066dd5486733d19b8d1f4756d606
te-in	Container image with the locale.	sha256:ebcbaec4e3a494099c7edf15b22acbd7b29347fe7cb4825d13
th-th	Container image with the locale.	sha256:bb239e9081d9cf4fffeee666346cc3c67ce83b9bda1d2e81a0
tr-tr	Container image with the locale.	sha256:68d554ae90b0a2202be5f09fb03a7d277f7d0cb0336cb0510b
zh-cn	Container image with the locale.	sha256:2ec742699abb843b91f9516cb863d66ecf5f38d5350c3c23c6
zh-hk	Container image with the locale.	sha256:34f21fed7129dbeaef6476b286e5d6741b635f034ea038b8fe
zh-tw	Container image with the locale.	sha256:f2e2dc638ac2e58177302947df30bea7448563a012deb3e4f4

Text-to-speech

The [Text-to-speech](#) container image can be found on the [mcr.microsoft.com](#) container registry syndicate. It resides within the [azure-cognitive-services/speechservices/](#) repository and is named [text-to-speech](#). The fully qualified container image name is, [mcr.microsoft.com/azure-cognitive-services/speechservices/text-to-speech](#).

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous version](#)

Release note for `1.15.0-amd64-<locale-and-voice>`:

Feature

- Upgrade to latest models.

LOCALES FOR V1.15.0	NOTES	DIGEST
<code>ar-eg-hoda</code>	Container image with the <code>ar-EG</code> locale and <code>ar-EG-Hoda</code> voice.	<code>sha256:61a154451bfef9766235f85fc7ca3698151244b04bf32cfc5a</code>
<code>ar-sa-naayf</code>	Container image with the <code>ar-SA</code> locale and <code>ar-SA-Naayf</code> voice.	<code>sha256:13cf045d959ce9362adfad114d8997e628f5e0d08e6e86a86e</code>
<code>bg-bg-ivan</code>	Container image with the <code>bg-BG</code> locale and <code>bg-BG-Ivan</code> voice.	<code>sha256:19f8c32f6723470c14c4b1731ff256853ee5c441a95a89faff</code>
<code>ca-es-herenarus</code>	Container image with the <code>ca-ES</code> locale and <code>ca-ES-HerenaRUS</code> voice.	<code>sha256:16835388036906af8b35238f05b7f17308b8fae92bf4c89199</code>
<code>cs-cz-jakub</code>	Container image with the <code>cs-CZ</code> locale and <code>cs-CZ-Jakub</code> voice.	<code>sha256:06af13ede8234c14f8a48b956017cd7858a1c0d984042a9a60</code>
<code>da-dk-hellerus</code>	Container image with the <code>da-DK</code> locale and <code>da-DK-He1leRUS</code> voice.	<code>sha256:1c6375ee05948ec9a9b2554e2423e2c2d68e7595f58d401bd2</code>
<code>de-at-michael</code>	Container image with the <code>de-AT</code> locale and <code>de-AT-Michael</code> voice.	<code>sha256:27e88c817ab91b2a4dbb5df1f88828708445993c1d657d974b</code>
<code>de-ch-karsten</code>	Container image with the <code>de-CH</code> locale and <code>de-CH-Karsten</code> voice.	<code>sha256:6b8ce6192783c1b158410a43a8fd9517cfe63c8b4a3cd0f111</code>
<code>de-de-heddarus</code>	Container image with the <code>de-DE</code> locale and <code>de-DE-Hedda</code> voice.	<code>sha256:28e1a6f0860165a4f3750b059334117240e0613ddf44d1e3c4</code>
<code>de-de-hedda</code>	Container image with the <code>de-DE</code> locale and <code>de-DE-Hedda</code> voice.	<code>sha256:28e1a6f0860165a4f3750b059334117240e0613ddf44d1e3c4</code>
<code>de-de-stefan-apollo</code>	Container image with the <code>de-DE</code> locale and <code>de-DE-Stefan-Apollo</code> voice.	<code>sha256:3730dfefb60f3a74df523e790738595b29e3dc694a16506a6d</code>
<code>el-gr-stefanos</code>	Container image with the <code>el-GR</code> locale and <code>el-GR-Stefanos</code> voice.	<code>sha256:dfce427d7c08bd26d38513fd4b5c85662fe4feeddefa75e124</code>
<code>en-au-catherine</code>	Container image with the <code>en-AU</code> locale and <code>en-AU-Catherine</code> voice.	<code>sha256:71a9a64adc48044e2ce81119bc118056a906db284311fc3761</code>
<code>en-au-hayleyrus</code>	Container image with the <code>en-AU</code> locale and <code>en-AU-HayleyRUS</code> voice.	<code>sha256:a42624ebf51afff052a0ed8518f474855d70b4a9245cd8e814</code>
<code>en-ca-heatherrus</code>	Container image with the <code>en-CA</code> locale and <code>en-CA-HeatherRUS</code> voice.	<code>sha256:b5f745bbf9de83f57ac4e6e2760049e10a8eaae362018c4d5a</code>
<code>en-ca-linda</code>	Container image with the <code>en-CA</code> locale and <code>en-CA-Linda</code> voice.	<code>sha256:6638a92b495c76ca16331c652b123fa52163242cfbd8f8298c</code>
<code>en-gb-george-apollo</code>	Container image with the <code>en-GB</code> locale and <code>en-GB-George-Apollo</code> voice.	<code>sha256:748c7042dfa3107f387c34ee29269fc2bd96f27af525f2dc7b</code>

LOCALES FOR V1.15.0	NOTES	DIGEST
en-gb-hazelrus	Container image with the en-GB locale and en-GB-HazeIRUS voice.	sha256:8a439470579f95645bf5831ee5f0643b872d6dbd7426cf572
en-gb-susan-apollo	Container image with the en-GB locale and en-GB-Susan-Apollo voice.	sha256:ad9c5741a2b19fc936ec740fa0bbd2700e09e361d7ce9df0bb
en-ie-sean	Container image with the en-IE locale and en-IE-Sean voice.	sha256:45d1d07f67c81b11f7b239f0e46bd22964d0e795b01e72e58
en-in-heera-apollo	Container image with the en-IN locale and en-IN-Heera-Apollo voice.	sha256:f4cd71fac26b0d1f0693ce91535a0fd14ac90e323c6f9d8239
en-in-priarus	Container image with the en-IN locale and en-IN-PriyaRUS voice.	sha256:5a228190a5fe62aaa5f8443ab4041d2a7af381e30236333a44
en-in-ravi-apollo	Container image with the en-IN locale and en-IN-Ravi-Apollo voice.	sha256:3f477ad93ff643f90adf268775c9b8cd8fb3b2cadf347b3663
en-us-aria24krus	Container image with the en-US locale and en-US-Aria24kRUS voice.	sha256:d4ece3a336171cd46068831b3203460c86e5cd7f053b56a8a7
en-us-ariarus	Container image with the en-US locale and en-US-AriaRUS voice.	sha256:d4ece3a336171cd46068831b3203460c86e5cd7f053b56a8a7
en-us-benjaminrus	Container image with the en-US locale and en-US-BenjaminRUS voice.	sha256:f668eb749ee51c01bcadf0df8e1a0b6fc000fb64a93bd12458
en-us-guy24krus	Container image with the en-US locale and en-US-Guy24kRUS voice.	sha256:50900ece25a078bc4e0a0fec845cc9516e975a7b90621e4fde
en-us-zirarus	Container image with the en-US locale and en-US-ZiraRUS voice.	sha256:772bdc81780a05f7400a88b1cddcef6ef0be153ce873df2391
es-es-helenarus	Container image with the es-ES locale and es-ES-HelenaRUS voice.	sha256:ab25fc60c8a8e095fcf63fe953bd2acf1f0569e6aaaf02e90d
es-es-laura-apollo	Container image with the es-ES locale and es-ES-Laura-Apollo voice.	sha256:11c144693d62b28e1444378638295e801c07888fd6ff70903b
es-es-pablo-apollo	Container image with the es-ES locale and es-ES-Pablo-Apollo voice.	sha256:56db18adc44ee4412fd64f2d9303960525627ecf9b6cd6c201
es-mx-hildarus	Container image with the es-MX locale and es-MX-HildaRUS voice.	sha256:80ad68c2ca58380ca3d88e509ad32a21f70ecc41fab701f629
es-mx-raul-apollo	Container image with the es-MX locale and es-MX-Raul-Apollo voice.	sha256:fd51dcc46ac5c81949d7ff3ceeacf7144fb6e516089fff645
fi-fi-heidirus	Container image with the fi-FI locale and fi-FI-HeidiRUS voice.	sha256:0ba17a99d35d4963110316d6bb7742082d0362f23490790bb8
fr-ca-caroline	Container image with the fr-CA locale and fr-CA-Caroline voice.	sha256:67304f764165b34c051104d8ef51202dcbaafcf3b88d5568ac
fr-ca-harmonierus	Container image with the fr-CA locale and fr-CA-HarmonieRUS voice.	sha256:9b428ec672b60e8e6f9642cc5f23741e84df5e68477bb5fd4f
fr-ch-guillaume	Container image with the fr-CH locale and fr-CH-Guillaume voice.	sha256:d3fedebf0321f9135335be369fec84be42a3653977f0834c6b

LOCALES FOR V1.15.0	NOTES	DIGEST
fr-fr-hortenserus	Container image with the fr-FR locale and fr-FR-HortenseRUS voice.	sha256:2d33762773d299ffd37a3103b3c32ce8d1b7f3f107daf6514b
fr-fr-julie-apollo	Container image with the fr-FR locale and fr-FR-Julie-Apollo voice.	sha256:54f762a2d68cc8a33049b18085fac44f5bad1750a1d85347d5
fr-fr-paul-apollo	Container image with the fr-FR locale and fr-FR-Paul-Apollo voice.	sha256:7d3e4a75495be2c503f55596d39a5bdfe75538b453a5fb7edb
he-il-asaf	Container image with the he-IL locale and he-IL-Asaf voice.	sha256:729bd1c6128ee059e89d04e2e2fd5cd925e59550014b901bf5
hi-in-hemant	Container image with the hi-IN locale and hi-IN-Hemant voice.	sha256:9ed035183c7c2a0debe44dc6bae67d097334b0be8f5bec643b
hi-in-kalpna-apollo	Container image with the hi-IN locale and hi-IN-Kalpna-Apollo voice.	sha256:f043d625788fd61bba7454a64502572a2e4fed310775c371c7
hi-in-kalpna	Container image with the hi-IN locale and hi-IN-Kalpna voice.	sha256:f043d625788fd61bba7454a64502572a2e4fed310775c371c7
hr-hr-matej	Container image with the hr-HR locale and hr-HR-Matej voice.	sha256:a320245b93af76b125386f4566383ec6e13a21c951a8468d1f
hu-hu-szabolcs	Container image with the hu-HU locale and hu-HU-Szabolcs voice.	sha256:94d86ae188bb08df0192de4221404132d631cae6aa6d4fc4bf
id-id-andika	Container image with the id-ID locale and id-ID-Andika voice.	sha256:8fee6f6d8552fae0ce050765ea5c842497a699f5feb700f705
it-it-cosimo-apollo	Container image with the it-IT locale and it-IT-Cosimo-Apollo voice.	sha256:1d99f0f538e0d61b527fbc77f9281e0f932bac7e6ba513b13e
it-it-luciarus	Container image with the it-IT locale and it-IT-LuciaRUS voice.	sha256:99db33a668e298c58be1c50b9d4b84aeb0949f0334187b0216
ja-jp-ayumi-apollo	Container image with the ja-JP locale and ja-JP-Ayumi-Apollo voice.	sha256:50d1e986d318692917968654008466fc3cca4911c3bcd36af6
ja-jp-harukarus	Container image with the ja-JP locale and ja-JP-HarukaRUS voice.	sha256:7736a87dcf3595056bb558c6cb38094d1732bb164406a99d87
ja-jp-ichiro-apollo	Container image with the ja-JP locale and ja-JP-Ichiro-Apollo voice.	sha256:6ce704a51150e0ee092f2197ba7cf4bcbf8473e5cd56a9a083
ko-kr-heamirus	Container image with the ko-KR locale and ko-KR-HeamiRUS voice.	sha256:ec5d75470dbae50cb5bc2f93ed642e404446b099cb2302499b3
ms-my-rizwan	Container image with the ms-MY locale and ms-MY-Rizwan voice.	sha256:e572b62f0b4153382318266dcd59d6e92daf8acc6f323e461d
nb-no-huldarus	Container image with the nb-NO locale and nb-NO-HuldaRUS voice.	sha256:691ef2ead95a0d4703cd6064bac9355e86a361fcffe5ad36a7
n1-nl-hannarus	Container image with the n1-NL locale and n1-NL-HannaRUS voice.	sha256:f52a717d4d8b7db39b18c9a9e448e2e6d6e19600093518002a
pl-pl-paulinarus	Container image with the pl-PL locale and pl-PL-PaulinaRUS voice.	sha256:1927ff28b40b7c37ee1b8d5f4efb2fd7d905affd35c2798394

LOCALES FOR V1.15.0	NOTES	DIGEST
pt-br-daniel-apollo	Container image with the pt-BR locale and pt-BR-Daniel-Apollo voice.	sha256:ebce3b7b51fb28fce4c446fbbf3607f4307b1cec3f9fa7abdd
pt-br-heloisarus	Container image with the pt-BR locale and pt-BR-HeloisaRUS voice.	sha256:195e719735768fdf6ea2f1fc829a40cae5af4d35b62e52d1c7
pt-pt-heliarus	Container image with the pt-PT locale and pt-PT-HeliaRUS voice.	sha256:f0ea6ec57615a55b13f491e6f96b3cc0e29092f63a981fd297
ro-ro-andrei	Container image with the ro-RO locale and ro-RO-Andrei voice.	sha256:deee319f2b6d8145f3ed567cfcdfa2ca718cd1b408f8d9fbf1
ru-ru-ekaterinarus	Container image with the ru-RU locale and ru-RU-EkaterinaRUS voice.	sha256:d0005c1363e197c0f85180a07d650655b473117de12170a631
ru-ru-irina-apollo	Container image with the ru-RU locale and ru-RU-Irina-Apollo voice.	sha256:53731218ed6e2bed227c25a2a2e1d528a19dbc078e2af55aa
ru-ru-pavel-apollo	Container image with the ru-RU locale and ru-RU-Pavel-Apollo voice.	sha256:81b2a56f72460a780466337136729b011ef1eac4689b1ec9ed
sk-sk-filip	Container image with the sk-SK locale and sk-SK-Filip voice.	sha256:e3d44c7ac30b1b9b186eaf1761ccadd89b17fcb4d4f63e1dab
sl-si-lado	Container image with the sl-SI locale and sl-SI-Lado voice.	sha256:8ecb2b3d0c60f4c88522090d24e55d84a6132b751d71b41a3d
sv-se-hedvigrus	Container image with the sv-SE locale and sv-SE-HedvigRUS voice.	sha256:5b61e4ebe696e7cee23403ec4aed299cbf4874c0eeb5a163a8
ta-in-valluvar	Container image with the ta-IN locale and ta-IN-Valluvar voice.	sha256:adf3c421feb6385ba3acb241750d909a42f41d09b5ebbc66db
te-in-chitra	Container image with the te-IN locale and te-IN-Chitra voice.	sha256:e9fc71faf37ca890a82e29bec29b6cfd94299e2d78aaed8c98
th-th-pattara	Container image with the th-TH locale and th-TH-Pattara voice.	sha256:b02cc2b23a7d1ec2f3f2d3917a51316fb009597d5d9606b5f1
tr-tr-sedarus	Container image with the tr-TR locale and tr-TR-SedaRUS voice.	sha256:961773f7f544cc0643590f4ed44d40f12e3fa23e44834afd19
vi-vn-an	Container image with the vi-VN locale and vi-VN-An voice.	sha256:f1fdda1c758a4361d2fb594f02d47be7cf88571e5a51fb845b
zh-cn-huihirus	Container image with the zh-CN locale and zh-CN-HuihuiRUS voice.	sha256:183125591097ab157bf57088fae3a8ab0af4472cabd3d1c7bd
zh-cn-kangkang-apollo	Container image with the zh-CN locale and zh-CN-Kangkang-Apollo voice.	sha256:72a77502eb91ebf407bfbbf068b442e1c281da33814e042b02
zh-cn-yaoyao-apollo	Container image with the zh-CN locale and zh-CN-Yaoyao-Apollo voice.	sha256:9a202b3172def1a35553d7adf5298af71b44dde10ee261752b
zh-hk-danny-apollo	Container image with the zh-HK locale and zh-HK-Danny-Apollo voice.	sha256:9bbba04f272231084b9c87d668e5a71ab7f61d464eeaaab50d4

LOCALES FOR V1.15.0	NOTES	DIGEST
zh-hk-tracy-apollo	Container image with the zh-HK locale and zh-HK-Tracy-Apollo voice.	sha256:048d335ea90493fde6ccce8715925e472fddb405c3208bba5a
zh-hk-tracyrus	Container image with the zh-HK locale and zh-HK-TracyRUS voice.	sha256:048d335ea90493fde6ccce8715925e472fddb405c3208bba5a
zh-tw-hanhanrus	Container image with the zh-TW locale and zh-TW-HanHanRUS voice.	sha256:fe30bb665c416d0a6cc3547425e1736802d7527eebdd919ee4
zh-tw-yating-apollo	Container image with the zh-TW locale and zh-TW-Yating-Apollo voice.	sha256:6308d4e4302d02bbb4043ec6cceb6e574b7e156a5d774bef09
zh-tw-zhiwei-apollo	Container image with the zh-TW locale and zh-TW-Zhiwei-Apollo voice.	sha256:e40dda8b5e9313a5962c260c1e9eb410b19e60fa74062ad069

Neural Text-to-speech

The [Neural Text-to-speech](#) container image can be found on the [mcr.microsoft.com](#) container registry syndicate. It resides within the [azure-cognitive-services/speechservices/](#) repository and is named [neural-text-to-speech](#). The fully qualified container image name is, [mcr.microsoft.com/azure-cognitive-services/speechservices/neural-text-to-speech](#).

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous version](#)

Release notes for [v1.9.0](#):

- Add 1 new [en-GB](#) and 9 (4 are preview) new [zh-CN](#) voices.

IMAGE TAGS	NOTES
latest	Container image with the en-US locale and en-US-AriaNeural voice.
1.9.0-amd64- <locale-and-voice>	Replace <locale> with one of the available locales, listed below. For example 1.9.0-amd64-en-us-arianeural .
V1.9.0 LOCALES AND VOICES	NOTES
de-de-conradneural	Container image with the de-DE locale and de-DE-ConradNeural voice.
de-de-katjneural	Container image with the de-DE locale and de-DE-KatjaNeural voice.
en-au-natashaneural	Container image with the en-AU locale and en-AU-NatashaNeural voice.
en-au-williamneural	Container image with the en-AU locale and en-AU-WilliamNeural voice.
en-ca-claraneural	Container image with the en-CA locale and en-CA-ClaraNeural voice.
en-ca-liamneural	Container image with the en-CA locale and en-CA-LiamNeural voice.
en-gb-libbyneural	Container image with the en-GB locale and en-GB-LibbyNeural voice.

V1.9.0 LOCALES AND VOICES	NOTES
<code>en-gb-ryanneural</code>	Container image with the <code>en-GB</code> locale and <code>en-GB-RyanNeural</code> voice.
<code>en-gb-sonianeural</code>	Container image with the <code>en-GB</code> locale and <code>en-GB-SoniaNeural</code> voice.
<code>en-us-arianeural</code>	Container image with the <code>en-US</code> locale and <code>en-US-AriaNeural</code> voice.
<code>en-us-guyneural</code>	Container image with the <code>en-US</code> locale and <code>en-US-GuyNeural</code> voice.
<code>en-us-jennyneural</code>	Container image with the <code>en-US</code> locale and <code>en-US-JennyNeural</code> voice.
<code>es-es-alvaroneural</code>	Container image with the <code>es-ES</code> locale and <code>es-ES-AlvaroNeural</code> voice.
<code>es-es-elviraneural</code>	Container image with the <code>es-ES</code> locale and <code>es-ES-ElviraNeural</code> voice.
<code>es-mx-dalianeural</code>	Container image with the <code>es-MX</code> locale and <code>es-MX-DaliaNeural</code> voice.
<code>es-mx-jorgeneural</code>	Container image with the <code>es-MX</code> locale and <code>es-MX-JorgeNeural</code> voice.
<code>fr-ca-antoineural</code>	Container image with the <code>fr-CA</code> locale and <code>fr-CA-AntoineNeural</code> voice.
<code>fr-ca-jeanneural</code>	Container image with the <code>fr-CA</code> locale and <code>fr-CA-JeanNeural</code> voice.
<code>fr-ca-sylvianeural</code>	Container image with the <code>fr-CA</code> locale and <code>fr-CA-SylvieNeural</code> voice.
<code>fr-fr-deniseneural</code>	Container image with the <code>fr-FR</code> locale and <code>fr-FR-DeniseNeural</code> voice.
<code>fr-fr-henrineural</code>	Container image with the <code>fr-FR</code> locale and <code>fr-FR-HenriNeural</code> voice.
<code>hi-in-madhurneural</code>	Container image with the <code>hi-IN</code> locale and <code>hi-IN-MadhurNeural</code> voice.
<code>hi-in-swaraneural</code>	Container image with the <code>hi-IN</code> locale and <code>hi-IN-Swaraneural</code> voice.
<code>it-it-diegoneural</code>	Container image with the <code>it-IT</code> locale and <code>it-IT-DiegoNeural</code> voice.
<code>it-it-elsaneural</code>	Container image with the <code>it-IT</code> locale and <code>it-IT-ElsaNeural</code> voice.
<code>it-it-isabellaneural</code>	Container image with the <code>it-IT</code> locale and <code>it-IT-IsabellaNeural</code> voice.
<code>ja-jp-keitaneural</code>	Container image with the <code>ja-JP</code> locale and <code>ja-JP-KeitaNeural</code> voice.
<code>ja-jp-nanamineural</code>	Container image with the <code>ja-JP</code> locale and <code>ja-JP-NanamiNeural</code> voice.

V1.9.0 LOCALES AND VOICES	NOTES
ko-kr-injooneural	Container image with the ko-KR locale and ko-KR-InJoonNeural voice.
ko-kr-sunhineural	Container image with the ko-KR locale and ko-KR-SunHiNeural voice.
pt-br-antonioneural	Container image with the pt-BR locale and pt-BR-AntonioNeural voice.
pt-br-franciscaneural	Container image with the pt-BR locale and pt-BR-FranciscaNeural voice.
tr-tr-ahmetneural	Container image with the tr-TR locale and tr-TR-AhmetNeural voice.
tr-tr-emelneural	Container image with the tr-TR locale and tr-TR-EmelNeural voice.
zh-cn-xiaoxiaoneural	Container image with the zh-CN locale and zh-CN-XiaoxiaoNeural voice.
zh-cn-xiaoyouneural	Container image with the zh-CN locale and zh-CN-XiaoYouNeural voice.
zh-cn-yunyangneural	Container image with the zh-CN locale and zh-CN-YunYangNeural voice.
zh-cn-yunyeneural	Container image with the zh-CN locale and zh-CN-YunYeNeural voice.
zh-cn-xiaochenneural-preview	Container image with the zh-CN locale and zh-CN-XiaoChenNeural voice.
zh-cn-xiaohanneural	Container image with the zh-CN locale and zh-CN-XiaoHanNeural voice.
zh-cn-xiaomoneural	Container image with the zh-CN locale and zh-CN-XiaoMoNeural voice.
zh-cn-xiaoqiuneural-preview	Container image with the zh-CN locale and zh-CN-XiaoQiuNeural voice.
zh-cn-xiaoruineural	Container image with the zh-CN locale and zh-CN-XiaoRuiNeural voice.
zh-cn-xiaoshuangneural-preview	Container image with the zh-CN locale and zh-CN-XiaoShuangNeural voice.
zh-cn-xiaoxuanneural	Container image with the zh-CN locale and zh-CN-XiaoXuanNeural voice.
zh-cn-xiaoyanneural-preview	Container image with the zh-CN locale and zh-CN-XiaoYanNeural voice.
zh-cn-yunxineural	Container image with the zh-CN locale and zh-CN-YunXiNeural voice.

Speech language detection

The [Speech language detection](#) container image can be found on the [mcr.microsoft.com](#) container registry syndicate. It resides within the [azure-cognitive-services/speechservices/](#) repository and is named [language-detection](#). The fully qualified container image name is, [mcr.microsoft.com/azure-cognitive-services/speechservices/language-detection](#).

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous versions](#)
- Release notes for version `1.3.0`:
 - Support for standalone language IDs with `SingleLanguage` and continuous mode.

IMAGE TAGS	NOTES
<code>latest</code>	
<code>1.3.0-amd64-preview</code>	

Key Phrase Extraction

container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/textanalytics/` repository and is named `keyphrase`. The fully qualified container image name is, `mcr.microsoft.com/azure-cognitive-services/textanalytics/keyphrase`.

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous versions](#)

IMAGE TAGS	NOTES
<code>latest</code>	
<code>1.1.013570001-amd64</code>	

Text language detection

The [Language Detection](#) container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/textanalytics/` repository and is named `language`. The fully qualified container image name is, `mcr.microsoft.com/azure-cognitive-services/textanalytics/language`.

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest versions](#)
- [Previous versions](#)

IMAGE TAGS	NOTES
<code>latest</code>	
<code>1.1.013570001-amd64</code>	

Sentiment analysis

The [Sentiment Analysis](#) container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/textanalytics/` repository and is named `sentiment`. The fully qualified container image name is, `mcr.microsoft.com/azure-cognitive-services/textanalytics/sentiment`.

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

IMAGE TAGS	NOTES
<code>latest</code>	
<code>3.0-en</code>	Sentiment Analysis v3 (English)
<code>3.0-es</code>	Sentiment Analysis v3 (Spanish)

IMAGE TAGS	NOTES
<code>3.0-fr</code>	Sentiment Analysis v3 (French)
<code>3.0-it</code>	Sentiment Analysis v3 (Italian)
<code>3.0-de</code>	Sentiment Analysis v3 (German)
<code>3.0-zh</code>	Sentiment Analysis v3 (Chinese - simplified)
<code>3.0-zht</code>	Sentiment Analysis v3 (Chinese - traditional)
<code>3.0-ja</code>	Sentiment Analysis v3 (Japanese)
<code>3.0-pt</code>	Sentiment Analysis v3 (Portuguese)
<code>3.0-nl</code>	Sentiment Analysis v3 (Dutch)
<code>2.1</code>	Sentiment Analysis v2

Text Analytics for health

The [Text Analytics for health](#) container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/textanalytics/` repository and is named `healthcare`. The fully qualified container image name is

```
mcr.microsoft.com/azure-cognitive-services/textanalytics/healthcare
```

This container image has the following tags available. You can also find a full list of [tags on the MCR](#).

- [Latest version](#)
- [Previous versions](#)

Release notes for `3.0.017010001-onprem-amd64`:

- You can now use the [Text Analytics for health container with the client library](#)

IMAGE TAGS	NOTES
<code>latest</code>	
<code>3.0.017010001-onprem-amd64</code>	

Translator

The [Translator](#) container image can be found on the `mcr.microsoft.com` container registry syndicate. It resides within the `azure-cognitive-services/translator` repository and is named `text-translation`. The fully qualified container image name is `mcr.microsoft.com/azure-cognitive-services/translator/text-translation`.

This container image has the following tags available.

IMAGE TAGS	NOTES
<code>latest</code>	

Create containers for reuse

3/30/2021 • 4 minutes to read • [Edit Online](#)

Use these container recipes to create Cognitive Services Containers that can be reused. Containers can be built with some or all configuration settings so that they are *not* needed when the container is started.

Once you have this new layer of container (with settings), and you have tested it locally, you can store the container in a container registry. When the container starts, it will only need those settings that are not currently stored in the container. The private registry container provides configuration space for you to pass those settings in.

Docker run syntax

Any `docker run` examples in this document assume a Windows console with a `^` line continuation character. Consider the following for your own use:

- Do not change the order of the arguments unless you are very familiar with docker containers.
- If you are using an operating system other than Windows, or a console other than Windows console, use the correct console/terminal, folder syntax for mounts, and line continuation character for your console and system. Because the Cognitive Services container is a Linux operating system, the target mount uses a Linux-style folder syntax.
- `docker run` examples use the directory off the `c:` drive to avoid any permission conflicts on Windows. If you need to use a specific directory as the input directory, you may need to grant the docker service permission.

Store no configuration settings in image

The example `docker run` commands for each service do not store any configuration settings in the container. When you start the container from a console or registry service, those configuration settings need to pass in. The private registry container provides configuration space for you to pass those settings in.

Reuse recipe: store all configuration settings with container

In order to store all configuration settings, create a `Dockerfile` with those settings.

Issues with this approach:

- The new container has a separate name and tag from the original container.
- In order to change these settings, you will have to change the values of the Dockerfile, rebuild the image, and republish to your registry.
- If someone gets access to your container registry or your local host, they can run the container and use the Cognitive Services endpoints.
- If your Cognitive Service doesn't require input mounts, don't add the `COPY` lines to your Dockerfile.

Create Dockerfile, pulling from the existing Cognitive Services container you want to use, then use docker commands in the Dockerfile to set or pull in information the container needs.

This example:

- Sets the billing endpoint, `{BILLING_ENDPOINT}` from the host's environment key using `ENV`.
- Sets the billing API-key, `{ENDPOINT_KEY}` from the host's environment key using `ENV`.

Reuse recipe: store billing settings with container

This example shows how to build the Text Analytics' sentiment container from a Dockerfile.

```
FROM mcr.microsoft.com/azure-cognitive-services/sentiment:latest
ENV billing={BILLING_ENDPOINT}
ENV apikey={ENDPOINT_KEY}
ENV EULA=accept
```

Build and run the container [locally](#) or from your [private registry container](#) as needed.

Reuse recipe: store billing and mount settings with container

This example shows how to use Language Understanding, saving billing and models from the Dockerfile.

- Copies the Language Understanding (LUIS) model file from the host's file system using `COPY`.
- The LUIS container supports more than one model. If all models are stored in the same folder, you all need one `COPY` statement.
- Run the docker file from the relative parent of the model input directory. For the following example, run the `docker build` and `docker run` commands from the relative parent of `/input`. The first `/input` on the `COPY` command is the host computer's directory. The second `/input` is the container's directory.

```
FROM <container-registry>/<cognitive-service-container-name>:<tag>
ENV billing={BILLING_ENDPOINT}
ENV apikey={ENDPOINT_KEY}
ENV EULA=accept
COPY /input /input
```

Build and run the container [locally](#) or from your [private registry container](#) as needed.

How to use container on your local host

To build the Docker file, replace `<your-image-name>` with the new name of the image, then use:

```
docker build -t <your-image-name> .
```

To run the image, and remove it when the container stops (`--rm`):

```
docker run --rm <your-image-name>
```

How to add container to private registry

Follow these steps to use the Dockerfile and place the new image in your private container registry.

1. Create a `Dockerfile` with the text from reuse recipe. A `Dockerfile` doesn't have an extension.
2. Replace any values in the angle brackets with your own values.
3. Build the file into an image at the command line or terminal, using the following command. Replace the values in the angle brackets, `<>`, with your own container name and tag.

The tag option, `-t`, is a way to add information about what you have changed for the container. For example, a container name of `modified-LUIS` indicates the original container has been layered. A tag name of `with-billing-and-model` indicates how the Language Understanding (LUIS) container has been modified.

```
docker build -t <your-new-container-name>:<your-new-tag-name> .
```

4. Sign in to Azure CLI from a console. This command opens a browser and requires authentication. Once authenticated, you can close the browser and continue working in the console.

```
az login
```

5. Sign in to your private registry with Azure CLI from a console.

Replace the values in the angle brackets, `<my-registry>`, with your own registry name.

```
az acr login --name <my-registry>
```

You can also sign in with docker login if you are assigned a service principal.

```
docker login <my-registry>.azurecr.io
```

6. Tag the container with the private registry location. Replace the values in the angle brackets, `<my-registry>`, with your own registry name.

```
docker tag <your-new-container-name>:<your-new-tag-name> <my-registry>.azurecr.io/<your-new-container-name-in-registry>:<your-new-tag-name>
```

If you don't use a tag name, `latest` is implied.

7. Push the new image to your private container registry. When you view your private container registry, the container name used in the following CLI command will be the name of the repository.

```
docker push <my-registry>.azurecr.io/<your-new-container-name-in-registry>:<your-new-tag-name>
```

Next steps

[Create and use Azure Container Instance](#)

Deploy and run container on Azure Container Instance

4/29/2021 • 7 minutes to read • [Edit Online](#)

With the following steps, scale Azure Cognitive Services applications in the cloud easily with Azure [Container Instances](#). Containerization helps you focus on building your applications instead of managing the infrastructure. For more information on using containers, see [features and benefits](#).

Prerequisites

The recipe works with any Cognitive Services container. The Cognitive Service resource must be created before using the recipe. Each Cognitive Service that supports containers has a "How to install" article for installing and configuring the service for a container. Some services require a file or set of files as input for the container, it is important that you understand and have used the container successfully before using this solution.

- An Azure resource for the Azure Cognitive Service you're using.
- Cognitive Service **endpoint URL** - review your specific service's "How to install" for the container, to find where the endpoint URL is from within the Azure portal, and what a correct example of the URL looks like. The exact format can change from service to service.
- Cognitive Service **key** - the keys are on the **Keys** page for the Azure resource. You only need one of the two keys. The key is a string of 32 alpha-numeric characters.
- A single Cognitive Services Container on your local host (your computer). Make sure you can:
 - Pull down the image with a `docker pull` command.
 - Run the local container successfully with all required configuration settings with a `docker run` command.
 - Call the container's endpoint, getting a response of HTTP 2xx and a JSON response back.

All variables in angle brackets, `<>`, need to be replaced with your own values. This replacement includes the angle brackets.

IMPORTANT

The LUIS container requires a `.gz` model file that is pulled in at runtime. The container must be able to access this model file via a volume mount from the container instance. To upload a model file, follow these steps:

1. [Create an Azure file share](#). Take note of the Azure Storage account name, key, and file share name as you'll need them later.
2. [export your LUIS model \(packaged app\) from the LUIS portal](#).
3. In the Azure portal, navigate to the **Overview** page of your storage account resource, and select **File shares**.
4. Select the file share name that you recently created, then select **Upload**. Then upload your packaged app.

- [Azure portal](#)
- [CLI](#)

Create an Azure Container Instance resource using the Azure portal

1. Go to the [Create](#) page for Container Instances.
2. On the **Basics** tab, enter the following details:

SETTING	VALUE
Subscription	Select your subscription.
Resource group	Select the available resource group or create a new one such as <code>cognitive-services</code> .
Container name	Enter a name such as <code>cognitive-container-instance</code> . The name must be in lower caps.
Location	Select a region for deployment.
Image type	If your container image is stored in a container registry that doesn't require credentials, choose <code>Public</code> . If accessing your container image requires credentials, choose <code>Private</code> . Refer to container repositories and images for details on whether or not the container image is <code>Public</code> or <code>Private</code> ("Public Preview").
Image name	<p>Enter the Cognitive Services container location. The location is what's used as an argument to the <code>docker pull</code> command. Refer to the container repositories and images for the available image names and their corresponding repository.</p> <p>The image name must be fully qualified specifying three parts. First, the container registry, then the repository, finally the image name:</p> <pre><container-registry>/<repository>/<image-name></pre> <p>Here is an example,</p> <pre>mcr.microsoft.com/azure-cognitive-services/keyphrase</pre> <p>would represent the Key Phrase Extraction image in the Microsoft Container Registry under the Azure Cognitive Services repository. Another example is,</p> <pre>containerpreview.azurecr.io/microsoft/cognitive-services-speech-to-text</pre> <p>which would represent the Speech to Text image in the Microsoft repository of the Container Preview container registry.</p>
OS type	<code>Linux</code>
Size	<p>Change size to the suggested recommendations for your specific Cognitive Service container:</p> <p>2 CPU cores 4 GB</p>

3. On the **Networking** tab, enter the following details:

SETTING	VALUE
---------	-------

SETTING	VALUE
Ports	Set the TCP port to <input type="text" value="5000"/> . Exposes the container on port 5000.

- On the **Advanced** tab, enter the required **Environment Variables** for the container billing settings of the Azure Container Instance resource:

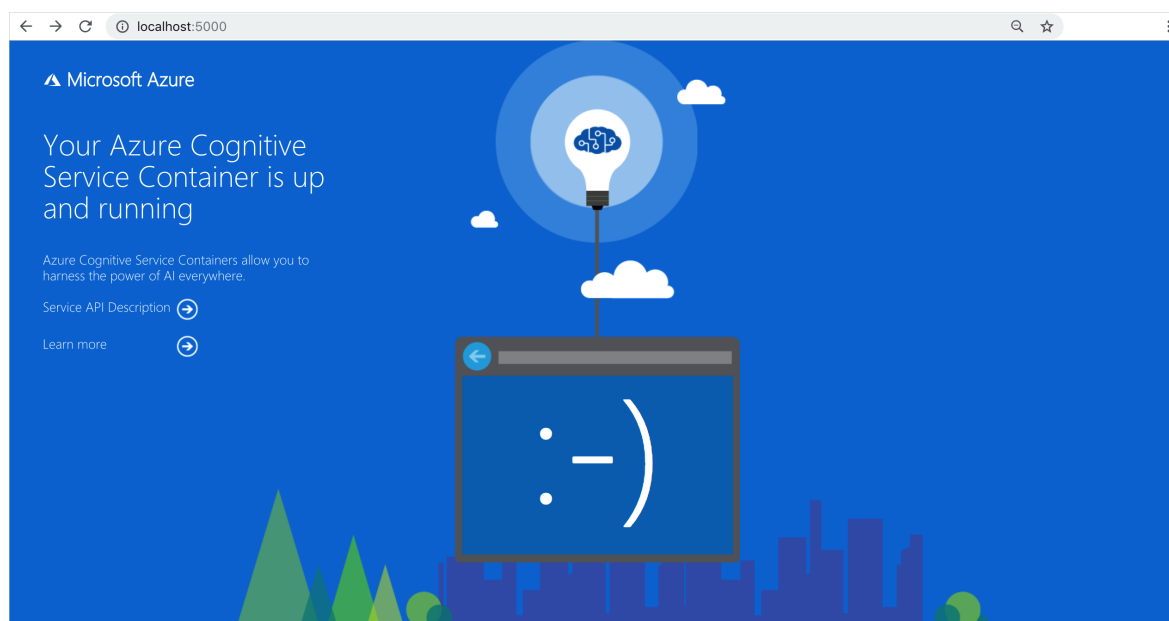
KEY	VALUE
<input type="text" value="ApiKey"/>	Copied from the Keys and endpoint page of the resource. It is a 32 alphanumeric-character string with no spaces or dashes, <input type="text" value="xx"/> .
<input type="text" value="Billing"/>	Your endpoint URL copied from the Keys and endpoint page of the resource.
<input type="text" value="Eula"/>	<input type="text" value="accept"/>

- Click **Review and Create**
- After validation passes, click **Create** to finish the creation process
- When the resource is successfully deployed, it's ready

Use the Container Instance

- [Azure portal](#)
- [CLI](#)

- Select the **Overview** and copy the IP address. It will be a numeric IP address such as .
- Open a new browser tab and use the IP address, for example, . You will see the container's home page, letting you know the container is running.



- Select **Service API Description** to view the swagger page for the container.
- Select any of the **POST** APIs and select **Try it out**. The parameters are displayed including the input. Fill

in the parameters.

5. Select **Execute** to send the request to your Container Instance.

You have successfully created and used Cognitive Services containers in Azure Container Instance.

Deploy the Text Analytics language detection container to Azure Kubernetes Service

3/5/2021 • 10 minutes to read • [Edit Online](#)

Learn how to deploy the language detection container. This procedure shows you how create the local Docker containers, push the containers to your own private container registry, run the container in a Kubernetes cluster, and test it in a web browser.

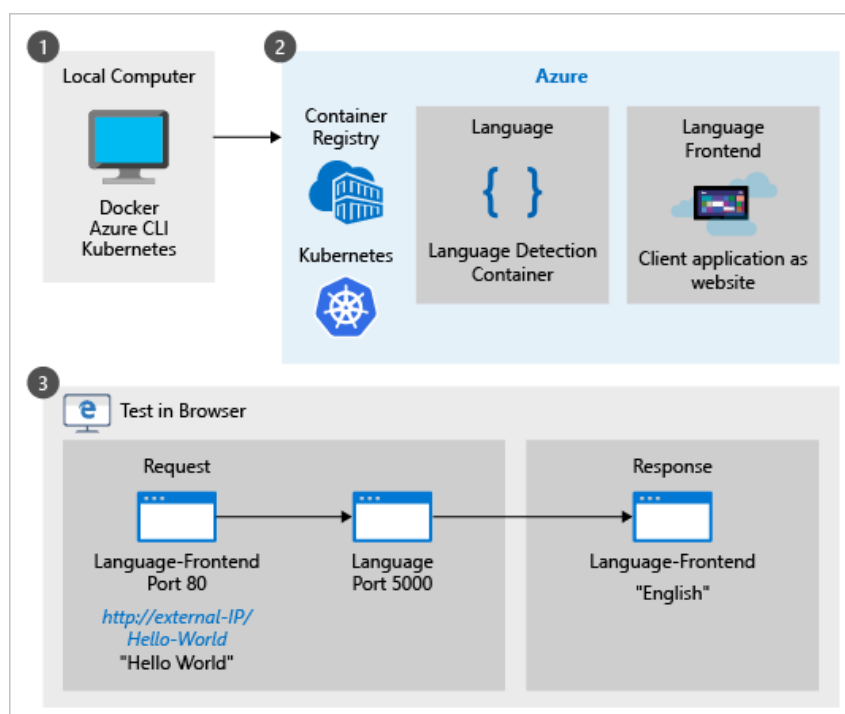
Prerequisites

This procedure requires several tools that must be installed and run locally. Do not use Azure Cloud shell.

- Use an Azure subscription. If you don't have an Azure subscription, create a [free account](#) before you begin.
- [Git](#) for your operating system so you can clone the [sample](#) used in this procedure.
- [Azure CLI](#).
- [Docker engine](#) and validate that the Docker CLI works in a console window.
- [kubectl](#).
- An Azure resource with the correct pricing tier. Not all pricing tiers work with this container:
 - **Text Analytics** resource with F0 or Standard pricing tiers only.
 - **Cognitive Services** resource with the S0 pricing tier.

Running the sample

This procedure loads and runs the Cognitive Services Container sample for language detection. The sample has two containers, one for the client application and one for the Cognitive Services container. We'll push both of these images to the Azure Container Registry. Once they are on your own registry, create an Azure Kubernetes Service to access these images and run the containers. When the containers are running, use the **kubectl** CLI to watch the containers performance. Access the client application with an HTTP request and see the results.



The sample containers

The sample has two container images, one for the frontend website. The second image is the language detection container returning the detected language (culture) of text. Both containers are accessible from an external IP when you are done.

The language-frontend container

This website is equivalent to your own client-side application that makes requests of the language detection endpoint. When the procedure is finished, you get the detected language of a string of characters by accessing the website container in a browser with `http://<external-IP>/<text-to-analyze>`. An example of this URL is `http://132.12.23.255/helloworld!`. The result in the browser is `English`.

The language container

The language detection container, in this specific procedure, is accessible to any external request. The container hasn't been changed in any way so the standard Cognitive Services container-specific language detection API is available.

For this container, that API is a POST request for language detection. As with all Cognitive Services containers, you can learn more about the container from its hosted Swagger information,

`http://<external-IP>:5000/swagger/index.html`.

Port 5000 is the default port used with the Cognitive Services containers.

Create Azure Container Registry service

To deploy the container to the Azure Kubernetes Service, the container images need to be accessible. Create your own Azure Container Registry service to host the images.

1. Sign in to the Azure CLI

```
az login
```

2. Create a resource group named `cogserv-container-rg` to hold every resource created in this procedure.

```
az group create --name cogserv-container-rg --location westus
```

3. Create your own Azure Container Registry with the format of your name then `registry`, such as `pattyregistry`. Do not use dashes or underline characters in the name.

```
az acr create --resource-group cogserv-container-rg --name pattyregistry --sku Basic
```

Save the results to get the `loginServer` property. This will be part of the hosted container's address, used later in the `language.yml` file.

```
az acr create --resource-group cogserv-container-rg --name pattyregistry --sku Basic
```

```
{
  "adminUserEnabled": false,
  "creationDate": "2019-01-02T23:49:53.783549+00:00",
  "id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/cogserv-container-rg/providers/Microsoft.ContainerRegistry/registries/pattyregistry",
  "location": "westus",
  "loginServer": "pattyregistry.azurecr.io",
  "name": "pattyregistry",
  "provisioningState": "Succeeded",
  "resourceGroup": "cogserv-container-rg",
  "sku": {
    "name": "Basic",
    "tier": "Basic"
  },
  "status": null,
  "storageAccount": null,
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries"
}
```

4. Sign in to your container registry. You need to login before you can push images to your registry.

```
az acr login --name pattyregistry
```

Get website Docker image

1. The sample code used in this procedure is in the Cognitive Services containers samples repository. Clone the repository to have a local copy of the sample.

```
git clone https://github.com/Azure-Samples/cognitive-services-containers-samples
```

Once the repository is on your local computer, find the website in the `\dotnet\Language\FrontendService` directory. This website acts as the client application calling the language detection API hosted in the language detection container.

2. Build the Docker image for this website. Make sure the console is in the `\FrontendService` directory where the Dockerfile is located when you run the following command:

```
docker build -t language-frontend -t pattyregistry.azurecr.io/language-frontend:v1 .
```

To track the version on your container registry, add the tag with a version format, such as `v1`.

3. Push the image to your container registry. This may take a few minutes.

```
docker push pattyregistry.azurecr.io/language-frontend:v1
```

If you get an `unauthorized: authentication required` error, login with the

```
az acr login --name <your-container-registry-name>
```

command.

When the process is done, the results should be similar to:

```
The push refers to repository [pattyregistry.azurecr.io/language-frontend]
82ff52ee6c73: Pushed
07599c047227: Pushed
816caf41a9a1: Pushed
2924be3aed17: Pushed
45b83a23806f: Pushed
ef68f6734aa4: Pushed
v1: digest: sha256:31930445deee181605c0cde53dab5a104528dc1ff57e5b3b34324f0d8a0eb286 size: 1580
```

Get language detection Docker image

1. Pull the latest version of the Docker image to the local machine. This may take a few minutes. If there is a newer version of this container, change the value from `1.1.006770001-amd64-preview` to the newer version.

```
docker pull mcr.microsoft.com/azure-cognitive-services/language:1.1.006770001-amd64-preview
```

2. Tag image with your container registry. Find the latest version and replace the version `1.1.006770001-amd64-preview` if you have a more recent version.

```
docker tag mcr.microsoft.com/azure-cognitive-services/language
pattyregistry.azurecr.io/language:1.1.006770001-amd64-preview
```

3. Push the image to your container registry. This may take a few minutes.

```
docker push pattyregistry.azurecr.io/language:1.1.006770001-amd64-preview
```

Get Container Registry credentials

The following steps are needed to get the required information to connect your container registry with the Azure Kubernetes Service you create later in this procedure.

1. Create service principal.

```
az ad sp create-for-rbac --skip-assignment
```

Save the results `appId` value for the assignee parameter in step 3, `<appId>`. Save the `password` for the next section's client-secret parameter `<client-secret>`.

```
{
  "appId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "displayName": "azure-cli-2018-12-31-18-39-32",
  "name": "http://azure-cli-2018-12-31-18-39-32",
  "password": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

2. Get your container registry ID.

```
az acr show --resource-group cogserv-container-rg --name pattyregistry --query "id" --o table
```

Save the output for the scope parameter value, `<acrId>`, in the next step. It looks like:

```
/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/cogserv-container-rg/providers/Microsoft.ContainerRegistry/registries/pattyregistry
```

Save the full value for step 3 in this section.

3. To grant the correct access for the AKS cluster to use images stored in your container registry, create a role assignment. Replace `<appId>` and `<acrId>` with the values gathered in the previous two steps.

```
az role assignment create --assignee <appId> --scope <acrId> --role Reader
```

Create Azure Kubernetes Service

1. Create the Kubernetes cluster. All the parameter values are from previous sections except the name parameter. Choose a name that indicates who created it and its purpose, such as `patty-kube`.

```
az aks create --resource-group cogserv-container-rg --name patty-kube --node-count 2 --service-principal <appId> --client-secret <client-secret> --generate-ssh-keys
```

This step may take a few minutes. The result is:

```
{
  "aadProfile": null,
  "addonProfiles": null,
  "agentPoolProfiles": [
    {
      "count": 2,
      "dnsPrefix": null,
      "fqdn": null,
      "maxPods": 110,
      "name": "nodepool1",
      "osDiskSizeGb": 30,
      "osType": "Linux",
      "ports": null,
      "storageProfile": "ManagedDisks",
      "vmSize": "Standard_DS1_v2",
      "vnetSubnetId": null
    }
  ],
  "dnsPrefix": "patty-kube--65a101",
  "enableRbac": true,
  "fqdn": "patty-kube--65a101-341f1f54.hcp.westus.azmk8s.io",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourcegroups/cogserv-container-rg/providers/Microsoft.ContainerService/managedClusters/patty-kube",
  "kubernetesVersion": "1.9.11",
  "linuxProfile": {
    "adminUsername": "azureuser",
    "ssh": {
      "publicKeys": [
        {
          "keyData": "ssh-rsa AAAAB3NzaC...ohR2d81mFC"
        }
      ]
    }
  },
  "location": "westus",
  "name": "patty-kube",
  "networkProfile": {
    "dnsServiceIp": "10.0.0.10",
    "dockerBridgeCidr": "172.17.0.1/16",
    "networkPlugin": "kubenet",
    "networkPolicy": null,
    "podCidr": "10.244.0.0/16",
    "serviceCidr": "10.0.0.0/16"
  },
  "nodeResourceGroup": "MC_patty_westus",
  "provisioningState": "Succeeded",
  "resourceGroup": "cogserv-container-rg",
  "servicePrincipalProfile": {
    "clientId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "keyVaultSecretRef": null,
    "secret": null
  },
  "tags": null,
  "type": "Microsoft.ContainerService/ManagedClusters"
}
```

The service is created but it doesn't have the website container or language detection container yet.

2. Get credentials of the Kubernetes cluster.

```
az aks get-credentials --resource-group cogserv-container-rg --name patty-kube
```

Load the orchestration definition into your Kubernetes service

This section uses the **kubectl** CLI to talk with the Azure Kubernetes Service.

1. Before loading the orchestration definition, check **kubectl** has access to the nodes.

```
kubectl get nodes
```

The response looks like:

NAME	STATUS	ROLES	AGE	VERSION
aks-nodpool11-13756812-0	Ready	agent	6m	v1.9.11
aks-nodpool11-13756812-1	Ready	agent	6m	v1.9.11

2. Copy the following file and name it `language.yml`. The file has a `service` section and a `deployment` section each for the two container types, the `language-frontend` website container and the `language` detection container.

```
# A service which exposes the .net frontend app container through a dependable hostname:
http://language-frontend:5000
apiVersion: v1
kind: Service
metadata:
  name: language-frontend
  labels:
    run: language-frontend
spec:
  selector:
    app: language-frontend
  type: LoadBalancer
  ports:
    - name: front
      port: 80
      targetPort: 80
      protocol: TCP
---
# A deployment declaratively indicating how many instances of the .net frontend app container we want
up
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: language-frontend
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: language-frontend
    spec:
      containers:
        - name: language-frontend
          image: # < URI of the Frontend App image >
          ports:
            - name: public-port
              containerPort: 80
          livenessProbe:
            httpGet:
              path: /status
              port: public-port
            initialDelaySeconds: 30
            timeoutSeconds: 1
            periodSeconds: 10
          imagePullSecrets:
            - name: # < Name of the registry secret providing access to the frontend image >
          automountServiceAccountToken: false
---
```

```
# A service which exposes the cognitive-service containers through a dependable hostname:
http://language:5000
apiVersion: v1
kind: Service
metadata:
  name: language
  labels:
    run: language
spec:
  selector:
    app: language
  type: LoadBalancer
  ports:
    - name: language
      port: 5000
      targetPort: 5000
      protocol: TCP
---
# A deployment declaratively indicating how many instances of the cognitive-service container we want
up
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: language
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: language
    spec:
      containers:
        - name: language
          image: # < URI of the Language Image >
          ports:
            - name: public-port
              containerPort: 5000
          livenessProbe:
            httpGet:
              path: /status
              port: public-port
            initialDelaySeconds: 30
            timeoutSeconds: 1
            periodSeconds: 10
          args:
            - "eula=accept"
            - "apikey=" # < API Key for the Language Service >
            - "billing=" # < Language billing endpoint URI >

          imagePullSecrets:
            - name: # < Name of the registry secret providing access to the Language image >

          automountServiceAccountToken: false
```

3. Change the language-frontend deployment lines of `language.yml` based on the following table to add your own container registry image names, client secret, and text analytics settings.

LANGUAGE-FRONTEND DEPLOYMENT SETTINGS	PURPOSE
Line 32 <code>image</code> property	Image location for the frontend image in your Container Registry <div><container-registry-name>.azurecr.io/language-frontend:v1</div>

LANGUAGE-FRONTEND DEPLOYMENT SETTINGS	PURPOSE
Line 44 <code>name</code> property	Container Registry secret for the image, referred to as <code><client-secret></code> in a previous section.

4. Change the language deployment lines of `language.yml` based on the following table to add your own container registry image names, client secret, and text analytics settings.

LANGUAGE DEPLOYMENT SETTINGS	PURPOSE
Line 78 <code>image</code> property	Image location for the language image in your Container Registry <code><container-registry-name>.azurecr.io/language:1.1.006770001-amd64-preview</code>
Line 95 <code>name</code> property	Container Registry secret for the image, referred to as <code><client-secret></code> in a previous section.
Line 91 <code>apiKey</code> property	Your text analytics resource key
Line 92 <code>billing</code> property	The billing endpoint for your text analytics resource. <code>https://westus.api.cognitive.microsoft.com/text/analytics/v2.1</code>

Because the `apiKey` and `billing endpoint` are set as part of the Kubernetes orchestration definition, the website container doesn't need to know about these or pass them as part of the request. The website container refers to the language detection container by its orchestrator name `language`.

5. Load the orchestration definition file for this sample from the folder where you created and saved the `language.yml`.

```
kubectl apply -f language.yml
```

The response is:

```
service "language-frontend" created
deployment.apps "language-frontend" created
service "language" created
deployment.apps "language" created
```

Get external IPs of containers

For the two containers, verify the `language-frontend` and `language` services are running and get the external IP address.

```
kubectl get all
```


NAME	READY	STATUS	RESTARTS	AGE
pod/language-586849d8dc-7zvz5	1/1	Running	0	13h
pod/language-frontend-68b9969969-bz9bg	1/1	Running	1	13h

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	14h
service/language	LoadBalancer	10.0.39.169	104.42.172.68	5000:30161/TCP	13h
service/language-frontend	LoadBalancer	10.0.42.136	104.42.37.219	80:30943/TCP	13h

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.extensions/language	1	1	1	1	13h
deployment.extensions/language-frontend	1	1	1	1	13h

NAME	DESIRED	CURRENT	READY	AGE
replicaset.extensions/language-586849d8dc	1	1	1	13h
replicaset.extensions/language-frontend-68b9969969	1	1	1	13h

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/language	1	1	1	1	13h
deployment.apps/language-frontend	1	1	1	1	13h

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/language-586849d8dc	1	1	1	13h
replicaset.apps/language-frontend-68b9969969	1	1	1	13h

If the `EXTERNAL-IP` for the service is shown as pending, rerun the command until the IP address is shown before moving to the next step.

Test the language detection container

Open a browser and navigate to the external IP of the `language` container from the previous section:

`http://<external-ip>:5000/swagger/index.html`. You can use the `Try it` feature of the API to test the language detection endpoint.

The screenshot shows the Swagger UI for the 'Language Detection Cognitive Service API V1'. The header is green with the Swagger logo and a dropdown menu showing 'Language Detection Cognitive Service API V1'. The main content area has a title 'Language Detection Cognitive Service API' with a 'v2' badge. Below the title is a link to '/swagger/v2/swagger.json' and a description: 'The API returns the detected language and a numeric score between 0 and 1. Scores close to 1 indicate 100% certainty that the identified language is true. A total of 120 languages are supported.' There are two expandable sections: 'Detect Language' with a green 'POST' button and the endpoint '/text/analytics/v2.0/languages Detect Language', and 'Status' with a blue 'GET' button and the endpoint '/status'.

Test the client application container

Change the URL in the browser to the external IP of the `language-frontend` container using the following format:

`http://<external-ip>/helloworld`. The English culture text of `helloworld` is predicted as `English`.

Clean up resources

When you are done with the cluster, delete the Azure resource group.

```
az group delete --name cogserv-container-rg
```

Related information

- [kubectl for Docker Users](#)

Next steps

[Cognitive Services Containers](#)

Use Docker Compose to deploy multiple containers

3/30/2021 • 5 minutes to read • [Edit Online](#)

This article shows you how to deploy multiple Azure Cognitive Services containers. Specifically, you'll learn how to use Docker Compose to orchestrate multiple Docker container images.

[Docker Compose](#) is a tool for defining and running multi-container Docker applications. In Compose, you use a YAML file to configure your application's services. Then, you create and start all the services from your configuration by running a single command.

It can be useful to orchestrate multiple container images on a single host computer. In this article, we'll pull together the Read and Form Recognizer containers.

Prerequisites

This procedure requires several tools that must be installed and run locally:

- An Azure subscription. If you don't have one, create a [free account](#) before you begin.
- [Docker Engine](#). Confirm that the Docker CLI works in a console window.
- An Azure resource with the correct pricing tier. Only the following pricing tiers work with this container:
 - **Computer Vision** resource with F0 or Standard pricing tier only.
 - **Form Recognizer** resource with F0 or Standard pricing tier only.
 - **Cognitive Services** resource with the S0 pricing tier.
- If you're using a gated preview container, You will need to complete the [online request form](#) to use it.

Docker Compose file

The YAML file defines all the services to be deployed. These services rely on either a `DockerFile` or an existing container image. In this case, we'll use two preview images. Copy and paste the following YAML file, and save it as *docker-compose.yaml*. Provide the appropriate **apikey**, **billing**, and **EndpointUri** values in the file.

```

version: '3.7'
services:
  forms:
    image: "mcr.microsoft.com/azure-cognitive-services/form-recognizer/layout"
    environment:
      eula: accept
      billing: # < Your form recognizer billing URL >
      apikey: # < Your form recognizer API key >
      FormRecognizer__ComputerVisionApiKey: # < Your form recognizer API key >
      FormRecognizer__ComputerVisionEndpointUri: # < Your form recognizer URI >
    volumes:
      - type: bind
        source: E:\publicpreview\output
        target: /output
      - type: bind
        source: E:\publicpreview\input
        target: /input
    ports:
      - "5010:5000"

  ocr:
    image: "mcr.microsoft.com/azure-cognitive-services/vision/read:3.1-preview"
    environment:
      eula: accept
      apikey: # < Your computer vision API key >
      billing: # < Your computer vision billing URL >
    ports:
      - "5021:5000"

```

IMPORTANT

Create the directories on the host machine that are specified under the **volumes** node. This approach is required because the directories must exist before you try to mount an image by using volume bindings.

Start the configured Docker Compose services

A Docker Compose file enables the management of all the stages in a defined service's life cycle: starting, stopping, and rebuilding services; viewing the service status; and log streaming. Open a command-line interface from the project directory (where the `docker-compose.yml` file is located).

NOTE

To avoid errors, make sure that the host machine correctly shares drives with Docker Engine. For example, if `E:\publicpreview` is used as a directory in the `docker-compose.yml` file, share drive E with Docker.

From the command-line interface, execute the following command to start (or restart) all the services defined in the `docker-compose.yml` file:

```
docker-compose up
```

The first time Docker executes the **docker-compose up** command by using this configuration, it pulls the images configured under the **services** node and then downloads and mounts them:

```
Pulling forms (mcr.microsoft.com/azure-cognitive-services/form-recognizer/layout:)...
latest: Pulling from azure-cognitive-services/form-recognizer/layout
743f2d6c1f65: Pull complete
72befba99561: Pull complete
2a40b9192d02: Pull complete
c7715c9d5c33: Pull complete
f0b33959f1c4: Pull complete
b8ab86c6ab26: Pull complete
41940c21ed3c: Pull complete
e3d37dd258d4: Pull complete
cdb5eb761109: Pull complete
fd93b5f95865: Pull complete
ef41dcbc5857: Pull complete
4d05c86a4178: Pull complete
34e811d37201: Pull complete
Pulling ocr (mcr.microsoft.com/azure-cognitive-services/vision/read:3.1-preview:)...
latest: Pulling from /azure-cognitive-services/vision/read:3.1-preview
f476d66f5408: Already exists
8882c27f669e: Already exists
d9af21273955: Already exists
f5029279ec12: Already exists
1a578849dcd1: Pull complete
45064b1ab0bf: Download complete
4bb846705268: Downloading [=====>] 187.1MB/222.8MB
c56511552241: Waiting
e91d2aa0f1ad: Downloading [=====>] 162.2MB/176.1MB
```

After the images are downloaded, the image services are started:

```

Starting docker_ocr_1 ... done
Starting docker_forms_1 ... doneAttaching to docker_ocr_1, docker_forms_1forms_1 | forms_1 | forms_1 |
Notice: This Preview is made available to you on the condition that you agree to the Supplemental Terms of
Use for Microsoft Azure Previews [https://go.microsoft.com/fwlink/?linkid=2018815], which supplement your
agreement [https://go.microsoft.com/fwlink/?linkid=2018657] governing your use of Azure. If you do not have
an existing agreement governing your use of Azure, you agree that your agreement governing use of Azure is
the Microsoft Online Subscription Agreement [https://go.microsoft.com/fwlink/?linkid=2018755] (which
incorporates the Online Services Terms [https://go.microsoft.com/fwlink/?linkid=2018760]). By using the
Preview you agree to these terms.
forms_1 |
forms_1 |
forms_1 | Using '/input' for reading models and other read-only data.
forms_1 | Using '/output/forms/812d811d1bcc' for writing logs and other output data.
forms_1 | Logging to console.
forms_1 | Submitting metering to 'https://westus2.api.cognitive.microsoft.com/'.
forms_1 | WARNING: No access control enabled!
forms_1 | warn: Microsoft.AspNetCore.Server.Kestrel[0]
forms_1 |      Overriding address(es) 'http://+:80'. Binding to endpoints defined in UseKestrel() instead.
forms_1 | Hosting environment: Production
forms_1 | Content root path: /app/forms
forms_1 | Now listening on: http://0.0.0.0:5000
forms_1 | Application started. Press Ctrl+C to shut down.
ocr_1 |
ocr_1 |
ocr_1 | Notice: This Preview is made available to you on the condition that you agree to the Supplemental
Terms of Use for Microsoft Azure Previews [https://go.microsoft.com/fwlink/?linkid=2018815], which
supplement your agreement [https://go.microsoft.com/fwlink/?linkid=2018657] governing your use of Azure. If
you do not have an existing agreement governing your use of Azure, you agree that your agreement governing
use of Azure is the Microsoft Online Subscription Agreement [https://go.microsoft.com/fwlink/?
linkid=2018755] (which incorporates the Online Services Terms [https://go.microsoft.com/fwlink/?
linkid=2018760]). By using the Preview you agree to these terms.
ocr_1 |
ocr_1 |
ocr_1 | Logging to console.
ocr_1 | Submitting metering to 'https://westcentralus.api.cognitive.microsoft.com/'.
ocr_1 | WARNING: No access control enabled!
ocr_1 | Hosting environment: Production
ocr_1 | Content root path: /
ocr_1 | Now listening on: http://0.0.0.0:5000
ocr_1 | Application started. Press Ctrl+C to shut down.

```

Verify the service availability

TIP

You can use the `docker images` command to list your downloaded container images. For example, the following command lists the ID, repository, and tag of each downloaded container image, formatted as a table:

```
docker images --format "table {{.ID}}\t{{.Repository}}\t{{.Tag}}"
```

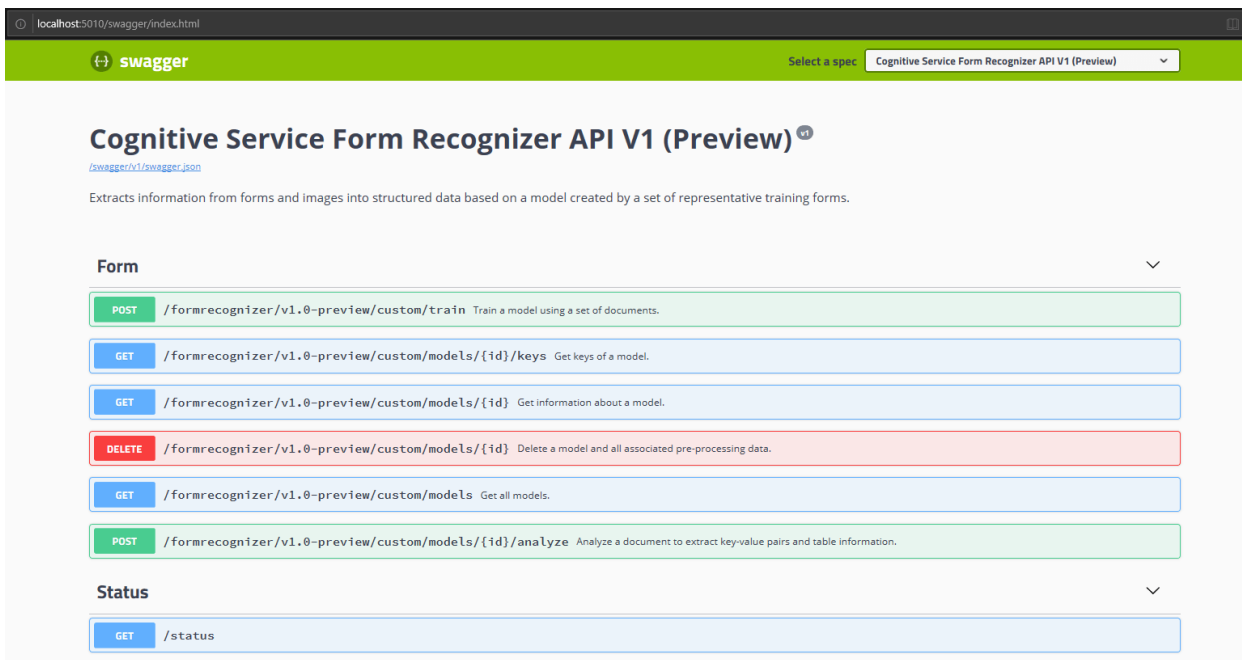
IMAGE ID	REPOSITORY	TAG
<image-id>	<repository-path/name>	<tag-name>

Here's some example output:

IMAGE ID	REPOSITORY	TAG
2ce533f88e80	mcr.microsoft.com/azure-cognitive-services/form-recognizer/layout	latest
4be104c126c5	mcr.microsoft.com/azure-cognitive-services/vision/read:3.1-preview	latest

Test containers

Open a browser on the host machine and go to **localhost** by using the specified port from the *docker-compose.yaml* file, such as `http://localhost:5021/swagger/index.html`. For example, you could use the **Try It** feature in the API to test the Form Recognizer endpoint. Both containers swagger pages should be available and testable.



The screenshot displays the Swagger UI for the Cognitive Service Form Recognizer API V1 (Preview). The browser address bar shows `localhost:5021/swagger/index.html`. The Swagger logo is in the top left, and a dropdown menu in the top right shows 'Cognitive Service Form Recognizer API V1 (Preview)'. The main heading is 'Cognitive Service Form Recognizer API V1 (Preview)' with a link to `/swagger/v1/swagger.json`. Below the heading is a description: 'Extracts information from forms and images into structured data based on a model created by a set of representative training forms.'

The endpoints are organized into two sections: 'Form' and 'Status'. The 'Form' section contains six endpoints, and the 'Status' section contains one endpoint.

Method	Endpoint	Description
POST	<code>/formrecognizer/v1.0-preview/custom/train</code>	Train a model using a set of documents.
GET	<code>/formrecognizer/v1.0-preview/custom/models/{id}/keys</code>	Get keys of a model.
GET	<code>/formrecognizer/v1.0-preview/custom/models/{id}</code>	Get information about a model.
DELETE	<code>/formrecognizer/v1.0-preview/custom/models/{id}</code>	Delete a model and all associated pre-processing data.
GET	<code>/formrecognizer/v1.0-preview/custom/models</code>	Get all models.
POST	<code>/formrecognizer/v1.0-preview/custom/models/{id}/analyze</code>	Analyze a document to extract key-value pairs and table information.
GET	<code>/status</code>	

Next steps

[Cognitive Services containers](#)

Tutorial: Create a container image for deployment to Azure Container Instances

11/2/2020 • 3 minutes to read • [Edit Online](#)

Azure Container Instances enables deployment of Docker containers onto Azure infrastructure without provisioning any virtual machines or adopting a higher-level service. In this tutorial, you package a small Node.js web application into a container image that can be run using Azure Container Instances.

In this article, part one of the series, you:

- Clone application source code from GitHub
- Create a container image from application source
- Test the image in a local Docker environment

In tutorial parts two and three, you upload your image to Azure Container Registry, and then deploy it to Azure Container Instances.

Before you begin

You must satisfy the following requirements to complete this tutorial:

Azure CLI: You must have Azure CLI version 2.0.29 or later installed on your local computer. Run `az --version` to find the version. If you need to install or upgrade, see [Install the Azure CLI](#).

Docker: This tutorial assumes a basic understanding of core Docker concepts like containers, container images, and basic `docker` commands. For a primer on Docker and container basics, see the [Docker overview](#).

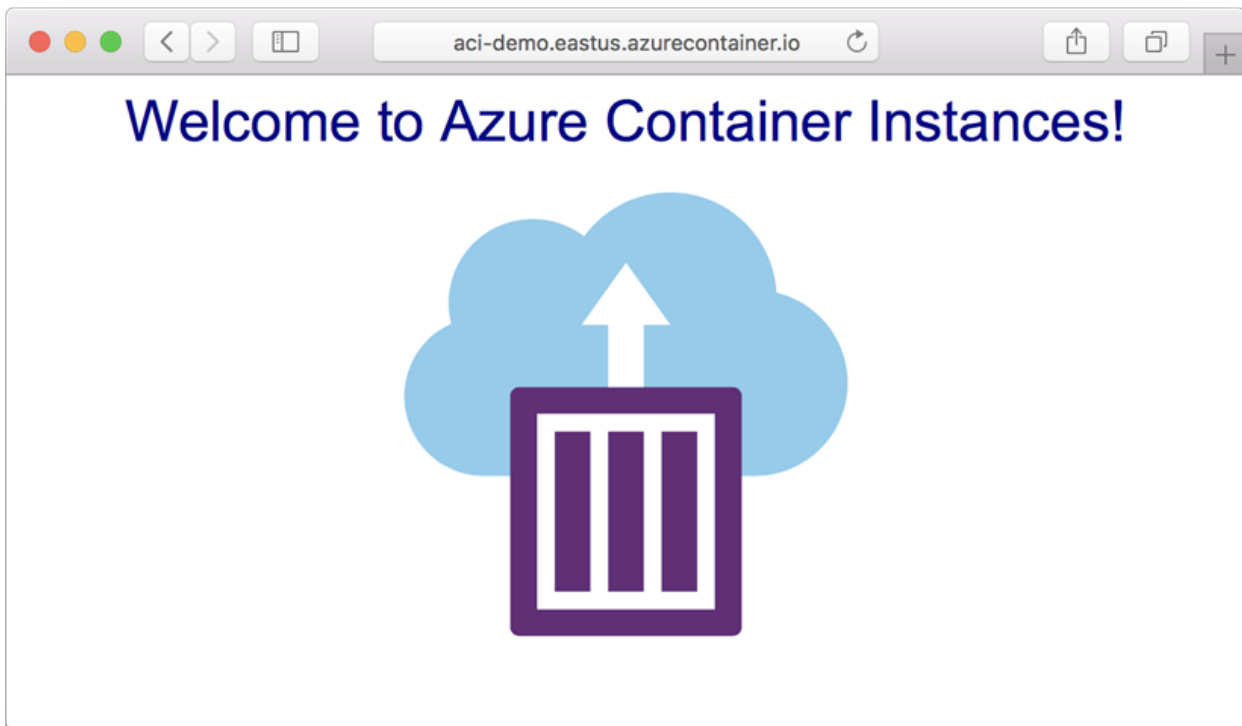
Docker: To complete this tutorial, you need Docker installed locally. Docker provides packages that configure the Docker environment on [macOS](#), [Windows](#), and [Linux](#).

IMPORTANT

Because the Azure Cloud shell does not include the Docker daemon, you *must* install both the Azure CLI and Docker Engine on your *local computer* to complete this tutorial. You cannot use the Azure Cloud Shell for this tutorial.

Get application code

The sample application in this tutorial is a simple web app built in [Node.js](#). The application serves a static HTML page, and looks similar to the following screenshot:



Use Git to clone the sample application's repository:

```
git clone https://github.com/Azure-Samples/aci-helloworld.git
```

You can also [download the ZIP archive](#) from GitHub directly.

Build the container image

The Dockerfile in the sample application shows how the container is built. It starts from an [official Node.js image](#) based on [Alpine Linux](#), a small distribution that is well suited for use with containers. It then copies the application files into the container, installs dependencies using the Node Package Manager, and finally, starts the application.

```
FROM node:8.9.3-alpine
RUN mkdir -p /usr/src/app
COPY ./app/ /usr/src/app/
WORKDIR /usr/src/app
RUN npm install
CMD node /usr/src/app/index.js
```

Use the [docker build](#) command to create the container image and tag it as *aci-tutorial-app*.

```
docker build ./aci-helloworld -t aci-tutorial-app
```

Output from the [docker build](#) command is similar to the following (truncated for readability):

```
$ docker build ./aci-helloworld -t aci-tutorial-app
Sending build context to Docker daemon 119.3kB
Step 1/6 : FROM node:8.9.3-alpine
8.9.3-alpine: Pulling from library/node
88286f41530e: Pull complete
84f3a4bf8410: Pull complete
d0d9b2214720: Pull complete
Digest: sha256:c73277ccc763752b42bb2400d1aaecb4e3d32e3a9dbedd0e49885c71bea07354
Status: Downloaded newer image for node:8.9.3-alpine
--> 90f5ee24bee2
...
Step 6/6 : CMD node /usr/src/app/index.js
--> Running in f4a1ea099eec
--> 6edad76d09e9
Removing intermediate container f4a1ea099eec
Successfully built 6edad76d09e9
Successfully tagged aci-tutorial-app:latest
```

Use the `docker images` command to see the built image:

```
docker images
```

Your newly built image should appear in the list:

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
aci-tutorial-app	latest	5c745774dfa9	39 seconds ago	68.1 MB

Run the container locally

Before you deploy the container to Azure Container Instances, use `docker run` to run it locally and confirm that it works. The `-d` switch lets the container run in the background, while `-p` allows you to map an arbitrary port on your computer to port 80 in the container.

```
docker run -d -p 8080:80 aci-tutorial-app
```

Output from the `docker run` command displays the running container's ID if the command was successful:

```
$ docker run -d -p 8080:80 aci-tutorial-app
a2e3e4435db58ab0c664ce521854c2e1a1bda88c9cf2fcff46aedf48df86cccf
```

Now, navigate to `http://localhost:8080` in your browser to confirm that the container is running. You should see a web page similar to the following:



Next steps

In this tutorial, you created a container image that can be deployed in Azure Container Instances, and verified that it runs locally. So far, you've done the following:

- Cloned the application source from GitHub
- Created a container image from the application source
- Tested the container locally

Advance to the next tutorial in the series to learn about storing your container image in Azure Container Registry:

[Push image to Azure Container Registry](#)

Quickstart: Create a private container registry using the Azure CLI

6/24/2021 • 4 minutes to read • [Edit Online](#)

Azure Container Registry is a private registry service for building, storing, and managing container images and related artifacts. In this quickstart, you create an Azure container registry instance with the Azure CLI. Then, use Docker commands to push a container image into the registry, and finally pull and run the image from your registry.

This quickstart requires that you are running the Azure CLI (version 2.0.55 or later recommended). Run `az --version` to find the version. If you need to install or upgrade, see [Install Azure CLI](#).

You must also have Docker installed locally. Docker provides packages that easily configure Docker on any [macOS](#), [Windows](#), or [Linux](#) system.

Because the Azure Cloud Shell doesn't include all required Docker components (the `dockerd` daemon), you can't use the Cloud Shell for this quickstart.

Create a resource group

Create a resource group with the [az group create](#) command. An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named *myResourceGroup* in the *eastus* location.

```
az group create --name myResourceGroup --location eastus
```

Create a container registry

In this quickstart you create a *Basic* registry, which is a cost-optimized option for developers learning about Azure Container Registry. For details on available service tiers, see [Container registry service tiers](#).

Create an ACR instance using the [az acr create](#) command. The registry name must be unique within Azure, and contain 5-50 alphanumeric characters. In the following example, *myContainerRegistry007* is used. Update this to a unique value.

```
az acr create --resource-group myResourceGroup \
  --name myContainerRegistry007 --sku Basic
```

When the registry is created, the output is similar to the following:

```
{
  "adminUserEnabled": false,
  "creationDate": "2019-01-08T22:32:13.175925+00:00",
  "id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myResourceGroup/providers/Microsoft.ContainerRegistry/registries/myContainerRegistry007",
  "location": "eastus",
  "loginServer": "mycontainerregistry007.azurecr.io",
  "name": "myContainerRegistry007",
  "provisioningState": "Succeeded",
  "resourceGroup": "myResourceGroup",
  "sku": {
    "name": "Basic",
    "tier": "Basic"
  },
  "status": null,
  "storageAccount": null,
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries"
}
```

Take note of `loginServer` in the output, which is the fully qualified registry name (all lowercase). Throughout the rest of this quickstart `<registry-name>` is a placeholder for the container registry name, and `<login-server>` is a placeholder for the registry's login server name.

TIP

In this quickstart, you create a *Basic* registry, which is a cost-optimized option for developers learning about Azure Container Registry. Choose other tiers for increased storage and image throughput, and capabilities such as connection using a [private endpoint](#). For details on available service tiers (SKUs), see [Container registry service tiers](#).

Log in to registry

Before pushing and pulling container images, you must log in to the registry. To do so, use the [az acr login](#) command. Specify only the registry resource name when logging in with the Azure CLI. Don't use the fully qualified login server name.

```
az acr login --name <registry-name>
```

Example:

```
az acr login --name mycontainerregistry
```

The command returns a `Login Succeeded` message once completed.

Push image to registry

To push an image to an Azure Container registry, you must first have an image. If you don't yet have any local container images, run the following [docker pull](#) command to pull an existing public image. For this example, pull the `hello-world` image from Microsoft Container Registry.

```
docker pull mcr.microsoft.com/hello-world
```

Before you can push an image to your registry, you must tag it with the fully qualified name of your registry

login server. The login server name is in the format `<registry-name>.azurecr.io` (must be all lowercase), for example, `mycontainerregistry.azurecr.io`.

Tag the image using the `docker tag` command. Replace `<login-server>` with the login server name of your ACR instance.

```
docker tag mcr.microsoft.com/hello-world <login-server>/hello-world:v1
```

Example:

```
docker tag mcr.microsoft.com/hello-world mycontainerregistry.azurecr.io/hello-world:v1
```

Finally, use `docker push` to push the image to the registry instance. Replace `<login-server>` with the login server name of your registry instance. This example creates the **hello-world** repository, containing the `hello-world:v1` image.

```
docker push <login-server>/hello-world:v1
```

After pushing the image to your container registry, remove the `hello-world:v1` image from your local Docker environment. (Note that this `docker rmi` command does not remove the image from the **hello-world** repository in your Azure container registry.)

```
docker rmi <login-server>/hello-world:v1
```

List container images

The following example lists the repositories in your registry:

```
az acr repository list --name <registry-name> --output table
```

Output:

```
Result
-----
hello-world
```

The following example lists the tags on the **hello-world** repository.

```
az acr repository show-tags --name <registry-name> --repository hello-world --output table
```

Output:

```
Result
-----
v1
```

Run image from registry

Now, you can pull and run the `hello-world:v1` container image from your container registry by using `docker`

run:

```
docker run <login-server>/hello-world:v1
```

Example output:

```
Unable to find image 'mycontainerregistry.azurecr.io/hello-world:v1' locally
v1: Pulling from hello-world
Digest: sha256:662dd8e65ef7ccf13f417962c2f77567d3b132f12c95909de6c85ac3c326a345
Status: Downloaded newer image for mycontainerregistry.azurecr.io/hello-world:v1

Hello from Docker!
This message shows that your installation appears to be working correctly.

[...]
```

Clean up resources

When no longer needed, you can use the [az group delete](#) command to remove the resource group, the container registry, and the container images stored there.

```
az group delete --name myResourceGroup
```

Next steps

In this quickstart, you created an Azure Container Registry with the Azure CLI, pushed a container image to the registry, and pulled and ran the image from the registry. Continue to the [Azure Container Registry tutorials](#) for a deeper look at ACR.

[Azure Container Registry tutorials](#)

[Azure Container Registry Tasks tutorials](#)

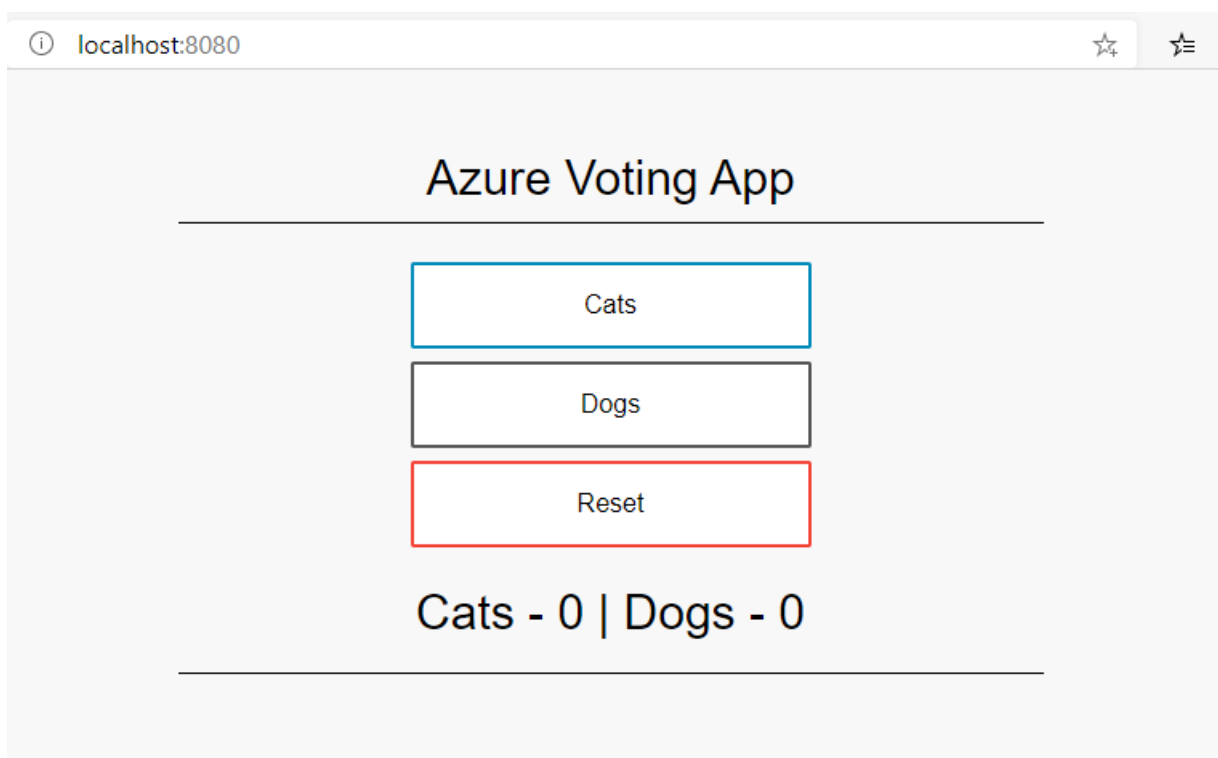
Tutorial: Prepare an application for Azure Kubernetes Service (AKS)

3/5/2021 • 3 minutes to read • [Edit Online](#)

In this tutorial, part one of seven, a multi-container application is prepared for use in Kubernetes. Existing development tools such as Docker Compose are used to locally build and test an application. You learn how to:

- Clone a sample application source from GitHub
- Create a container image from the sample application source
- Test the multi-container application in a local Docker environment

Once completed, the following application runs in your local development environment:



In later tutorials, the container image is uploaded to an Azure Container Registry, and then deployed into an AKS cluster.

Before you begin

This tutorial assumes a basic understanding of core Docker concepts such as containers, container images, and `docker` commands. For a primer on container basics, see [Get started with Docker](#).

To complete this tutorial, you need a local Docker development environment running Linux containers. Docker provides packages that configure Docker on a [Mac](#), [Windows](#), or [Linux](#) system.

NOTE

Azure Cloud Shell does not include the Docker components required to complete every step in these tutorials. Therefore, we recommend using a full Docker development environment.

Get application code

The [sample application](#) used in this tutorial is a basic voting app consisting of a front-end web component and a back-end Redis instance. The web component is packaged into a custom container image. The Redis instance uses an unmodified image from Docker Hub.

Use [git](#) to clone the sample application to your development environment:

```
git clone https://github.com/Azure-Samples/azure-voting-app-redis.git
```

Change into the cloned directory.

```
cd azure-voting-app-redis
```

Inside the directory is the application source code, a pre-created Docker compose file, and a Kubernetes manifest file. These files are used throughout the tutorial set. The contents and structure of the directory are as follows:

```
azure-voting-app-redis
├── azure-vote-all-in-one-redis.yaml
├── docker-compose.yaml
├── LICENSE
├── README.md
├── azure-vote
│   ├── app_init.supervisord.conf
│   ├── Dockerfile
│   ├── Dockerfile-for-app-service
│   └── sshd_config
│   └── azure-vote
│       ├── config_file.cfg
│       ├── main.py
│       ├── static
│       │   └── default.css
│       └── templates
│           └── index.html
└── jenkins-tutorial
    ├── config-jenkins.sh
    └── deploy-jenkins-vm.sh
```

Create container images

[Docker Compose](#) can be used to automate building container images and the deployment of multi-container applications.

Use the sample `docker-compose.yaml` file to create the container image, download the Redis image, and start the application:

```
docker-compose up -d
```

When completed, use the [docker images](#) command to see the created images. Three images have been downloaded or created. The *azure-vote-front* image contains the front-end application and uses the *nginx-flask* image as a base. The *redis* image is used to start a Redis instance.

```
$ docker images
```

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
mcr.microsoft.com/azuredocs/azure-vote-front 944MB	v1	84b41c268ad9	9 seconds ago
mcr.microsoft.com/oss/bitnami/redis 103MB	6.0.8	3a54a920bb6c	2 days ago
tiangolo/uwsgi-nginx-flask 944MB	python3.6	a16ce562e863	6 weeks ago

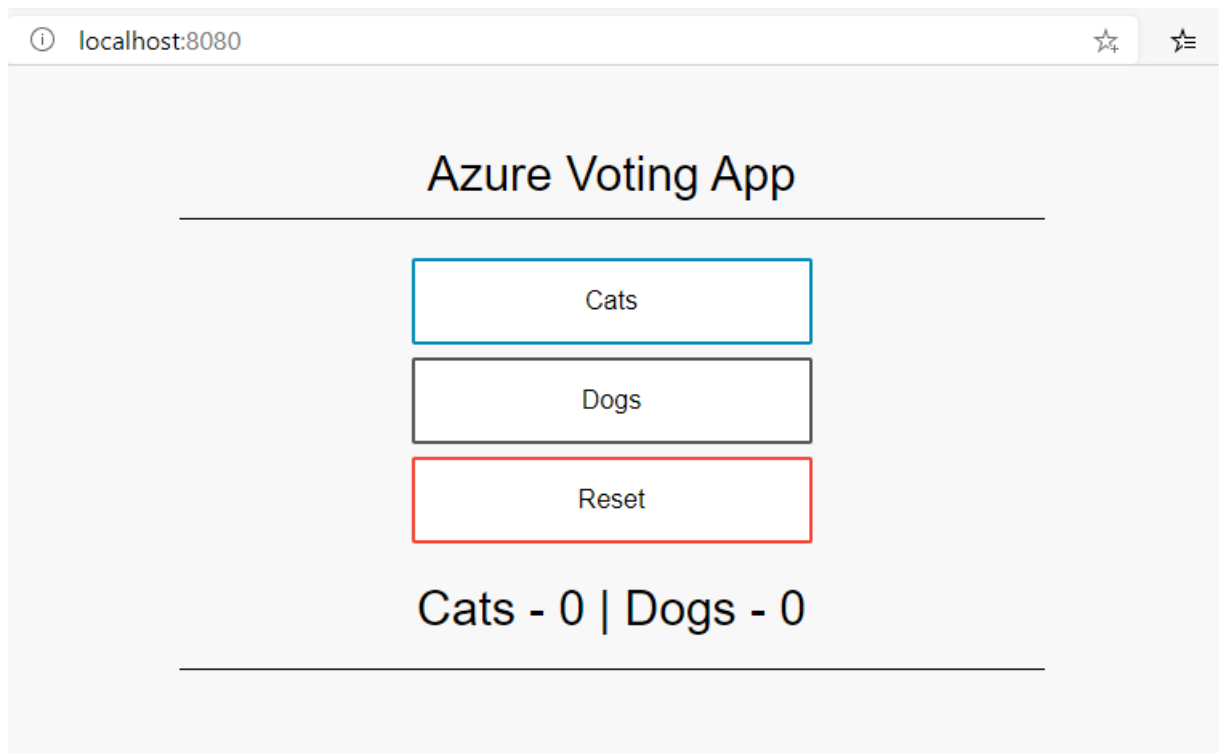
Run the [docker ps](#) command to see the running containers:

```
$ docker ps
```

CONTAINER ID STATUS	IMAGE PORTS	NAMES	COMMAND	CREATED
d10e5244f237 Up 3 minutes	mcr.microsoft.com/azuredocs/azure-vote-front:v1 443/tcp, 0.0.0.0:8080->80/tcp	azure-vote-front	"/entrypoint.sh /sta..."	3 minutes ago
21574cb38c1f Up 3 minutes	mcr.microsoft.com/oss/bitnami/redis:6.0.8 0.0.0.0:6379->6379/tcp	azure-vote-back	"/opt/bitnami/script..."	3 minutes ago

Test application locally

To see the running application, enter `http://localhost:8080` in a local web browser. The sample application loads, as shown in the following example:



Clean up resources

Now that the application's functionality has been validated, the running containers can be stopped and removed. **Do not delete the container images** - in the next tutorial, the *azure-vote-front* image is uploaded to an Azure Container Registry instance.

Stop and remove the container instances and resources with the [docker-compose down](#) command:

```
docker-compose down
```

When the local application has been removed, you have a Docker image that contains the Azure Vote application, *azure-vote-front*, for use with the next tutorial.

Next steps

In this tutorial, an application was tested and container images created for the application. You learned how to:

- Clone a sample application source from GitHub
- Create a container image from the sample application source
- Test the multi-container application in a local Docker environment

Advance to the next tutorial to learn how to store container images in Azure Container Registry.

[Push images to Azure Container Registry](#)

Azure Cognitive Services security

9/3/2021 • 5 minutes to read • [Edit Online](#)

Security should be considered a top priority when developing any and all applications. With the onset of artificial intelligence enabled applications, security is even more important. In this article various aspects of Azure Cognitive Services security are outlined, such as the use of transport layer security, authentication, securely configuring sensitive data, and Customer Lockbox for customer data access.

Transport Layer Security (TLS)

All of the Cognitive Services endpoints exposed over HTTP enforce TLS 1.2. With an enforced security protocol, consumers attempting to call a Cognitive Services endpoint should adhere to these guidelines:

- The client Operating System (OS) needs to support TLS 1.2
- The language (and platform) used to make the HTTP call need to specify TLS 1.2 as part of the request
 - Depending on the language and platform, specifying TLS is done either implicitly or explicitly

For .NET users, consider the [Transport Layer Security best practices](#).

Authentication

When discussing authentication, there are several common misconceptions. Authentication and authorization are often confused for one another. Identity is also a major component in security. An identity is a collection of information about a [principal](#). Identity providers (IdP) provide identities to authentication services.

Authentication is the act of verifying a user's identity. Authorization is the specification of access rights and privileges to resources for a given identity. Several of the Cognitive Services offerings, include Azure role-based access control (Azure RBAC). Azure RBAC could be used to simplify some of the ceremony involved with manually managing principals. For more details, see [Azure role-based access control for Azure resources](#).

For more information on authentication with subscription keys, access tokens and Azure Active Directory (AAD), see [authenticate requests to Azure Cognitive Services](#).

Environment variables and application configuration

Environment variables are name-value pairs, stored within a specific environment. A more secure alternative to using hardcoded values for sensitive data, is to use environment variables. Hardcoded values are insecure and should be avoided.

Caution

Do **not** use hardcoded values for sensitive data, doing so is a major security vulnerability.

NOTE

While environment variables are stored in plain text, they are isolated to an environment. If an environment is compromised, so too are the variables with the environment.

Set environment variable

To set environment variables, use one the following commands - where the `ENVIRONMENT_VARIABLE_KEY` is the named key and `value` is the value stored in the environment variable.

- [Command Line](#)

- [PowerShell](#)
- [Bash](#)

Create and assign persisted environment variable, given the value.

```
:: Assigns the env var to the value
setx ENVIRONMENT_VARIABLE_KEY="value"
```

In a new instance of the **Command Prompt**, read the environment variable.

```
:: Prints the env var value
echo %ENVIRONMENT_VARIABLE_KEY%
```

TIP

After setting an environment variable, restart your integrated development environment (IDE) to ensure that newly added environment variables are available.

Get environment variable

To get an environment variable, it must be read into memory. Depending on the language you're using, consider the following code snippets. These code snippets demonstrate how to get environment variable given the

`ENVIRONMENT_VARIABLE_KEY` and assign to a variable named `value`.

- [C#](#)
- [C++](#)
- [Java](#)
- [Node.js](#)
- [Python](#)
- [Objective-C](#)

For more information, see `Environment.GetEnvironmentVariable`.

```
using static System.Environment;

class Program
{
    static void Main()
    {
        // Get the named env var, and assign it to the value variable
        var value =
            GetEnvironmentVariable(
                "ENVIRONMENT_VARIABLE_KEY");
    }
}
```

Customer Lockbox

[Customer Lockbox for Microsoft Azure](#) provides an interface for customers to review, and approve or reject customer data access requests. It is used in cases where a Microsoft engineer needs to access customer data during a support request. For information on how Customer Lockbox requests are initiated, tracked, and stored for later reviews and audits, see [Customer Lockbox](#).

Customer Lockbox is available for this service:

- [Translator](#)

For the following services, Microsoft engineers will not access any customer data in the E0 tier:

- [Language Understanding](#)
- [Face](#)
- [Content Moderator](#)
- [Personalizer](#)

To request the ability to use the E0 SKU, fill out and submit this [request Form](#). It will take approximately 3-5 business days to hear back on the status of your request. Depending on demand, you may be placed in a queue and approved as space becomes available. Once approved for using the E0 SKU with LUIS, you'll need to create a new resource from the Azure portal and select E0 as the Pricing Tier. Users won't be able to upgrade from the F0 to the new E0 SKU.

The Speech service doesn't currently support Customer Lockbox. However, customer data can be stored using bring your own storage (BYOS), allowing you to achieve similar data controls to Customer Lockbox. Keep in mind that Speech service data stays and is processed in the region where the Speech resource was created. This applies to any data at rest and data in transit. When using customization features, like Custom Speech and Custom Voice, all customer data is transferred, stored, and processed in the same region where your BYOS (if used) and Speech service resource reside.

IMPORTANT

Microsoft **does not** use customer data to improve its Speech models. Additionally, if endpoint logging is disabled and no customizations are used, then no customer data is stored.

Next steps

- Explore the various [Cognitive Services](#)
- Learn more about [Cognitive Services Virtual Networks](#)

Configure Azure Cognitive Services virtual networks

9/3/2021 • 16 minutes to read • [Edit Online](#)

Azure Cognitive Services provides a layered security model. This model enables you to secure your Cognitive Services accounts to a specific subset of networks. When network rules are configured, only applications requesting data over the specified set of networks can access the account. You can limit access to your resources with request filtering. Allowing only requests originating from specified IP addresses, IP ranges or from a list of subnets in [Azure Virtual Networks](#).

An application that accesses a Cognitive Services resource when network rules are in effect requires authorization. Authorization is supported with [Azure Active Directory](#) (Azure AD) credentials or with a valid API key.

IMPORTANT

Turning on firewall rules for your Cognitive Services account blocks incoming requests for data by default. In order to allow requests through, one of the following conditions needs to be met:

- The request should originate from a service operating within an Azure Virtual Network (VNet) on the allowed subnet list of the target Cognitive Services account. The endpoint in requests originated from VNet needs to be set as the [custom subdomain](#) of your Cognitive Services account.
- Or the request should originate from an allowed list of IP addresses.

Requests that are blocked include those from other Azure services, from the Azure portal, from logging and metrics services, and so on.

NOTE

This article has been updated to use the Azure Az PowerShell module. The Az PowerShell module is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Scenarios

To secure your Cognitive Services resource, you should first configure a rule to deny access to traffic from all networks (including internet traffic) by default. Then, you should configure rules that grant access to traffic from specific VNets. This configuration enables you to build a secure network boundary for your applications. You can also configure rules to grant access to traffic from select public internet IP address ranges, enabling connections from specific internet or on-premises clients.

Network rules are enforced on all network protocols to Azure Cognitive Services, including REST and WebSocket. To access data using tools such as the Azure test consoles, explicit network rules must be configured. You can apply network rules to existing Cognitive Services resources, or when you create new Cognitive Services resources. Once network rules are applied, they're enforced for all requests.

Supported regions and service offerings

Virtual networks (VNETs) are supported in [regions where Cognitive Services are available](#). Currently multi-service resource does not support VNET. Cognitive Services supports service tags for network rules

configuration. The services listed below are included in the **CognitiveServicesManagement** service tag.

- Anomaly Detector
- Computer Vision
- Content Moderator
- Custom Vision
- Face
- Language Understanding (LUIS)
- Personalizer
- Speech Services
- Text Analytics
- QnA Maker
- Translator Text

NOTE

If you're using LUIS or Speech Services, the **CognitiveServicesManagement** tag only enables you use the service using the SDK or REST API. To access and use LUIS portal and/or Speech Studio from a virtual network, you will need to use the following tags:

- **AzureActiveDirectory**
- **AzureFrontDoor.Frontend**
- **AzureResourceManager**
- **CognitiveServicesManagement**

Change the default network access rule

By default, Cognitive Services resources accept connections from clients on any network. To limit access to selected networks, you must first change the default action.

WARNING

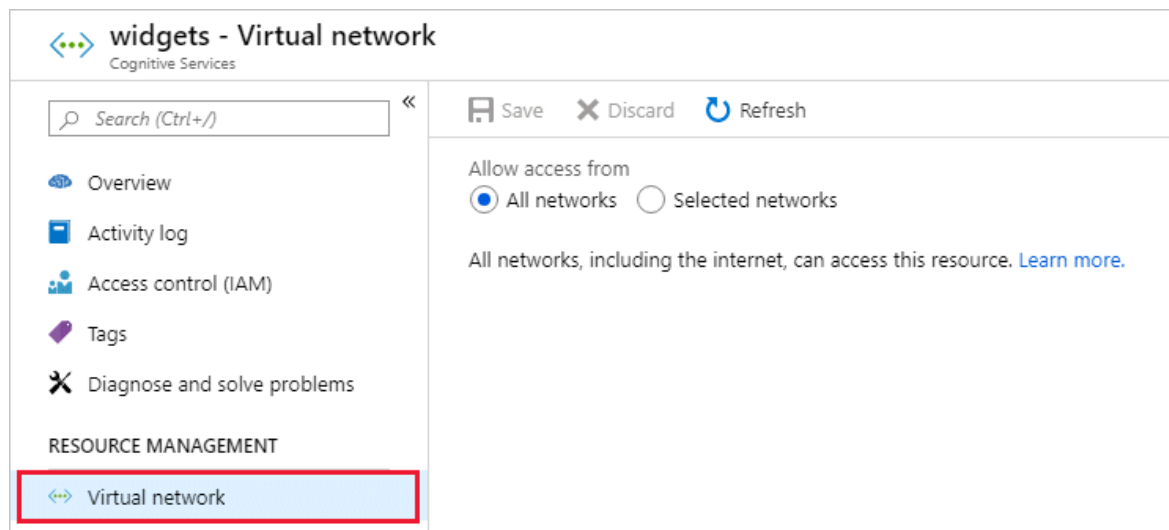
Making changes to network rules can impact your applications' ability to connect to Azure Cognitive Services. Setting the default network rule to **deny** blocks all access to the data unless specific network rules that **grant** access are also applied. Be sure to grant access to any allowed networks using network rules before you change the default rule to deny access. If you are allow listing IP addresses for your on-premises network, be sure to add all possible outgoing public IP addresses from your on-premises network.

Managing default network access rules

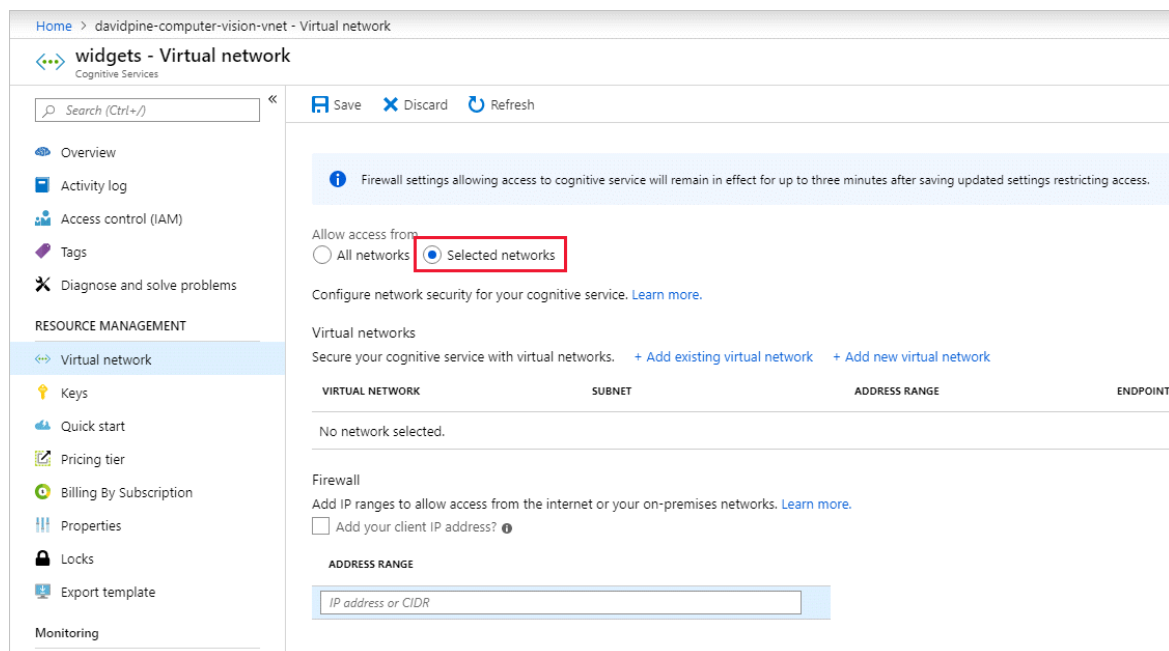
You can manage default network access rules for Cognitive Services resources through the Azure portal, PowerShell, or the Azure CLI.

- [Azure portal](#)
- [PowerShell](#)
- [Azure CLI](#)

1. Go to the Cognitive Services resource you want to secure.
2. Select the **RESOURCE MANAGEMENT** menu called **Virtual network**.



3. To deny access by default, choose to allow access from **Selected networks**. With the **Selected networks** setting alone, unaccompanied by configured **Virtual networks** or **Address ranges** - all access is effectively denied. When all access is denied, requests attempting to consume the Cognitive Services resource aren't permitted. The Azure portal, Azure PowerShell or, Azure CLI can still be used to configure the Cognitive Services resource.
4. To allow traffic from all networks, choose to allow access from **All networks**.



5. Select **Save** to apply your changes.

Grant access from a virtual network

You can configure Cognitive Services resources to allow access only from specific subnets. The allowed subnets may belong to a VNet in the same subscription, or in a different subscription, including subscriptions belonging to a different Azure Active Directory tenant.

Enable a [service endpoint](#) for Azure Cognitive Services within the VNet. The service endpoint routes traffic from the VNet through an optimal path to the Azure Cognitive Services service. The identities of the subnet and the virtual network are also transmitted with each request. Administrators can then configure network rules for the Cognitive Services resource that allow requests to be received from specific subnets in a VNet. Clients granted access via these network rules must continue to meet the authorization requirements of the Cognitive Services resource to access the data.

Each Cognitive Services resource supports up to 100 virtual network rules, which may be combined with [IP network rules](#).

Required permissions

To apply a virtual network rule to a Cognitive Services resource, the user must have the appropriate permissions for the subnets being added. The required permission is the default *Contributor* role, or the *Cognitive Services Contributor* role. Required permissions can also be added to custom role definitions.

Cognitive Services resource and the virtual networks granted access may be in different subscriptions, including subscriptions that are a part of a different Azure AD tenant.

NOTE

Configuration of rules that grant access to subnets in virtual networks that are a part of a different Azure Active Directory tenant are currently only supported through Powershell, CLI and REST APIs. Such rules cannot be configured through the Azure portal, though they may be viewed in the portal.

Managing virtual network rules

You can manage virtual network rules for Cognitive Services resources through the Azure portal, PowerShell, or the Azure CLI.

- [Azure portal](#)
- [PowerShell](#)
- [Azure CLI](#)

1. Go to the Cognitive Services resource you want to secure.
2. Select the **RESOURCE MANAGEMENT** menu called **Virtual network**.
3. Check that you've selected to allow access from **Selected networks**.
4. To grant access to a virtual network with an existing network rule, under **Virtual networks**, select **Add existing virtual network**.

The screenshot shows the 'Virtual network' configuration page in the Azure portal. At the top, there's a header 'widgets - Virtual network' with 'Cognitive Services' below it. Below the header are buttons for 'Save', 'Discard', and 'Refresh'. A blue information banner states: 'Firewall settings allowing access to cognitive service will remain in effect for up to three minutes after saving updated settings restricting access.' Under 'Allow access from', the 'Selected networks' radio button is selected. A link 'Learn more.' is provided for configuring network security. In the 'Virtual networks' section, the text 'Secure your cognitive service with virtual networks.' is followed by two buttons: '+ Add existing virtual network' (highlighted with a red box) and '+ Add new virtual network'. Below this is a table with columns 'VIRTUAL NETWORK', 'SUBNET', and 'ADDRESS RANGE'. The table currently shows 'No network selected.' At the bottom, there's a 'Firewall' section with a link 'Learn more.' and a checkbox 'Add your client IP address?' which is unchecked. Below the checkbox is an 'ADDRESS RANGE' label and a text input field containing the placeholder 'IP address or CIDR'.

5. Select the **Virtual networks** and **Subnets** options, and then select **Enable**.

Create virtual network

Name

widgets-vnet

Address space

10.1.0.0/16

10.1.0.0 - 10.1.255.255 (65536 addresses)

Subscription

widgets-subscription

Resource group

widgets-resource-group

Create new

Location

(US) West US 2

Subnet

Name

default

Address range

10.1.0.0/24

10.1.0.0 - 10.1.0.255 (256 addresses)

DDoS protection

☒ Basic
 ☐ Standard

Service endpoint

Microsoft.CognitiveServices

Firewall

☒ Disabled
 ☐ Enabled

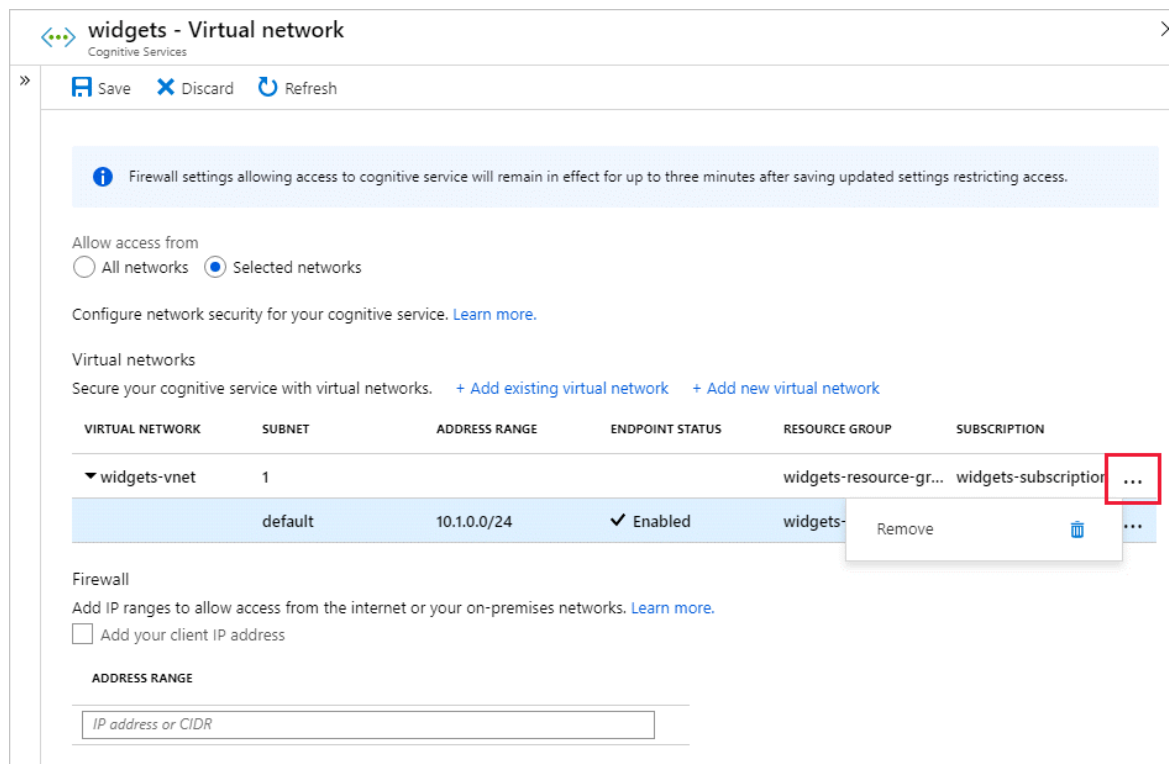
Create

NOTE

If a service endpoint for Azure Cognitive Services wasn't previously configured for the selected virtual network and subnets, you can configure it as part of this operation.

Presently, only virtual networks belonging to the same Azure Active Directory tenant are shown for selection during rule creation. To grant access to a subnet in a virtual network belonging to another tenant, please use Powershell, CLI or REST APIs.

- To remove a virtual network or subnet rule, select ... to open the context menu for the virtual network or subnet, and select **Remove**.



9. Select **Save** to apply your changes.

IMPORTANT

Be sure to [set the default rule to deny](#), or network rules have no effect.

Grant access from an internet IP range

You can configure Cognitive Services resources to allow access from specific public internet IP address ranges. This configuration grants access to specific services and on-premises networks, effectively blocking general internet traffic.

Provide allowed internet address ranges using [CIDR notation](#) in the form `16.17.18.0/24` or as individual IP addresses like `16.17.18.19`.

TIP

Small address ranges using "/31" or "/32" prefix sizes are not supported. These ranges should be configured using individual IP address rules.

IP network rules are only allowed for **public internet** IP addresses. IP address ranges reserved for private networks (as defined in [RFC 1918](#)) aren't allowed in IP rules. Private networks include addresses that start with `10.*`, `172.16.* - 172.31.*`, and `192.168.*`.

Only IPV4 addresses are supported at this time. Each Cognitive Services resource supports up to 100 IP network rules, which may be combined with [Virtual network rules](#).

Configuring access from on-premises networks

To grant access from your on-premises networks to your Cognitive Services resource with an IP network rule, you must identify the internet facing IP addresses used by your network. Contact your network administrator for help.

If you're using [ExpressRoute](#) on-premises for public peering or Microsoft peering, you'll need to identify the NAT

IP addresses. For public peering, each ExpressRoute circuit by default uses two NAT IP addresses. Each is applied to Azure service traffic when the traffic enters the Microsoft Azure network backbone. For Microsoft peering, the NAT IP addresses that are used are either customer provided or are provided by the service provider. To allow access to your service resources, you must allow these public IP addresses in the resource IP firewall setting. To find your public peering ExpressRoute circuit IP addresses, [open a support ticket with ExpressRoute](#) via the Azure portal. Learn more about [NAT for ExpressRoute public and Microsoft peering](#).

Managing IP network rules

You can manage IP network rules for Cognitive Services resources through the Azure portal, PowerShell, or the Azure CLI.

- [Azure portal](#)
- [PowerShell](#)
- [Azure CLI](#)

1. Go to the Cognitive Services resource you want to secure.
2. Select the **RESOURCE MANAGEMENT** menu called **Virtual network**.
3. Check that you've selected to allow access from **Selected networks**.
4. To grant access to an internet IP range, enter the IP address or address range (in [CIDR format](#)) under **Firewall > Address Range**. Only valid public IP (non-reserved) addresses are accepted.

widgets - Virtual network
Cognitive Services

» Save Discard Refresh

Firewall settings allowing access to cognitive service will remain in effect for up to three minutes after saving updated settings restricting access.

Allow access from
☐ All networks ☒ Selected networks

Configure network security for your cognitive service. [Learn more.](#)

Virtual networks
Secure your cognitive service with virtual networks. [+ Add existing virtual network](#) [+ Add new virtual network](#)

VIRTUAL NETWORK	SUBNET	ADDRESS RANGE
No network selected.		

Firewall
Add IP ranges to allow access from the internet or your on-premises networks. [Learn more.](#)

☐ Add your client IP address? ⓘ

ADDRESS RANGE

✓ 🗑️

IP address or CIDR

5. To remove an IP network rule, select the trash can icon next to the address range.

widgets - Virtual network
Cognitive Services

Save Discard Refresh

Firewall settings allowing access to cognitive service will remain in effect for up to three minutes after saving updated settings restricting access.

Allow access from
☐ All networks ☒ Selected networks

Configure network security for your cognitive service. [Learn more.](#)

Virtual networks
Secure your cognitive service with virtual networks. [+ Add existing virtual network](#) [+ Add new virtual network](#)

VIRTUAL NETWORK	SUBNET	ADDRESS RANGE
No network selected.		

Firewall
Add IP ranges to allow access from the internet or your on-premises networks. [Learn more.](#)

☐ Add your client IP address? ⓘ

ADDRESS RANGE

173.0.0.0/16

IP address or CIDR

6. Select **Save** to apply your changes.

IMPORTANT

Be sure to [set the default rule to deny](#), or network rules have no effect.

Use private endpoints

You can use [private endpoints](#) for your Cognitive Services resources to allow clients on a virtual network (VNet) to securely access data over a [Private Link](#). The private endpoint uses an IP address from the VNet address space for your Cognitive Services resource. Network traffic between the clients on the VNet and the resource traverses the VNet and a private link on the Microsoft backbone network, eliminating exposure from the public internet.

Private endpoints for Cognitive Services resources let you:

- Secure your Cognitive Services resource by configuring the firewall to block all connections on the public endpoint for the Cognitive Services service.
- Increase security for the VNet, by enabling you to block exfiltration of data from the VNet.
- Securely connect to Cognitive Services resources from on-premises networks that connect to the VNet using [VPN](#) or [ExpressRoutes](#) with private-peering.

Conceptual overview

A private endpoint is a special network interface for an Azure resource in your [VNet](#). Creating a private endpoint for your Cognitive Services resource provides secure connectivity between clients in your VNet and your resource. The private endpoint is assigned an IP address from the IP address range of your VNet. The connection between the private endpoint and the Cognitive Services service uses a secure private link.

Applications in the VNet can connect to the service over the private endpoint seamlessly, using the same connection strings and authorization mechanisms that they would use otherwise. The exception is the Speech Services, which require a separate endpoint. See the section on [Private endpoints with the Speech Services](#). Private endpoints can be used with all protocols supported by the Cognitive Services resource, including REST.

Private endpoints can be created in subnets that use [Service Endpoints](#). Clients in a subnet can connect to one

Cognitive Services resource using private endpoint, while using service endpoints to access others.

When you create a private endpoint for a Cognitive Services resource in your VNet, a consent request is sent for approval to the Cognitive Services resource owner. If the user requesting the creation of the private endpoint is also an owner of the resource, this consent request is automatically approved.

Cognitive Services resource owners can manage consent requests and the private endpoints, through the '*Private endpoints*' tab for the Cognitive Services resource in the [Azure portal](#).

Private endpoints

When creating the private endpoint, you must specify the Cognitive Services resource it connects to. For more information on creating a private endpoint, see:

- [Create a private endpoint using the Private Link Center in the Azure portal](#)
- [Create a private endpoint using Azure CLI](#)
- [Create a private endpoint using Azure PowerShell](#)

Connecting to private endpoints

Clients on a VNet using the private endpoint should use the same connection string for the Cognitive Services resource as clients connecting to the public endpoint. The exception is the Speech Services, which require a separate endpoint. See the section on [Private endpoints with the Speech Services](#). We rely upon DNS resolution to automatically route the connections from the VNet to the Cognitive Services resource over a private link.

We create a [private DNS zone](#) attached to the VNet with the necessary updates for the private endpoints, by default. However, if you're using your own DNS server, you may need to make additional changes to your DNS configuration. The section on [DNS changes](#) below describes the updates required for private endpoints.

Private endpoints with the Speech Services

See [Using Speech Services with private endpoints provided by Azure Private Link](#).

DNS changes for private endpoints

When you create a private endpoint, the DNS CNAME resource record for the Cognitive Services resource is updated to an alias in a subdomain with the prefix '*privatelink*'. By default, we also create a [private DNS zone](#), corresponding to the '*privatelink*' subdomain, with the DNS A resource records for the private endpoints.

When you resolve the endpoint URL from outside the VNet with the private endpoint, it resolves to the public endpoint of the Cognitive Services resource. When resolved from the VNet hosting the private endpoint, the endpoint URL resolves to the private endpoint's IP address.

This approach enables access to the Cognitive Services resource using the same connection string for clients in the VNet hosting the private endpoints and clients outside the VNet.

If you are using a custom DNS server on your network, clients must be able to resolve the fully qualified domain name (FQDN) for the Cognitive Services resource endpoint to the private endpoint IP address. Configure your DNS server to delegate your private link subdomain to the private DNS zone for the VNet.

TIP

When using a custom or on-premises DNS server, you should configure your DNS server to resolve the Cognitive Services resource name in the '*privatelink*' subdomain to the private endpoint IP address. You can do this by delegating the '*privatelink*' subdomain to the private DNS zone of the VNet, or configuring the DNS zone on your DNS server and adding the DNS A records.

For more information on configuring your own DNS server to support private endpoints, refer to the following articles:

- [Name resolution for resources in Azure virtual networks](#)
- [DNS configuration for private endpoints](#)

Pricing

For pricing details, see [Azure Private Link pricing](#).

Next steps

- Explore the various [Azure Cognitive Services](#)
- Learn more about [Azure Virtual Network Service Endpoints](#)

Configure customer-managed keys with Azure Key Vault for Cognitive Services

8/17/2021 • 2 minutes to read • [Edit Online](#)

The process to enable Customer-Managed Keys with Azure Key Vault for Cognitive Services varies by product. Use these links for service-specific instructions:

Vision

- [Custom Vision encryption of data at rest](#)
- [Face Services encryption of data at rest](#)
- [Form Recognizer encryption of data at rest](#)

Language

- [Language Understanding service encryption of data at rest](#)
- [QnA Maker encryption of data at rest](#)
- [Translator encryption of data at rest](#)

Speech

- [Speech encryption of data at rest](#)

Decision

- [Content Moderator encryption of data at rest](#)
- [Personalizer encryption of data at rest](#)

Next steps

- [What is Azure Key Vault?](#)
- [Cognitive Services Customer-Managed Key Request Form](#)

Authenticate requests to Azure Cognitive Services

7/22/2021 • 8 minutes to read • [Edit Online](#)

Each request to an Azure Cognitive Service must include an authentication header. This header passes along a subscription key or access token, which is used to validate your subscription for a service or group of services. In this article, you'll learn about three ways to authenticate a request and the requirements for each.

- Authenticate with a [single-service](#) or [multi-service](#) subscription key
- Authenticate with a [token](#)
- Authenticate with [Azure Active Directory \(AAD\)](#)

Prerequisites

Before you make a request, you need an Azure account and an Azure Cognitive Services subscription. If you already have an account, go ahead and skip to the next section. If you don't have an account, we have a guide to get you set up in minutes: [Create a Cognitive Services account for Azure](#).

You can get your subscription key from the [Azure portal](#) after [creating your account](#).

Authentication headers

Let's quickly review the authentication headers available for use with Azure Cognitive Services.

HEADER	DESCRIPTION
Ocp-Apim-Subscription-Key	Use this header to authenticate with a subscription key for a specific service or a multi-service subscription key.
Ocp-Apim-Subscription-Region	This header is only required when using a multi-service subscription key with the Translator service . Use this header to specify the subscription region.
Authorization	Use this header if you are using an authentication token. The steps to perform a token exchange are detailed in the following sections. The value provided follows this format: <code>Bearer <TOKEN></code> .

Authenticate with a single-service subscription key

The first option is to authenticate a request with a subscription key for a specific service, like Translator. The keys are available in the Azure portal for each resource that you've created. To use a subscription key to authenticate a request, it must be passed along as the `Ocp-Apim-Subscription-Key` header.

These sample requests demonstrates how to use the `Ocp-Apim-Subscription-Key` header. Keep in mind, when using this sample you'll need to include a valid subscription key.

This is a sample call to the Bing Web Search API:

```
curl -X GET 'https://api.cognitive.microsoft.com/bing/v7.0/search?q=Welsch%20Pembroke%20Corgis' \  
-H 'Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY' | json_pp
```

This is a sample call to the Translator service:

```
curl -X POST 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=en&to=de' \
-H 'Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY' \
-H 'Content-Type: application/json' \
--data-raw ' [{ "text": "How much for the cup of coffee?" } ]' | json_pp
```

The following video demonstrates using a Cognitive Services key.

Authenticate with a multi-service subscription key

WARNING

At this time, the multi-service key doesn't support: QnA Maker, Immersive Reader, Personalizer, and Anomaly Detector.

This option also uses a subscription key to authenticate requests. The main difference is that a subscription key is not tied to a specific service, rather, a single key can be used to authenticate requests for multiple Cognitive Services. See [Cognitive Services pricing](#) for information about regional availability, supported features, and pricing.

The subscription key is provided in each request as the `Ocp-Apim-Subscription-Key` header.



Supported regions

When using the multi-service subscription key to make a request to `api.cognitive.microsoft.com`, you must include the region in the URL. For example: `westus.api.cognitive.microsoft.com`.

When using multi-service subscription key with the Translator service, you must specify the subscription region with the `Ocp-Apim-Subscription-Region` header.

Multi-service authentication is supported in these regions:

- `australiaeast`
- `brazilsouth`
- `canadacentral`
- `centralindia`
- `eastasia`

- eastus
- japaneast
- northeurope
- southcentralus
- southeastasia
- uksouth
- westcentralus
- westeurope
- westus
- westus2
- francecentral
- koreacentral
- northcentralus
- southafricanorth
- uanorth
- switzerlandnorth

Sample requests

This is a sample call to the Bing Web Search API:

```
curl -X GET 'https://YOUR-REGION.api.cognitive.microsoft.com/bing/v7.0/search?q=Welsch%20Pembroke%20Corgis' \
-H 'Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY' | json_pp
```

This is a sample call to the Translator service:

```
curl -X POST 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=en&to=de' \
-H 'Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY' \
-H 'Ocp-Apim-Subscription-Region: YOUR_SUBSCRIPTION_REGION' \
-H 'Content-Type: application/json' \
--data-raw ' [{ "text": "How much for the cup of coffee?" } ]' | json_pp
```

Authenticate with an authentication token

Some Azure Cognitive Services accept, and in some cases require, an authentication token. Currently, these services support authentication tokens:

- Text Translation API
- Speech Services: Speech-to-text REST API
- Speech Services: Text-to-speech REST API

NOTE

QnA Maker also uses the Authorization header, but requires an endpoint key. For more information, see [QnA Maker: Get answer from knowledge base](#).

WARNING

The services that support authentication tokens may change over time, please check the API reference for a service before using this authentication method.

Both single service and multi-service subscription keys can be exchanged for authentication tokens. Authentication tokens are valid for 10 minutes.

Authentication tokens are included in a request as the `Authorization` header. The token value provided must be preceded by `Bearer`, for example: `Bearer YOUR_AUTH_TOKEN`.

Sample requests

Use this URL to exchange a subscription key for an authentication token:

```
https://YOUR-REGION.api.cognitive.microsoft.com/sts/v1.0/issueToken
```

```
curl -v -X POST \
  "https://YOUR-REGION.api.cognitive.microsoft.com/sts/v1.0/issueToken" \
  -H "Content-type: application/x-www-form-urlencoded" \
  -H "Content-length: 0" \
  -H "Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY"
```

These multi-service regions support token exchange:

- `australiaeast`
- `brazilsouth`
- `canadacentral`
- `centralindia`
- `eastasia`
- `eastus`
- `japaneast`
- `northeurope`
- `southcentralus`
- `southeastasia`
- `uksouth`
- `westcentralus`
- `westeurope`
- `westus`
- `westus2`

After you get an authentication token, you'll need to pass it in each request as the `Authorization` header. This is a sample call to the Translator service:

```
curl -X POST 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=en&to=de' \
  -H 'Authorization: Bearer YOUR_AUTH_TOKEN' \
  -H 'Content-Type: application/json' \
  --data-raw ' [{ "text": "How much for the cup of coffee?" } ]' | json_pp
```

Authenticate with Azure Active Directory

AAD authentication always needs to be used together with custom subdomain name of your Azure resource. [Regional endpoints](#) do not support AAD authentication.

In the following sections, you'll use either the Azure Cloud Shell environment or the Azure CLI to create a subdomain, assign roles, and obtain a bearer token to call the Azure Cognitive Services. If you get stuck, links are provided in each section with all available options for each command in Azure Cloud Shell/Azure CLI.

The first step is to create a custom subdomain. If you want to use an existing Cognitive Services resource which does not have custom subdomain name, follow the instructions in [Cognitive Services Custom Subdomains](#) to enable custom subdomain for your resource.

- ```
Set-AzContext -SubscriptionName <SubscriptionName>
```

- ```
$account = New-AzCognitiveServicesAccount -ResourceGroupName <RESOURCE_GROUP_NAME> -name
<ACCOUNT_NAME> -Type <ACCOUNT_TYPE> -SkuName <SUBSCRIPTION_TYPE> -Location <REGION> -
CustomSubdomainName <UNIQUE SUBDOMAIN>
```

- Keep in mind that Azure role assignments may take up to five minutes to propagate.

- ```
$SecureStringPassword = ConvertTo-SecureString -String <YOUR_PASSWORD> -AsPlainText -Force

$app = New-AzADApplication -DisplayName <APP_DISPLAY_NAME> -IdentifierUri <APP_URI> -Password $SecureStringPassword
```

```
New-AzADServicePrincipal -ApplicationId <APPLICATION ID>
```

#### NOTE

If you register an application in the Azure portal, this step is completed for you.

3. The last step is to [assign the "Cognitive Services User" role](#) to the service principal (scoped to the resource). By assigning a role, you're granting service principal access to this resource. You can grant the same service principal access to multiple resources in your subscription.

#### NOTE

The ObjectId of the service principal is used, not the ObjectId for the application. The ACCOUNT\_ID will be the Azure resource Id of the Cognitive Services account you created. You can find Azure resource Id from "properties" of the resource in Azure portal.

```
New-AzRoleAssignment -ObjectId <SERVICE_PRINCIPAL_OBJECTID> -Scope <ACCOUNT_ID> -RoleDefinitionName "Cognitive Services User"
```

### Sample request

In this sample, a password is used to authenticate the service principal. The token provided is then used to call the Computer Vision API.

1. Get your TenantId:

```
$context=Get-AzContext
$context.Tenant.Id
```

2. Get a token:

#### NOTE

If you're using Azure Cloud Shell, the `SecureClientSecret` class isn't available.

- [PowerShell](#)
- [Azure Cloud Shell](#)

```
$authContext = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext" -
ArgumentList "https://login.windows.net/<TENANT_ID>"
$secureSecretObject = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.SecureClientSecret" -
ArgumentList $SecureStringPassword
$clientCredential = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.ClientCredential" -
ArgumentList $app.ApplicationId, $secureSecretObject
$token=$authContext.AcquireTokenAsync("https://cognitiveservices.azure.com/",
$clientCredential).Result
$token
```

3. Call the Computer Vision API:

```
$url = $account.Endpoint+"vision/v1.0/models"
$result = Invoke-RestMethod -Uri $url -Method Get -Headers
@{"Authorization"=$token.CreateAuthorizationHeader()} -Verbose
$result | ConvertTo-Json
```



Alternatively, the service principal can be authenticated with a certificate. Besides service principal, user principal is also supported by having permissions delegated through another AAD application. In this case, instead of passwords or certificates, users would be prompted for two-factor authentication when acquiring token.

## Authorize access to managed identities

Cognitive Services support Azure Active Directory (Azure AD) authentication with [managed identities for Azure resources](#). Managed identities for Azure resources can authorize access to Cognitive Services resources using Azure AD credentials from applications running in Azure virtual machines (VMs), function apps, virtual machine scale sets, and other services. By using managed identities for Azure resources together with Azure AD authentication, you can avoid storing credentials with your applications that run in the cloud.

### Enable managed identities on a VM

Before you can use managed identities for Azure resources to authorize access to Cognitive Services resources from your VM, you must enable managed identities for Azure resources on the VM. To learn how to enable managed identities for Azure Resources, see:

- [Azure portal](#)
- [Azure PowerShell](#)
- [Azure CLI](#)
- [Azure Resource Manager template](#)
- [Azure Resource Manager client libraries](#)

For more information about managed identities, see [Managed identities for Azure resources](#).

## See also

- [What is Cognitive Services?](#)
- [Cognitive Services pricing](#)
- [Custom subdomains](#)

# Configure data loss prevention for Azure Cognitive Services

9/3/2021 • 2 minutes to read • [Edit Online](#)

Cognitive Services data loss prevention capabilities allow customers to configure the list of outbound URLs their Cognitive Services resources are allowed to access. This creates another level of control for customers to prevent data loss. In this article, we'll cover the steps required to enable the data loss prevention feature for Cognitive Services resources.

## Prerequisites

Before you make a request, you need an Azure account and an Azure Cognitive Services subscription. If you already have an account, go ahead and skip to the next section. If you don't have an account, we have a guide to get you set up in minutes: [Create a Cognitive Services account for Azure](#).

You can get your subscription key from the [Azure portal](#) after [creating your account](#).

## Enabling data loss prevention

There are two parts to enable data loss prevention. First the property `restrictOutboundNetworkAccess` must be set to true. When this is set to true, you also need to provide the list of approved URLs. The list of URLs is added to the `allowedFqdnList` property. The `allowedFqdnList` property contains an array of comma-separated URLs.

### NOTE

- The `allowedFqdnList` property value supports a maximum of 1000 URLs.
- The property supports both IP addresses and fully qualified domain names i.e., `www.microsoft.com`, values.
- It can take up to 15 minutes for the updated list to take effect.

- [Azure CLI](#)
- [PowerShell](#)

1. Install the [Azure CLI](#) and [sign in](#), or select [Try it](#).
2. View the details of the Cognitive Services resource.

```
az cognitiveservices account show \
-g "myresourcegroup" -n "myaccount" \
```

3. View the current properties of the Cognitive Services resource.

```
az rest -m get \
-u /subscriptions/{subscription ID}/resourceGroups/{resource
group}/providers/Microsoft.CognitiveServices/accounts/{account name}?api-version=2021-04-30 \
```

4. Configure the `restrictOutboundNetworkAccess` property and update the `allowedFqdnList` with the approved URLs

```
az rest -m patch \
 -u /subscriptions/{subscription ID}}/resourceGroups/{resource
group}/providers/Microsoft.CognitiveServices/accounts/{account name}?api-version=2021-04-30 \
 -b '{"properties": { "restrictOutboundNetworkAccess": true, "allowedFqdnList": ["microsoft.com"
] }}'
```

## Supported services

The following services support data loss prevention configuration:

- Computer Vision
- Content Moderator
- Custom Vision
- Face
- Speech Service
- QnA Maker

## Next steps

- [Configure Virtual Networks](#)

# Azure Policy Regulatory Compliance controls for Azure Cognitive Services

9/17/2021 • 11 minutes to read • [Edit Online](#)

[Regulatory Compliance in Azure Policy](#) provides Microsoft created and managed initiative definitions, known as *built-ins*, for the **compliance domains** and **security controls** related to different compliance standards. This page lists the **compliance domains** and **security controls** for Azure Cognitive Services. You can assign the built-ins for a **security control** individually to help make your Azure resources compliant with the specific standard.

The title of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Policy Version** column to view the source on the [Azure Policy GitHub repo](#).

## IMPORTANT

Each control below is associated with one or more [Azure Policy](#) definitions. These policies may help you [assess compliance](#) with the control; however, there often is not a one-to-one or complete match between a control and one or more policies. As such, **Compliant** in Azure Policy refers only to the policies themselves; this doesn't ensure you're fully compliant with all requirements of a control. In addition, the compliance standard includes controls that aren't addressed by any Azure Policy definitions at this time. Therefore, compliance in Azure Policy is only a partial view of your overall compliance status. The associations between controls and Azure Policy Regulatory Compliance definitions for these compliance standards may change over time.

## Azure Security Benchmark

The [Azure Security Benchmark](#) provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Azure Security Benchmark, see the [Azure Security Benchmark mapping files](#).

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Azure Security Benchmark](#).

| DOMAIN           | CONTROL ID | CONTROL TITLE                           | POLICY<br>(AZURE PORTAL)                                                                              | POLICY VERSION<br>(GITHUB) |
|------------------|------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------|----------------------------|
| Network Security | NS-1       | Implement security for internal traffic | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0                      |
| Network Security | NS-1       | Implement security for internal traffic | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0                      |
| Data Protection  | DP-5       | Encrypt sensitive data at rest          | <a href="#">Cognitive Services accounts should enable data encryption with a customer-managed key</a> | 2.0.0                      |

## CMMC Level 3

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CMMC Level 3](#). For more information about this compliance standard, see [Cybersecurity Maturity Model Certification \(CMMC\)](#).

| DOMAIN                   | CONTROL ID | CONTROL TITLE                                                                                                                                           | POLICY (AZURE PORTAL)                                                            | POLICY VERSION (GITHUB) |
|--------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|-------------------------|
| Access Control           | AC.1.001   | Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems). | <a href="#">Cognitive Services accounts should disable public network access</a> | 2.0.0                   |
| Access Control           | AC.1.001   | Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems). | <a href="#">Cognitive Services accounts should restrict network access</a>       | 2.0.0                   |
| Access Control           | AC.1.002   | Limit information system access to the types of transactions and functions that authorized users are permitted to execute.                              | <a href="#">Cognitive Services accounts should disable public network access</a> | 2.0.0                   |
| Access Control           | AC.1.002   | Limit information system access to the types of transactions and functions that authorized users are permitted to execute.                              | <a href="#">Cognitive Services accounts should restrict network access</a>       | 2.0.0                   |
| Access Control           | AC.2.016   | Control the flow of CUI in accordance with approved authorizations.                                                                                     | <a href="#">Cognitive Services accounts should disable public network access</a> | 2.0.0                   |
| Access Control           | AC.2.016   | Control the flow of CUI in accordance with approved authorizations.                                                                                     | <a href="#">Cognitive Services accounts should restrict network access</a>       | 2.0.0                   |
| Configuration Management | CM.3.068   | Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.                                              | <a href="#">Cognitive Services accounts should disable public network access</a> | 2.0.0                   |

| DOMAIN                               | CONTROL ID | CONTROL TITLE                                                                                                                                                                                        | POLICY                                                                                | POLICY VERSION |
|--------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|----------------|
| Configuration Management             | CM.3.068   | Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.                                                                                           | Cognitive Services accounts should restrict network access                            | 2.0.0          |
| System and Communications Protection | SC.1.175   | Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems. | Cognitive Services accounts should disable public network access                      | 2.0.0          |
| System and Communications Protection | SC.1.175   | Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems. | Cognitive Services accounts should restrict network access                            | 2.0.0          |
| System and Communications Protection | SC.3.177   | Employ FIPS-validated cryptography when used to protect the confidentiality of CUI.                                                                                                                  | Cognitive Services accounts should enable data encryption with a customer-managed key | 2.0.0          |
| System and Communications Protection | SC.3.183   | Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).                                                          | Cognitive Services accounts should disable public network access                      | 2.0.0          |

| DOMAIN                               | CONTROL ID | CONTROL TITLE                                                                                                                               | POLICY                                                                     | POLICY VERSION |
|--------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|----------------|
| System and Communications Protection | SC.3.183   | Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception). | <a href="#">Cognitive Services accounts should restrict network access</a> | 2.0.0          |

## FedRAMP High

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - FedRAMP High](#). For more information about this compliance standard, see [FedRAMP High](#).

| DOMAIN         | CONTROL ID | CONTROL TITLE                       | POLICY<br>(AZURE PORTAL)                                                                      | POLICY VERSION<br>(GITHUB) |
|----------------|------------|-------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------|
| Access Control | AC-2       | Account Management                  | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-2 (1)   | Automated System Account Management | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-2 (7)   | Role-based Schemes                  | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-3       | Access Enforcement                  | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-4       | Information Flow Enforcement        | <a href="#">Cognitive Services accounts should disable public network access</a>              | 2.0.0                      |
| Access Control | AC-4       | Information Flow Enforcement        | <a href="#">Cognitive Services accounts should restrict network access</a>                    | 2.0.0                      |
| Access Control | AC-4       | Information Flow Enforcement        | <a href="#">Cognitive Services should use private link</a>                                    | 2.0.0                      |
| Access Control | AC-17      | Remote Access                       | <a href="#">Cognitive Services should use private link</a>                                    | 2.0.0                      |

| DOMAIN                               | CONTROL ID | CONTROL TITLE                                            | POLICY                                                                                                | POLICY VERSION |
|--------------------------------------|------------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------------|
| Access Control                       | AC-17 (1)  | Automated Monitoring / Control                           | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| Identification and Authentication    | IA-2       | Identification and Authentication (organizational Users) | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a>         | 1.0.0          |
| Identification and Authentication    | IA-4       | Identifier Management                                    | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a>         | 1.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                                      | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                                      | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                                      | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                            | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                            | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                            | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| System and Communications Protection | SC-12      | Cryptographic Key Establishment and Management           | <a href="#">Cognitive Services accounts should enable data encryption with a customer-managed key</a> | 2.0.0          |

## FedRAMP Moderate

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - FedRAMP Moderate](#). For more information about this compliance standard, see [FedRAMP Moderate](#).



| DOMAIN                               | CONTROL ID | CONTROL TITLE                                            | POLICY<br>(AZURE PORTAL)                                                      | POLICY VERSION<br>(GITHUB) |
|--------------------------------------|------------|----------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------|
| Access Control                       | AC-2       | Account Management                                       | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-2 (1)   | Automated System Account Management                      | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-2 (7)   | Role-based Schemes                                       | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-3       | Access Enforcement                                       | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-4       | Information Flow Enforcement                             | Cognitive Services accounts should disable public network access              | 2.0.0                      |
| Access Control                       | AC-4       | Information Flow Enforcement                             | Cognitive Services accounts should restrict network access                    | 2.0.0                      |
| Access Control                       | AC-4       | Information Flow Enforcement                             | Cognitive Services should use private link                                    | 2.0.0                      |
| Access Control                       | AC-17      | Remote Access                                            | Cognitive Services should use private link                                    | 2.0.0                      |
| Access Control                       | AC-17 (1)  | Automated Monitoring / Control                           | Cognitive Services should use private link                                    | 2.0.0                      |
| Identification and Authentication    | IA-2       | Identification and Authentication (organizational Users) | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Identification and Authentication    | IA-4       | Identifier Management                                    | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| System and Communications Protection | SC-7       | Boundary Protection                                      | Cognitive Services accounts should disable public network access              | 2.0.0                      |

| DOMAIN                               | CONTROL ID | CONTROL TITLE                                  | POLICY                                                                                                | POLICY VERSION |
|--------------------------------------|------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------------|
| System and Communications Protection | SC-7       | Boundary Protection                            | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                            | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                  | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                  | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                  | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| System and Communications Protection | SC-12      | Cryptographic Key Establishment and Management | <a href="#">Cognitive Services accounts should enable data encryption with a customer-managed key</a> | 2.0.0          |

## New Zealand ISM Restricted

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - New Zealand ISM Restricted](#). For more information about this compliance standard, see [New Zealand ISM Restricted](#).

| DOMAIN           | CONTROL ID | CONTROL TITLE                                               | POLICY<br>(AZURE PORTAL)                                                                              | POLICY VERSION<br>(GITHUB) |
|------------------|------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------------------------|
| Cryptography     | CR-3       | 17.1.46 Reducing storage and physical transfer requirements | <a href="#">Cognitive Services accounts should enable data encryption with a customer-managed key</a> | 2.0.0                      |
| Gateway security | GS-2       | 19.1.11 Using Gateways                                      | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0                      |

| DOMAIN           | CONTROL ID | CONTROL TITLE                        | POLICY                                                                     | POLICY VERSION |
|------------------|------------|--------------------------------------|----------------------------------------------------------------------------|----------------|
| Gateway security | GS-3       | 19.1.12<br>Configuration of Gateways | <a href="#">Cognitive Services accounts should restrict network access</a> | 2.0.0          |

## NIST SP 800-53 Rev. 4

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 4](#). For more information about this compliance standard, see [NIST SP 800-53 Rev. 4](#).

| DOMAIN         | CONTROL ID | CONTROL TITLE                       | POLICY<br>(AZURE PORTAL)                                                                      | POLICY VERSION<br>(GITHUB) |
|----------------|------------|-------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------|
| Access Control | AC-2       | Account Management                  | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-2 (1)   | Automated System Account Management | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-2 (7)   | Role-based Schemes                  | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-3       | Access Enforcement                  | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a> | 1.0.0                      |
| Access Control | AC-4       | Information Flow Enforcement        | <a href="#">Cognitive Services accounts should disable public network access</a>              | 2.0.0                      |
| Access Control | AC-4       | Information Flow Enforcement        | <a href="#">Cognitive Services accounts should restrict network access</a>                    | 2.0.0                      |
| Access Control | AC-4       | Information Flow Enforcement        | <a href="#">Cognitive Services should use private link</a>                                    | 2.0.0                      |
| Access Control | AC-17      | Remote Access                       | <a href="#">Cognitive Services should use private link</a>                                    | 2.0.0                      |

| DOMAIN                               | CONTROL ID | CONTROL TITLE                                            | POLICY                                                                                                | POLICY VERSION |
|--------------------------------------|------------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------------|
| Access Control                       | AC-17 (1)  | Automated Monitoring / Control                           | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| Identification and Authentication    | IA-2       | Identification and Authentication (organizational Users) | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a>         | 1.0.0          |
| Identification and Authentication    | IA-4       | Identifier Management                                    | <a href="#">Cognitive Services accounts should have local authentication methods disabled</a>         | 1.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                                      | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                                      | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                                      | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                            | <a href="#">Cognitive Services accounts should disable public network access</a>                      | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                            | <a href="#">Cognitive Services accounts should restrict network access</a>                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                            | <a href="#">Cognitive Services should use private link</a>                                            | 2.0.0          |
| System and Communications Protection | SC-12      | Cryptographic Key Establishment and Management           | <a href="#">Cognitive Services accounts should enable data encryption with a customer-managed key</a> | 2.0.0          |

## NIST SP 800-53 Rev. 5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 5](#). For more information about this compliance standard, see [NIST SP 800-53 Rev. 5](#).

| DOMAIN                               | CONTROL ID | CONTROL TITLE                                            | POLICY<br>(AZURE PORTAL)                                                      | POLICY VERSION<br>(GITHUB) |
|--------------------------------------|------------|----------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------|
| Access Control                       | AC-2       | Account Management                                       | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-2 (1)   | Automated System Account Management                      | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-2 (7)   | Privileged User Accounts                                 | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-3       | Access Enforcement                                       | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Access Control                       | AC-4       | Information Flow Enforcement                             | Cognitive Services accounts should disable public network access              | 2.0.0                      |
| Access Control                       | AC-4       | Information Flow Enforcement                             | Cognitive Services accounts should restrict network access                    | 2.0.0                      |
| Access Control                       | AC-4       | Information Flow Enforcement                             | Cognitive Services should use private link                                    | 2.0.0                      |
| Access Control                       | AC-17      | Remote Access                                            | Cognitive Services should use private link                                    | 2.0.0                      |
| Access Control                       | AC-17 (1)  | Monitoring and Control                                   | Cognitive Services should use private link                                    | 2.0.0                      |
| Identification and Authentication    | IA-2       | Identification and Authentication (organizational Users) | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| Identification and Authentication    | IA-4       | Identifier Management                                    | Cognitive Services accounts should have local authentication methods disabled | 1.0.0                      |
| System and Communications Protection | SC-7       | Boundary Protection                                      | Cognitive Services accounts should disable public network access              | 2.0.0                      |

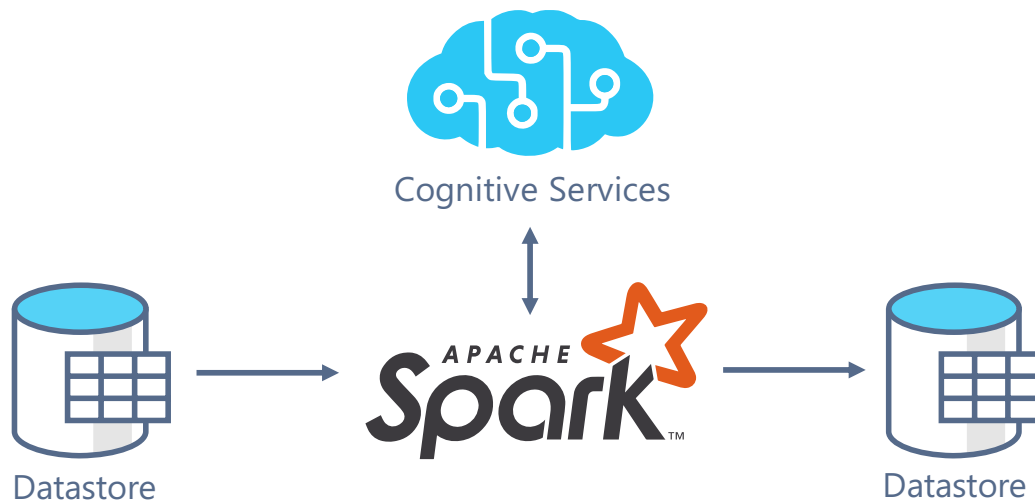
| DOMAIN                               | CONTROL ID | CONTROL TITLE                                  | POLICY                                                                                | POLICY VERSION |
|--------------------------------------|------------|------------------------------------------------|---------------------------------------------------------------------------------------|----------------|
| System and Communications Protection | SC-7       | Boundary Protection                            | Cognitive Services accounts should restrict network access                            | 2.0.0          |
| System and Communications Protection | SC-7       | Boundary Protection                            | Cognitive Services should use private link                                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                  | Cognitive Services accounts should disable public network access                      | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                  | Cognitive Services accounts should restrict network access                            | 2.0.0          |
| System and Communications Protection | SC-7 (3)   | Access Points                                  | Cognitive Services should use private link                                            | 2.0.0          |
| System and Communications Protection | SC-12      | Cryptographic Key Establishment and Management | Cognitive Services accounts should enable data encryption with a customer-managed key | 2.0.0          |

## Next steps

- Learn more about [Azure Policy Regulatory Compliance](#).
- See the built-ins on the [Azure Policy GitHub repo](#).

# Azure Cognitive Services for Big Data

3/5/2021 • 5 minutes to read • [Edit Online](#)



The Azure Cognitive Services for Big Data lets users channel terabytes of data through Cognitive Services using [Apache Spark™](#). With the Cognitive Services for Big Data, it's easy to create large-scale intelligent applications with any datastore.

With Cognitive Services for Big Data you can embed continuously improving, intelligent models directly into Apache Spark™ and SQL computations. These tools liberate developers from low-level networking details, so that they can focus on creating smart, distributed applications.

## Features and benefits

Cognitive Services for Big Data can use services from any region in the world, as well as [containerized Cognitive Services](#). Containers support low or no connectivity deployments with ultra-low latency responses.

Containerized Cognitive Services can be run locally, directly on the worker nodes of your Spark cluster, or on an external orchestrator like Kubernetes.

## Supported services

[Cognitive Services](#), accessed through APIs and SDKs, help developers build intelligent applications without having AI or data science skills. With Cognitive Services you can make your applications see, hear, speak, understand, and reason. To use the Cognitive Services, your application must send data to the service over the network. Once received, the service sends an intelligent response in return. The following services are available for big data workloads:

### Vision

| SERVICE NAME                    | SERVICE DESCRIPTION                                                                                                          |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Computer Vision</a> | The Computer Vision service provides you with access to advanced algorithms for processing images and returning information. |
| <a href="#">Face</a>            | The Face service provides access to advanced face algorithms, enabling face attribute detection and recognition.             |

## Speech

| SERVICE NAME                   | SERVICE DESCRIPTION                                                                                                                                        |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Speech service</a> | The Speech service provides access to features like speech recognition, speech synthesis, speech translation, and speaker verification and identification. |

## Decision

| SERVICE NAME                     | SERVICE DESCRIPTION                                                                                             |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <a href="#">Anomaly Detector</a> | The Anomaly Detector (Preview) service allows you to monitor and detect abnormalities in your time series data. |

## Language

| SERVICE NAME                   | SERVICE DESCRIPTION                                                                                                                                   |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Text Analytics</a> | The Text Analytics service provides natural language processing over raw text for sentiment analysis, key-phrases extraction, and language detection. |

## Search

| SERVICE NAME                      | SERVICE DESCRIPTION                                                                                      |
|-----------------------------------|----------------------------------------------------------------------------------------------------------|
| <a href="#">Bing Image Search</a> | The Bing Image Search service returns a display of images determined to be relevant to the user's query. |

# Supported programming languages for Cognitive Services for Big Data

The Cognitive Services for Big Data are built on Apache Spark. Apache Spark is a distributed computing library that supports Java, Scala, Python, R, and many other languages. These languages are currently supported.

## Python

We provide a PySpark API in the `mm1spark.cognitive` namespace of [Microsoft ML for Apache Spark](#). For more information, see the [Python Developer API](#). For usage examples, see the [Python Samples](#).

## Scala and Java

We provide a Scala and Java-based Spark API in the `com.microsoft.ml.spark.cognitive` namespace of [Microsoft ML for Apache Spark](#). For more information, see the [Scala Developer API](#). For usage examples, see the [Scala Samples](#).

# Supported platforms and connectors

The Cognitive Services for Big Data requires Apache Spark. There are several Apache Spark platforms that support the Cognitive Services for Big Data.

## Azure Databricks

[Azure Databricks](#) is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform. It provides one-click setup, streamlined work-flows, and an interactive workspace that supports collaboration between data scientists, data engineers, and business analysts.



## Azure Synapse Analytics

[Azure Synapse Analytics](#) is an enterprise data warehouse that uses massive parallel processing. With Synapse Analytics, you can quickly run complex queries across petabytes of data. Azure Synapse Analytics provides managed Spark Pools to run Spark Jobs with an intuitive Jupyter Notebook Interface.

## Azure Kubernetes Service

[Azure Kubernetes Service \(AKS\)](#) orchestrates Docker Containers and distributed applications at massive scales. AKS is a managed Kubernetes offering that simplifies using Kubernetes in Azure. Kubernetes can enable fine-grained control of Cognitive Service scale, latency, and networking. However, we recommend using Azure Databricks or Azure Synapse Analytics if you're unfamiliar with Apache Spark.

## Data Connectors

Once you have a Spark Cluster, the next step is connecting to your data. Apache Spark has a broad collection of database connectors. These connectors allow applications to work with large datasets no matter where they're stored. For more information about supported databases and connectors, see the [list of supported datasources for Azure Databricks](#).

# Concepts

## Spark

[Apache Spark™](#) is a unified analytics engine for large-scale data processing. Its parallel processing framework boosts performance of big data and analytic applications. Spark can operate as both a batch and stream processing system, without changing core application code.

The basis of Spark is the DataFrame: a tabular collection of data distributed across the Apache Spark worker nodes. A Spark DataFrame is like a table in a relational database or a data frame in R/Python, but with limitless scale. DataFrames can be constructed from many sources such as: structured data files, tables in Hive, or external databases. Once your data is in a Spark DataFrame, you can:

- Do SQL-style computations such as join and filter tables.
- Apply functions to large datasets using MapReduce style parallelism.
- Apply Distributed Machine Learning using Microsoft Machine Learning for Apache Spark.
- Use the Cognitive Services for Big Data to enrich your data with ready-to-use intelligent services.

## Microsoft Machine Learning for Apache Spark (MMLSpark)

[Microsoft Machine Learning for Apache Spark](#) (MMLSpark) is an open-source, distributed machine learning library (ML) built on Apache Spark. The Cognitive Services for Big Data is included in this package. Additionally, MMLSpark contains several other ML tools for Apache Spark, such as LightGBM, Vowpal Wabbit, OpenCV, LIME, and more. With MMLSpark, you can build powerful predictive and analytical models from any Spark datasource.

## HTTP on Spark

Cognitive Services for Big Data is an example of how we can integrate intelligent web services with big data. Web services power many applications across the globe and most services communicate through the Hypertext Transfer Protocol (HTTP). To work with *arbitrary* web services at large scales, we provide HTTP on Spark. With HTTP on Spark, you can pass terabytes of data through any web service. Under the hood, we use this technology to power Cognitive Services for Big Data.

## Developer samples

- [Recipe: Predictive Maintenance](#)
- [Recipe: Intelligent Art Exploration](#)

## Blog posts

- [Learn more about how Cognitive Services work on Apache Spark™](#)
- [Saving Snow Leopards with Deep Learning and Computer Vision on Spark](#)
- [Microsoft Research Podcast: MMLSpark, empowering AI for Good with Mark Hamilton](#)
- [Academic Whitepaper: Large Scale Intelligent Microservices](#)

## Webinars and videos

- [The Azure Cognitive Services on Spark: Clusters with Embedded Intelligent Services](#)
- [Spark Summit Keynote: Scalable AI for Good](#)
- [The Cognitive Services for Big Data in Cosmos DB](#)
- [Lightning Talk on Large Scale Intelligent Microservices](#)

## Next steps

- [Getting Started with the Cognitive Services for Big Data](#)
- [Simple Python Examples](#)
- [Simple Scala Examples](#)

# Getting started

7/19/2021 • 4 minutes to read • [Edit Online](#)

Setting up your environment is the first step to building a pipeline for your data. After your environment is ready, running a sample is quick and easy.

In this article, we'll perform these steps to get you started:

1. [Create a Cognitive Services resource](#)
2. [Create an Apache Spark Cluster](#)
3. [Try a sample](#)

## Create a Cognitive Services resource

To use the Big Data Cognitive Services, you must first create a Cognitive Service for your workflow. There are two main types of Cognitive Services: cloud services hosted in Azure and containerized services managed by users. We recommend starting with the simpler cloud-based Cognitive Services.

### Cloud services

Cloud-based Cognitive Services are intelligent algorithms hosted in Azure. These services are ready for use without training, you just need an internet connection. You can [create a Cognitive Service in the Azure portal](#) or with the [Azure CLI](#).

### Containerized services (optional)

If your application or workload uses large datasets, requires private networking, or can't contact the cloud, communicating with cloud services might be impossible. In this situation, containerized Cognitive Services have these benefits:

- **Low Connectivity:** You can deploy containerized Cognitive Services in any computing environment, both on-cloud and off. If your application can't contact the cloud, consider deploying containerized Cognitive Services on your application.
- **Low Latency:** Because containerized services don't require the round-trip communication to/from the cloud, responses are returned with much lower latencies.
- **Privacy and Data Security:** You can deploy containerized services into private networks, so that sensitive data doesn't leave the network.
- **High Scalability:** Containerized services don't have "rate limits" and run on user-managed computers. So, you can scale Cognitive Services without end to handle much larger workloads.

Follow [this guide](#) to create a containerized Cognitive Service.

## Create an Apache Spark cluster

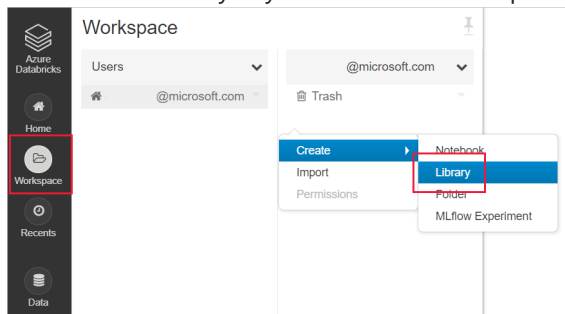
[Apache Spark™](#) is a distributed computing framework designed for big-data data processing. Users can work with Apache Spark in Azure with services like Azure Databricks, Azure Synapse Analytics, HDInsight, and Azure Kubernetes Services. To use the Big Data Cognitive Services, you must first create a cluster. If you already have a Spark cluster, feel free to try an example.

### Azure Databricks

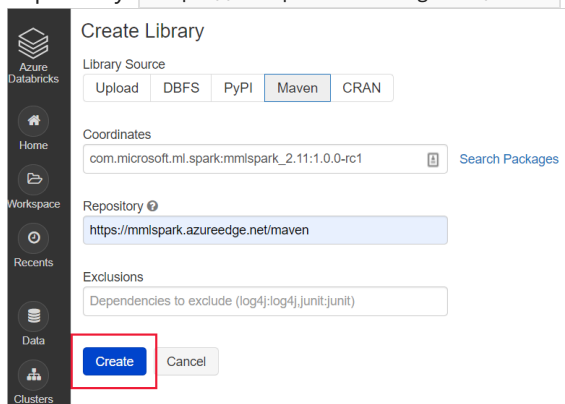
Azure Databricks is an Apache Spark-based analytics platform with a one-click setup, streamlined workflows,

and an interactive workspace. It's often used to collaborate between data scientists, engineers, and business analysts. To use the Big Data Cognitive Services on Azure Databricks, follow these steps:

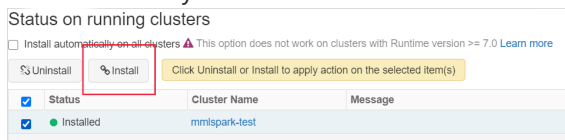
1. [Create an Azure Databricks workspace](#)
  2. [Create a Spark cluster in Databricks](#)
  3. Install the Big Data Cognitive Services
- Create a new library in your databricks workspace



- Input the following maven coordinates Coordinates: `com.microsoft.ml.spark:mm1spark_2.11:1.0.0-rc3`  
Repository: `https://mm1spark.azureedge.net/maven`



- Install the library onto a cluster



### Azure Synapse Analytics (optional)

Optionally, you can use Synapse Analytics to create a spark cluster. Azure Synapse Analytics brings together enterprise data warehousing and big data analytics. It gives you the freedom to query data on your terms, using either serverless on-demand or provisioned resources at scale. To get started using Azure Synapse Analytics, follow these steps:

1. [Create a Synapse Workspace \(preview\)](#).
2. [Create a new serverless Apache Spark pool \(preview\) using the Azure portal](#).

In Azure Synapse Analytics, Big Data for Cognitive Services is installed by default.

### Azure Kubernetes Service

If you're using containerized Cognitive Services, one popular option for deploying Spark alongside containers is the Azure Kubernetes Service.

To get started on Azure Kubernetes Service, follow these steps:

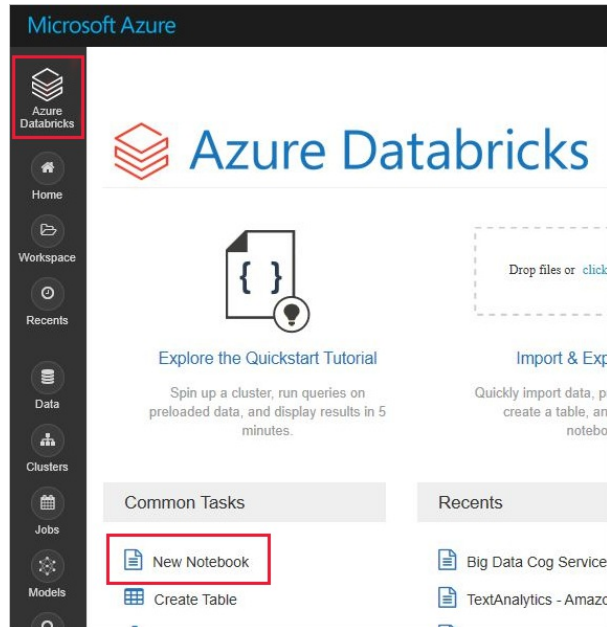
1. [Deploy an Azure Kubernetes Service \(AKS\) cluster using the Azure portal](#)
2. [Install the Apache Spark 2.4.0 helm chart](#)
3. [Install a cognitive service container using Helm](#)

## Try a sample

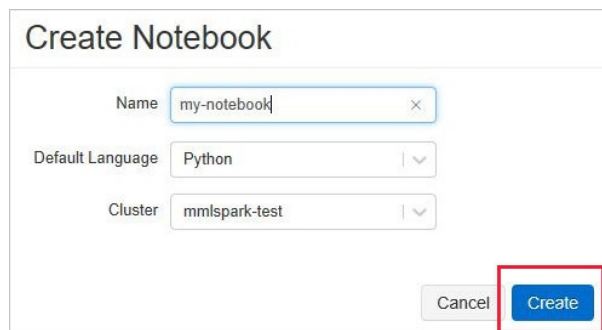
After you set up your Spark cluster and environment, you can run a short sample. This section demonstrates how to use the Big Data for Cognitive Services in Azure Databricks.

First, you can create a notebook in Azure Databricks. For other Spark cluster providers, use their notebooks or Spark Submit.

1. Create a new Databricks notebook, by choosing **New Notebook** from the **Azure Databricks** menu.



2. In the **Create Notebook** dialog box, enter a name, select **Python** as the language, and select the Spark cluster that you created earlier.



Select **Create**.

3. Paste this code snippet into your new notebook.

```

from mmlspark.cognitive import *
from pyspark.sql.functions import col

Add your subscription key from Text Analytics (or a general Cognitive Service key)
service_key = "ADD-SUBSCRIPTION-KEY-HERE"

df = spark.createDataFrame([
 ("I am so happy today, its sunny!", "en-US"),
 ("I am frustrated by this rush hour traffic", "en-US"),
 ("The cognitive services on spark aint bad", "en-US"),
], ["text", "language"])

sentiment = (TextSentiment()
 .setTextCol("text")
 .setLocation("eastus")
 .setSubscriptionKey(service_key)
 .setOutputCol("sentiment")
 .setErrorCol("error")
 .setLanguageCol("language"))

results = sentiment.transform(df)

Show the results in a table
display(results.select("text", col("sentiment")[0].getItem("score").alias("sentiment")))

```

1. Get your subscription key from the **Keys and Endpoint** menu from your Text Analytics dashboard in the Azure portal.
2. Replace the subscription key placeholder in your Databricks notebook code with your subscription key.
3. Select the play, or triangle, symbol in the upper right of your notebook cell to run the sample. Optionally, select **Run All** at the top of your notebook to run all cells. The answers will display below the cell in a table.

### Expected results

| TEXT                                      | SENTIMENT |
|-------------------------------------------|-----------|
| I am so happy today, its sunny!           | 0.978959  |
| I am frustrated by this rush hour traffic | 0.0237956 |
| The cognitive services on spark aint bad  | 0.888896  |

## Next steps

- [Short Python Examples](#)
- [Short Scala Examples](#)
- [Recipe: Predictive Maintenance](#)
- [Recipe: Intelligent Art Exploration](#)

# Python Samples for Cognitive Services for Big Data

3/5/2021 • 6 minutes to read • [Edit Online](#)

The following snippets are ready to run and will help get you started with using Cognitive Services on Spark with Python.

The samples in this article use these Cognitive Services:

- Text Analytics - get the sentiment (or mood) of a set of sentences.
- Computer Vision - get the tags (one-word descriptions) associated with a set of images.
- Bing Image Search - search the web for images related to a natural language query.
- Speech-to-text - transcribe audio files to extract text-based transcripts.
- Anomaly Detector - detect anomalies within a time series data.

## Prerequisites

1. Follow the steps in [Getting started](#) to set up your Azure Databricks and Cognitive Services environment. This tutorial shows you how to install MMLSpark and how to create your Spark cluster in Databricks.
2. After you create a new notebook in Azure Databricks, copy the **Shared code** below and paste into a new cell in your notebook.
3. Choose a service sample, below, and copy paste it into a second new cell in your notebook.
4. Replace any of the service subscription key placeholders with your own key.
5. Choose the run button (triangle icon) in the upper right corner of the cell, then select **Run Cell**.
6. View results in a table below the cell.

## Shared code

To get started, we'll need to add this code to the project:

```
from mmlspark.cognitive import *

A general Cognitive Services key for Text Analytics and Computer Vision (or use separate keys that belong
to each service)
service_key = "ADD_YOUR_SUBSCRIPTION_KEY"
A Bing Search v7 subscription key
bing_search_key = "ADD_YOUR_SUBSCRIPTION_KEY"
An Anomaly Dectector subscription key
anomaly_key = "ADD_YOUR_SUBSCRIPTION_KEY"

Validate the key
assert service_key != "ADD_YOUR_SUBSCRIPTION_KEY"
```

## Text Analytics sample

The [Text Analytics](#) service provides several algorithms for extracting intelligent insights from text. For example, we can find the sentiment of given input text. The service will return a score between 0.0 and 1.0 where low scores indicate negative sentiment and high score indicates positive sentiment. This sample uses three simple sentences and returns the sentiment for each.

```

from pyspark.sql.functions import col

Create a dataframe that's tied to it's column names
df = spark.createDataFrame([
 ("I am so happy today, its sunny!", "en-US"),
 ("I am frustrated by this rush hour traffic", "en-US"),
 ("The cognitive services on spark aint bad", "en-US"),
], ["text", "language"])

Run the Text Analytics service with options
sentiment = (TextSentiment()
 .setTextCol("text")
 .setLocation("eastus")
 .setSubscriptionKey(service_key)
 .setOutputCol("sentiment")
 .setErrorCol("error")
 .setLanguageCol("language"))

Show the results of your text query in a table format
display(sentiment.transform(df).select("text", col("sentiment")[0].getItem("sentiment").alias("sentiment")))

```

### Expected result

| TEXT                                      | SENTIMENT |
|-------------------------------------------|-----------|
| I am so happy today, its sunny!           | positive  |
| I am frustrated by this rush hour traffic | negative  |
| The cognitive services on spark aint bad  | positive  |

## Computer Vision sample

[Computer Vision](#) analyzes images to identify structure such as faces, objects, and natural-language descriptions. In this sample, we tag a list of images. Tags are one-word descriptions of things in the image like recognizable objects, people, scenery, and actions.

```

Create a dataframe with the image URLs
df = spark.createDataFrame([
 ("https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg",),
 ("https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg",),
 ("https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg",)
], ["image",])

Run the Computer Vision service. Analyze Image extracts information from/about the images.
analysis = (AnalyzeImage()
 .setLocation("eastus")
 .setSubscriptionKey(service_key)
 .setVisualFeatures(["Categories", "Color", "Description", "Faces", "Objects", "Tags"])
 .setOutputCol("analysis_results")
 .setImageUrlCol("image")
 .setErrorCol("error"))

Show the results of what you wanted to pull out of the images.
display(analysis.transform(df).select("image", "analysis_results.description.tags"))

```



## Expected result

| IMAGE                                                                                                                                                                                                                                                                     | TAGS                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg">https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg</a> | ['skating' 'person' 'man' 'outdoor' 'riding' 'sport' 'skateboard' 'young' 'board' 'shirt' 'air' 'black' 'park' 'boy' 'side' 'jumping' 'trick' 'ramp' 'doing' 'flying']                                                                                                                             |
| <a href="https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg">https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg</a>         | ['dog' 'outdoor' 'fence' 'wooden' 'small' 'brown' 'building' 'sitting' 'front' 'bench' 'standing' 'table' 'walking' 'board' 'beach' 'white' 'holding' 'bridge' 'track']                                                                                                                            |
| <a href="https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg">https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg</a>     | ['outdoor' 'grass' 'house' 'building' 'old' 'home' 'front' 'small' 'church' 'stone' 'large' 'grazing' 'yard' 'green' 'sitting' 'leading' 'sheep' 'brick' 'bench' 'street' 'white' 'country' 'clock' 'sign' 'parked' 'field' 'standing' 'garden' 'water' 'red' 'horse' 'man' 'tall' 'fire' 'group'] |

## Bing Image Search sample

[Bing Image Search](#) searches the web to retrieve images related to a user's natural language query. In this sample, we use a text query that looks for images with quotes. It returns a list of image URLs that contain photos related to our query.

```
from pyspark.ml import PipelineModel

Number of images Bing will return per query
imgsPerBatch = 10
A list of offsets, used to page into the search results
offsets = [(i*imgsPerBatch,) for i in range(100)]
Since web content is our data, we create a dataframe with options on that data: offsets
bingParameters = spark.createDataFrame(offsets, ["offset"])

Run the Bing Image Search service with our text query
bingSearch = (BingImageSearch()
 .setSubscriptionKey(bing_search_key)
 .setOffsetCol("offset")
 .setQuery("Martin Luther King Jr. quotes")
 .setCount(imgsPerBatch)
 .setOutputCol("images"))

Transformer that extracts and flattens the richly structured output of Bing Image Search into a simple URL column
getUrl = BingImageSearch.getUrlTransformer("images", "url")

This displays the full results returned, uncomment to use
display(bingSearch.transform(bingParameters))

Since we have two services, they are put into a pipeline
pipeline = PipelineModel(stages=[bingSearch, getUrl])

Show the results of your search: image URLs
display(pipeline.transform(bingParameters))
```

## Expected result

| URL                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://iheartintelligence.com/wp-content/uploads/2019/01/powerful-quotes-martin-luther-king-jr.jpg">https://iheartintelligence.com/wp-content/uploads/2019/01/powerful-quotes-martin-luther-king-jr.jpg</a> |

| URL                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="http://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr-Quotes-16.jpg">http://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr-Quotes-16.jpg</a>                                                 |
| <a href="http://www.sofreshandsogreen.com/wp-content/uploads/2012/01/martin-luther-king-jr-quote-sofreshandsogreendotcom.jpg">http://www.sofreshandsogreen.com/wp-content/uploads/2012/01/martin-luther-king-jr-quote-sofreshandsogreendotcom.jpg</a> |
| <a href="https://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr-Quotes-18.jpg">https://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr-Quotes-18.jpg</a>                                               |
| <a href="https://tsal-eszuskq0bptlfh8awbb.stackpathdns.com/wp-content/uploads/2018/01/MartinLutherKingQuotes.jpg">https://tsal-eszuskq0bptlfh8awbb.stackpathdns.com/wp-content/uploads/2018/01/MartinLutherKingQuotes.jpg</a>                         |

## Speech-to-Text sample

The [Speech-to-text](#) service converts streams or files of spoken audio to text. In this sample, we transcribe two audio files. The first file is easy to understand, and the second is more challenging.

```
Create a dataframe with our audio URLs, tied to the column called "url"
df = spark.createDataFrame([("https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav",),
 ("https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3",)
], ["url"])

Run the Speech-to-text service to translate the audio into text
speech_to_text = (SpeechToTextSDK()
 .setSubscriptionKey(service_key)
 .setLocation("eastus")
 .setOutputCol("text")
 .setAudioDataCol("url")
 .setLanguage("en-US")
 .setProfanity("Masked"))

Show the results of the translation
display(speech_to_text.transform(df).select("url", "text.DisplayText"))
```

### Expected result

| URL                                                                                                                                               | DISPLAYTEXT                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav">https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav</a> | Custom speech provides tools that allow you to visually inspect the recognition quality of a model by comparing audio data with the corresponding recognition result from the custom speech portal. You can playback uploaded audio and determine if the provided recognition result is correct. This tool allows you to quickly inspect quality of Microsoft's baseline speech to text model or a trained custom model without having to transcribe any audio data. |
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3">https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3</a> | Add a gentleman Sir thinking visual check.                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3">https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3</a> | I hear me.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3">https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3</a> | I like the reassurance for radio that I can hear it as well.                                                                                                                                                                                                                                                                                                                                                                                                         |

## Anomaly Detector sample

[Anomaly Detector](#) is great for detecting irregularities in your time series data. In this sample, we use the service to find anomalies in the entire time series.

```
from pyspark.sql.functions import lit

Create a dataframe with the point data that Anomaly Detector requires
df = spark.createDataFrame([
 ("1972-01-01T00:00:00Z", 826.0),
 ("1972-02-01T00:00:00Z", 799.0),
 ("1972-03-01T00:00:00Z", 890.0),
 ("1972-04-01T00:00:00Z", 900.0),
 ("1972-05-01T00:00:00Z", 766.0),
 ("1972-06-01T00:00:00Z", 805.0),
 ("1972-07-01T00:00:00Z", 821.0),
 ("1972-08-01T00:00:00Z", 20000.0),
 ("1972-09-01T00:00:00Z", 883.0),
 ("1972-10-01T00:00:00Z", 898.0),
 ("1972-11-01T00:00:00Z", 957.0),
 ("1972-12-01T00:00:00Z", 924.0),
 ("1973-01-01T00:00:00Z", 881.0),
 ("1973-02-01T00:00:00Z", 837.0),
 ("1973-03-01T00:00:00Z", 9000.0)
], ["timestamp", "value"]).withColumn("group", lit("series1"))

Run the Anomaly Detector service to look for irregular data
anomaly_detector = (SimpleDetectAnomalies()
 .setSubscriptionKey(anomaly_key)
 .setLocation("eastus")
 .setTimestampCol("timestamp")
 .setValueCol("value")
 .setOutputCol("anomalies")
 .setGroupbyCol("group")
 .setGranularity("monthly"))

Show the full results of the analysis with the anomalies marked as "True"
display(anomaly_detector.transform(df).select("timestamp", "value", "anomalies.isAnomaly"))
```

### Expected result

| TIMESTAMP            | VALUE | ISANOMALY |
|----------------------|-------|-----------|
| 1972-01-01T00:00:00Z | 826   | False     |
| 1972-02-01T00:00:00Z | 799   | False     |
| 1972-03-01T00:00:00Z | 890   | False     |
| 1972-04-01T00:00:00Z | 900   | False     |
| 1972-05-01T00:00:00Z | 766   | False     |
| 1972-06-01T00:00:00Z | 805   | False     |
| 1972-07-01T00:00:00Z | 821   | False     |
| 1972-08-01T00:00:00Z | 20000 | True      |
| 1972-09-01T00:00:00Z | 883   | False     |

| TIMESTAMP            | VALUE | ISANOMALY |
|----------------------|-------|-----------|
| 1972-10-01T00:00:00Z | 898   | False     |
| 1972-11-01T00:00:00Z | 957   | False     |
| 1972-12-01T00:00:00Z | 924   | False     |
| 1973-01-01T00:00:00Z | 881   | False     |
| 1973-02-01T00:00:00Z | 837   | False     |
| 1973-03-01T00:00:00Z | 9000  | True      |

## Arbitrary web APIs

With HTTP on Spark, any web service can be used in your big data pipeline. In this example, we use the [World Bank API](#) to get information about various countries around the world.

```
from requests import Request
from mm1spark.io.http import HTTPTransformer, http_udf
from pyspark.sql.functions import udf, col

Use any requests from the python requests library
def world_bank_request(country):
 return Request("GET", "http://api.worldbank.org/v2/country/{}?format=json".format(country))

Create a dataframe with spcificies which countries we want data on
df = (spark.createDataFrame([("br",), ("usa",)], ["country"]))
 .withColumn("request", http_udf(world_bank_request)(col("country"))))

Much faster for big data because of the concurrency :)
client = (HTTPTransformer()
 .setConcurrency(3)
 .setInputCol("request")
 .setOutputCol("response"))

Get the body of the response
def get_response_body(resp):
 return resp.entity.content.decode()

Show the details of the country data returned
display(client.transform(df).select("country", udf(get_response_body)(col("response")).alias("response")))
```

### Expected result

| COUNTRY | RESPONSE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| br      | {   "page":1,"pages":1,"per_page":"50","total":1},   {"id":"BRA","iso2Code":"BR","name":"Brazil","region":   {"id":"LCN","iso2code":"ZJ","value":"Latin America & Caribbean"},   "adminregion":{"id":"LAC","iso2code":"XJ","value":"Latin America & Caribbean (excluding high income)"},   "incomeLevel":{"id":"UMC","iso2code":"XT","value":"Upper middle income"},   "lendingType":{"id":"IBD","iso2code":"XF","value":"IBRD"},   "capitalCity":"Brasilia",   "longitude":"-47.9292","latitude":"-15.7801"} } |

| COUNTRY | RESPONSE                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| usa     | [{"page":1,"pages":1,"per_page":"50","total":1},<br>[{"id":"USA","iso2Code":"US","name":"United States","region":<br>{"id":"NAC","iso2code":"XU","value":"North<br>America"},"adminregion":<br>{"id":"","iso2code":"","value":""},"incomeLevel":<br>{"id":"HIC","iso2code":"XD","value":"High<br>income"},"lendingType":<br>{"id":"LNX","iso2code":"XX","value":"Not<br>classified"},"capitalCity":"Washington D.C. ","longitude":"-<br>77.032","latitude":"38.8895"}]] |

## See also

- [Recipe: Anomaly Detection](#)
- [Recipe: Art Explorer](#)

# Quick Examples

3/5/2021 • 5 minutes to read • [Edit Online](#)

The following snippets are ready to run and will help get you started with using Cognitive Services on Spark. The samples below are in Scala.

The samples use these Cognitive Services:

- Text Analytics - get the sentiment (or mood) of a set of sentences.
- Computer Vision - get the tags (one-word descriptions) associated with a set of images.
- Bing Image Search - search the web for images related to a natural language query.
- Speech-to-text - transcribe audio files to extract text-based transcripts.
- Anomaly Detector - detect anomalies within a time series data.

## Prerequisites

1. Follow the steps in [Getting started](#) to set up your Azure Databricks and Cognitive Services environment. This tutorial will include how to install MMLSpark and how to create your Spark cluster in Databricks.
2. After you create a new notebook in Azure Databricks, copy the **Shared code** below and paste into a new cell in your notebook.
3. Choose a service sample, below, and copy paste it into a second new cell in your notebook.
4. Replace any of the service subscription key placeholders with your own key.
5. Choose the run button (triangle icon) in the upper right corner of the cell, then select **Run Cell**.
6. View results in a table below the cell.

## Shared code

To get started, add this code to your project:

```
import com.microsoft.ml.spark.cognitive._
import spark.implicits._

val serviceKey = "ADD-YOUR-SUBSCRIPTION-KEY"
val location = "eastus"
```

## Text Analytics

The [Text Analytics](#) service provides several algorithms for extracting intelligent insights from text. For example, we can find the sentiment of given input text. The service will return a score between `0.0` and `1.0` where low scores indicate negative sentiment and high score indicates positive sentiment. The sample below uses three simple sentences and returns the sentiment score for each.

```
import org.apache.spark.sql.functions.col

val df = Seq(
 ("I am so happy today, its sunny!", "en-US"),
 ("I am frustrated by this rush hour traffic", "en-US"),
 ("The cognitive services on spark aint bad", "en-US")
).toDF("text", "language")

val sentiment = new TextSentiment()
 .setTextCol("text")
 .setLocation(location)
 .setSubscriptionKey(serviceKey)
 .setOutputCol("sentiment")
 .setErrorCol("error")
 .setLanguageCol("language")

display(sentiment.transform(df).select(col("text"), col("sentiment"))
(0).getItem("score").alias("sentiment")))
```

### Expected result

| TEXT                                      | SENTIMENT            |
|-------------------------------------------|----------------------|
| I am so happy today, its sunny!           | 0.9789592027664185   |
| I am frustrated by this rush hour traffic | 0.023795604705810547 |
| The cognitive services on spark aint bad  | 0.8888956308364868   |

## Computer Vision

[Computer Vision](#) analyzes images to identify structure such as faces, objects, and natural-language descriptions. In this sample, we tag a list of images. Tags are one-word descriptions of things in the image like recognizable objects, people, scenery, and actions.

```
// Create a dataframe with the image URLs
val df = Seq(
 ("https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg"),
 ("https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg"),
 ("https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg")
).toDF("image")

// Run the Computer Vision service. Analyze Image extracts information from/about the images.
val analysis = new AnalyzeImage()
 .setLocation(location)
 .setSubscriptionKey(serviceKey)
 .setVisualFeatures(Seq("Categories", "Color", "Description", "Faces", "Objects", "Tags"))
 .setOutputCol("results")
 .setImageUrlCol("image")
 .setErrorCol("error")

// Show the results of what you wanted to pull out of the images.
display(analysis.transform(df).select(col("image"),
col("results").getItem("tags").getItem("name").alias("results"))))

// Uncomment for full results with all visual feature requests
//display(analysis.transform(df).select(col("image"), col("results")))
```

## Expected result

| IMAGE                                                                                                                                                                                                                                                                     | TAGS                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg">https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg</a> | ['skating' 'person' 'man' 'outdoor' 'riding' 'sport' 'skateboard' 'young' 'board' 'shirt' 'air' 'black' 'park' 'boy' 'side' 'jumping' 'trick' 'ramp' 'doing' 'flying']                                                                                                                             |
| <a href="https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg">https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg</a>         | ['dog' 'outdoor' 'fence' 'wooden' 'small' 'brown' 'building' 'sitting' 'front' 'bench' 'standing' 'table' 'walking' 'board' 'beach' 'white' 'holding' 'bridge' 'track']                                                                                                                            |
| <a href="https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg">https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg</a>     | ['outdoor' 'grass' 'house' 'building' 'old' 'home' 'front' 'small' 'church' 'stone' 'large' 'grazing' 'yard' 'green' 'sitting' 'leading' 'sheep' 'brick' 'bench' 'street' 'white' 'country' 'clock' 'sign' 'parked' 'field' 'standing' 'garden' 'water' 'red' 'horse' 'man' 'tall' 'fire' 'group'] |

## Bing Image Search

[Bing Image Search](#) searches the web to retrieve images related to a user's natural language query. In this sample, we use a text query that looks for images with quotes. It returns a list of image URLs that contain photos related to our query.

```
import org.apache.spark.ml.Pipeline

// Number of images Bing will return per query
val imgsPerBatch = 10

// A list of offsets, used to page into the search results
val df = (0 until 100).map(o => Tuple1(o*imgsPerBatch)).toSeq.toDF("offset")

// Run the Bing Image Search service with our text query
val bingSearch = new BingImageSearch()
 .setSubscriptionKey(bingSearchKey)
 .setOffsetCol("offset")
 .setQuery("Martin Luther King Jr. quotes")
 .setCount(imgsPerBatch)
 .setOutputCol("images")

// Transformer that extracts and flattens the richly structured output of Bing Image Search into a simple
// URL column
val getUrls = BingImageSearch.getUrlTransformer("images", "url")

// This displays the full results returned, uncomment to use
// display(bingSearch.transform(bingParameters))

// Since we have two services, they are put into a pipeline
val pipeline = new Pipeline().setStages(Array(bingSearch, getUrls))

// Show the results of your search: image URLs
display(pipeline.fit(df).transform(df))
```

## Expected result

| URL                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://iheartintelligence.com/wp-content/uploads/2019/01/powerful-quotes-martin-luther-king-jr.jpg">https://iheartintelligence.com/wp-content/uploads/2019/01/powerful-quotes-martin-luther-king-jr.jpg</a> |
| <a href="http://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr.-Quotes-16.jpg">http://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr.-Quotes-16.jpg</a>               |



| URL                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="http://www.sofreshandsogreen.com/wp-content/uploads/2012/01/martin-luther-king-jr-quote-sofreshandsogreendotcom.jpg">http://www.sofreshandsogreen.com/wp-content/uploads/2012/01/martin-luther-king-jr-quote-sofreshandsogreendotcom.jpg</a> |
| <a href="https://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr.-Quotes-18.jpg">https://everydaypowerblog.com/wp-content/uploads/2014/01/Martin-Luther-King-Jr.-Quotes-18.jpg</a>                                             |
| <a href="https://tsal-eszuskq0bptlfh8awbb.stackpathdns.com/wp-content/uploads/2018/01/MartinLutherKingQuotes.jpg">https://tsal-eszuskq0bptlfh8awbb.stackpathdns.com/wp-content/uploads/2018/01/MartinLutherKingQuotes.jpg</a>                         |

## Speech-to-Text

The [Speech-to-text](#) service converts streams or files of spoken audio to text. In this sample, we transcribe two audio files. The first file is easy to understand, and the second is more challenging.

```
import org.apache.spark.sql.functions.col

// Create a dataframe with audio URLs, tied to the column called "url"
val df = Seq(("https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav"),
 ("https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3")).toDF("url")

// Run the Speech-to-text service to translate the audio into text
val speechToText = new SpeechToTextSDK()
 .setSubscriptionKey(serviceKey)
 .setLocation("eastus")
 .setOutputCol("text")
 .setAudioDataCol("url")
 .setLanguage("en-US")
 .setProfanity("Masked")

// Show the results of the translation
display(speechToText.transform(df).select(col("url"), col("text").getItem("DisplayText")))
```

### Expected result

| URL                                                                                                                                               | DISPLAYTEXT                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav">https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav</a> | Custom speech provides tools that allow you to visually inspect the recognition quality of a model by comparing audio data with the corresponding recognition result from the custom speech portal. You can playback uploaded audio and determine if the provided recognition result is correct. This tool allows you to quickly inspect quality of Microsoft's baseline speech to text model or a trained custom model without having to transcribe any audio data. |
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3">https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3</a> | Add a gentleman Sir thinking visual check.                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3">https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3</a> | I hear me.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3">https://mmlspark.blob.core.windows.net/datasets/Speech/audio3.mp3</a> | I like the reassurance for radio that I can hear it as well.                                                                                                                                                                                                                                                                                                                                                                                                         |

## Anomaly Detector

[Anomaly Detector](#) is great for detecting irregularities in your time series data. In this sample, we use the service to find anomalies in the entire time series.

```

import org.apache.spark.sql.functions.{col, lit}

val anomalyKey = "84a2c303cc7e49f6a44d692c27fb9967"

val df = Seq(
 ("1972-01-01T00:00:00Z", 826.0),
 ("1972-02-01T00:00:00Z", 799.0),
 ("1972-03-01T00:00:00Z", 890.0),
 ("1972-04-01T00:00:00Z", 900.0),
 ("1972-05-01T00:00:00Z", 766.0),
 ("1972-06-01T00:00:00Z", 805.0),
 ("1972-07-01T00:00:00Z", 821.0),
 ("1972-08-01T00:00:00Z", 20000.0),
 ("1972-09-01T00:00:00Z", 883.0),
 ("1972-10-01T00:00:00Z", 898.0),
 ("1972-11-01T00:00:00Z", 957.0),
 ("1972-12-01T00:00:00Z", 924.0),
 ("1973-01-01T00:00:00Z", 881.0),
 ("1973-02-01T00:00:00Z", 837.0),
 ("1973-03-01T00:00:00Z", 9000.0)
).toDF("timestamp", "value").withColumn("group", lit("series1"))

// Run the Anomaly Detector service to look for irregular data
val anomalyDetector = new SimpleDetectAnomalies()
 .setSubscriptionKey(anomalyKey)
 .setLocation("eastus")
 .setTimestampCol("timestamp")
 .setValueCol("value")
 .setOutputCol("anomalies")
 .setGroupbyCol("group")
 .setGranularity("monthly")

// Show the full results of the analysis with the anomalies marked as "True"
display(anomalyDetector.transform(df).select("timestamp", "value", "anomalies.isAnomaly"))

```

## Expected result

| TIMESTAMP            | VALUE | ISANOMALY |
|----------------------|-------|-----------|
| 1972-01-01T00:00:00Z | 826   | False     |
| 1972-02-01T00:00:00Z | 799   | False     |
| 1972-03-01T00:00:00Z | 890   | False     |
| 1972-04-01T00:00:00Z | 900   | False     |
| 1972-05-01T00:00:00Z | 766   | False     |
| 1972-06-01T00:00:00Z | 805   | False     |
| 1972-07-01T00:00:00Z | 821   | False     |
| 1972-08-01T00:00:00Z | 20000 | True      |
| 1972-09-01T00:00:00Z | 883   | False     |
| 1972-10-01T00:00:00Z | 898   | False     |

| TIMESTAMP            | VALUE | ISANOMALY |
|----------------------|-------|-----------|
| 1972-11-01T00:00:00Z | 957   | False     |
| 1972-12-01T00:00:00Z | 924   | False     |
| 1973-01-01T00:00:00Z | 881   | False     |
| 1973-02-01T00:00:00Z | 837   | False     |
| 1973-03-01T00:00:00Z | 9000  | True      |

# Recipe: Predictive maintenance with the Cognitive Services for Big Data

3/22/2021 • 3 minutes to read • [Edit Online](#)

This recipe shows how you can use Azure Synapse Analytics and Cognitive Services on Apache Spark for predictive maintenance of IoT devices. We'll follow along with the [CosmosDB and Synapse Link](#) sample. To keep things simple, in this recipe we'll read the data straight from a CSV file rather than getting streamed data through CosmosDB and Synapse Link. We strongly encourage you to look over the Synapse Link sample.

## Hypothetical scenario

The hypothetical scenario is a Power Plant, where IoT devices are monitoring [steam turbines](#). The IoTSignals collection has Revolutions per minute (RPM) and Megawatts (MW) data for each turbine. Signals from steam turbines are being analyzed and anomalous signals are detected.

There could be outliers in the data in random frequency. In those situations, RPM values will go up and MW output will go down, for circuit protection. The idea is to see the data varying at the same time, but with different signals.

## Prerequisites

- An Azure subscription - [Create one for free](#)
- [Azure Synapse workspace](#) configured with a [serverless Apache Spark pool](#)

## Setup

### Create an Anomaly Detector resource

Azure Cognitive Services are represented by Azure resources that you subscribe to. Create a resource for Translator using the [Azure portal](#) or [Azure CLI](#). You can also:

- View an existing resource in the [Azure portal](#).

Make note of the endpoint and the key for this resource, you'll need it in this guide.

## Enter your service keys

Let's start by adding your key and location.

```
service_key = None # Paste your anomaly detector key here
location = None # Paste your anomaly detector location here

assert (service_key is not None)
assert (location is not None)
```

## Read data into a DataFrame

Next, let's read the IoTSignals file into a DataFrame. Open a new notebook in your Synapse workspace and create a DataFrame from the file.

```
df_signals = spark.read.csv("wasbs://publicwasb@mmlspark.blob.core.windows.net/iot/IoTSignals.csv",
header=True, inferSchema=True)
```

## Run anomaly detection using Cognitive Services on Spark

The goal is to find instances where the signals from the IoT devices were outputting anomalous values so that we can see when something is going wrong and do predictive maintenance. To do that, let's use Anomaly Detector on Spark:

```
from pyspark.sql.functions import col, struct
from mmlspark.cognitive import SimpleDetectAnomalies
from mmlspark.core.spark import FluentAPI

detector = (SimpleDetectAnomalies()
 .setSubscriptionKey(service_key)
 .setLocation(location)
 .setOutputCol("anomalies")
 .setGroupbyCol("grouping")
 .setSensitivity(95)
 .setGranularity("secondly"))

df_anomaly = (df_signals
 .where(col("unitSymbol") == 'RPM')
 .withColumn("timestamp", col("dateTime").cast("string"))
 .withColumn("value", col("measureValue").cast("double"))
 .withColumn("grouping", struct("deviceId"))
 .mlTransform(detector)).cache()

df_anomaly.createOrReplaceTempView('df_anomaly')
```

Let's take a look at the data:

```
df_anomaly.select("timestamp", "value", "deviceId", "anomalies.isAnomaly").show(3)
```

This cell should yield a result that looks like:

| TIMESTAMP           | VALUE | DEVICEID | ISANOMALY |
|---------------------|-------|----------|-----------|
| 2020-05-01 18:33:51 | 3174  | dev-7    | False     |
| 2020-05-01 18:33:52 | 2976  | dev-7    | False     |
| 2020-05-01 18:33:53 | 2714  | dev-7    | False     |

## Visualize anomalies for one of the devices

IoTSignals.csv has signals from multiple IoT devices. We'll focus on a specific device and visualize anomalous outputs from the device.

```
df_anomaly_single_device = spark.sql("""
select
 timestamp,
 measureValue,
 anomalies.expectedValue,
 anomalies.expectedValue + anomalies.upperMargin as expectedUpperValue,
 anomalies.expectedValue - anomalies.lowerMargin as expectedLowerValue,
 case when anomalies.isAnomaly=true then 1 else 0 end as isAnomaly
from
 df_anomaly
where deviceid = 'dev-1' and timestamp < '2020-04-29'
order by timestamp
limit 200""")
```

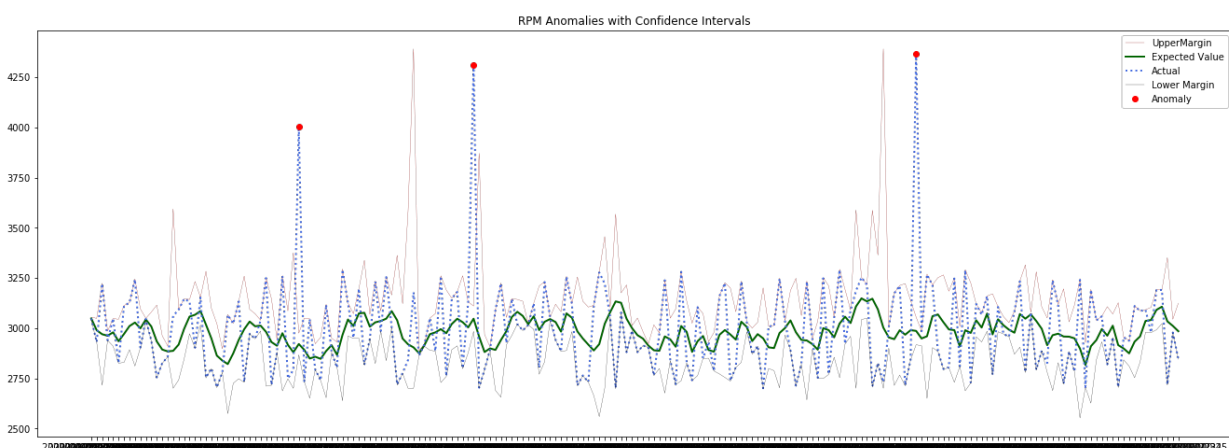
Now that we have created a dataframe that represents the anomalies for a particular device, we can visualize these anomalies:

```
import matplotlib.pyplot as plt
from pyspark.sql.functions import col

adf = df_anomaly_single_device.toPandas()
adf_subset = df_anomaly_single_device.where(col("isAnomaly") == 1).toPandas()

plt.figure(figsize=(23,8))
plt.plot(adf['timestamp'],adf['expectedUpperValue'], color='darkred', linestyle='solid', linewidth=0.25,
label='UpperMargin')
plt.plot(adf['timestamp'],adf['expectedValue'], color='darkgreen', linestyle='solid', linewidth=2,
label='Expected Value')
plt.plot(adf['timestamp'],adf['measureValue'], 'b', color='royalblue', linestyle='dotted', linewidth=2,
label='Actual')
plt.plot(adf['timestamp'],adf['expectedLowerValue'], color='black', linestyle='solid', linewidth=0.25,
label='Lower Margin')
plt.plot(adf_subset['timestamp'],adf_subset['measureValue'], 'ro', label = 'Anomaly')
plt.legend()
plt.title('RPM Anomalies with Confidence Intervals')
plt.show()
```

If successful, your output will look like this:



## Next steps

Learn how to do predictive maintenance at scale with Azure Cognitive Services, Azure Synapse Analytics, and Azure CosmosDB. For more information, see the full sample on [GitHub](#).

# Recipe: Intelligent Art Exploration with the Cognitive Services for Big Data

3/5/2021 • 2 minutes to read • [Edit Online](#)

In this example, we'll use the Cognitive Services for Big Data to add intelligent annotations to the Open Access collection from the Metropolitan Museum of Art (MET). This will enable us to create an intelligent search engine using Azure Search even without manual annotations.

## Prerequisites

- You must have a subscription key for Computer Vision and Cognitive Search. Follow the instructions in [Create a Cognitive Services account](#) to subscribe to Computer Vision and get your key.

### NOTE

For pricing information, see [Azure Cognitive Search](#).

## Import Libraries

Run the following command to import libraries for this recipe.

```
import os, sys, time, json, requests
from pyspark.ml import Transformer, Estimator, Pipeline
from pyspark.ml.feature import SQLTransformer
from pyspark.sql.functions import lit, udf, col, split
```

## Set up Subscription Keys

Run the following command to set up variables for service keys. Insert your subscription keys for Computer Vision and Azure Cognitive Search.

```
VISION_API_KEY = 'INSERT_COMPUTER_VISION_SUBSCRIPTION_KEY'
AZURE_SEARCH_KEY = 'INSERT_AZURE_COGNITIVE_SEARCH_SUBSCRIPTION_KEY'
search_service = "mmlspark-azure-search"
search_index = "test"
```

## Read the Data

Run the following command to load data from the MET's Open Access collection.

```
data = spark.read\
 .format("csv")\
 .option("header", True)\
 .load("wasbs://publicwasb@mmlspark.blob.core.windows.net/metartworks_sample.csv")\
 .withColumn("searchAction", lit("upload"))\
 .withColumn("Neighbors", split(col("Neighbors"), ",").cast("array<string>"))\
 .withColumn("Tags", split(col("Tags"), ",").cast("array<string>"))\
 .limit(25)
```

# Analyze the Images

Run the following command to use Computer Vision on the MET's Open Access artworks collection. As a result, you'll get visual features from the artworks.

```
from mmlspark.cognitive import AnalyzeImage
from mmlspark.stages import SelectColumns

#define pipeline
describeImage = (AnalyzeImage()
 .setSubscriptionKey(VISION_API_KEY)
 .setLocation("eastus")
 .setImageUrlCol("PrimaryImageUrl")
 .setOutputCol("RawImageDescription")
 .setErrorCol("Errors")
 .setVisualFeatures(["Categories", "Tags", "Description", "Faces", "ImageType", "Color", "Adult"])
 .setConcurrency(5))

df2 = describeImage.transform(data)\
 .select("*", "RawImageDescription.*").drop("Errors", "RawImageDescription")
```

## Create the Search Index

Run the following command to write the results to Azure Search to create a search engine of the artworks with enriched metadata from Computer Vision.

```
from mmlspark.cognitive import *
df2.writeToAzureSearch(
 subscriptionKey=AZURE_SEARCH_KEY,
 actionCol="searchAction",
 serviceName=search_service,
 indexName=search_index,
 keyCol="ObjectID"
)
```

## Query the Search Index

Run the following command to query the Azure Search index.

```
url = 'https://{}.search.windows.net/indexes/{}/docs/search?api-version=2019-05-06'.format(search_service,
search_index)
requests.post(url, json={"search": "Glass"}, headers = {"api-key": AZURE_SEARCH_KEY}).json()
```

## Next steps

Learn how to use [Cognitive Services for Big Data for Anomaly Detection](#).



# Cognitive Services development options

9/3/2021 • 7 minutes to read • [Edit Online](#)

This document provides a high-level overview of development and deployment options to help you get started with Azure Cognitive Services.

Azure Cognitive Services are cloud-based AI services that allow developers to build intelligence into their applications and products without deep knowledge of machine learning. With Cognitive Services, you have access to AI capabilities or models that are built, trained, and updated by Microsoft - ready to be used in your applications. In many cases, you also have the option to customize the models for your business needs.

Cognitive Services are organized into four categories: Decision, Language, Speech, and Vision. Typically you would access these services through REST APIs, client libraries, and custom tools (like command-line interfaces) provided by Microsoft. However, this is only one path to success. Through Azure, you also have access to several development options, such as:

- Automation and integration tools like Logic Apps and Power Automate.
- Deployment options such as Azure Functions and the App Service.
- Cognitive Services Docker containers for secure access.
- Tools like Apache Spark, Azure Databricks, Azure Synapse Analytics, and Azure Kubernetes Service for Big Data scenarios.

Before we jump in, it's important to know that the Cognitive Services are primarily used for two distinct tasks. Based on the task you want to perform, you have different development and deployment options to choose from.

- [Development options for prediction and analysis](#)
- [Tools to customize and configure models](#)

## Development options for prediction and analysis

The tools that you will use to customize and configure models are different from those that you'll use to call the Cognitive Services. Out of the box, most Cognitive Services allow you to send data and receive insights without any customization. For example:

- You can send an image to the Computer Vision service to detect words and phrases or count the number of people in the frame
- You can send an audio file to the Speech service and get transcriptions and translate the speech to text at the same time

Azure offers a wide range of tools that are designed for different types of users, many of which can be used with Cognitive Services. Designer-driven tools are the easiest to use, and are quick to set up and automate, but may have limitations when it comes to customization. Our REST APIs and client libraries provide users with more control and flexibility, but require more effort, time, and expertise to build a solution. If you use REST APIs and client libraries, there is an expectation that you're comfortable working with modern programming languages like C#, Java, Python, JavaScript, or another popular programming language.

Let's take a look at the different ways that you can work with the Cognitive Services.

### Client libraries and REST APIs

Cognitive Services client libraries and REST APIs provide you direct access to your service. These tools provide programmatic access to the Cognitive Services, their baseline models, and in many cases allow you to

programmatically customize your models and solutions.

- **Target user(s):** Developers and data scientists
- **Benefits:** Provides the greatest flexibility to call the services from any language and environment.
- **UI:** N/A - Code only
- **Subscription(s):** Azure account + Cognitive Services resources

If you want to learn more about available client libraries and REST APIs, use our [Cognitive Services overview](#) to pick a service and get started with one of our quickstarts for vision, decision, language, and speech.

### Cognitive Services for Big Data

With Cognitive Services for Big Data you can embed continuously improving, intelligent models directly into Apache Spark™ and SQL computations. These tools liberate developers from low-level networking details, so that they can focus on creating smart, distributed applications. Cognitive Services for Big Data support the following platforms and connectors: Azure Databricks, Azure Synapse, Azure Kubernetes Service, and Data Connectors.

- **Target user(s):** Data scientists and data engineers
- **Benefits:** The Azure Cognitive Services for Big Data let users channel terabytes of data through Cognitive Services using Apache Spark™. It's easy to create large-scale intelligent applications with any datastore.
- **UI:** N/A - Code only
- **Subscription(s):** Azure account + Cognitive Services resources

If you want to learn more about Big Data for Cognitive Services, a good place to start is with the [overview](#). If you're ready to start building, try our [Python](#) or [Scala](#) samples.

### Azure Functions and Azure Service Web Jobs

[Azure Functions](#) and [Azure App Service Web Jobs](#) both provide code-first integration services designed for developers and are built on [Azure App Services](#). These products provide serverless infrastructure for writing code. Within that code you can make calls to our services using our client libraries and REST APIs.

- **Target user(s):** Developers and data scientists
- **Benefits:** Serverless compute service that lets you run event-triggered code.
- **UI:** Yes
- **Subscription(s):** Azure account + Cognitive Services resource + Azure Functions subscription

### Azure Logic Apps

[Azure Logic Apps](#) share the same workflow designer and connectors as Power Automate but provide more advanced control, including integrations with Visual Studio and DevOps. Power Automate makes it easy to integrate with your Cognitive Services resources through service-specific connectors that provide a proxy or wrapper around the APIs. These are the same connectors as those available in Power Automate.

- **Target user(s):** Developers, integrators, IT pros, DevOps
- **Benefits:** Designer-first (declarative) development model providing advanced options and integration in a low-code solution
- **UI:** Yes
- **Subscription(s):** Azure account + Cognitive Services resource + Logic Apps deployment

### Power Automate

Power Automate is a service in the [Power Platform](#) that helps you create automated workflows between apps and services without writing code. We offer several connectors to make it easy to interact with your Cognitive Services resource in a Power Automate solution. Power Automate is built on top of Logic Apps.

- **Target user(s):** Business users (analysts) and SharePoint administrators

- **Benefits:** Automate repetitive manual tasks simply by recording mouse clicks, keystrokes and copy paste steps from your desktop!
- **UI tools:** Yes - UI only
- **Subscription(s):** Azure account + Cognitive Services resource + Power Automate Subscription + Office 365 Subscription

## AI Builder

[AI Builder](#) is a Microsoft Power Platform capability you can use to improve business performance by automating processes and predicting outcomes. AI Builder brings the power of AI to your solutions through a point-and-click experience. Many cognitive services such as Text Analytics, and Computer Vision have been directly integrated here and you don't need to create your own Cognitive Services.

- **Target user(s):** Business users (analysts) and SharePoint administrators
- **Benefits:** A turnkey solution that brings the power of AI through a point-and-click experience. No coding or data science skills required.
- **UI tools:** Yes - UI only
- **Subscription(s):** AI Builder

## Continuous integration and deployment

You can use Azure DevOps and GitHub actions to manage your deployments. In the [section below](#), we have two examples of CI/CD integrations to train and deploy custom models for Speech and the Language Understanding (LUIS) service.

- **Target user(s):** Developers, data scientists, and data engineers
- **Benefits:** Allows you to continuously adjust, update, and deploy applications and models programmatically. There is significant benefit when regularly using your data to improve and update models for Speech, Vision, Language, and Decision.
- **UI tools:** N/A - Code only
- **Subscription(s):** Azure account + Cognitive Services resource + GitHub account

## Tools to customize and configure models

As you progress on your journey building an application or workflow with the Cognitive Services, you may find that you need to customize the model to achieve the desired performance. Many of our services allow you to build on top of the pre-built models to meet your specific business needs. For all our customizable services, we provide both a UI-driven experience for walking through the process as well as APIs for code-driven training. For example:

- You want to train a Custom Speech model to correctly recognize medical terms with a word error rate (WER) below 3 percent
- You want to build an image classifier with Custom Vision that can tell the difference between coniferous and deciduous trees
- You want to build a custom neural voice with your personal voice data for an improved automated customer experience

The tools that you will use to train and configure models are different from those that you'll use to call the Cognitive Services. In many cases, Cognitive Services that support customization provide portals and UI tools designed to help you train, evaluate, and deploy models. Let's quickly take a look at a few options:

| PILLAR | SERVICE | CUSTOMIZATION UI | QUICKSTART |
|--------|---------|------------------|------------|
|--------|---------|------------------|------------|

| PILLAR   | SERVICE                       | CUSTOMIZATION UI                                                                                                                    | QUICKSTART |
|----------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|------------|
| Vision   | Custom Vision                 | <a href="https://www.customvision.ai/">https://www.customvision.ai/</a>                                                             | Quickstart |
| Decision | Content Moderator             | <a href="https://contentmoderator.cognitive.microsoft.com/dashboard">https://contentmoderator.cognitive.microsoft.com/dashboard</a> | Quickstart |
| Decision | Personalizer                  | UI is available in the Azure portal under your Personalizer resource.                                                               | Quickstart |
| Language | Language Understanding (LUIS) | <a href="https://www.luis.ai/">https://www.luis.ai/</a>                                                                             |            |
| Language | QnA Maker                     | <a href="https://www.qnamaker.ai/">https://www.qnamaker.ai/</a>                                                                     | Quickstart |
| Language | Translator/Custom Translator  | <a href="https://portal.customtranslator.azure.ai/">https://portal.customtranslator.azure.ai/</a>                                   | Quickstart |
| Speech   | Custom Commands               | <a href="https://speech.microsoft.com/">https://speech.microsoft.com/</a>                                                           | Quickstart |
| Speech   | Custom Speech                 | <a href="https://speech.microsoft.com/">https://speech.microsoft.com/</a>                                                           | Quickstart |
| Speech   | Custom Voice                  | <a href="https://speech.microsoft.com/">https://speech.microsoft.com/</a>                                                           | Quickstart |

### Continuous integration and delivery with DevOps and GitHub Actions

Language Understanding and the Speech service offer continuous integration and continuous deployment solutions that are powered by Azure DevOps and GitHub actions. These tools are used for automated training, testing, and release management of custom models.

- [CI/CD for Custom Speech](#)
- [CI/CD for LUIS](#)

## On-premises containers

Many of the Cognitive Services can be deployed in containers for on-premises access and use. Using these containers gives you the flexibility to bring Cognitive Services closer to your data for compliance, security, or other operational reasons. For a complete list of Cognitive Services containers, see [On-premises containers for Cognitive Services](#).

## Next steps

- [Create a Cognitive Services resource and start building](#)

# Cognitive Services and machine learning

9/10/2021 • 5 minutes to read • [Edit Online](#)

Cognitive Services provides machine learning capabilities to solve general problems such as analyzing text for emotional sentiment or analyzing images to recognize objects or faces. You don't need special machine learning or data science knowledge to use these services.

[Cognitive Services](#) is a group of services, each supporting different, generalized prediction capabilities. The services are divided into different categories to help you find the right service.

| SERVICE CATEGORY         | PURPOSE                                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Decision</a> | Build apps that surface recommendations for informed and efficient decision-making.                                                                     |
| <a href="#">Language</a> | Allow your apps to process natural language with pre-built scripts, evaluate sentiment and learn how to recognize what users want.                      |
| <a href="#">Search</a>   | Add Bing Search APIs to your apps and harness the ability to comb billions of webpages, images, videos, and news with a single API call.                |
| <a href="#">Speech</a>   | Convert speech into text and text into natural-sounding speech. Translate from one language to another and enable speaker verification and recognition. |
| <a href="#">Vision</a>   | Recognize, identify, caption, index, and moderate your pictures, videos, and digital ink content.                                                       |

Use Cognitive Services when you:

- Can use a generalized solution.
- Access solution from a programming REST API or SDK.

Use another machine-learning solution when you:

- Need to choose the algorithm and need to train on very specific data.

## What is machine learning?

Machine learning is a concept where you bring together data and an algorithm to solve a specific need. Once the data and algorithm are trained, the output is a model that you can use again with different data. The trained model provides insights based on the new data.

The process of building a machine learning system requires some knowledge of machine learning or data science.

Machine learning is provided using [Azure Machine Learning \(AML\) products and services](#).

## What is a Cognitive Service?

A Cognitive Service provides part or all of the components in a machine learning solution: data, algorithm, and

trained model. These services are meant to require general knowledge about your data without needing experience with machine learning or data science. These services provide both REST API(s) and language-based SDKs. As a result, you need to have programming language knowledge to use the services.

## How are Cognitive Services and Azure Machine Learning (AML) similar?

Both have the end-goal of applying artificial intelligence (AI) to enhance business operations, though how each provides this in the respective offerings is different.

Generally, the audiences are different:

- Cognitive Services are for developers without machine-learning experience.
- Azure Machine Learning is tailored for data scientists.

## How is a Cognitive Service different from machine learning?

A Cognitive Service provides a trained model for you. This brings data and an algorithm together, available from a REST API(s) or SDK. You can implement this service within minutes, depending on your scenario. A Cognitive Service provides answers to general problems such as key phrases in text or item identification in images.

Machine learning is a process that generally requires a longer period of time to implement successfully. This time is spent on data collection, cleaning, transformation, algorithm selection, model training, and deployment to get to the same level of functionality provided by a Cognitive Service. With machine learning, it is possible to provide answers to highly specialized and/or specific problems. Machine learning problems require familiarity with the specific subject matter and data of the problem under consideration, as well as expertise in data science.

## What kind of data do you have?

Cognitive Services, as a group of services, can require none, some, or all custom data for the trained model.

### **No additional training data required**

Services that provide a fully-trained model can be treated as a *opaque box*. You don't need to know how they work or what data was used to train them. You bring your data to a fully trained model to get a prediction.

### **Some or all training data required**

Some services allow you to bring your own data, then train a model. This allows you to extend the model using the Service's data and algorithm with your own data. The output matches your needs. When you bring your own data, you may need to tag the data in a way specific to the service. For example, if you are training a model to identify flowers, you can provide a catalog of flower images along with the location of the flower in each image to train the model.

A service may *allow* you to provide data to enhance its own data. A service may *require* you to provide data.

### **Real-time or near real-time data required**

A service may need real-time or near-real time data to build an effective model. These services process significant amounts of model data.

## Service requirements for the data model

The following data categorizes each service by which kind of data it allows or requires.

| COGNITIVE SERVICE                              | NO TRAINING DATA REQUIRED | YOU PROVIDE SOME OR ALL TRAINING DATA | REAL-TIME OR NEAR REAL-TIME DATA COLLECTION |
|------------------------------------------------|---------------------------|---------------------------------------|---------------------------------------------|
| <a href="#">Anomaly Detector</a>               | x                         | x                                     | x                                           |
| Bing Search                                    | x                         |                                       |                                             |
| <a href="#">Computer Vision</a>                | x                         |                                       |                                             |
| <a href="#">Content Moderator</a>              | x                         |                                       | x                                           |
| <a href="#">Custom Vision</a>                  |                           | x                                     |                                             |
| <a href="#">Face</a>                           | x                         | x                                     |                                             |
| <a href="#">Ink Recognizer</a>                 | x                         | x                                     |                                             |
| <a href="#">Language Understanding (LUIS)</a>  |                           | x                                     |                                             |
| <a href="#">Personalizer</a>                   | x*                        | x*                                    | x                                           |
| <a href="#">QnA Maker</a>                      |                           | x                                     |                                             |
| <a href="#">Speaker Recognizer</a>             |                           | x                                     |                                             |
| <a href="#">Speech Text-to-speech (TTS)</a>    | x                         | x                                     |                                             |
| <a href="#">Speech Speech-to-text (STT)</a>    | x                         | x                                     |                                             |
| <a href="#">Speech Translation</a>             | x                         |                                       |                                             |
| <a href="#">Text Analytics</a>                 | x                         |                                       |                                             |
| <a href="#">Translator</a>                     | x                         |                                       |                                             |
| <a href="#">Translator - custom translator</a> |                           | x                                     |                                             |

\*Personalizer only needs training data collected by the service (as it operates in real-time) to evaluate your policy and data. Personalizer does not need large historical datasets for up-front or batch training.

## Where can you use Cognitive Services?

The services are used in any application that can make REST API(s) or SDK calls. Examples of applications include web sites, bots, virtual or mixed reality, desktop and mobile applications.

## How is Azure Cognitive Search related to Cognitive Services?

[Azure Cognitive Search](#) is a separate cloud search service that optionally uses Cognitive Services to add image and natural language processing to indexing workloads. Cognitive Services is exposed in Azure Cognitive Search through [built-in skills](#) that wrap individual APIs. You can use a free resource for walkthroughs, but plan on creating and attaching a [billable resource](#) for larger volumes.

# How can you use Cognitive Services?

Each service provides information about your data. You can combine services together to chain solutions such as converting speech (audio) to text, translating the text into many languages, then using the translated languages to get answers from a knowledge base. While Cognitive Services can be used to create intelligent solutions on their own, they can also be combined with traditional machine learning projects to supplement models or accelerate the development process.

Cognitive Services that provide exported models for other machine learning tools:

| COGNITIVE SERVICE             | MODEL INFORMATION                                                                        |
|-------------------------------|------------------------------------------------------------------------------------------|
| <a href="#">Custom Vision</a> | <a href="#">Export</a> for Tensorflow for Android, CoreML for iOS11, ONNX for Windows ML |

## Learn more

- [Architecture Guide - What are the machine learning products at Microsoft?](#)
- [Machine learning - Introduction to deep learning vs. machine learning](#)

## Next steps

- Create your Cognitive Service account in the [Azure portal](#) or with [Azure CLI](#).
- Learn how to [authenticate](#) to a Cognitive Service.
- Use [diagnostic logging](#) for issue identification and debugging.
- Deploy a Cognitive Service in a Docker [container](#).
- Keep up to date with [service updates](#).



# Custom subdomain names for Cognitive Services

3/5/2021 • 2 minutes to read • [Edit Online](#)

Azure Cognitive Services use custom subdomain names for each resource created through the [Azure portal](#), [Azure Cloud Shell](#), or [Azure CLI](#). Unlike regional endpoints, which were common for all customers in a specific Azure region, custom subdomain names are unique to the resource. Custom subdomain names are required to enable features like Azure Active Directory (Azure AD) for authentication.

## How does this impact existing resources?

Cognitive Services resources created before July 1, 2019 will use the regional endpoints for the associated service. These endpoints will work with existing and new resources.

If you'd like to migrate an existing resource to leverage custom subdomain names, so that you can enable features like Azure AD, follow these instructions:

1. Sign in to the Azure portal and locate the Cognitive Services resource that you'd like to add a custom subdomain name to.
2. In the **Overview** blade, locate and select **Generate Custom Domain Name**.
3. This opens a panel with instructions to create a unique custom subdomain for your resource.

### WARNING

After you've created a custom subdomain name it **cannot** be changed.

## Do I need to update my existing resources?

No. The regional endpoint will continue to work for new and existing Cognitive Services and the custom subdomain name is optional. Even if a custom subdomain name is added the regional endpoint will continue to work with the resource.

## What if an SDK asks me for the region for a resource?

### WARNING

Speech Services use custom subdomains with [private endpoints](#) only. In all other cases use **regional endpoints** with Speech Services and associated SDKs.

Regional endpoints and custom subdomain names are both supported and can be used interchangeably. However, the full endpoint is required.

Region information is available in the **Overview** blade for your resource in the [Azure portal](#). For the full list of regional endpoints, see [Is there a list of regional endpoints?](#)

## Are custom subdomain names regional?

Yes. Using a custom subdomain name doesn't change any of the regional aspects of your Cognitive Services resource.

## What are the requirements for a custom subdomain name?

A custom subdomain name is unique to your resource. The name can only include alphanumeric characters and the `-` character; it must be between 2 and 64 characters in length and cannot end with a `-`.

## Can I change a custom domain name?

No. After a custom subdomain name is created and associated with a resource it cannot be changed.

## Can I reuse a custom domain name?

Each custom subdomain name is unique, so in order to reuse a custom subdomain name that you've assigned to a Cognitive Services resource, you'll need to delete the existing resource. After the resource has been deleted, you can reuse the custom subdomain name.

## Is there a list of regional endpoints?

Yes. This is a list of regional endpoints that you can use with Azure Cognitive Services resources.

### NOTE

The Translator service and Bing Search APIs use global endpoints.

| ENDPOINT TYPE | REGION                     | ENDPOINT                                                          |
|---------------|----------------------------|-------------------------------------------------------------------|
| Public        | Global (Translator & Bing) | <code>https://api.cognitive.microsoft.com</code>                  |
|               | Australia East             | <code>https://australiaeast.api.cognitive.microsoft.com</code>    |
|               | Brazil South               | <code>https://brazilsouth.api.cognitive.microsoft.com</code>      |
|               | Canada Central             | <code>https://canadacentral.api.cognitive.microsoft.com</code>    |
|               | Central US                 | <code>https://centralus.api.cognitive.microsoft.com</code>        |
|               | East Asia                  | <code>https://eastasia.api.cognitive.microsoft.com</code>         |
|               | East US                    | <code>https://eastus.api.cognitive.microsoft.com</code>           |
|               | East US 2                  | <code>https://eastus2.api.cognitive.microsoft.com</code>          |
|               | France Central             | <code>https://francecentral.api.cognitive.microsoft.com</code>    |
|               | India Central              | <code>https://centralindia.api.cognitive.microsoft.com</code>     |
|               | Japan East                 | <code>https://japaneast.api.cognitive.microsoft.com</code>        |
|               | Korea Central              | <code>https://koreacentral.api.cognitive.microsoft.com</code>     |
|               | North Central US           | <code>https://northcentralus.api.cognitive.microsoft.com</code>   |
|               | North Europe               | <code>https://northeurope.api.cognitive.microsoft.com</code>      |
|               | South Africa North         | <code>https://southafricanorth.api.cognitive.microsoft.com</code> |
|               | South Central US           | <code>https://southcentralus.api.cognitive.microsoft.com</code>   |
|               | Southeast Asia             | <code>https://southeastasia.api.cognitive.microsoft.com</code>    |
|               | UK South                   | <code>https://uksouth.api.cognitive.microsoft.com</code>          |

| ENDPOINT TYPE | REGION          | ENDPOINT                                                      |
|---------------|-----------------|---------------------------------------------------------------|
|               | West Central US | <code>https://westcentralus.api.cognitive.microsoft.cc</code> |
|               | West Europe     | <code>https://westeurope.api.cognitive.microsoft.com</code>   |
|               | West US         | <code>https://westus.api.cognitive.microsoft.com</code>       |
|               | West US 2       | <code>https://westus2.api.cognitive.microsoft.com</code>      |
| US Gov        | US Gov Virginia | <code>https://virginia.api.cognitive.microsoft.us</code>      |
| China         | China East 2    | <code>https://chinaeast2.api.cognitive.azure.cn</code>        |
|               | China North     | <code>https://chinanorth.api.cognitive.azure.cn</code>        |

## See also

- [What are the Cognitive Services?](#)
- [Authentication](#)

# Azure Policy built-in policy definitions for Azure Cognitive Services

9/17/2021 • 2 minutes to read • [Edit Online](#)

This page is an index of [Azure Policy](#) built-in policy definitions for Azure Cognitive Services. For additional Azure Policy built-ins for other services, see [Azure Policy built-in definitions](#).

The name of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Version** column to view the source on the [Azure Policy GitHub repo](#).

## Azure Cognitive Services

| NAME<br>(AZURE PORTAL)                                                                                | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | EFFECT(S)             | VERSION<br>(GITHUB)   |
|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-----------------------|
| <a href="#">Cognitive Services accounts should disable public network access</a>                      | Disabling public network access improves security by ensuring that Cognitive Services account isn't exposed on the public internet. Creating private endpoints can limit exposure of Cognitive Services account. Learn more at: <a href="https://go.microsoft.com/fwlink/?linkid=2129800">https://go.microsoft.com/fwlink/?linkid=2129800</a> .                                                                                                                                            | Audit, Deny, Disabled | <a href="#">2.0.0</a> |
| <a href="#">Cognitive Services accounts should enable data encryption with a customer-managed key</a> | Customer-managed keys are commonly required to meet regulatory compliance standards. Customer-managed keys enable the data stored in Cognitive Services to be encrypted with an Azure Key Vault key created and owned by you. You have full control and responsibility for the key lifecycle, including rotation and management. Learn more about customer-managed keys at <a href="https://go.microsoft.com/fwlink/?linkid=2121321">https://go.microsoft.com/fwlink/?linkid=2121321</a> . | Audit, Deny, Disabled | <a href="#">2.0.0</a> |
| <a href="#">Cognitive Services accounts should have local authentication methods disabled</a>         | Disabling local authentication methods improves security by ensuring that Cognitive Services accounts require Azure Active Directory identities exclusively for authentication. Learn more at: <a href="https://aka.ms/cs/auth">https://aka.ms/cs/auth</a> .                                                                                                                                                                                                                               | Audit, Deny, Disabled | <a href="#">1.0.0</a> |

| NAME                                                          | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | EFFECT(S)             | VERSION |
|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|---------|
| Cognitive Services accounts should restrict network access    | Network access to Cognitive Services accounts should be restricted. Configure network rules so only applications from allowed networks can access the Cognitive Services account. To allow connections from specific internet or on-premises clients, access can be granted to traffic from specific Azure virtual networks or to public internet IP address ranges.                                                                                                                                             | Audit, Deny, Disabled | 2.0.0   |
| Cognitive Services accounts should use a managed identity     | Assigning a managed identity to your Cognitive Service account helps ensure secure authentication. This identity is used by this Cognitive service account to communicate with other Azure services, like Azure Key Vault, in a secure way without you having to manage any credentials.                                                                                                                                                                                                                         | Audit, Deny, Disabled | 1.0.0   |
| Cognitive Services accounts should use customer owned storage | Use customer owned storage to control the data stored at rest in Cognitive Services. To learn more about customer owned storage, visit <a href="https://aka.ms/cogsvc-cmk">https://aka.ms/cogsvc-cmk</a> .                                                                                                                                                                                                                                                                                                       | Audit, Deny, Disabled | 2.0.0   |
| Cognitive Services should use private link                    | Azure Private Link lets you connect your virtual networks to Azure services without a public IP address at the source or destination. The Private Link platform handles the connectivity between the consumer and services over the Azure backbone network. By mapping private endpoints to Cognitive Services, you'll reduce the potential for data leakage. Learn more about private links at: <a href="https://go.microsoft.com/fwlink/?linkid=2129800">https://go.microsoft.com/fwlink/?linkid=2129800</a> . | Audit, Disabled       | 2.0.0   |

| NAME                                                                                          | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                      | EFFECT(S)                   | VERSION |
|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|---------|
| <a href="#">Configure Cognitive Services accounts to disable local authentication methods</a> | Disable local authentication methods so that your Cognitive Services accounts require Azure Active Directory identities exclusively for authentication. Learn more at: <a href="https://aka.ms/cs/auth">https://aka.ms/cs/auth</a> .                                                                                                                                             | Modify, Disabled            | 1.0.0   |
| <a href="#">Configure Cognitive Services accounts to disable public network access</a>        | Disable public network access for your Cognitive Services resource so that it's not accessible over the public internet. This can reduce data leakage risks. Learn more at: <a href="https://go.microsoft.com/fwlink/?linkid=2129800">https://go.microsoft.com/fwlink/?linkid=2129800</a> .                                                                                      | Disabled, Modify            | 2.0.0   |
| <a href="#">Configure Cognitive Services accounts with private endpoints</a>                  | Private endpoints connect your virtual networks to Azure services without a public IP address at the source or destination. By mapping private endpoints to Cognitive Services, you'll reduce the potential for data leakage. Learn more about private links at: <a href="https://go.microsoft.com/fwlink/?linkid=2129800">https://go.microsoft.com/fwlink/?linkid=2129800</a> . | DeployIfNotExists, Disabled | 2.0.0   |

## Next steps

- See the built-ins on the [Azure Policy GitHub repo](#).
- Review the [Azure Policy definition structure](#).
- Review [Understanding policy effects](#).

# Azure Cognitive Services support and help options

8/13/2021 • 2 minutes to read • [Edit Online](#)

Are you just starting to explore the functionality of Azure Cognitive Services? Perhaps you are implementing a new feature in your application. Or after using the service, do you have suggestions on how to improve it? Here are options for where you can get support, stay up-to-date, give feedback, and report bugs for Cognitive Services.

## Create an Azure support request



Explore the range of [Azure support options and choose the plan](#) that best fits, whether you're a developer just starting your cloud journey or a large organization deploying business-critical, strategic applications. Azure customers can create and manage support requests in the Azure portal.

- [Azure portal](#)
- [Azure portal for the United States government](#)

## Post a question on Microsoft Q&A

For quick and reliable answers on your technical product questions from Microsoft Engineers, Azure Most Valuable Professionals (MVPs), or our expert community, engage with us on [Microsoft Q&A](#), Azure's preferred destination for community support.

If you can't find an answer to your problem using search, submit a new question to Microsoft Q&A. Use one of the following tags when you ask your question:

- [Cognitive Services](#)

### Vision

- [Computer Vision](#)
- [Custom Vision](#)
- [Face](#)
- [Form Recognizer](#)
- [Video Indexer](#)

### Language

- [Immersive Reader](#)
- [Language Understanding \(LUIS\)](#)
- [QnA Maker](#)
- [Text Analytics](#)
- [Translator](#)

### Speech

- [Speech service](#)

### Decision

- [Anomaly Detector](#)

- [Content Moderator](#)
- [Metrics Advisor \(preview\)](#)
- [Personalizer](#)

## Post a question to Stack Overflow

For answers on your developer questions from the largest community developer ecosystem, ask your question on Stack Overflow.

If you do submit a new question to Stack Overflow, please use one or more of the following tags when you create the question:

- [Cognitive Services](#)

### Vision

- [Computer Vision](#)
- [Custom Vision](#)
- [Face](#)
- [Form Recognizer](#)
- [Video Indexer](#)

### Language

- [Immersive Reader](#)
- [Language Understanding \(LUIS\)](#)
- [QnA Maker](#)
- [Text Analytics](#)
- [Translator](#)

### Speech

- [Speech service](#)

### Decision

- [Anomaly Detector](#)
- [Content Moderator](#)
- [Metrics Advisor \(preview\)](#)
- [Personalizer](#)

## Submit feedback

To request new features, post them on <https://feedback.azure.com>. Share your ideas for making Cognitive Services and its APIs work better for the applications you develop.

- [Cognitive Services](#)

### Vision

- [Computer Vision](#)
- [Custom Vision](#)
- [Face](#)
- [Form Recognizer](#)



- [Video Indexer](#)

## Language

- [Immersive Reader](#)
- [Language Understanding \(LUIS\)](#)
- [QnA Maker](#)
- [Text Analytics](#)
- [Translator](#)

## Speech

- [Speech service](#)

## Decision

- [Anomaly Detector](#)
- [Content Moderator](#)
- [Metrics Advisor \(preview\)](#)
- [Personalizer](#)

## Stay informed

Staying informed about features in a new release or news on the Azure blog can help you find the difference between a programming error, a service bug, or a feature not yet available in Cognitive Services.

- Learn more about product updates, roadmap, and announcements in [Azure Updates](#).
- See what Cognitive Services articles have recently been added or updated in [What's new in docs?](#)
- News about Cognitive Services is shared in the [Azure blog](#).
- [Join the conversation on Reddit](#) about Cognitive Services.

## Next steps

[What are Azure Cognitive Services?](#)

# Responsible use of AI with Cognitive Services

6/16/2021 • 2 minutes to read • [Edit Online](#)

Azure Cognitive Services provides information and guidelines on how to responsibly use artificial intelligence in applications. Below are the links to articles that provide this guidance for the different services within the Cognitive Services suite.

## Computer Vision - OCR

- [Transparency note and use cases](#)
- [Characteristics and limitations](#)
- [Integration and responsible use](#)
- [Data, privacy, and security](#)

## Computer Vision - Spatial Analysis

- [Transparency note and use cases](#)
- [Characteristics and limitations](#)
- [Responsible use in AI deployment](#)
- [Disclosure design guidelines](#)
- [Research insights](#)
- [Data, privacy, and security](#)

## QnA Maker

- [Transparency note and use cases](#)
- [Characteristics and limitations](#)
- [Integration and responsible use](#)
- [Data, privacy, and security](#)

## Text Analytics

- [Transparency note and use cases](#)
- [Integration and responsible use](#)
- [Data, privacy, and security](#)

## Speech - Pronunciation Assessment

- [Transparency note and use cases](#)
- [Characteristics and limitations](#)

## Speech - Custom Neural Voice

- [Transparency note and use cases](#)
- [Characteristics and limitations](#)
- [Responsible deployment of synthetic speech](#)
- [Disclosure of voice talent](#)

- [Disclosure of design guidelines](#)
- [Disclosure of design patterns](#)
- [Code of conduct](#)
- [Data, privacy, and security](#)

## Anomaly Detector

- [Transparency note and use cases](#)
- [Characteristics and limitations](#)
- [Integration and responsible use](#)
- [Data, privacy, and security](#)

# Compare Azure Government and global Azure

9/15/2021 • 23 minutes to read • [Edit Online](#)

Microsoft Azure Government uses same underlying technologies as global Azure, which includes the core components of [Infrastructure-as-a-Service \(IaaS\)](#), [Platform-as-a-Service \(PaaS\)](#), and [Software-as-a-Service \(SaaS\)](#). Both Azure and Azure Government have the same comprehensive security controls in place and the same Microsoft commitment on the safeguarding of customer data. Whereas both cloud environments are assessed and authorized at the FedRAMP High impact level, Azure Government provides an extra layer of protection to customers through contractual commitments regarding storage of customer data in the United States and limiting potential access to systems processing customer data to [screened US persons](#). These commitments may be of interest to customers using the cloud to store or process data subject to US export control regulations.

## Export control implications

You are responsible for designing and deploying your applications to meet [US export control requirements](#) such as the requirements prescribed in the EAR, ITAR, and DoE 10 CFR Part 810. In doing so, you should not include sensitive or restricted information in Azure resource names, as explained in [Considerations for naming Azure resources](#).

## Guidance for developers

Azure Government services operate the same way as the corresponding services in global Azure, which is why most of the existing online Azure documentation applies equally well to Azure Government. However, there are some key differences that developers working on applications hosted in Azure Government must be aware of. For more information, see [Guidance for developers](#). As a developer, you must know how to connect to Azure Government and once you connect you will mostly have the same experience as in global Azure.

You can use AzureCLI or PowerShell to obtain Azure Government endpoints for services you provisioned:

- Use **Azure CLI** to run the [az cloud show](#) command and provide `AzureUSGovernment` as the name of the target cloud environment. For example,

```
az cloud show --name AzureUSGovernment
```

should get you different endpoints for Azure Government.

- Use a **PowerShell** cmdlet such as [Get-AzureEnvironment](#) (or [Get-AzureRmEnvironment](#)) to get endpoints and metadata for an instance of Azure service. For example,

```
Get-AzureEnvironment -Name AzureUSGovernment
```

should get you properties for Azure Government. These cmdlets get environments from your subscription data file.

Table below lists API endpoints in Azure vs. Azure Government for accessing and managing some of the more common services. If you provisioned a service that isn't listed in the table below, see the Azure CLI and PowerShell examples above for suggestions on how to obtain the corresponding Azure Government endpoint.

| SERVICE CATEGORY      | SERVICE NAME           | AZURE PUBLIC                            | AZURE GOVERNMENT                                                                                                                                                                                                                                                                                                                                          | NOTES                           |
|-----------------------|------------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| AI + Machine Learning | Azure Bot Service      | *.botframework.com                      | *.botframework.azure.us                                                                                                                                                                                                                                                                                                                                   |                                 |
|                       | Azure Form Recognizer  | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | Computer Vision        | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | Custom Vision          | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us<br><a href="#">Portal</a>                                                                                                                                                                                                                                                                                                    |                                 |
|                       | Content Moderator      | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | Face API               | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | Language Understanding | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us<br><a href="#">Portal</a>                                                                                                                                                                                                                                                                                                    |                                 |
|                       | Personalizer           | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | QnA Maker              | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | Speech service         | See <a href="#">STT API docs</a>        | <a href="#">Speech Studio</a><br>See <a href="#">Speech service endpoints</a><br><br><b>Speech translation endpoints</b><br>Virginia:<br><a href="https://usgovvirginia.s2s.speech.azure.us">https://usgovvirginia.s2s.speech.azure.us</a><br>Arizona:<br><a href="https://usgovarizona.s2s.speech.azure.us">https://usgovarizona.s2s.speech.azure.us</a> |                                 |
|                       | Text Analytics         | *.cognitiveservices.azure.com           | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
|                       | Translator             | See <a href="#">Translator API docs</a> | *.cognitiveservices.azure.us                                                                                                                                                                                                                                                                                                                              |                                 |
| Analytics             | HDInsight              | *.azurehdinsight.net                    | *.azurehdinsight.us                                                                                                                                                                                                                                                                                                                                       |                                 |
|                       | Power BI               | app.powerbi.com                         | app.powerbigov.us                                                                                                                                                                                                                                                                                                                                         | <a href="#">Power BI US Gov</a> |
| Compute               | Batch                  | *.batch.azure.com                       | *.batch.usgovcloudappi.net                                                                                                                                                                                                                                                                                                                                |                                 |
|                       | Cloud Services         | *.cloudapp.net                          | *.usgovcloudapp.net                                                                                                                                                                                                                                                                                                                                       |                                 |

| SERVICE CATEGORY | SERVICE NAME | AZURE PUBLIC | AZURE GOVERNMENT | NOTES |
|------------------|--------------|--------------|------------------|-------|
|------------------|--------------|--------------|------------------|-------|

|                           |                               |                                    |                                       |                                                    |
|---------------------------|-------------------------------|------------------------------------|---------------------------------------|----------------------------------------------------|
|                           | Azure Functions               | *.azurewebsites.net                | *.azurewebsites.us                    |                                                    |
|                           | Service Fabric                | *.cloudapp.azure.com               | *.cloudapp.usgovcloudapi.net          |                                                    |
| Containers                | Container Registry Suffix     | *.azurecr.io                       | *.azurecr.us                          |                                                    |
| Databases                 | Azure Cache for Redis         | *.redis.cache.windows.net          | *.redis.cache.usgovcloudapi.net       | See <a href="#">How to connect to other clouds</a> |
|                           | Azure Cosmos DB               | *.documents.azure.com              | *.documents.azure.us                  |                                                    |
|                           | Azure Database for MariaDB    | *.mariadb.database.azure.com       | *.mariadb.database.usgovcloudapi.net  |                                                    |
|                           | Azure Database for MySQL      | *.mysql.database.azure.com         | *.mysql.database.usgovcloudapi.net    |                                                    |
|                           | Azure Database for PostgreSQL | *.postgres.database.azure.com      | *.postgres.database.usgovcloudapi.net |                                                    |
|                           | Azure SQL Database            | *.database.windows.net             | *.database.usgovcloudapi.net          |                                                    |
| Integration               | Service Bus                   | *.servicebus.windows.net           | *.servicebus.usgovcloudapi.net        |                                                    |
| Internet of Things        | Azure Event Hubs              | *.servicebus.windows.net           | *.servicebus.usgovcloudapi.net        |                                                    |
|                           | Azure IoT Hub                 | *.azure-devices.net                | *.azure-devices.us                    |                                                    |
|                           | Azure Maps                    | atlas.microsoft.com                | atlas.azure.us                        |                                                    |
|                           | Notification Hubs             | *.servicebus.windows.net           | *.servicebus.usgovcloudapi.net        |                                                    |
| Management and Governance | Azure Monitor logs            | mms.microsoft.com                  | oms.microsoft.us                      | Log Analytics workspace portal                     |
|                           |                               | workspace.ods.opinsights.azure.com | workspace.ods.opinsights.azure.us     | <a href="#">Data collector API</a>                 |
|                           |                               | *.ods.opinsights.azure.com         | *.ods.opinsights.azure.us             |                                                    |
|                           |                               | *.oms.opinsights.azure.com         | *.oms.opinsights.azure.us             |                                                    |

| SERVICE CATEGORY | SERVICE NAME           | AZURE PUBLIC                             | AZURE GOVERNMENT                        | NOTES                  |
|------------------|------------------------|------------------------------------------|-----------------------------------------|------------------------|
|                  |                        | portal.loganalytics.io                   | portal.loganalytics.us                  |                        |
|                  |                        | api.loganalytics.io                      | api.loganalytics.us                     |                        |
|                  |                        | docs.loganalytics.io                     | docs.loganalytics.us                    |                        |
|                  | Azure Automation       | *.azure-automation.net                   | *.azure-automation.us                   |                        |
|                  | Portal and Cloud Shell | https://portal.azure.com                 | https://portal.azure.us                 |                        |
|                  | Gallery URL            | https://gallery.azure.com/               | https://gallery.azure.us/               |                        |
| Migration        | Azure Site Recovery    | *.hypervrecoverymanager.windowsazure.com | *.hypervrecoverymanager.windowsazure.us | Site Recovery service  |
|                  |                        | *.backup.windowsazure.com/               | *.backup.windowsazure.us/               | Protection service     |
|                  |                        | *.blob.core.windows.net/                 | *.blob.core.usgovcloudapi.net/          | Storing VM snapshots   |
|                  |                        | <a href="#">Public download MySQL</a>    | <a href="#">Gov download MySQL</a>      | Download MySQL         |
| Networking       | Traffic Manager        | *.trafficmanager.net                     | *.usgovtrafficmanager.net               |                        |
| Security         | Azure Active Directory | https://login.microsoftonline.com        | https://login.microsoftonline.us        |                        |
|                  | Key Vault              | *.vault.azure.net                        | *.vault.usgovcloudapi.net               | Endpoint               |
|                  |                        | cfa8b339-82a2-471a-a3c9-0fc0be7a4093     | 7e7c393b-45d0-48b1-a35e-2905ddf8183c    | Service Principal ID   |
|                  |                        | Azure Key Vault                          | Azure Key Vault                         | Service Principal Name |
| Storage          | Blob                   | *.blob.core.windows.net                  | *.blob.core.usgovcloudapi.net           |                        |
|                  | Queue                  | *.queue.core.windows.net                 | *.queue.core.usgovcloudapi.net          |                        |
|                  | Table                  | *.table.core.windows.net                 | *.table.core.usgovcloudapi.net          |                        |

| SERVICE CATEGORY | SERVICE NAME              | AZURE PUBLIC                         | AZURE GOVERNMENT                     | NOTES                |
|------------------|---------------------------|--------------------------------------|--------------------------------------|----------------------|
|                  | File                      | *.file.core.windows.net              | *.file.core.usgovcloudapi.net        |                      |
| Web              | API Management            | management.azure.com                 | management.usgovcloudapi.net         |                      |
|                  | API Management Gateway    | *.azure-api.net                      | *.azure-api.us                       |                      |
|                  | API Management Portal     | *.portal.azure-api.net               | *.portal.azure-api.us                |                      |
|                  | API Management management | *.management.azure-api.net           | *.management.azure-api.us            |                      |
|                  | App Configuration         | *.azconfig.io                        | *.azconfig.azure.us                  | Endpoint             |
|                  | App Service               | *.azurewebsites.net                  | *.azurewebsites.us                   | Endpoint             |
|                  |                           | abfa0a7c-a6b6-4736-8310-5855508787cd | 6a02c803-dafd-4136-b4c3-5a6f318b4714 | Service Principal ID |
|                  | Azure Cognitive Search    | *.search.windows.net                 | *.search.windows.us                  |                      |

## Service availability

Microsoft's goal for Azure Government is to match service availability in Azure. For service availability in Azure Government, see [Products available by region](#). Services available in Azure Government are listed by category and whether they are Generally Available or available through Preview. If a service is available in Azure Government, that fact is not reiterated in the rest of this article. Instead, you are encouraged to review [Products available by region](#) for the latest, up-to-date information on service availability.

In general, service availability in Azure Government implies that all corresponding service features are available to you. Variations to this approach and other applicable limitations are tracked and explained in this article based on the main service categories outlined in the [online directory of Azure services](#). Other considerations for service deployment and usage in Azure Government are also provided.

## AI + machine learning

This section outlines variations and considerations when using **Azure Bot Service**, **Azure Machine Learning**, and **Cognitive Services** in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure Bot Service

The following Azure Bot Service **features are not currently available** in Azure Government (updated 8/16/2021):

- Bot Framework Composer integration
- Channels (due to availability of dependent services)
  - Teams Channel



- Direct Line Speech Channel
- Telephony Channel (Preview)
- Microsoft Search Channel (Preview)
- Kik Channel (deprecated)

For more information, see [How do I create a bot that uses US Government data center.](#)

## Azure Machine Learning

For feature variations and limitations, see [Azure Machine Learning sovereign cloud parity.](#)

### Cognitive Services: Content Moderator

The following Content Moderator **features are not currently available** in Azure Government:

- Review UI and Review APIs.

### Cognitive Services: Language Understanding (LUIS)

The following Language Understanding **features are not currently available** in Azure Government:

- Speech Requests
- Prebuilt Domains

### Cognitive Services: Speech

For feature variations and limitations, including API endpoints, see [Speech service in sovereign clouds.](#)

### Cognitive Services: Translator

The following Translator **features are not currently available** in Azure Government:

- Custom Translator
- Translator Hub

## Analytics

This section outlines variations and considerations when using Analytics services in the Azure Government environment. For service availability, see [Products available by region.](#)

### Azure HDInsight

The following HDInsight **features are not currently available** in Azure Government:

- HDInsight on Windows
- Azure Data Lake Storage

Azure Blob Storage is the only available storage option currently.

For secured virtual networks, you will want to allow network security groups (NSGs) access to certain IP addresses and ports. For Azure Government, you should allow the following IP addresses (all with an Allowed port of 443):

| REGION         | ALLOWED IP ADDRESSES             | ALLOWED PORT |
|----------------|----------------------------------|--------------|
| US DoD Central | 52.180.249.174<br>52.180.250.239 | 443          |
| US DoD East    | 52.181.164.168<br>52.181.164.151 | 443          |

| REGION          | ALLOWED IP ADDRESSES                                          | ALLOWED PORT |
|-----------------|---------------------------------------------------------------|--------------|
| US Gov Texas    | 52.238.116.212<br>52.238.112.86                               | 443          |
| US Gov Virginia | 13.72.49.126<br>13.72.55.55<br>13.72.184.124<br>13.72.190.110 | 443          |
| US Gov Arizona  | 52.127.3.176<br>52.127.3.178                                  | 443          |

You can see a demo on how to build data-centric solutions on Azure Government using [HDInsight](#).

### Power BI

The following Power BI **features are not currently available** in Azure Government:

- Portal support

You can see a demo on [how to build data-centric solutions on Azure Government using Power BI](#).

#### NOTE

The content pack that typically makes activity logs and such available is not intended for use on Government tenants. The intention is to use Log Analytics for the purpose of the logs that aren't available through the content pack.

### Power BI Embedded

The following Power BI Embedded **features are not yet available** in Azure Government:

- Portal support

## Compute

This section outlines variations and considerations when using Compute services in the Azure Government environment. For service availability, see [Products available by region](#).

### Virtual Machines

The following Virtual Machines **features are not currently available** in Azure Government:

- Settings
  - Continuous delivery
- Operations
  - Auto shutdown
- Monitoring
  - Application Insights
- Support + troubleshooting
  - Ubuntu Advantage support plan

### Azure Functions

The following Functions **features are not currently available** in Azure Government:

- Running .NET 5 apps

When connecting your Functions app to Application Insights in Azure Government, make sure you use

`APPLICATIONINSIGHTS_CONNECTION_STRING`, which lets you customize the Application Insights endpoint.

## Containers

This section outlines variations and considerations when using Container services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure Kubernetes Service

The following Azure Kubernetes Service **features are not currently available** in Azure Government:

- [Customize node configuration](#) for Azure Kubernetes Service node pools

## Databases

This section outlines variations and considerations when using Databases services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure Database for MySQL

The following Azure Database for MySQL **features are not currently available** in Azure Government:

- Advanced Threat Protection

### Azure Database for PostgreSQL

The following Azure Database for PostgreSQL **features are not currently available** in Azure Government:

- Hyperscale (Citrus) and Flexible server deployment options
- The following features of the Single server deployment option
  - Advanced Threat Protection
  - Backup with long-term retention

### Azure SQL Managed Instance

The following Azure SQL Managed Instance **features are not currently available** in Azure Government:

- Long-term retention

## Developer Tools

This section outlines variations and considerations when using Developer Tools services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure DevTest Labs

The following Azure DevTest Labs **features are not currently available** in Azure Government:

- Auto shutdown feature for Azure Compute VMs; however, setting auto shutdown for [Labs](#) and [Lab Virtual Machines](#) is available.

## Identity

This section outlines variations and considerations when using Identity services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure Active Directory Premium P1 and P2

The following features have known limitations in Azure Government:

- Limitations with B2B Collaboration in supported Azure US Government tenants:
  - B2B Collaboration is available in most Azure US Government tenants created after June 2019. Over

time, more tenants will get access to this functionality. See [How can I tell if B2B collaboration is available in my Azure US Government tenant?](#)

- B2B collaboration is supported between tenants that are both within Azure US Government cloud and that both support B2B collaboration. Azure US Government tenants that support B2B collaboration can also collaborate with social users using Microsoft, Google accounts, or email one-time passcode accounts. If you invite a user outside of these groups (for example, if the user is in a tenant that isn't part of the Azure US Government cloud or doesn't yet support B2B collaboration), the invitation will fail or the user will be unable to redeem the invitation.
- B2B collaboration via Power BI is not supported. When you invite a guest user from within Power BI, the B2B flow is not used and the guest user won't appear in the tenant's user list. If a guest user is invited through other means, they'll appear in the Power BI user list, but any sharing request to the user will fail and display a 403 Forbidden error.
- Microsoft 365 Groups are not supported for B2B users and can't be enabled.
- Some SQL tools such as SQL Server Management Studio (SSMS) require you to set the appropriate cloud parameter. In the tool's Azure service setup options, set the cloud parameter to Azure Government.
- Limitations with multifactor authentication:
  - Hardware OATH tokens are not available in Azure Government.
  - Trusted IPs are not supported in Azure Government. Instead, use Conditional Access policies with named locations to establish when multifactor authentication should and should not be required based off the user's current IP address.
- Limitations with Azure AD join:
  - Enterprise state roaming for Windows 10 devices is not available

## Management and governance

This section outlines variations and considerations when using Management and Governance services in the Azure Government environment. For service availability, see [Products available by region](#).

### NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

### Azure Advisor

The following Azure Advisor recommendation **features are not currently available** in Azure Government:

- High Availability
  - Configure your VPN gateway to active-active for connection resilience
  - Create Azure Service Health alerts to be notified when Azure issues affect you
  - Configure Traffic Manager endpoints for resiliency
  - Use soft delete for your Azure Storage Account
- Performance
  - Improve App Service performance and reliability
  - Reduce DNS time to live on your Traffic Manager profile to fail over to healthy endpoints faster
  - Improve Azure Synapse Analytics performance
  - Use Premium Storage

- Migrate your Storage Account to Azure Resource Manager
- Cost
  - Buy reserved virtual machines instances to save money over pay-as-you-go costs
  - Eliminate unprovisioned ExpressRoute circuits
  - Delete or reconfigure idle virtual network gateways

The calculation for recommending that you should right-size or shut down underutilized virtual machines in Azure Government is as follows:

- Advisor monitors your virtual machine usage for seven days and identifies low-utilization virtual machines.
- Virtual machines are considered low utilization if their CPU utilization is 5% or less and their network utilization is less than 2%, or if the current workload can be accommodated by a smaller virtual machine size.

If you want to be more aggressive at identifying underutilized virtual machines, you can adjust the CPU utilization rule on a per subscription basis.

### Azure Cost Management and Billing

The following Azure Cost Management + Billing **features are not currently available** in Azure Government:

- Cost Management + Billing for cloud solution providers (CSPs)

### Azure Lighthouse

The following Azure Lighthouse **features are not currently available** in Azure Government:

- Managed Service offers published to Azure Marketplace

### Azure Monitor

The following Azure Monitor **features are not currently available** in Azure Government:

- Solutions that are in preview in Microsoft Azure, including:
  - Windows 10 Upgrade Analytics solution
  - Application Insights solution
  - Azure Networking Security Group Analytics solution
  - Azure Automation Analytics solution
  - Key Vault Analytics solution
- Solutions and features that require updates to on-premises software, including:
  - Surface Hub solution
- Features that are in preview in global Azure, including:
  - Export of data to Power BI
- Azure metrics and Azure diagnostics

The following Azure Monitor **features behave differently** in Azure Government:

- To connect your System Center Operations Manager management group to Azure Monitor logs, you need to download and import updated management packs.
  - System Center Operations Manager 2016
    1. Install [Update Rollup 2 for System Center Operations Manager 2016](#).
    2. Import the management packs included as part of Update Rollup 2 into Operations Manager. For information about how to import a management pack from a disk, see [How to import an Operations Manager Management Pack](#).
    3. To connect Operations Manager to Azure Monitor logs, follow the steps in [Connect Operations Manager to Azure Monitor](#).
  - System Center Operations Manager 2012 R2 UR3 (or later) / Operations Manager 2012 SP1 UR7 (or later)

1. Download and save the [updated management packs](#).
  2. Unzip the file that you downloaded.
  3. Import the management packs into Operations Manager. For information about how to import a management pack from a disk, see [How to import an Operations Manager Management Pack](#).
  4. To connect Operations Manager to Azure Monitor logs, follow the steps in [Connect Operations Manager to Azure Monitor](#).
- For more information about using computer groups from Configuration Manager, see [Connect Configuration Manager to Azure Monitor](#).

## Frequently asked questions

- Can I migrate data from Azure Monitor logs in Azure to Azure Government?
  - No. It is not possible to move data or your workspace from Azure to Azure Government.
- Can I switch between Azure and Azure Government workspaces from the Operations Management Suite portal?
  - No. The portals for Azure and Azure Government are separate and do not share information.

## Application Insights

This section describes the supplemental configuration that is required to use Application Insights (part of Azure Monitor) in Azure Government.

### Enable Application Insights for [ASP.NET](#) & [ASP.NET Core](#) with Visual Studio

In Azure Government, you can enable Application Insights with a [codeless agent](#) for your Azure App Services hosted applications or via the traditional **Add Applications Insights Telemetry** button in Visual Studio, which requires a small manual workaround. If you are experiencing the associated issue, you may see error messages like "There is no Azure subscription associated with this account" or "The selected subscription does not support Application Insights" even though the `microsoft.insights` resource provider has a status of registered for the subscription. To mitigate this issue, you must perform the following steps:

1. Switch Visual Studio to [target the Azure Government cloud](#).
2. Create (or if already existing, set) the User Environment variable for `AzureGraphApiVersion` as follows:
  - Variable name: `AzureGraphApiVersion`
  - Variable value: `2014-04-01`

To create a User Environment variable, go to **Control Panel > System > Advanced system settings > Advanced > Environment Variables**.

3. Make the appropriate Application Insights SDK endpoint modifications for either [ASP.NET](#) or [ASP.NET Core](#) depending on your project type.

**Snapshot Debugger** is now available for Azure Government customers. To use Snapshot Debugger, the only other prerequisite is to ensure that you are using [Snapshot Collector version 1.3.5](#) or later. Then follow the standard [Snapshot Debugger documentation](#).

**SDK endpoint modifications** - In order to send data from Application Insights to the Azure Government region, you will need to modify the default endpoint addresses that are used by the Application Insights SDKs. Each SDK requires slightly different modifications, as described in [Application Insights overriding default endpoints](#).

#### NOTE

[Connection strings](#) are the new preferred method of setting custom endpoints within Application Insights.

**Firewall exceptions** - Application Insights uses several IP addresses. You might need to know these addresses if the app that you are monitoring is hosted behind a firewall.

#### NOTE

Although these addresses are static, it's possible that we will need to change them from time to time. All Application Insights traffic represents outbound traffic except for availability monitoring and webhooks, which require inbound firewall rules.

You need to open some **outgoing ports** in your server's firewall to allow the Application Insights SDK and/or Status Monitor to send data to the portal:

| PURPOSE   | URL                       | IP ADDRESS  | PORTS |
|-----------|---------------------------|-------------|-------|
| Telemetry | dc.applicationinsights.us | 23.97.4.113 | 443   |

## Media

This section outlines variations and considerations when using Media services in the Azure Government environment. For service availability, see [Products available by region](#). For Azure Media Services v3 availability, see [Azure clouds and regions in which Media Services v3 exists](#).

### Media Services

For information on how to connect to Media Services v2, see [Access the Azure Media Services API with Azure AD authentication](#). The following Media Services **features are not currently available** in Azure Government:

- Analyzing – the Azure Media Indexer 2 Preview Azure Media Analytics media processor is not available in Azure Government.
- CDN integration – there is no CDN integration with streaming endpoints in Azure Government data centers.

### Media Services Video Indexer

For more information, see [Create a Video Indexer account](#).

## Migration

This section outlines variations and considerations when using Migration services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure Migrate

The following Azure Migrate **features are not currently available** in Azure Government:

- Dependency visualization functionality as Azure Migrate depends on Service Map for dependency visualization, which is currently unavailable in Azure Government.
- You can only create assessments for Azure Government as target regions and using Azure Government offers.

## Networking

This section outlines variations and considerations when using Networking services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure ExpressRoute

Azure ExpressRoute is used to create private connections between Azure Government datacenters and your on-premises infrastructure or a colocation facility. ExpressRoute connections do not go over the public Internet—

they offer optimized pathways (shortest hops, lowest latency, highest performance, and so on) to Azure Government geo-redundant regions.

- By default, all Azure Government ExpressRoute connectivity is configured active-active redundant with support for bursting, and it delivers up to 10 G circuit capacity (smallest is 50 MB).
- Microsoft owns and operates all fiber infrastructure between Azure Government regions and Azure Government ExpressRoute Meet-Me locations.
- Azure Government ExpressRoute provides connectivity to Microsoft Azure, Microsoft 365, and Dynamics 365 cloud services.

Aside from ExpressRoute, you can also use an [IPSec protected VPN](#) (site-to-site for a typical organization) to connect securely from your on-premises infrastructure to Azure Government. For network services to support Azure Government customer applications and solutions, it is recommended that ExpressRoute (private connectivity) is implemented to connect to Azure Government. If you use VPN connections, you should consider the following recommendations:

- You should contact your authorizing official/agency to determine whether private connectivity or other secure connection mechanism is required and to identify any extra restrictions to consider.
- You should decide whether to mandate that the site-to-site VPN is routed through a private connectivity zone.
- You should obtain either a Multi-Protocol Label Switching (MPLS) circuit or VPN with a licensed private connectivity access provider.

If you utilize a private connectivity architecture, you should validate that an appropriate implementation is established and maintained for your connection to the Gateway Network/Internet (GN/I) edge router demarcation point for Azure Government. Similarly, your organization must establish network connectivity between your on-premises environment and Gateway Network/Customer (GN/C) edge router demarcation point for Azure Government.

#### **BGP communities**

This section provides an overview of how BGP communities are used with ExpressRoute in Azure Government. Microsoft advertises routes in the public peering and Microsoft peering paths, with routes tagged with appropriate community values. The rationale for doing so and the details on community values are described below.

If you are connecting to Microsoft through ExpressRoute at any one peering location within the Azure Government region, you will have access to all Microsoft cloud services across all regions within the government boundary. For example, if you connected to Microsoft in Washington D.C. through ExpressRoute, you would have access to all Microsoft cloud services hosted in Azure Government. [ExpressRoute overview](#) provides details on locations and partners and a list of peering locations for Azure Government.

You can purchase more than one ExpressRoute circuit. Having multiple connections offers you significant benefits on high availability due to geo-redundancy. In cases where you have multiple ExpressRoute circuits, you will receive the same set of prefixes advertised from Microsoft on the public peering and Microsoft peering paths. This arrangement means you will have multiple paths from your network into Microsoft, which can potentially cause suboptimal routing decisions to be made within your network. As a result, you may experience suboptimal connectivity experiences to different services.

Microsoft tags prefixes advertised through public peering and Microsoft peering with appropriate BGP community values indicating the region the prefixes are hosted in. You can rely on the community values to make appropriate routing decisions to offer optimal routing to customers. For more information, see [Optimize ExpressRoute routing](#).



| AZURE GOVERNMENT REGION | BGP COMMUNITY VALUE |
|-------------------------|---------------------|
| US Gov Arizona          | 12076:51106         |
| US Gov Virginia         | 12076:51105         |
| US Gov Texas            | 12076:51108         |
| US DoD Central          | 12076:51209         |
| US DoD East             | 12076:51205         |

All routes advertised from Microsoft are tagged with the appropriate community value.

In addition to the above, Microsoft also tags prefixes based on the service they belong to. This tagging applies only to the Microsoft peering. The table below provides a mapping of service to BGP community value.

| SERVICE IN NATIONAL CLOUDS       | BGP COMMUNITY VALUE |
|----------------------------------|---------------------|
| Exchange Online                  | 12076:5110          |
| SharePoint Online                | 12076:5120          |
| Skype for Business Online        | 12076:5130          |
| Dynamics 365                     | 12076:5140          |
| Other Office 365 Online services | 12076:5200          |

#### NOTE

Microsoft does not honor any BGP community values that you set on the routes advertised to Microsoft.

### Private Link

For Private Link services availability, see [Azure Private Link availability](#).

### Traffic Manager

Traffic Manager health checks can originate from certain IP addresses for Azure Government. Review the [IP addresses in the JSON file](#) to ensure that incoming connections from these IP addresses are allowed at the endpoints to check its health status.

## Security

This section outlines variations and considerations when using Security services in the Azure Government environment. For service availability, see [Products available by region](#).

### Azure Defender for IoT

For feature variations and limitations, see [Cloud feature availability for US Government customers](#).

### Azure Information Protection

Azure Information Protection Premium is part of the [Enterprise Mobility + Security](#) suite. For details on this service and how to use it, see the [Azure Information Protection Premium Government Service Description](#).

## Azure Security Center

For feature variations and limitations, see [Cloud feature availability for US Government customers](#).

## Azure Sentinel

For feature variations and limitations, see [Cloud feature availability for US Government customers](#).

# Storage

This section outlines variations and considerations when using Storage services in the Azure Government environment. For service availability, see [Products available by region](#).

## Azure Backup

The following Azure Backup features are not currently available in Azure Government:

- Azure Disk Backup, as documented in [Azure Disk Backup support matrix](#).

## Azure managed disks

The following Azure managed disks features are not currently available in Azure Government:

- Zone-redundant storage (ZRS)

## Azure Storage

For a Quickstart that will help you get started with Storage in Azure Government, see [Develop with Storage API on Azure Government](#).

### Storage pairing in Azure Government

Azure relies on [paired regions](#) to deliver [geo-redundant storage](#). The following table shows the primary and secondary region pairings in Azure Government.

| GEOGRAPHY     | REGIONAL PAIR A | REGIONAL PAIR B |
|---------------|-----------------|-----------------|
| US Government | US Gov Arizona  | US Gov Texas    |
| US Government | US Gov Virginia | US Gov Texas    |

Table in [Guidance for developers](#) section shows URL endpoints for main Azure Storage services.

#### NOTE

All your scripts and code need to account for the appropriate endpoints. See [Configure Azure Storage Connection Strings](#).

For more information on APIs, see [Cloud Storage Account Constructor](#).

The endpoint suffix to use in these overloads is *core.usgovcloudapi.net*.

#### NOTE

If error 53 ("The network path was not found") is returned while you're [mounting the file share](#), a firewall might be blocking the outbound port. Try mounting the file share on VM that's in the same Azure subscription as the storage account.

When you're deploying the **StorSimple** Manager service, use the <https://portal.azure.us/> URL for the Azure Government portal. For deployment instructions for [StorSimple Virtual Array](#), see StorSimple Virtual Array system requirements. For the StorSimple 8000 series, see [StorSimple software, high availability, and networking](#)

[requirements](#) and go to the **Deploy** section from the left menu. For more information on StorSimple, see the [StorSimple documentation](#).

### Azure Import/Export

With Import/Export jobs for US Gov Arizona or US Gov Texas, the mailing address is for US Gov Virginia. The data is loaded into selected storage accounts from the US Gov Virginia region.

For all jobs, we recommend that you rotate your storage account keys after the job is complete to remove any access granted during the process. For more information, see [Manage storage account access keys](#).

## Web

This section outlines variations and considerations when using Web services in the Azure Government environment. For service availability, see [Products available by region](#).

### API Management

The following API Management **features are not currently available** in Azure Government:

- Azure AD B2C integration

### App Service

The following App Service **resources are not currently available** in Azure Government:

- App Service Certificate
- App Service Managed Certificate
- App Service Domain

The following App Service **features are not currently available** in Azure Government:

- Deployment
  - Deployment options: only Local Git Repository and External Repository are available
- Development tools
  - Resource explorer
- Azure Government portal
  - Private endpoints for Web Apps cannot be configured in the UI; however, private endpoints are enabled in Azure Government and you can use the Private Link Center if you need the UI.

## Next steps

Learn more about Azure Government:

- [Acquiring and accessing Azure Government](#)

Start using Azure Government:

- [Guidance for developers](#)
- [Connect with the Azure Government portal](#)