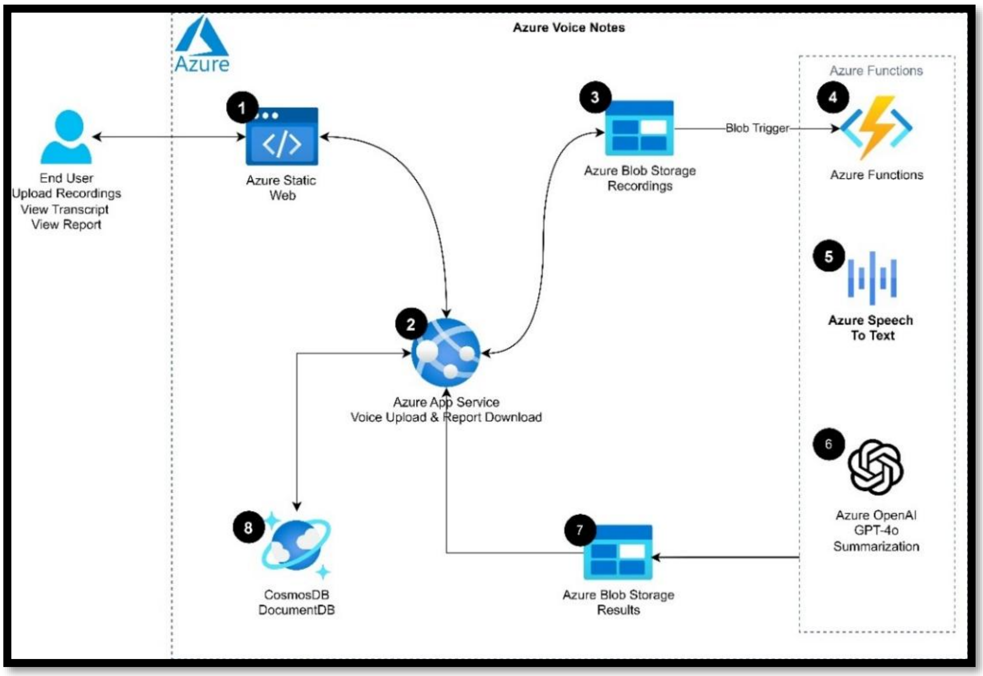
Leicestershire County Council

Version 1.2

Azure AI Transcription Accelerator Deployment User Guide

Introduction

This user guide provides instruction on manually deploying the Transcription accelerator developed by Microsoft, which will transcribe an audio recording and output it into a template using AI. If you are deploying this and want to use existing resources, then you can just deploy the additional resources you require and configure them accordingly. This deployment guide takes approximately 2.5 hours to completion.



- **Azure Static Web Apps**: Provides a web interface for user interaction.
- **Azure App Service**: Handles backend logic for user management, file handling, and workflow execution.
- **Azure Blob Storage**: Stores audio recordings, transcriptions, and reports.
- **Azure Functions**: Processes voice recordings asynchronously, handling transcription and summarization tasks.
- **Azure Speech-to-Text API**: Converts audio into structured text.
- **Azure OpenAI GPT-4o**: Summarizes transcriptions and refines text output.
- **CosmosDB (Document, Serverless)**: Manages metadata, logs, and user records.

Contents

1. Prerequisites

Before deploying the Transcription solution, the following items are required.

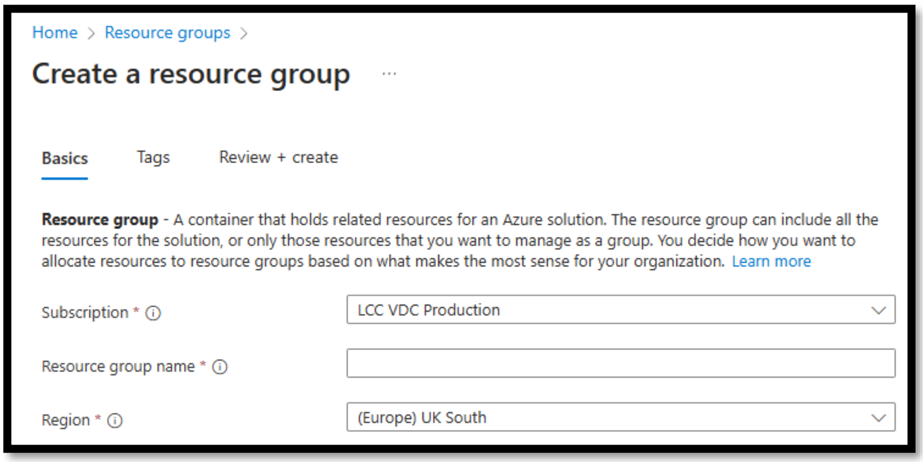
- Azure CLI – required to run via command prompt How to install the Azure CLI | Microsoft Learn
- Node.js – required to run via command prompt https://nodejs.org/en
- Static Web App CLI – run via command prompt (npm install -g @azure/static-web-apps-cli)
- Git Bash, included with GIT - https://git-scm.com/downloads)
- Azure Repos or GitHub to deploy the frontend code – required by the Static Web App

2. Creating a Resource Group

Resource groups are essential for managing and organizing your Azure resources.

Everything required for this Transcription Accelerator solution will be created inside the resource group you create. This will allow you to easily manage the solution and monitor the costs.

1. Log in to the Azure portal - https://portal.azure.com
2. In the left-hand menu, select Resource groups.
3. Click on + Create.



Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Subscription * ⓘ LCC VDC Production

Resource group name * ⓘ

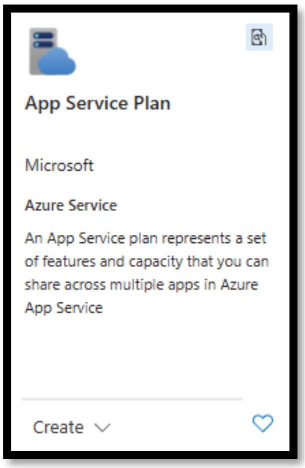
Region * ⓘ (Europe) UK South

4. Fill in the necessary details, such as Subscription, Resource group name and Region, then click Next.
5. Add any Tags you may require, then click on Review + create.
6. Once validated, click on Create.

3. Creating an App Service Plan

The App Service Plan acts as a container for your apps, providing the necessary resources and configurations to run them efficiently.

1. In the Resource group you have created, click “Create” on the top bar.
2. In the marketplace, search for “App Service Plan.”
3. Click "Create" and enter the required information. Choose **Linux** as the operating system and select your chosen Pricing plan.

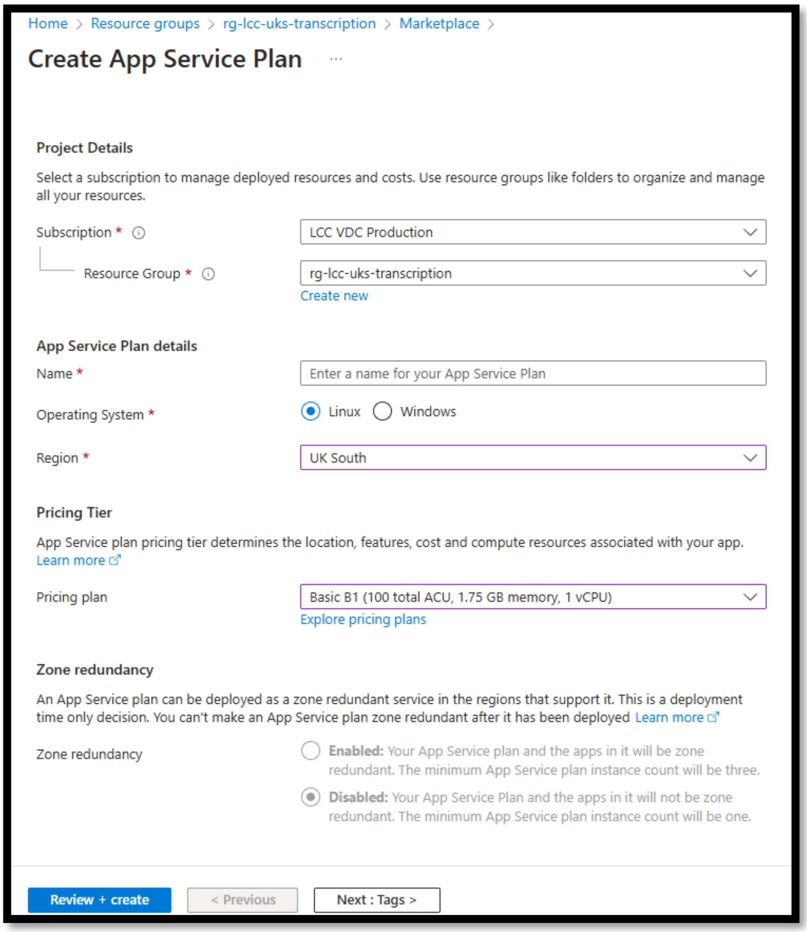


App Service Plan

Microsoft

Azure Service

An App Service plan represents a set of features and capacity that you can share across multiple apps in Azure App Service



Home > Resource groups > rg-lcc-uks-transcription > Marketplace >

Create App Service Plan

Project Details

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

Subscription * : LCC VDC Production

Resource Group * : rg-lcc-uks-transcription

App Service Plan details

Name * : Enter a name for your App Service Plan

Operating System * : ● Linux ○ Windows

Region * : UK South

Pricing Tier

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Pricing plan: Basic B1 (100 total ACU, 1.75 GB memory, 1 vCPU)

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed Learn more

Zone redundancy:

○ **Enabled:** Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three.

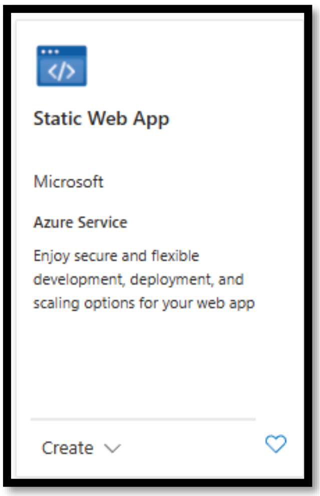
● **Disabled:** Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

4. Add any Tags you may require, then review and create the resource.

4. Creating a Static Web App

An Azure Static Web App is a service that lets you host websites that don’t change often. This is what hosts the front end that you use to interact with the solution.

1. In the Resource group, click “Create” on the top bar.
2. In the marketplace, search for “Static Web Apps.”
3. Click "Create" and enter the required information. Select your chosen ‘Hosting plan’ and deployment ‘Source’.

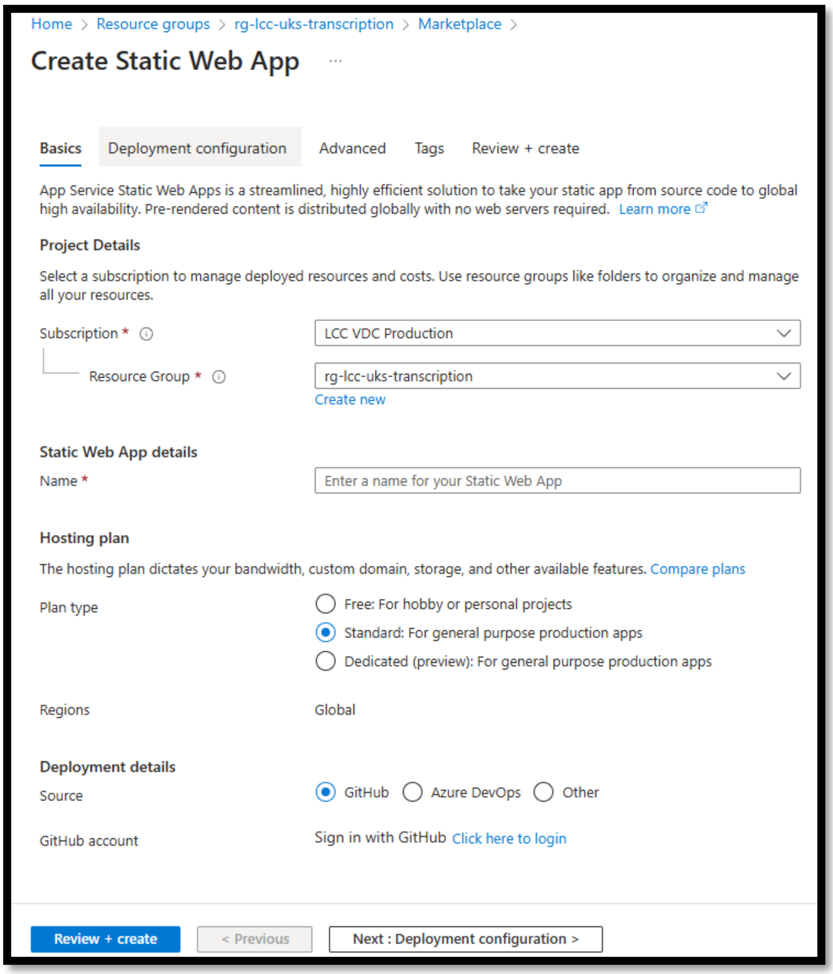


Static Web App

Microsoft

Azure Service

Enjoy secure and flexible development, deployment, and scaling options for your web app



Home > Resource groups > rg-lcc-uks-transcription > Marketplace >

Create Static Web App ...

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered content is distributed globally with no web servers required. Learn more

Project Details

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

Subscription * LCC VDC Production

Resource Group * rg-lcc-uks-transcription

Create new

Static Web App details

Name * Enter a name for your Static Web App

Hosting plan

The hosting plan dictates your bandwidth, custom domain, storage, and other available features. Compare plans

Plan type
○ Free: For hobby or personal projects
● Standard: For general purpose production apps
○ Dedicated (preview): For general purpose production apps

Regions Global

Deployment details

Source ● GitHub ○ Azure DevOps ○ Other

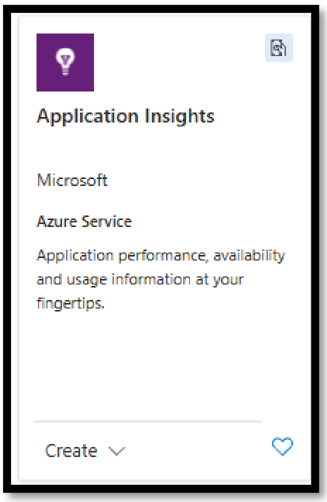
GitHub account Sign in with GitHub Click here to login

4. Provide a Name and select the Plan you want.
5. Choose your Deployment source and fill in the details as needed.
6. For ‘Build Presets’ choose next.js
7. Click on Next and setup your deployment config.
8. Click next, in the Advanced tab select the region.
9. Add any Tags you may require, then review and create the resource.

5. Create Application Insights

Application Insights provides monitoring and tracking for the solution.

1. Click on "Create a resource" in the top left.
2. Search for "Application Insights" in the marketplace.
3. Click "Create" and fill in the required details, such as Subscription, Resource group, and Region. Select an existing Log Analytics Workspace of your choice or create a new one as part of this deployment.

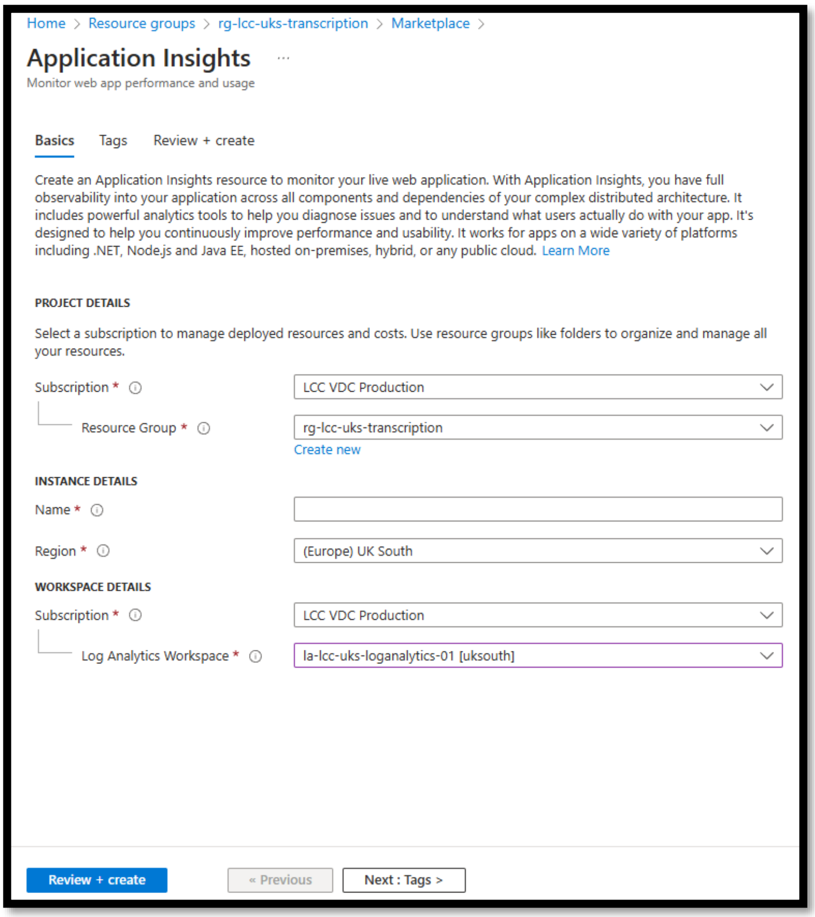


Application Insights

Microsoft

Azure Service

Application performance, availability and usage information at your fingertips.



Home > Resource groups > rg-lcc-uks-transcription > Marketplace >

Application Insights

Monitor web app performance and usage

Basics
Tags
Review + create

Create an Application Insights resource to monitor your live web application. With Application Insights, you have full observability into your application across all components and dependencies of your complex distributed architecture. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and Java EE, hosted on-premises, hybrid, or any public cloud. Learn More

PROJECT DETAILS

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

Subscription * LCC VDC Production

Resource Group * rg-lcc-uks-transcription

Create new

INSTANCE DETAILS

Name *

Region * (Europe) UK South

WORKSPACE DETAILS

Subscription * LCC VDC Production

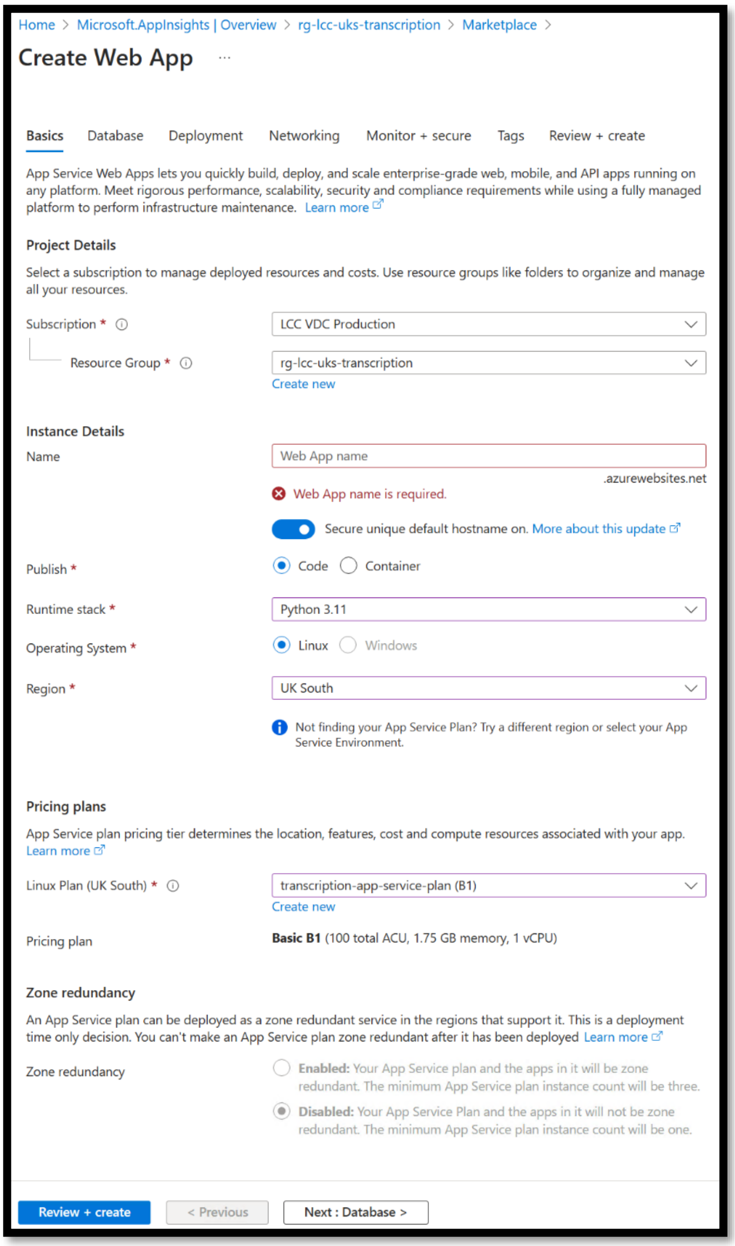
Log Analytics Workspace * la-lcc-uks-loganalytics-01 [uksouth]

5. Add any Tags you may require, then review and create the resource.

6. Creating a Web App

Web Apps allow you to create and host websites or web applications using various frameworks. This is assigned to the App Service Plan.

1. In the Resource group, click “Create” on the top bar.
2. In the marketplace, search for “Web App.”
3. Click "Create" and fill in the required details, such as Subscription, Resource group, and Region. Select the Runtime stack **Python 3.11**.



Create Web App ...

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. Learn more

Project Details

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

Subscription * : LCC VDC Production

Resource Group * : rg-lcc-uks-transcription

Create new

Instance Details

Name: Web App name

.azurewebsites.net

Web App name is required.

Secure unique default hostname on. More about this update

Publish * : ● Code ○ Container

Runtime stack * : Python 3.11

Operating System * : ● Linux ○ Windows

Region * : UK South

Not finding your App Service Plan? Try a different region or select your App Service Environment.

Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Linux Plan (UK South) * : transcription-app-service-plan (B1)

Create new

Pricing plan: **Basic B1** (100 total ACU, 1.75 GB memory, 1 vCPU)

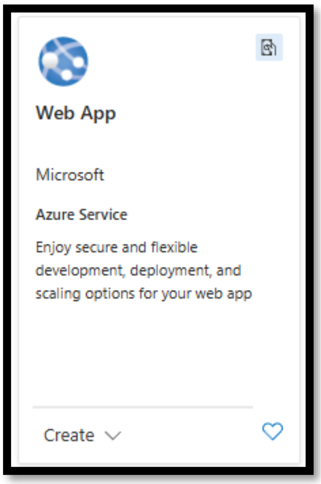
Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed Learn more

Zone redundancy:

○ **Enabled:** Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three.

● **Disabled:** Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.



Web App

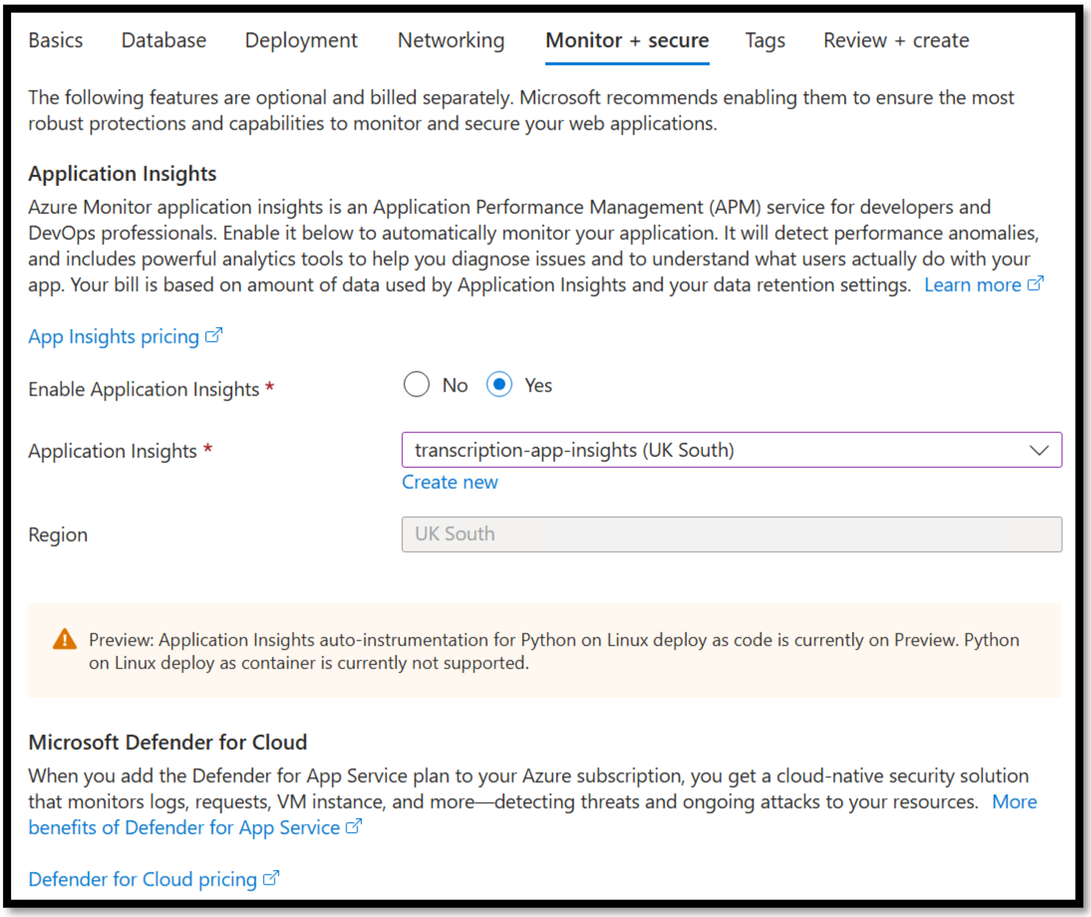
Microsoft

Azure Service

Enjoy secure and flexible development, deployment, and scaling options for your web app

4. Select the name of the ‘App Service Plan’ created earlier as the Linux Plan.
5. Click next, setting up a Database is not required.

6. Click next, a Continuous deployment is not required. Basic authentication can be set to **'Disable'**.
7. Click next, setup networking as required.
8. Click next, enable Application insights and select the Application Insights resource created previously (recommended).



The following features are optional and billed separately. Microsoft recommends enabling them to ensure the most robust protections and capabilities to monitor and secure your web applications.

Application Insights

Azure Monitor application insights is an Application Performance Management (APM) service for developers and DevOps professionals. Enable it below to automatically monitor your application. It will detect performance anomalies, and includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. Your bill is based on amount of data used by Application Insights and your data retention settings. Learn more

App Insights pricing

Enable Application Insights * ○ No ● Yes

Application Insights * transcription-app-insights (UK South)

Create new

Region UK South

Preview: Application Insights auto-instrumentation for Python on Linux deploy as code is currently on Preview. Python on Linux deploy as container is currently not supported.

Microsoft Defender for Cloud

When you add the Defender for App Service plan to your Azure subscription, you get a cloud-native security solution that monitors logs, requests, VM instance, and more—detecting threats and ongoing attacks to your resources. More benefits of Defender for App Service

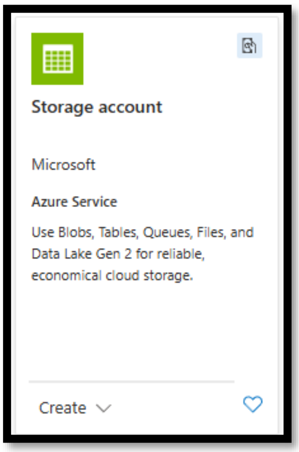
Defender for Cloud pricing

6. Add any Tags you may require, then review and create the resource.

7. Creating a Storage Account

Storage accounts provided various means to store both structured and unstructured data. For this solution, Blob storage is used which is required to store the recordings, transcripts and the summary output.

1. In the Resource group, click “Create” on the top bar.
2. In the marketplace, search for “Storage account”

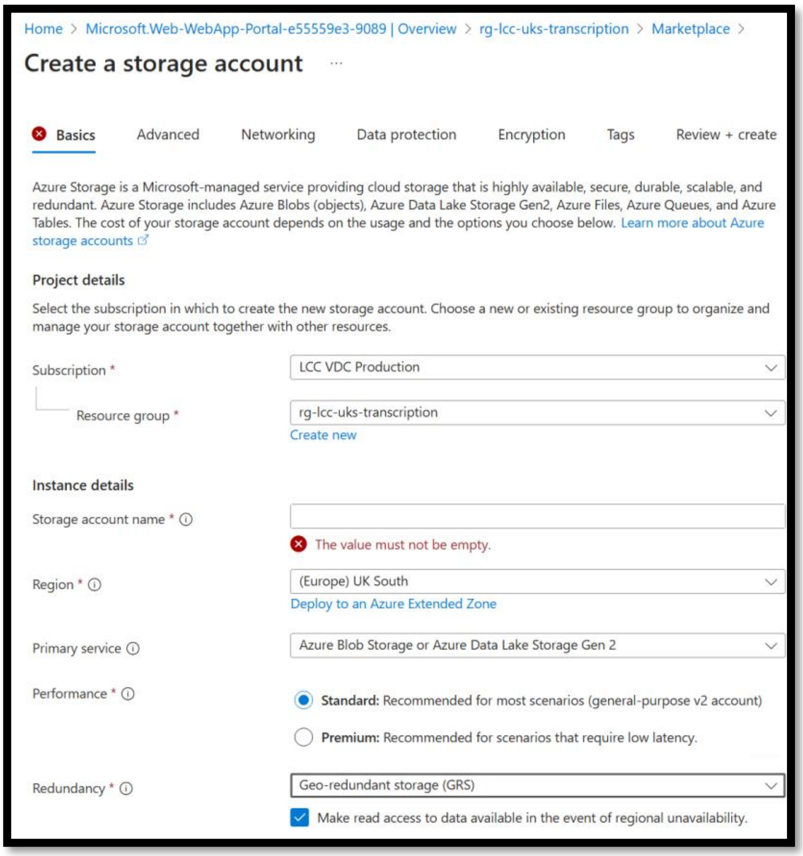


Storage account

Microsoft

Azure Service

Use Blobs, Tables, Queues, Files, and Data Lake Gen 2 for reliable, economical cloud storage.



Home > Microsoft.Web-WebApp-Portal-e55559e3-9089 | Overview > rg-lcc-uks-transcription > Marketplace >

Create a storage account ...

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

LCC VDC Production

Resource group *

rg-lcc-uks-transcription

Create new

Instance details

Storage account name *

The value must not be empty.

Region *

(Europe) UK South

Deploy to an Azure Extended Zone

Primary service

Azure Blob Storage or Azure Data Lake Storage Gen 2

Performance *
- ● **Standard:** Recommended for most scenarios (general-purpose v2 account)
- ○ **Premium:** Recommended for scenarios that require low latency.

Redundancy *

Geo-redundant storage (GRS)

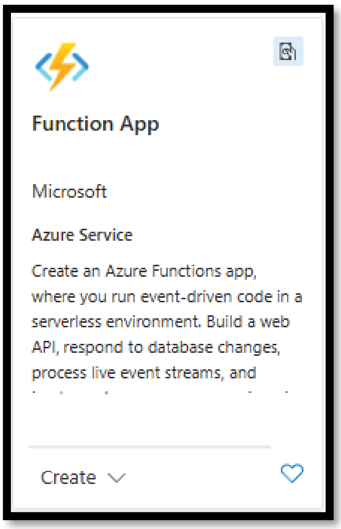
☑ Make read access to data available in the event of regional unavailability.

3. Click "Create" and fill in the required details, such as Subscription, Resource group, and Region.
4. Choose “**Azure blob Storage** or **Azure Data Lake Storage Gen 2**” as the Primary service.
5. For the Performance and Redundancy options, configure these as per your requirements.
6. Click next, configure Advanced, Networking, Data protection and Encryption as required – the current Microsoft default settings can work for this deployment.
7. Add any Tags you may require, then review and create the resource.

8. Creating a Function App

Function Apps allow you to run event-driven code without managing infrastructure.

1. In the Resource group, click “Create” on the top bar.
2. In the marketplace, search for “Function App” and select the ‘App Service’ as the Hosting plan.

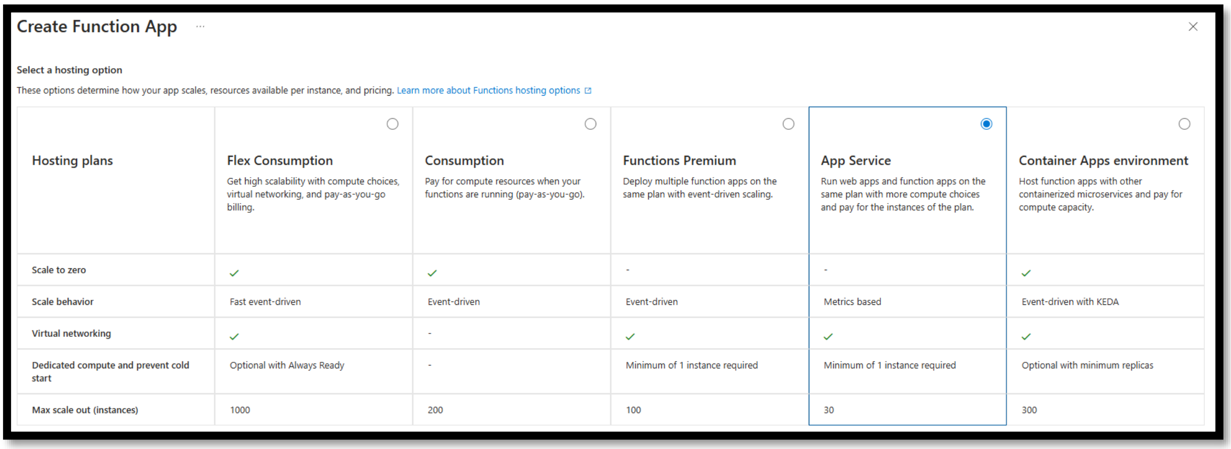


Function App

Microsoft

Azure Service

Create an Azure Functions app, where you run event-driven code in a serverless environment. Build a web API, respond to database changes, process live event streams, and



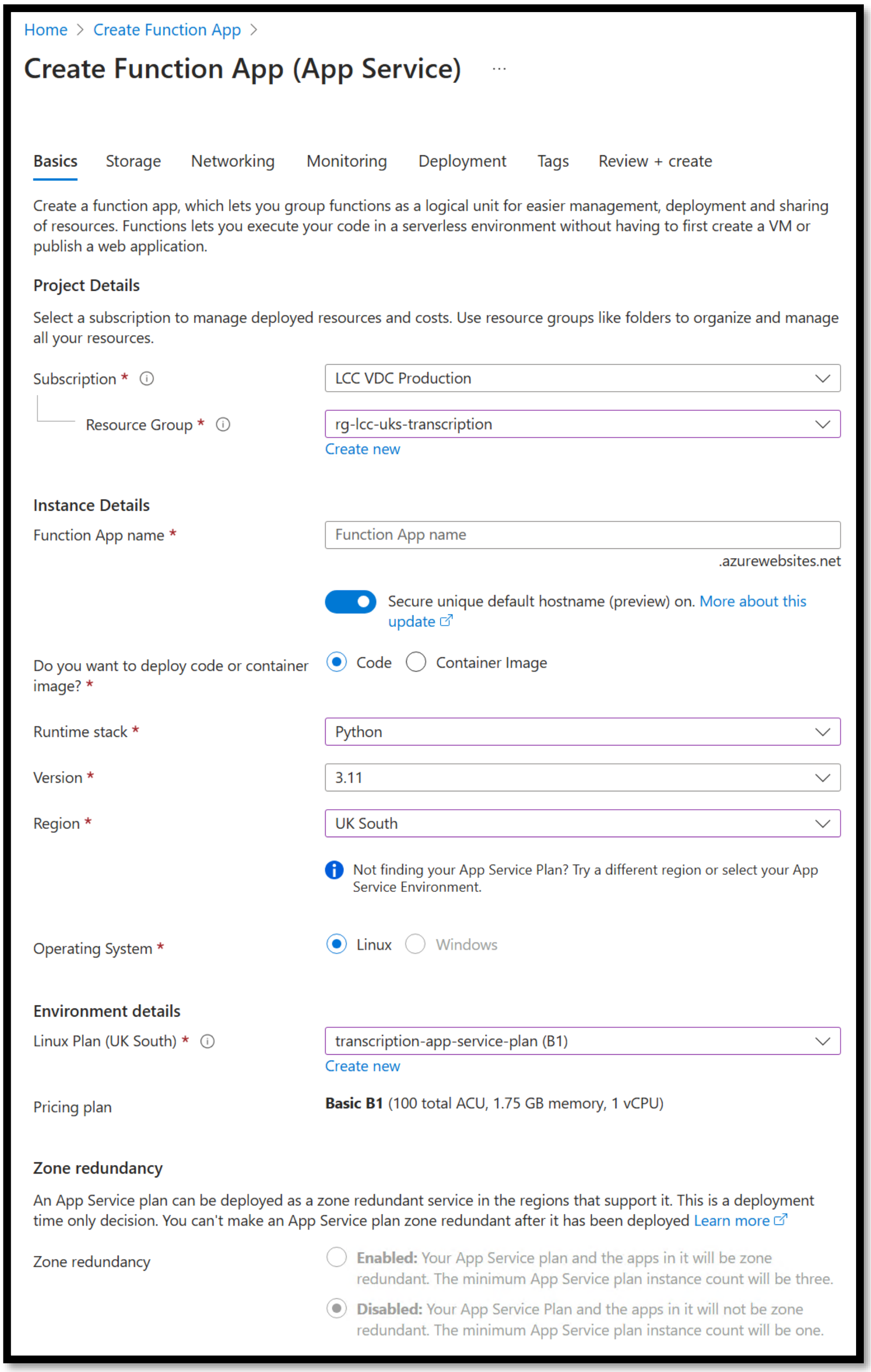
Create Function App

Select a hosting option

These options determine how your app scales, resources available per instance, and pricing. Learn more about Functions hosting options

Hosting plans	○ Flex Consumption Get high scalability with compute choices, virtual networking, and pay-as-you-go billing.	○ Consumption Pay for compute resources when your functions are running (pay-as-you-go).	○ Functions Premium Deploy multiple function apps on the same plan with event-driven scaling.	● App Service Run web apps and function apps on the same plan with more compute choices and pay for the instances of the plan.	○ Container Apps environment Host function apps with other containerized microservices and pay for compute capacity.
Scale to zero	✓	✓	-	-	✓
Scale behavior	Fast event-driven	Event-driven	Event-driven	Metrics based	Event-driven with KEDA
Virtual networking	✓	-	✓	✓	✓
Dedicated compute and prevent cold start	Optional with Always Ready	-	Minimum of 1 instance required	Minimum of 1 instance required	Optional with minimum replicas
Max scale out (instances)	1000	200	100	30	300

3. Click "Create" and fill in the required details, such as Subscription, Resource group, and Region.
   - Provide a unique Name, and for “Do you want to deploy code or container image?” select **Code**.
   - Select the Runtime stack as **Python** and Version as **3.11**.
   - For Linux Plan, select the app service plan you created earlier.



Home > Create Function App >

Create Function App (App Service) ...

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

Project Details

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

Subscription * : LCC VDC Production

Resource Group * : rg-lcc-uks-transcription
Create new

Instance Details

Function App name * : Function App name
.azurewebsites.net

Secure unique default hostname (preview) on. More about this update

Do you want to deploy code or container image? * : ● Code ○ Container Image

Runtime stack * : Python

Version * : 3.11

Region * : UK South

Not finding your App Service Plan? Try a different region or select your App Service Environment.

Operating System * : ● Linux ○ Windows

Environment details

Linux Plan (UK South) * : transcription-app-service-plan (B1)
Create new

Pricing plan : **Basic B1** (100 total ACU, 1.75 GB memory, 1 vCPU)

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed Learn more

Zone redundancy

○ **Enabled:** Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three.

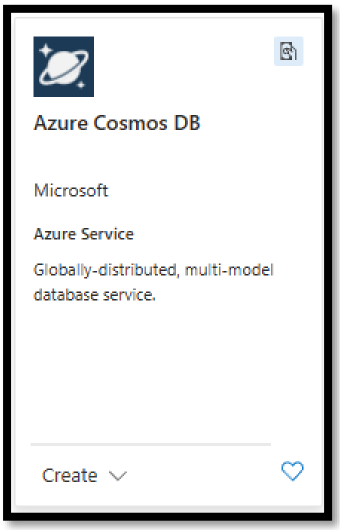
● **Disabled:** Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

4. Click next, under Storage select the storage account you created earlier.
5. Click next, setup Networking configure this as required.
6. Click next, under Monitoring setup application insights using the one you created earlier. This will be helpful for debugging.
7. Click next, setup continuous deployment if needed but this is not required.
8. Add any Tags you may require, then review and create the resource.

9. Creating a Cosmos DB

Cosmos DB provides globally distributed, multi-model database capabilities for your applications. This is used for the authentication, the prompts, and recording and tracking jobs.

1. In the Resource group, click “Create” on the top bar.
2. In the marketplace, search for “Azure Cosmos DB.” Click Create and then create the “Azure Cosmos DB for NoSQL”

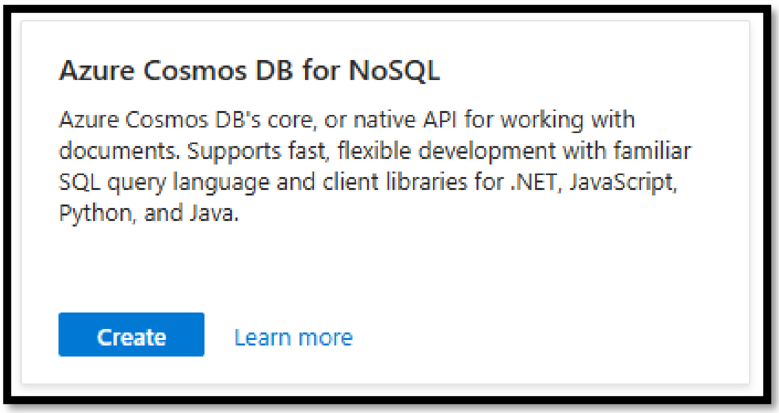


Azure Cosmos DB

Microsoft

Azure Service

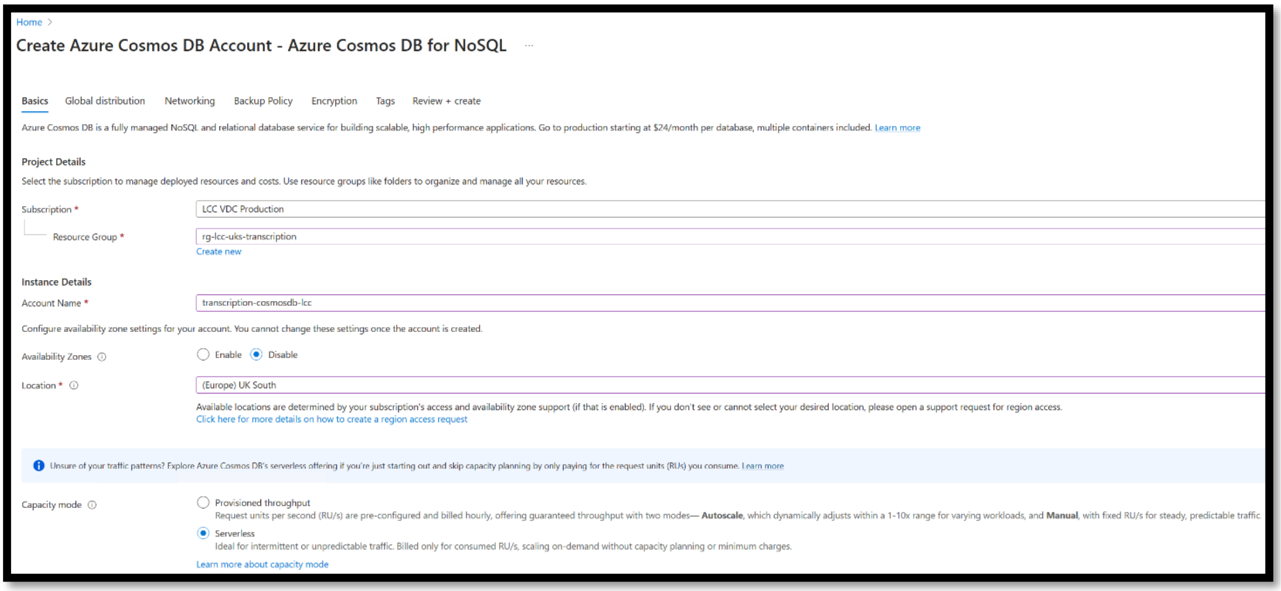
Globally-distributed, multi-model database service.



Azure Cosmos DB for NoSQL

Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

3. Click "Create" and fill in the required details, such as Subscription, Resource group, and Region. Choose the Capacity mode you require.



Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. Go to production starting at $24/month per database, multiple containers included. 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 *
LCC VDC Production

Resource Group *
rg-lcc-uks-transcription

Create new

Instance Details

Account Name *
transcription-cosmosdb-lcc

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones ○ Enable ● Disable

Location *
(Europe) UK South

Available locations are determined by your subscription's access and availability zone support (if that is enabled). If you don't see or cannot select your desired location, please open a support request for region access.
Click here for more details on how to create a region access request

Unsure of your traffic patterns? Explore Azure Cosmos DB's serverless offering if you're just starting out and skip capacity planning by only paying for the request units (RUs) you consume. Learn more

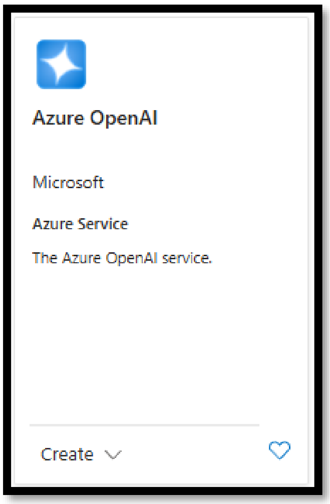
Capacity mode
○ Provisioned throughput
Request units per second (RU/s) are pre-configured and billed hourly, offering guaranteed throughput with two modes— **Autoscale**, which dynamically adjusts within a 1-10x range for varying workloads, and **Manual**, with fixed RU/s for steady, predictable traffic
● Serverless
Ideal for intermittent or unpredictable traffic. Billed only for consumed RU/s, scaling on-demand without capacity planning or minimum charges.
Learn more about capacity mode

4. Click next, configure Global distribution, Networking, and Backup Policy as you require.
5. Click next, under Encryption select **Service-Managed Keys**.
6. Add any Tags you may require, then review and create the resource.

10. Deploying the Azure OpenAI Service

Azure OpenAI services is where you deploy an LLM model e.g. GPT4o-mini that will be used to integrate the transcription solution with.

1. In the Resource group, click "Create" on the top bar.
2. In the marketplace, search for "Azure OpenAI."
3. Choose your Subscription, Resource group and Region. Set your chosen pricing tier.

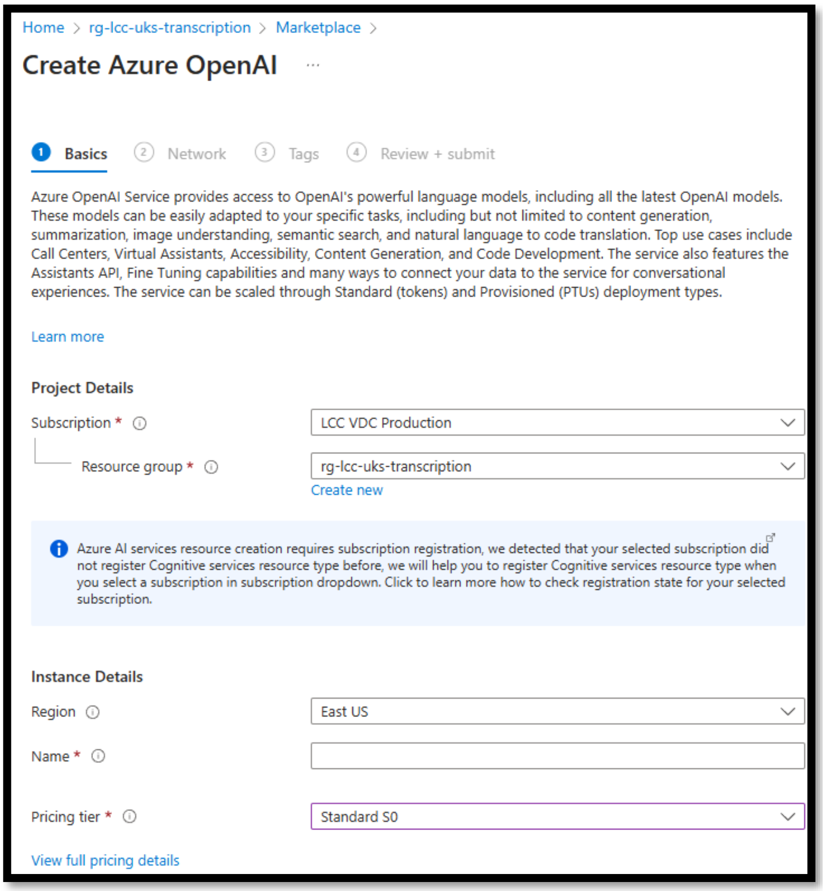


Azure OpenAI

Microsoft

Azure Service

The Azure OpenAI service.



Home > rg-lcc-uks-transcription > Marketplace >

Create Azure OpenAI ...

Azure OpenAI Service provides access to OpenAI's powerful language models, including all the latest OpenAI models. These models can be easily adapted to your specific tasks, including but not limited to content generation, summarization, image understanding, semantic search, and natural language to code translation. Top use cases include Call Centers, Virtual Assistants, Accessibility, Content Generation, and Code Development. The service also features the Assistants API, Fine Tuning capabilities and many ways to connect your data to the service for conversational experiences. The service can be scaled through Standard (tokens) and Provisioned (PTUs) deployment types.

Learn more

Project Details

Subscription * LCC VDC Production

Resource group * rg-lcc-uks-transcription

Create new

Azure AI services resource creation requires subscription registration, we detected that your selected subscription did not register Cognitive services resource type before, we will help you to register Cognitive services resource type when you select a subscription in subscription dropdown. Click to learn more how to check registration state for your selected subscription.

Instance Details

Region East US

Name *

Pricing tier * Standard S0

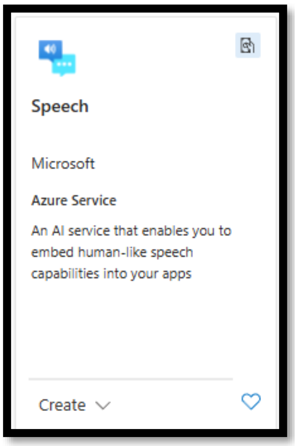
View full pricing details

4. Click next, configure Networking as you require.
5. Add any Tags you may require, then review and create the resource.

11. Deploying the Speech Service

The Speech Service offers speech-to-text, text-to-speech, and speech translation capabilities. This is used to transcribe the recording only.

1. In the Resource group, click “Create” on the top bar.
2. In the marketplace, search for “Speech” and create the resource.
3. Choose your Subscription, Resource group, Region and select your preferred Pricing tier.

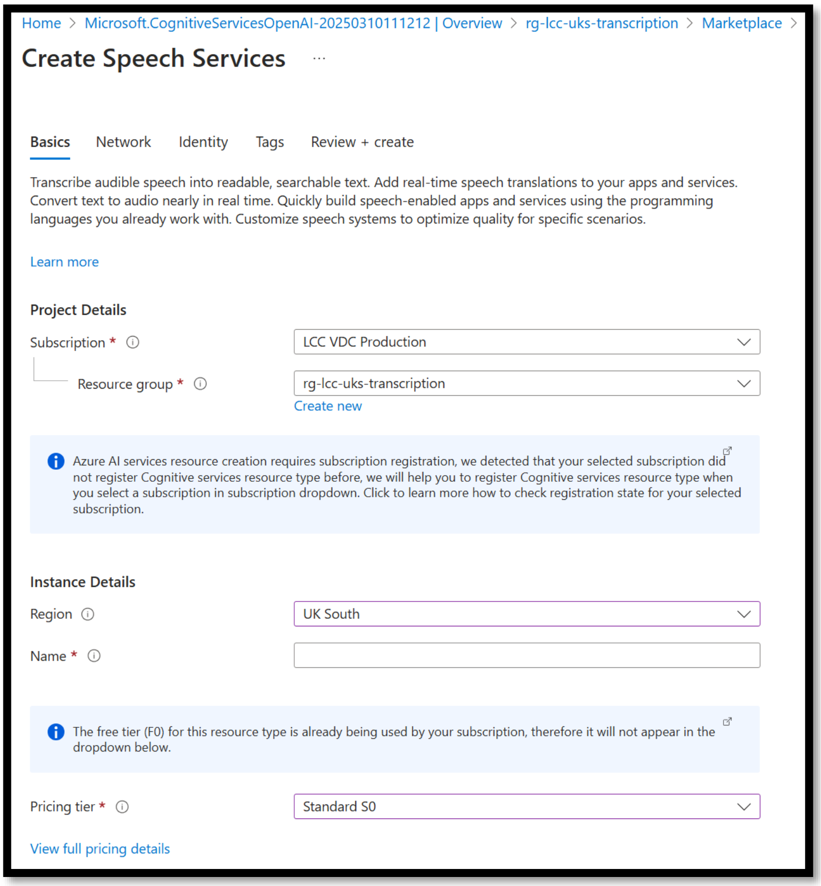


Speech

Microsoft

Azure Service

An AI service that enables you to embed human-like speech capabilities into your apps



Home > Microsoft.CognitiveServicesOpenAI-20250310111212 | Overview > rg-lcc-uks-transcription > Marketplace >

Create Speech Services ...

Basics
Network
Identity
Tags
Review + create

Transcribe audible speech into readable, searchable text. Add real-time speech translations to your apps and services. Convert text to audio nearly in real time. Quickly build speech-enabled apps and services using the programming languages you already work with. Customize speech systems to optimize quality for specific scenarios.

Learn more

Project Details

Subscription * LCC VDC Production

Resource group * rg-lcc-uks-transcription

Create new

Azure AI services resource creation requires subscription registration, we detected that your selected subscription did not register Cognitive services resource type before, we will help you to register Cognitive services resource type when you select a subscription in subscription dropdown. Click to learn more how to check registration state for your selected subscription.

Instance Details

Region UK South

Name *

The free tier (F0) for this resource type is already being used by your subscription, therefore it will not appear in the dropdown below.

Pricing tier * Standard S0

View full pricing details

4. Click next, configure Networking as you require.
5. Click next, under Identity enable **System assigned identity**.
6. Add any Tags you may require, then review and create the resource.

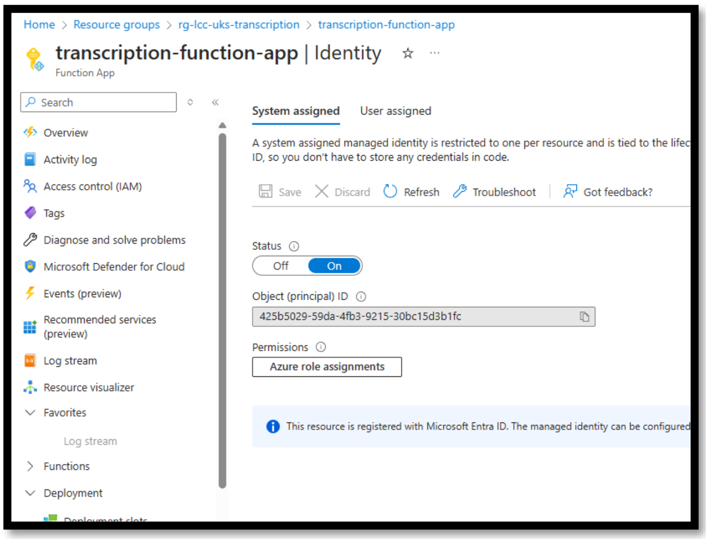
12. Resource Permissions

System Assigned Managed Identity needs to be enabled for the following resources, which will allow resources created above to securely access each other where required:

- WebApp
- Cosmos
- Speech service
- Function App

Function App Permissions

1. On the function app, in the left-hand menu, select "Identity" under the "Settings" section. Under the "System assigned" tab, switch the Status to "On" and save the changes. This action will create a managed identity for the Azure service.



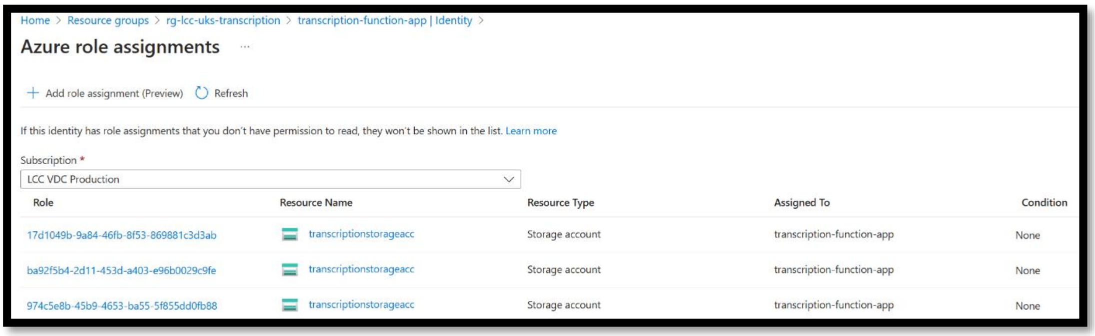
A system assigned managed identity is restricted to one per resource and is tied to the lifec
ID, so you don't have to store any credentials in code.

Status: Off ● On

Object (principal) ID: 425b5029-59da-4fb3-9215-30bc15d3b1fc

This resource is registered with Microsoft Entra ID. The managed identity can be configured

2. Under 'Azure role assignment' in the same location, click on 'Add role assignment'. Set the scope to storage, choose your subscription, choose your resource. Then set the roles
   - Storage Account Contributor
   - Storage Blob Data contributor
   - Storage Queue Data Contributor


Home > Resource groups > rg-lcc-uks-transcription > transcription-function-app | Identity >
Azure role assignments
Add role assignment (Preview)
Refresh
Learn more

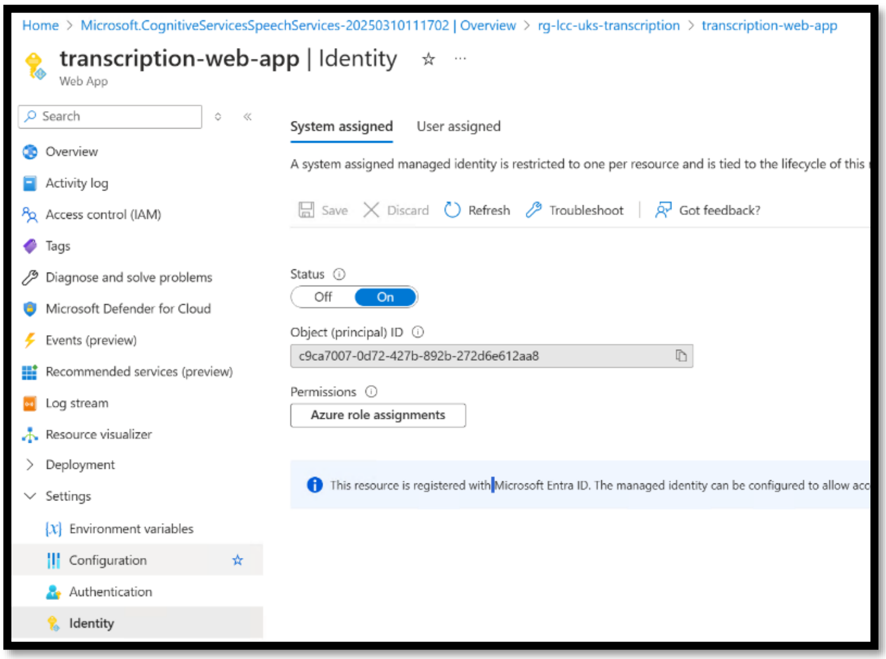
If this identity has role assignments that you don't have permission to read, they won't be shown in the list.

Subscription *: LCC VDC Production

Role	Resource Name	Resource Type	Assigned To	Condition
17d1049b-9a84-46fb-8f53-869881c3d3ab	transcriptionstorageacc	Storage account	transcription-function-app	None
ba92f5b4-2d11-453d-a403-e96b0029c9fe	transcriptionstorageacc	Storage account	transcription-function-app	None
974c5e8b-45b9-4653-ba55-5f855dd0fb88	transcriptionstorageacc	Storage account	transcription-function-app	None

WebApp Permissions

1. On the function app, in the left-hand menu, select "Identity" under the "Settings" section. Under the "System assigned" tab, switch the Status to "On" and save the changes. This action will create a managed identity for the Azure service.



transcription-web-app | Identity

Web App

System assigned | User assigned

A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this

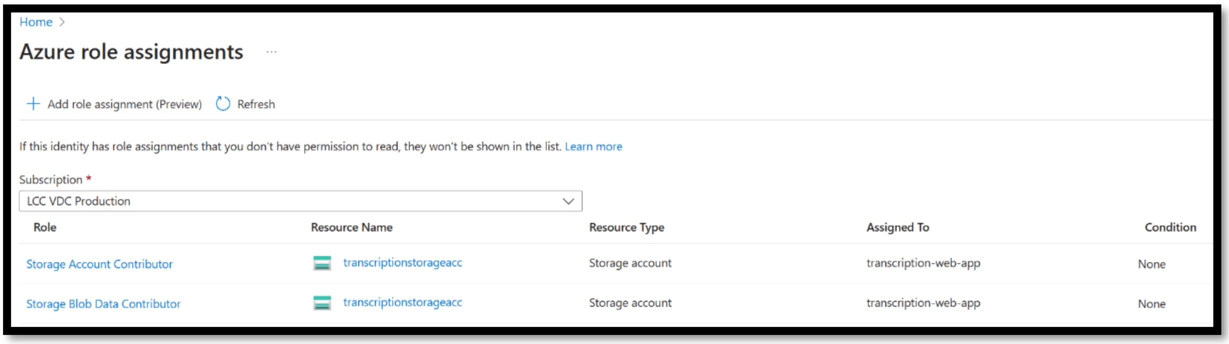
Status: Off **On**

Object (principal) ID: c9ca7007-0d72-427b-892b-272d6e612aa8

Permissions

This resource is registered with Microsoft Entra ID. The managed identity can be configured to allow acc

3. Under ‘Azure role assignment’ in the same location, click on ‘Add role assignment’. Set the scope to storage, choose your subscription, choose your resource. Then set the roles
   - Storage Account Contributor
   - Storage Blob Data Contributor



Azure role assignments

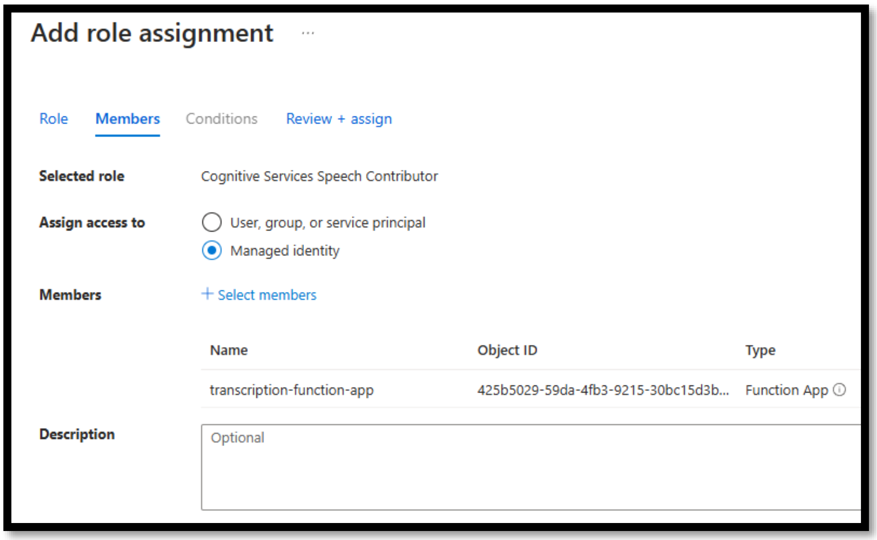
If this identity has role assignments that you don't have permission to read, they won't be shown in the list. Learn more

Subscription *: LCC VDC Production

Role	Resource Name	Resource Type	Assigned To	Condition
Storage Account Contributor	transcriptionstorageacc	Storage account	transcription-web-app	None
Storage Blob Data Contributor	transcriptionstorageacc	Storage account	transcription-web-app	None

Speech Service Permissions

1. On the Speech Service resource, navigate to Access Control (IAM), click add, then “Add role Assignment”. Find the “Cognitive Services Speech Contributor” role and click it to highlight it then click next. Change “Assign access” to “Managed Identity” then click “Select members”, set “Managed identity” to “Function App” and find the function app you created earlier. Select it then click “select” and then click “review & assigned” twice.



Add role assignment ···

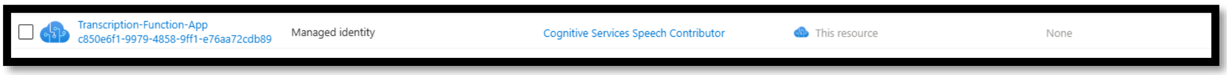
Selected role: Cognitive Services Speech Contributor

Assign access to: ○ User, group, or service principal ● Managed identity

Members: + Select members

Name	Object ID	Type
transcription-function-app	425b5029-59da-4fb3-9215-30bc15d3b...	Function App ⓘ

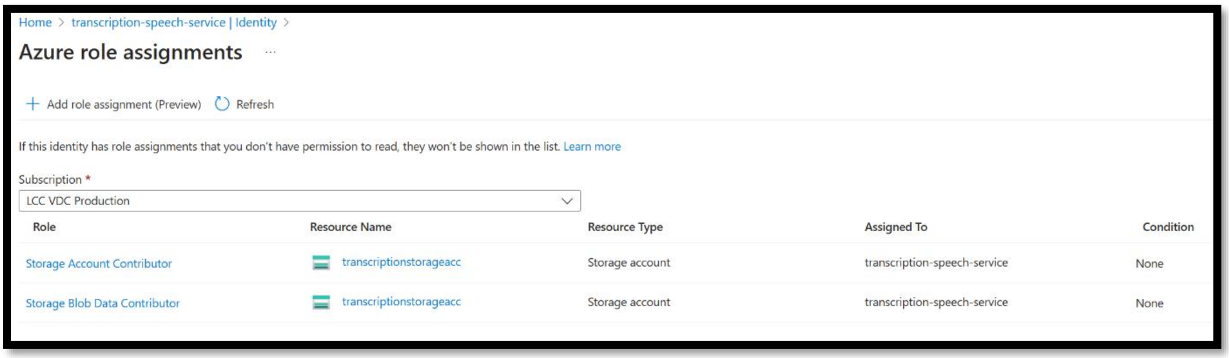
Description: Optional



□ Transcription-Function-App c850e6f1-9979-4858-9ff1-e76aa72cdb89	Managed identity	Cognitive Services Speech Contributor	This resource	None

2. Under Resource Management, select Identity and then Azure role assignments. You will need to set scope to storage, choose your subscription, choose your resource. Then set the roles

- Storage Account Contributor
- Storage Blob Data Contributor



Home > transcription-speech-service | Identity >

Azure role assignments ···

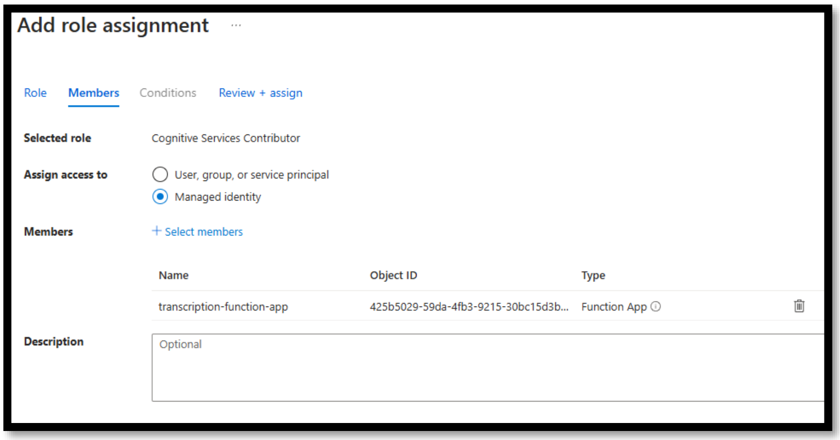
If this identity has role assignments that you don't have permission to read, they won't be shown in the list. Learn more

Subscription *: LCC VDC Production

Role	Resource Name	Resource Type	Assigned To	Condition
Storage Account Contributor	transcriptionstorageacc	Storage account	transcription-speech-service	None
Storage Blob Data Contributor	transcriptionstorageacc	Storage account	transcription-speech-service	None

Azure OpenAI Service Permissions

1. On the Azure OpenAI Service resource, navigate to Access Control (IAM), click add, then "Add role Assignment". Find the "Cognitive Services Contributor" role and click it to highlight it then click next. Change "Assign access" to "Managed Identity" then click "Select members", set "Managed identity" to "Function App" and find the function app you created earlier. Select it then click "select" and then click "review & assigned" twice.



Add role assignment ···

Selected role: Cognitive Services Contributor

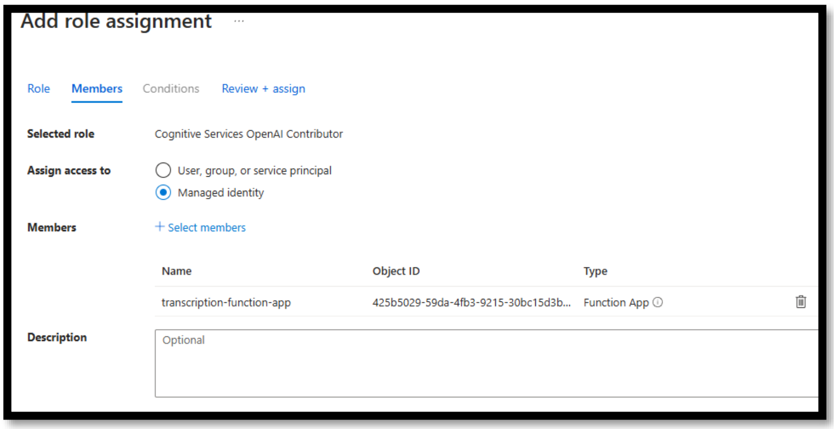
Assign access to: ○ User, group, or service principal ● Managed identity

Members: + Select members

Name	Object ID	Type
transcription-function-app	425b5029-59da-4fb3-9215-30bc15d3b...	Function App ⓘ

Description: Optional

2. Add another role in the same way for "Cognitive Services OpenAI Contributor" role.



Add role assignment ···

Selected role: Cognitive Services OpenAI Contributor

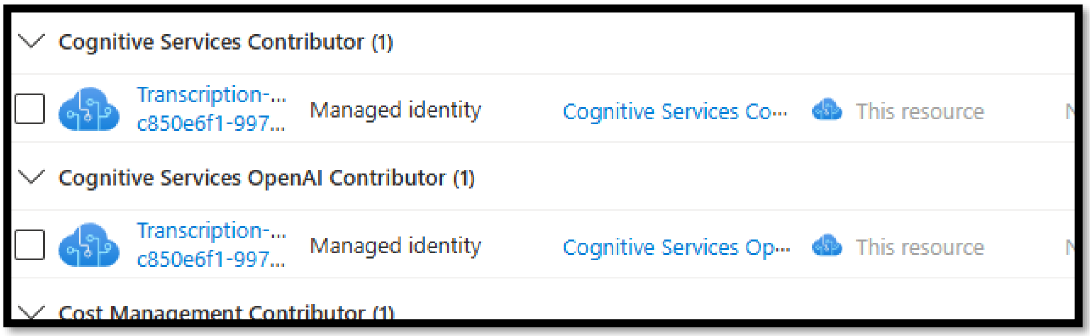
Assign access to: ○ User, group, or service principal ● Managed identity

Members: + Select members

Name	Object ID	Type
transcription-function-app	425b5029-59da-4fb3-9215-30bc15d3b...	Function App ⓘ

Description: Optional

Once the permissions have been added, it should look like the following:



Cognitive Services Contributor (1)

□ Transcription-... c850e6f1-997... | Managed identity | Cognitive Services Co... | This resource

Cognitive Services OpenAI Contributor (1)

□ Transcription-... c850e6f1-997... | Managed identity | Cognitive Services Op... | This resource

Cost Management Contributor (1)

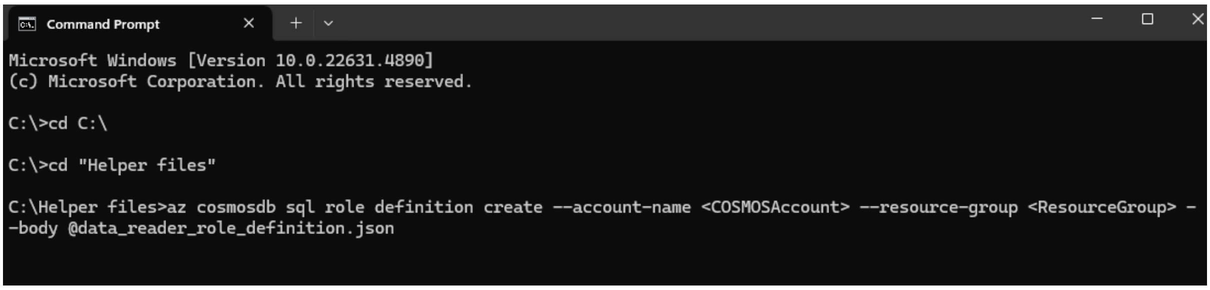
Setting Additional Permissions via Azure CLI

The next set of permissions are set using the Azure CLI, to provision resource access to Cosmos. After applying these permissions via the Azure CLI, they will not appear in the Azure Portal. *NB: For tenants with newer Azure subscriptions, you may be able to set these permission sets using the Azure Portal GUI.*

Assign Data Reader Role to Azure Cosmos DB Account

1. Create the 'Data Reader' role on the Azure Cosmos DB Account. edit the file provided named 'data_reader_role_definition.json' to add your details, then run the command below to create.
2. Open Command Prompt, change directory to the location of the file you have saved above, and run the following command to create the role:

   az cosmosdb sql role definition create --account-name ***<COSMOSAccount>*** *--resource-group* ***<ResourceGroup>*** *--body @data_reader_role_definition.json*



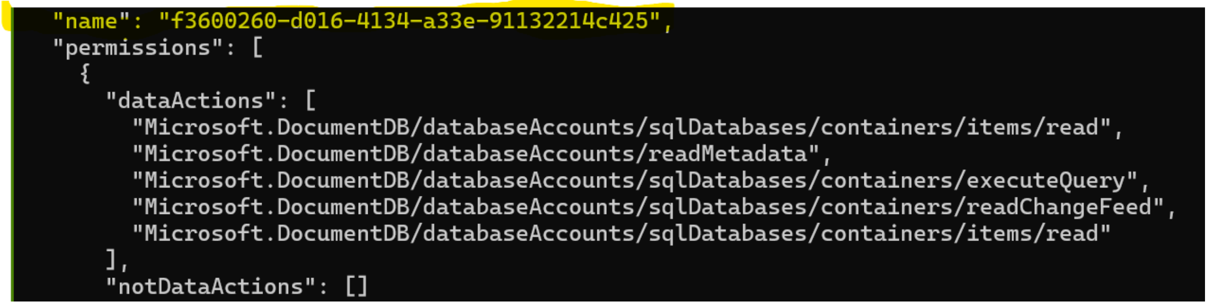
```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\>cd C:\

C:\>cd "Helper files"

C:\Helper files>az cosmosdb sql role definition create --account-name <COSMOSAccount> --resource-group <ResourceGroup> -
-body @data_reader_role_definition.json
```

3. Take note of the **name** value from the output.



```
    "name": "f3600260-d016-4134-a33e-91132214c425",
    "permissions": [
      {
        "dataActions": [
          "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/items/read",
          "Microsoft.DocumentDB/databaseAccounts/readMetadata",
          "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/executeQuery",
          "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/readChangeFeed",
          "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/items/read"
        ],
        "notDataActions": []
```

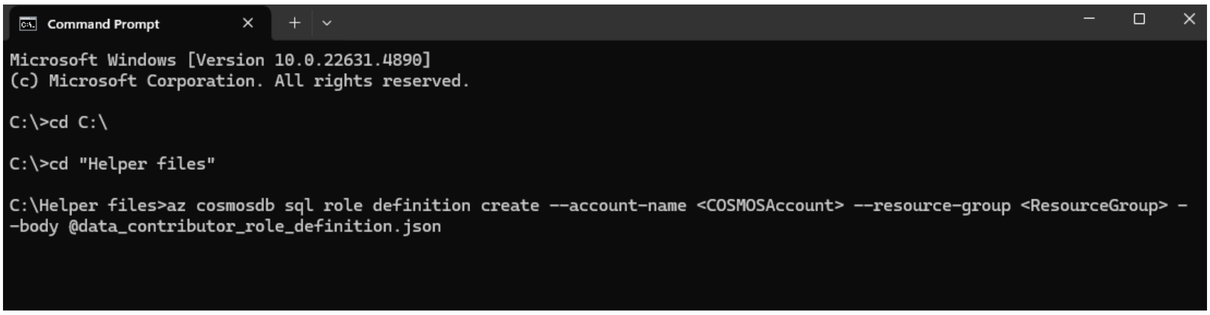
   Run the following command to assign the **Data Reader** role to the function app.

4. az cosmosdb sql role assignment create --account-name **<cosmosaccountname>** --resource-group **<resourcegroupname>** --role-definition-id /subscriptions/**<subscriptionID>**/resourceGroups/**<resourcegroupname>**/providers/Microsoft.DocumentDB/databaseAccounts/**<cosmosaccountname>**/sqlRoleDefinitions/**<INSERT NAME VALUE>** --principal-id **<function app principle id>** --scope /subscriptions/**<subscriptionID>**/resourceGroups/**<resourcegroupname>**/providers/Microsoft.DocumentDB/databaseAccounts/**<cosmosaccountname>**

Assign Data Contributor Role to Azure Cosmos DB Account

1. Create the 'Data Contributor role on the Azure Cosmos DB Account. Edit the file provided named 'data_contributor_role_definition.json' to add your details, then run the command below to create.
2. Using Azure CLI, change directory to the location of the file you have saved above, and create the role and run the following command:

   az cosmosdb sql role definition create --account-name <COSMOSAccount> --resource-group <ResourceGroup> --body @data_contributor_role_definition.json



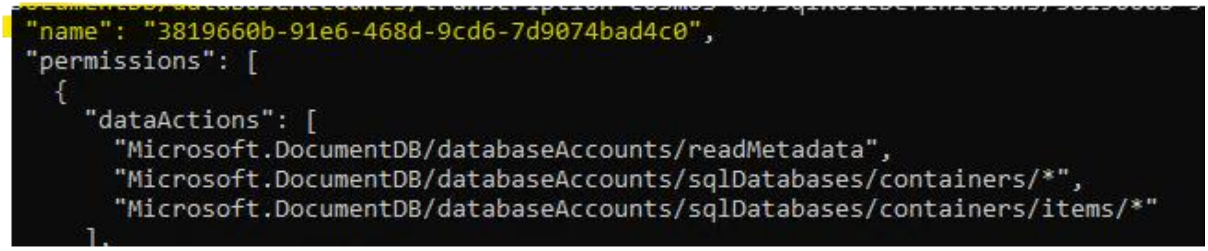
```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\>cd C:\

C:\>cd "Helper files"

C:\Helper files>az cosmosdb sql role definition create --account-name <COSMOSAccount> --resource-group <ResourceGroup> -
-body @data_contributor_role_definition.json
```

3. Take note of the **name** value from the output.



```
"name": "3819660b-91e6-468d-9cd6-7d9074bad4c0",
"permissions": [
  {
    "dataActions": [
      "Microsoft.DocumentDB/databaseAccounts/readMetadata",
      "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/*",
      "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/items/*"
    ],
```

4. Run the following command to assign the **Data Contributor** role to the function app.

az cosmosdb sql role assignment create --account-name **<cosmosaccountname>** --resource-group **<resourcegroupname>** --role-definition-id /subscriptions/**<subscriptionID>**/resourceGroups/**<resourcegroupname>**/providers/Microsoft.DocumentDB/databaseAccounts/**<cosmosaccountname>**/sqlRoleDefinitions/**<INSERT NAME VALUE>** --principal-id **<function app principle id>** --scope /subscriptions/**<subscriptionID>**/resourceGroups/**<resourcegroupname>**/providers/Microsoft.DocumentDB/databaseAccounts/**<cosmosaccountname>**

Assign the Cosmos Data Reader & Contributor Roles to the WebApp

1. Run the following Azure CLI command to add Cosmos DB Data Contributor role.

   az cosmosdb sql role assignment create --resource-group **<resourcegroupname>** *--account-name* **<cosmosaccountname>** *--role-definition-id "00000000-0000-0000-0000-000000000002" --principal-id* ***<web app Principle ID>*** *--scope "/subscriptions/****<subscriptionID>****/resourceGroups/*****<resourcegroupname>*****/providers/Microsoft.DocumentDB/databaseAccounts/*****<cosmosaccountname>*****"*

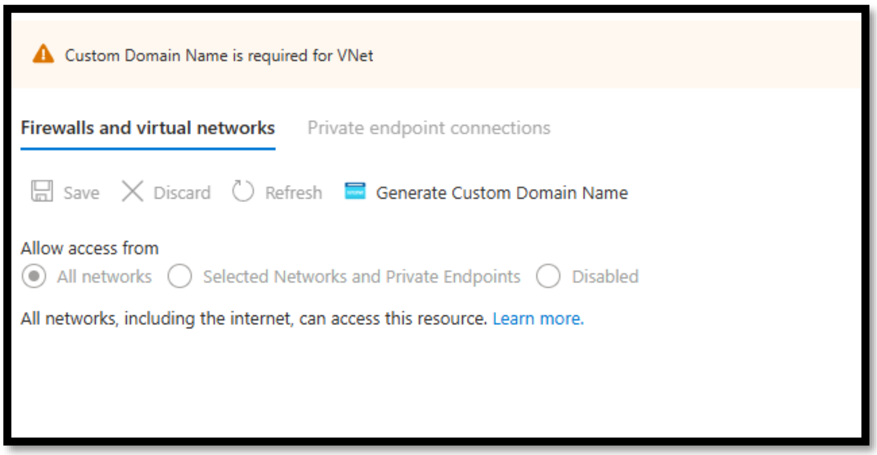
2. Run the following Azure CLI command to add Cosmos DB Data Reader role.

   az cosmosdb sql role assignment create --resource-group **<resourcegroupname>** *--account-name* **<cosmosaccountname>** *--role-definition-id "00000000-0000-0000-0000-000000000001" --principal-id* ***<web app Principle ID>*** *--scope*

*"/subscriptions/<**subscriptionID**>/resourceGroups/<**resourcegroupname**>/providers/Microsoft.DocumentDB/databaseAccounts/<**cosmosaccountname**>"*

13. Configuring the Speech Service

On the Speech Service resource, go to Networking and create a custom domain. Enter a name for the domain name and click save.



Custom Domain Name is required for VNet

Firewalls and virtual networks
Private endpoint connections
Save
Discard
Refresh
Generate Custom Domain Name

Allow access from

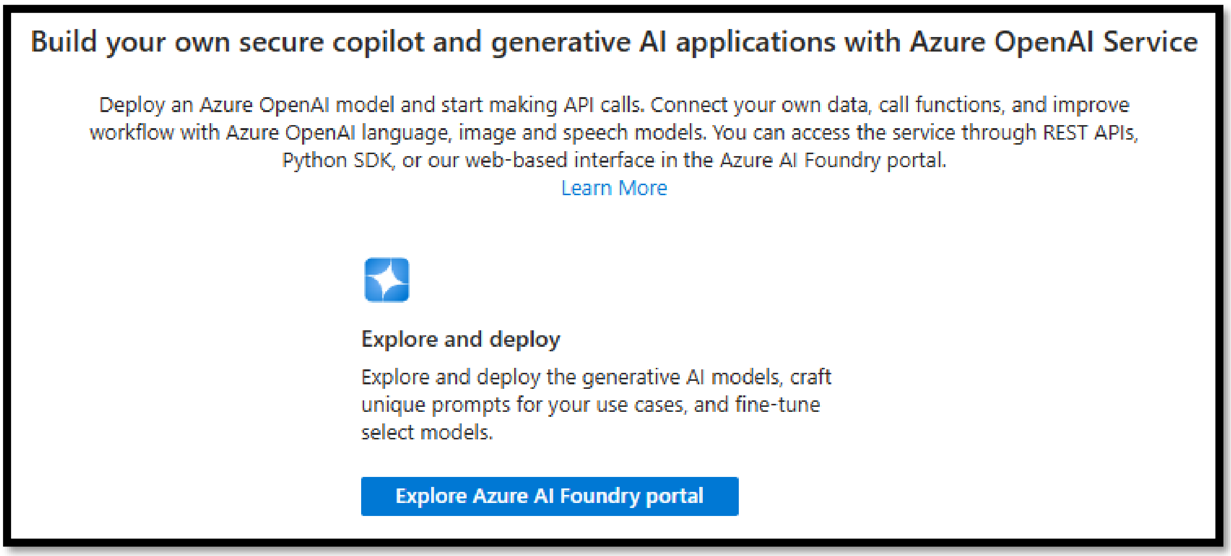
● All networks ○ Selected Networks and Private Endpoints ○ Disabled

All networks, including the internet, can access this resource. Learn more.

14. Configuring the Azure OpenAI Service

This section covers the deployment of the Azure OpenAI GPT model that will be used.

1. Go to the Azure OpenAI resource and click on 'Explore Azure AI Foundry portal'.



Build your own secure copilot and generative AI applications with Azure OpenAI Service

Deploy an Azure OpenAI model and start making API calls. Connect your own data, call functions, and improve workflow with Azure OpenAI language, image and speech models. You can access the service through REST APIs, Python SDK, or our web-based interface in the Azure AI Foundry portal.
Learn More

Explore and deploy

Explore and deploy the generative AI models, craft unique prompts for your use cases, and fine-tune select models.

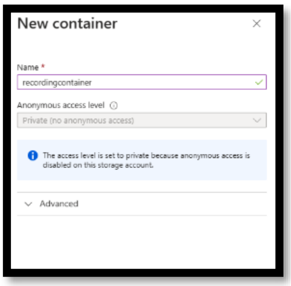
2. Under Deployments, click Deploy a model and then Deploy a Base Model. Select your chosen GPT model and click confirm. Select your preferred deployment options and click create resource and deploy.

3. Once created take note of the api version, which can be found at the end of the Target URL. This can be found under the endpoint heading. “https://<Servicename>.openai.azure.com/openai/deployments/<deployment>/chat/completions?api-version=2024-10-21”

NB: When testing the content filters on the GPT deployment may affect your results. You may need to request access to change the inbuilt content filters from Microsoft directly, which can be configured in the Azure AI Foundry portal.

15. Configuring the Storage Account

1. Navigate to the storage account resource, under ‘containers’ add a new container for example ‘recordingcontainer’.

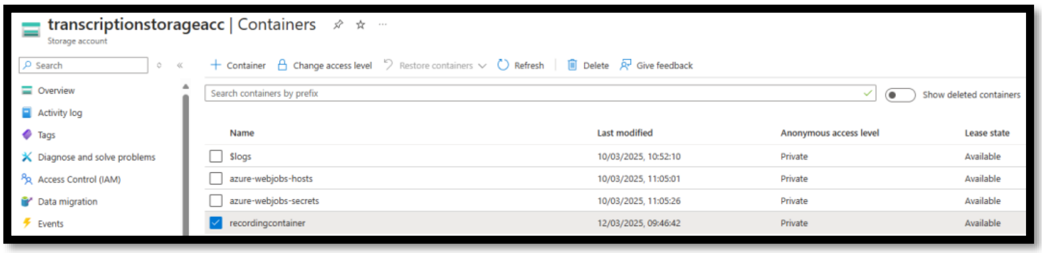


New container

Name *: recordingcontainer

Anonymous access level: Private (no anonymous access)

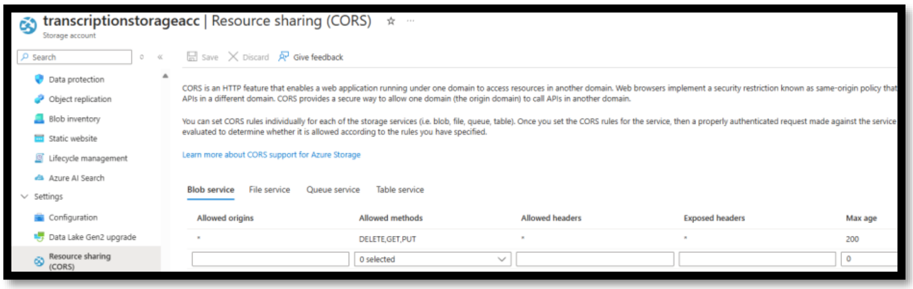
The access level is set to private because anonymous access is disabled on this storage account.



transcriptionstorageacc | Containers
Storage account

	Name	Last modified	Anonymous access level	Lease state
□	$logs	10/03/2025, 10:52:10	Private	Available
□	azure-webjobs-hosts	10/03/2025, 11:05:01	Private	Available
□	azure-webjobs-secrets	10/03/2025, 11:05:26	Private	Available
☑	recordingcontainer	12/03/2025, 09:46:42	Private	Available

2. On the storage account, under ‘Settings’ select ‘Resource sharing (CORS)’, and set the following for the Blob service:
   - Allowed Origins set to *
   - Allowed methods set to DELETE, GET and PUT
   - Allowed headers set to *
   - Exposed headers set to *
   - Max age set to 200



transcriptionstorageacc | Resource sharing (CORS)
Storage account

CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain. Web browsers implement a security restriction known as same-origin policy that APIs in a different domain. CORS provides a secure way to allow one domain (the origin domain) to call APIs in another domain.

You can set CORS rules individually for each of the storage services (i.e. blob, file, queue, table). Once you set the CORS rules for the service, then a properly authenticated request made against the service evaluated to determine whether it is allowed according to the rules you have specified.

Learn more about CORS support for Azure Storage

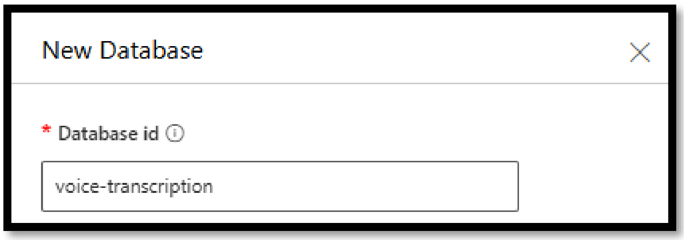
Allowed origins	Allowed methods	Allowed headers	Exposed headers	Max age
*	DELETE,GET,PUT	*	*	200
	0 selected			0

3. Save CORS settings

16. Configuring the Azure Cosmos DB Account

Creating a Database and Containers

1. Create a new Database within your Cosmos account e.g. “voice-transcription”. Navigate to “Data Explorer” click the arrow next to “New container” and then selected “New Database”.

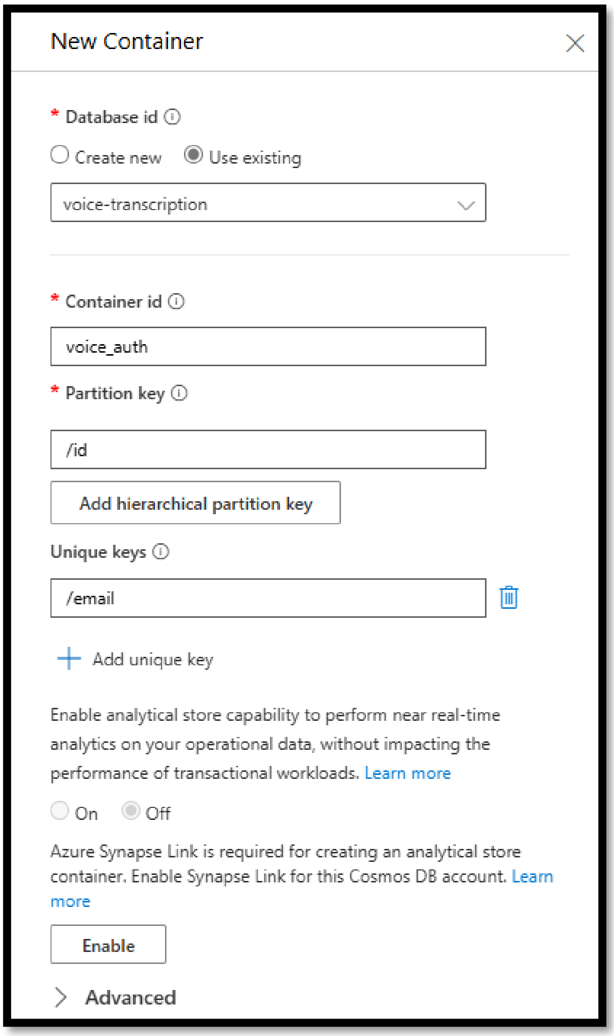


New Database

* Database id

voice-transcription

2. Create a voice_auth container in the new database.
   - Database id – use existing and select the database created
   - Container id – voice_auth
   - Partition key - /id
   - Unique keys - /email



New Container

* Database id ⓘ

○ Create new ● Use existing

voice-transcription

* Container id ⓘ

voice_auth

* Partition key ⓘ

/id

Unique keys ⓘ

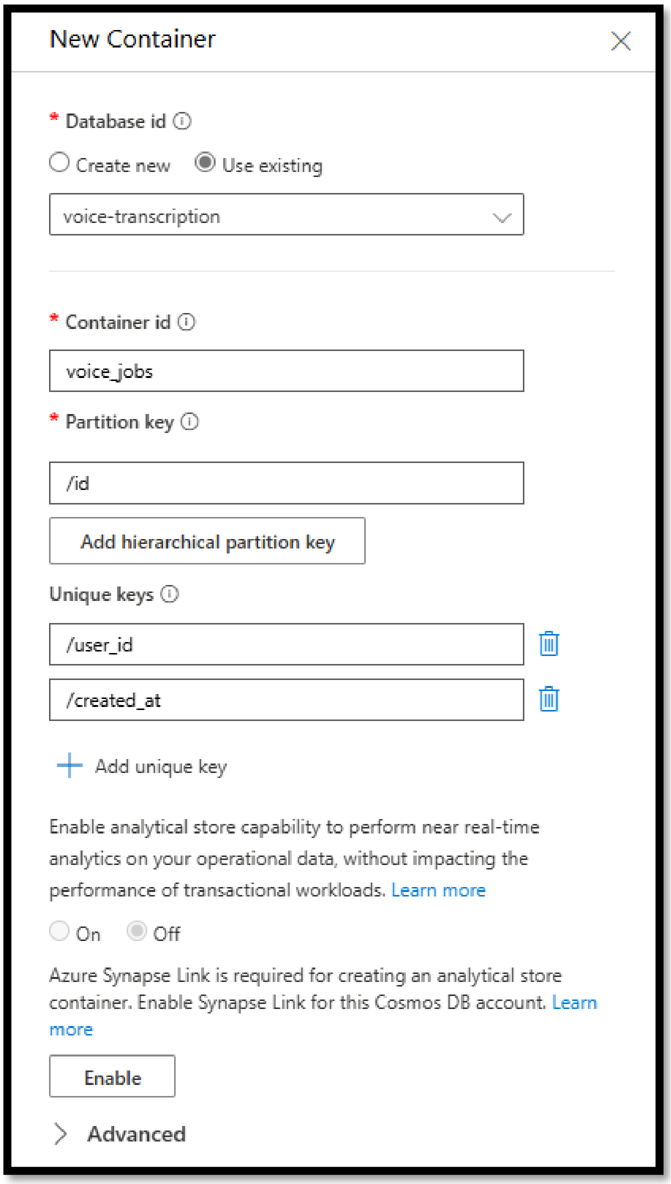
/email

Enable analytical store capability to perform near real-time analytics on your operational data, without impacting the performance of transactional workloads. Learn more

○ On ● Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. Learn more

3. Create a voice_jobs container in the new database.
    - Database id – use existing and select the database created
    - Container id – voice_jobs
    - Partition key - /id
    - Unique keys - /user_id and /created_at



New Container

* Database id ⓘ

○ Create new ● Use existing

voice-transcription

* Container id ⓘ

voice_jobs

* Partition key ⓘ

/id

Add hierarchical partition key

Unique keys ⓘ

/user_id

/created_at

+ Add unique key

Enable analytical store capability to perform near real-time analytics on your operational data, without impacting the performance of transactional workloads. Learn more

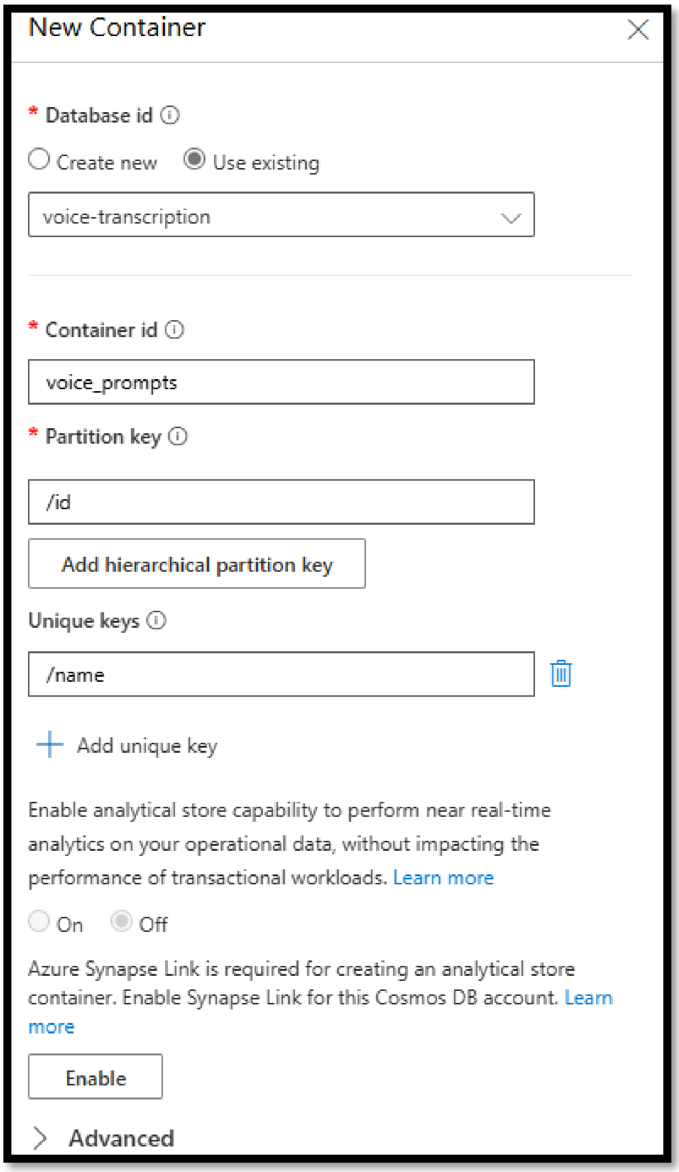
○ On ● Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. Learn more

Enable

> Advanced

4. Create a voice_prompts container in the new database.
    - Database id – use existing and select the database created
    - Container id – voice_prompts
    - Partition key - /id
    - Unique keys - /name



New Container

* Database id ⓘ

○ Create new ● Use existing

voice-transcription

* Container id ⓘ

voice_prompts

* Partition key ⓘ

/id

Add hierarchical partition key

Unique keys ⓘ

/name

+ Add unique key

Enable analytical store capability to perform near real-time analytics on your operational data, without impacting the performance of transactional workloads. Learn more

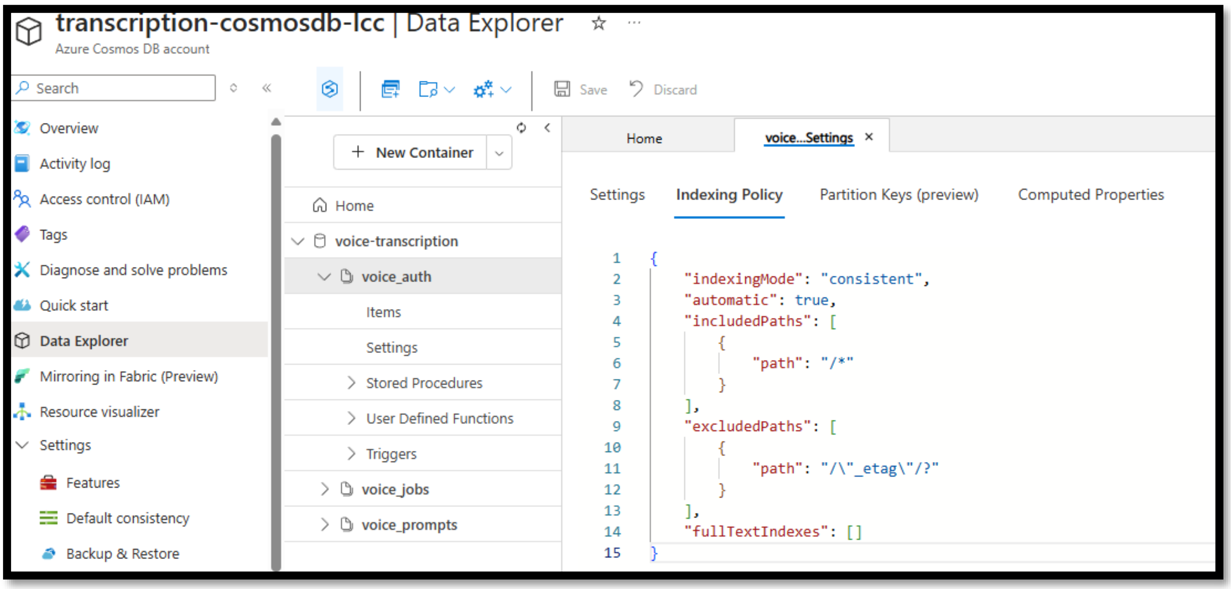
○ On ● Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. Learn more

Enable

> Advanced

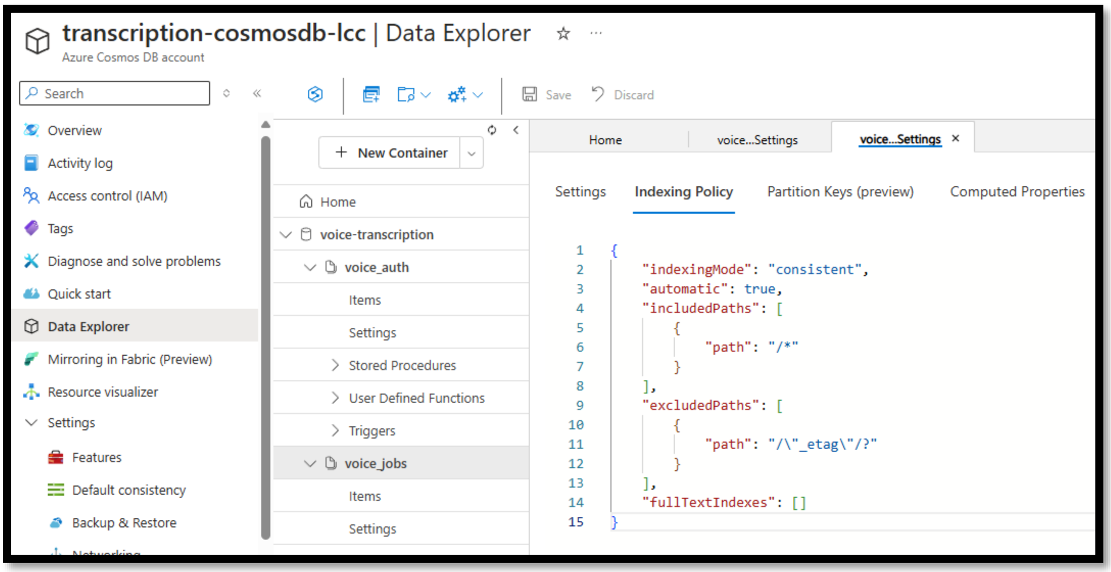
5. Create the indexing policy for the voice_auth container. On this container, under settings, click ‘Indexing Policy’ and paste in the following code and click save.



```
{
    "indexingMode": "consistent",
    "automatic": true,
    "includedPaths": [
        {
            "path": "/*"
        }
    ],
    "excludedPaths": [
        {
            "path": "/\"_etag\"/?"
        }
    ],
    "fullTextIndexes": []
}
```

```
{
    "indexingMode": "consistent",
    "automatic": true,
    "includedPaths": [
        {
            "path": "/email/?"
        },
        {
            "path": "/*"
        }
    ],
    "excludedPaths": [
        {
            "path": "/_etag/?"
        }
    ],
    "fullTextIndexes": []
}
```

6. Create the indexing policy for the voice_jobs container. On this container, under settings, click ‘Indexing Policy’ and paste in the following code and click save.

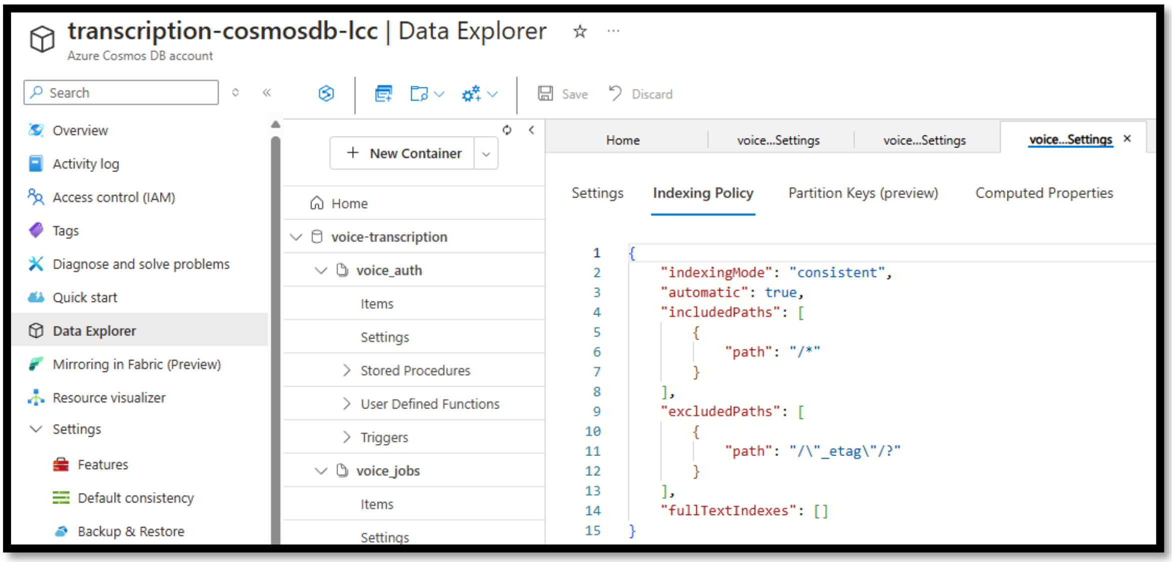


```
{
    "indexingMode": "consistent",
    "automatic": true,
    "includedPaths": [
        {
            "path": "/*"
        }
    ],
    "excludedPaths": [
        {
            "path": "/\"_etag\"/?"
        }
    ],
    "fullTextIndexes": []
}
```

```
{
  "indexingMode": "consistent",
  "automatic": true,
  "includedPaths": [
    {
      "path": "/user_id/?"
    },
    {
      "path": "/prompt_category_id/?"
    },
    {
      "path": "/prompt_subcategory_id/?"
    },
    {
      "path": "/status/?"
    },
    {
      "path": "/created_at/?"
    },
    {
      "path": "/*"
    }
  ],
  "excludedPaths": [
    {
      "path": "/_etag/?"
    }
  ],
  "fullTextIndexes": []
}
```

Creating Indexing Polices

7. Create the indexing policy for the voice_prompts container. On this container, under settings, click ‘Indexing Policy’ and paste in the following code and click save.

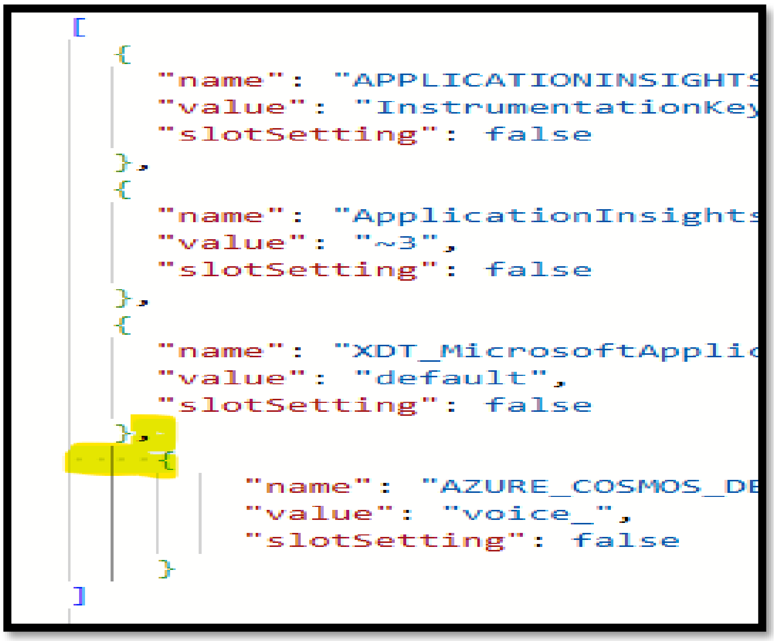


```
{
    "indexingMode": "consistent",
    "automatic": true,
    "includedPaths": [
        {
            "path": "/*"
        }
    ],
    "excludedPaths": [
        {
            "path": "/\"_etag\"/?"
        }
    ],
    "fullTextIndexes": []
}
```

```
{
    "indexingMode": "consistent",
    "automatic": true,
    "includedPaths": [
        {
            "path": "/type/?"
        },
        {
            "path": "/category_id/?"
        },
        {
            "path": "/name/?"
        },
        {
            "path": "/*"
        }
    ],
    "excludedPaths": [
        {
            "path": "/_etag/?"
        }
    ],
    "fullTextIndexes": []
}
```

17. Configuring the WebApp

1. Open the file web-app-environmentvariables.txt and fill in the “Value” for each of the variables.
2. Navigate to environment variables and click “advanced edit”. Paste the variables from the file ‘web-app-environment-variables.txt’ at the end of the current variables. Ensure that you add a “,” after the last “}” as highlighted in the screenshot below before pasting the additional variables from the file.



```
[
    {
        "name": "APPLICATIONINSIGHTS
        "value": "InstrumentationKey
        "slotSetting": false
    },
    {
        "name": "ApplicationInsights
        "value": "~3",
        "slotSetting": false
    },
    {
        "name": "XDT_MicrosoftApplic
        "value": "default",
        "slotSetting": false
    },
        {
            "name": "AZURE_COSMOS_DE
            "value": "voice_",
            "slotSetting": false
        }
]
```

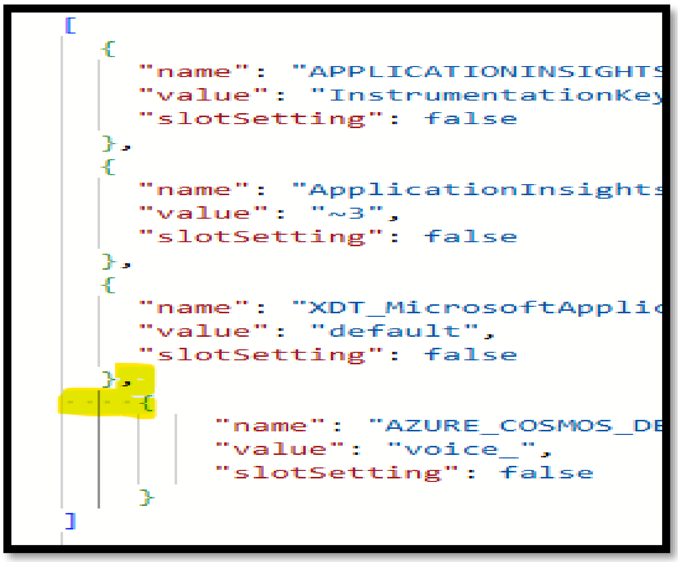
3. Press ok and then apply.
4. Navigate to Configuration and add the following startup command, click save and then continue.

python3 -m gunicorn -w 2 -k uvicorn.workers.UvicornWorker app.main:app --bind 0.0.0.0:8000 --reload --log-level debug

5. Now Navigate to CORS, in Allowed Origins, add * then click save.

18. Configuring the Function App

1. Open the file function-app-environmentvariables.txt and fill in the “Value” for each of the variables.
2. Navigate to environment variables and click “advanced edit”. Paste the variables from the file ‘function-app-environment-variables.txt’ at the end of the current variables. Ensure that you add a “,” after the last “}” as highlighted in the screenshot below before pasting the additional variables from the file.



```
[
    {
        "name": "APPLICATIONINSIGHTS
        "value": "InstrumentationKey
        "slotSetting": false
    },
    {
        "name": "ApplicationInsights
        "value": "~3",
        "slotSetting": false
    },
    {
        "name": "XDT_MicrosoftApplic
        "value": "default",
        "slotSetting": false
    },
    {
            "name": "AZURE_COSMOS_DB
            "value": "voice_",
            "slotSetting": false
        }
]
```

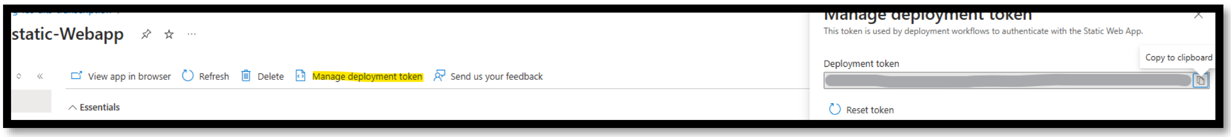
Note, the Azure_Speech_Deployment is the name of the custom endpoint chosen for the speech service defined earlier. You can find this on the speech service under Endpoint, found after https:// and before .cognitiveservices.

3. After adding the environment variables go to CORS in Allowed Origins, add * then click save.

19. Configuring & Deploying the Static Web App

1. Open CMD and change directory to \echo-brief\infra\frontend_app

Access the Azure portal and navigate to your Static Web App service and copy the deployment token.



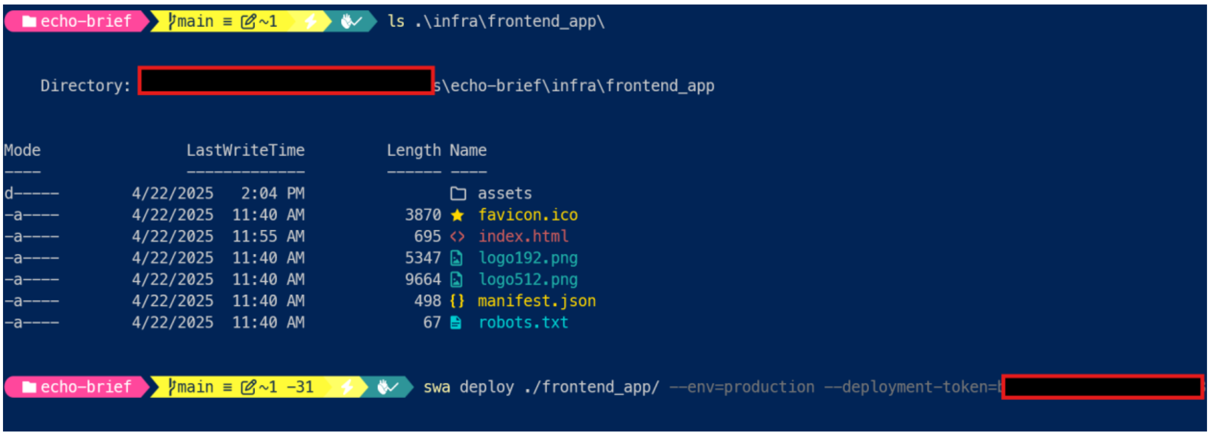
Manage deployment token

This token is used by deployment workflows to authenticate with the Static Web App.

Deployment token [redacted]

In the same command prompt run this command.

swa deploy ./out --env=production --deployment-token=**<deployment token>**



```
 echo-brief  main ≡ ~1    ls .\infra\frontend_app\

    Directory: [redacted]s\echo-brief\infra\frontend_app

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/22/2025   2:04 PM                assets
-a----         4/22/2025  11:40 AM           3870 favicon.ico
-a----         4/22/2025  11:55 AM            695 index.html
-a----         4/22/2025  11:40 AM           5347 logo192.png
-a----         4/22/2025  11:40 AM           9664 logo512.png
-a----         4/22/2025  11:40 AM            498 manifest.json
-a----         4/22/2025  11:40 AM             67 robots.txt

 echo-brief  main ≡ ~1 -31    swa deploy ./frontend_app/ --env=production --deployment-token=[redacted]
```

if you get "Failure Reason: Could not load StaticSitesClient metadata from remote. Please check your internet connection." Your firewall may be blocking access to the Azure static Web app CLI.

20. Deploying the Function App

To deploy the Function App:

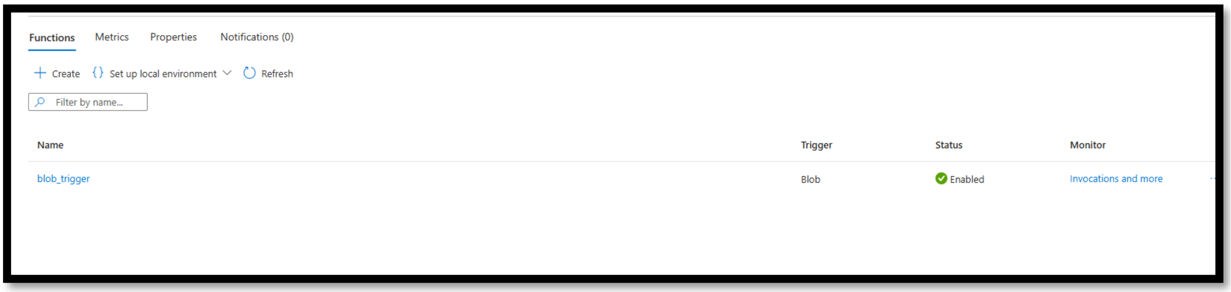
1. Open CMD, Change directory to \echo-brief\az-func-audio and run this command to zip the function app.

   ```
   tar -a -c -f ../output.zip *
   ```

2. then change directory to \echo-brief and then run this Azure CLI deployment command to deploy the Function App.

   ```
   az functionapp deployment source config-zip --subscription <subscriptionID> --resource-group <resourcegroupname> --name <functionappname> --src output.zip --build-remote true
   ```

3. Navigate to the Function app in the Azure Portal, on the overview page you should see the function "blob_trigger"



Name	Trigger	Status	Monitor
blob_trigger	Blob	Enabled	Invocations and more

21. Deploying the Backend App

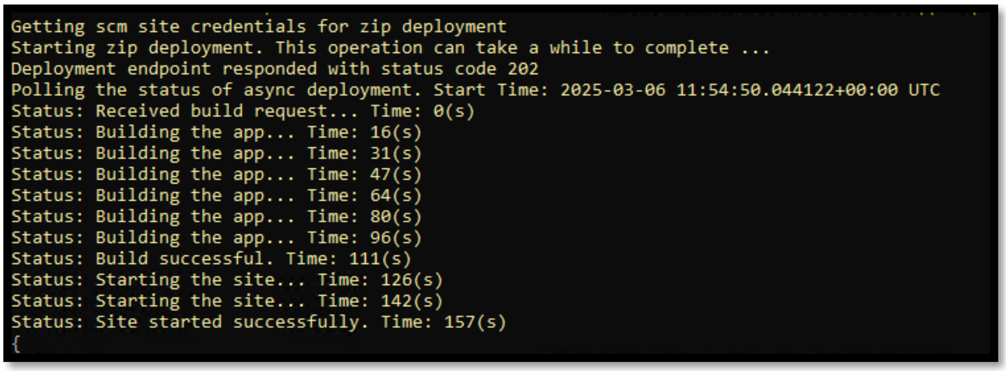
1. Open Command prompt and change directory to \echo-brief\backend_app and run this command to create a zip file.

   ```
   tar -a -c -f ../backend.zip app requirements.txt
   ```

2. change directory to \echo-brief\ and then run the following command to deploy the backend code to the web app.

   ```
   az webapp deployment source config-zip --resource-group <resource-group-name> --name <web-app-name> --src backend.zip
   ```

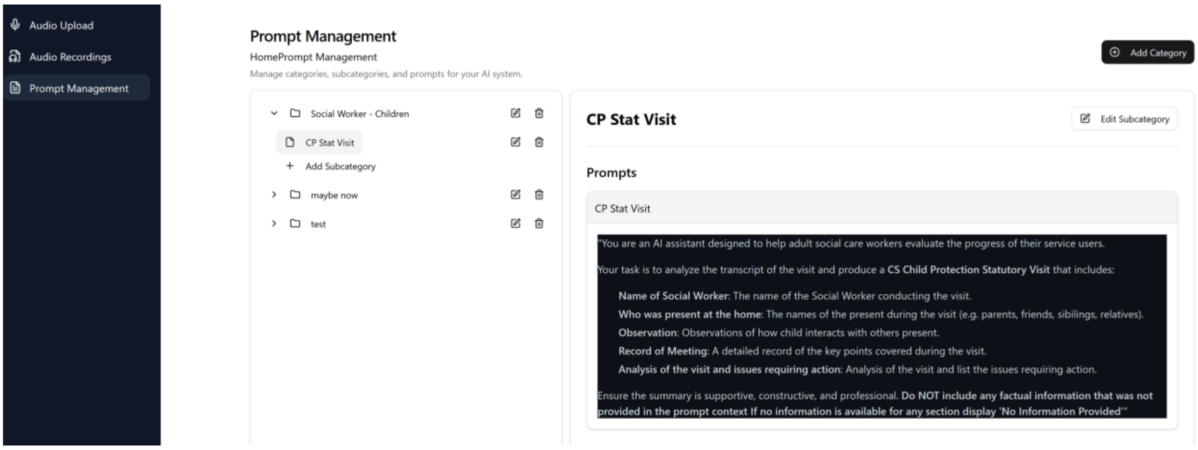
Once deployed you should get a 'Site Started Successfully' message.



```
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
Polling the status of async deployment. Start Time: 2025-03-06 11:54:50.044122+00:00 UTC
Status: Received build request... Time: 0(s)
Status: Building the app... Time: 16(s)
Status: Building the app... Time: 31(s)
Status: Building the app... Time: 47(s)
Status: Building the app... Time: 64(s)
Status: Building the app... Time: 80(s)
Status: Building the app... Time: 96(s)
Status: Build successful. Time: 111(s)
Status: Starting the site... Time: 126(s)
Status: Starting the site... Time: 142(s)
Status: Site started successfully. Time: 157(s)
{
```

22. Adding Prompts for Transcription Summary (Optional)

1. Using UI



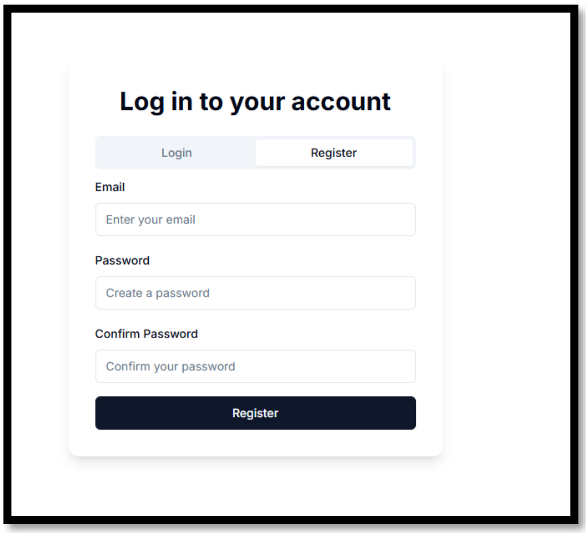
"You are an AI assistant designed to help adult social care workers evaluate the progress of their service users.

Your task is to analyze the transcript of the visit and produce a **CS Child Protection Statutory Visit** that includes:

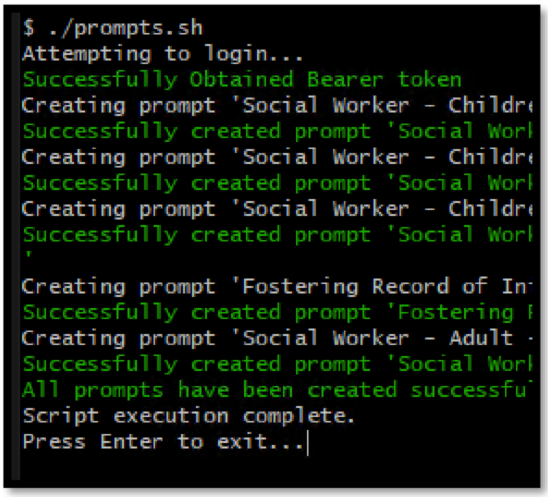
Name of Social Worker: The name of the Social Worker conducting the visit.
Who was present at the home: The names of the present during the visit (e.g. parents, friends, sibilings, relatives).
Observation: Observations of how child interacts with others present.
Record of Meeting: A detailed record of the key points covered during the visit.
Analysis of the visit and issues requiring action: Analysis of the visit and list the issues requiring action.

Ensure the summary is supportive, constructive, and professional. **Do NOT include any factual information that was not provided in the prompt context If no information is available for any section display 'No Information Provided'"**

2. (Using Cli) Navigate to the transcription services frontend and register an account. This can be found on the overview page on the Static WebApp.



3. Edit the prompts.sh file downloaded as part of the repo. Fill in the details (backend url, username and password.) Right click an empty space in the folder where prompts.sh is and click “open git bash here”. Then run “./prompts.sh”.



```
$ ./prompts.sh
Attempting to login...
Successfully Obtained Bearer token
Creating prompt 'Social Worker - Childr
Successfully created prompt 'Social Wor
Creating prompt 'Social Worker - Childr
Successfully created prompt 'Social Wor
Creating prompt 'Social Worker - Childr
Successfully created prompt 'Social Wor
'

Creating prompt 'Fostering Record of In
Successfully created prompt 'Fostering
Creating prompt 'Social Worker - Adult -
Successfully created prompt 'Social Wor
All prompts have been created successfu
Script execution complete.
Press Enter to exit...
```