**Play Book: Cardinals Name Pronunciation Tool.**

**Introduction:** Cardinals team worked on the **“Name Pronunciation Tool”** hackathon 2022 challenge.

The intent of the application is to either

**Synopsis:**

1. Translate the name (entered in the Text Box) using the Microsoft Cognitive Services voice SDK **(Standard Translation)** and play it back in the audio format from the UI.
2. Save an employee data into the Yuga byte database and an audio Blob into the Azure blob storage.
3. Fetch the employee’s audio (when an Email or UID) is typed in the text box, go to the database if employee record matches, then go to the Azure blob service to fetch the related audio and send it back to the UI to play the employee audio recording.

**General Application Flow and Considerations:**

Please note that we have a working code committed to the GitHub repository.

**GitHub:** [**https://github.com/grandorottcod/pronunciationNameUI**](https://github.com/grandorottcod/pronunciationNameUI)

**Spring Boot API:** [**https://github.com/cardinalsteam/azure-demo-app**](https://github.com/cardinalsteam/azure-demo-app)

**Demo Video:** [**https://cardinalsblobstorage.blob.core.windows.net/cardinalsblobstoragecontainer/Name%20Pronunciation%20Tool/final\_presentation.mp4**](https://cardinalsblobstorage.blob.core.windows.net/cardinalsblobstoragecontainer/Name%20Pronunciation%20Tool/final_presentation.mp4)

Please note that application is working when we run it using our local machine. There is a known issue with the Microsoft client SDK in the cloud deployment. So, we were unable to run the application in the cloud as the client SDK was throwing exceptions. **This issue was escalated to the Hackathon Azure Support team, and it was also escalated to the Microsoft.**

It was told to us to record the demo from our local machine and present it along with other supporting docs. Running this application is NOT a requirement a (though it will be desirable).

I am attaching all those email conversations, to make it clear that the issue was on the Microsoft Azure side and not from our application.

**Email attachment** stating the Issue and a reply from one of Hackathon’s Azure Support resource.



**Tech Stack:**

Java 11

Spring Boot 2.6.7

Microsoft client SDK 1.14.0

Azure Blob Storage

Yuga Byte database.

Angular User Interface.

Linux / Windows for local machine will work fine.

**How to run the Application on the local Machine:**

**Pre-Requisites:**

Java 11 must be installed and present on the class path of your local machine.

Open the application in your favorite IDE, (we are using IntelliJ), build it (gradle build) and run it (bootRun).

Once the application is running.

**Scenario #1: Text to Standard Speech conversion.**

Write the employee’s name in the text box and click on Search button. This will call the Spring Boot API and the Spring Boot will call the Azure Text to Speech service to get the standard pronunciation in audio format and will send it back to the User Interface (UI).

Diagram, schematic

Description automatically generated

**Scenario #2: Saving an Employee Record with the Audio.**

Let us say the employee wants to override the standard pronunciation and wants to record his/her own audio. In this case employees must save record in the Yuga Byte database. Please note that the employee information is saved in the database along with the reference to the audio blob. The actual audio blob will be saved in the Azure Blob storage service.

Once the audio is recorded an employee can play his/her audio just to test and then if satisfied, can click on the submit button to save the data. Clicking the Submit button will call the Spring Boot application and (uid, email, name, audio) will be stored in the Yuga byte database and the actual audio blob will be stored in the Azure blob storage service which can be retrieved later using UID or Email address.

Diagram

Description automatically generated

**Scenario #3: Searching an Employee with Audio.**

Let us say that in the scenario #2, we have saved the employee information with his/her voice. Now it is time to search it. We can search an employee using either UID or the Email address as both are PRIMARY KEYS in the Yuga byte database.

Write employee UID or Email address in the first form and click search. The call Will go to the Yuga byte database via Spring Boot application. It will fetch the employee record and will also fetch the audio blob previously stored in the Azure blob storage. Spring Boot app will send back the audio bytes to the calling application to play it. In our case the calling application is our Angular User Interface app.

**Yuga Byte DB schema:**

Table

Description automatically generated

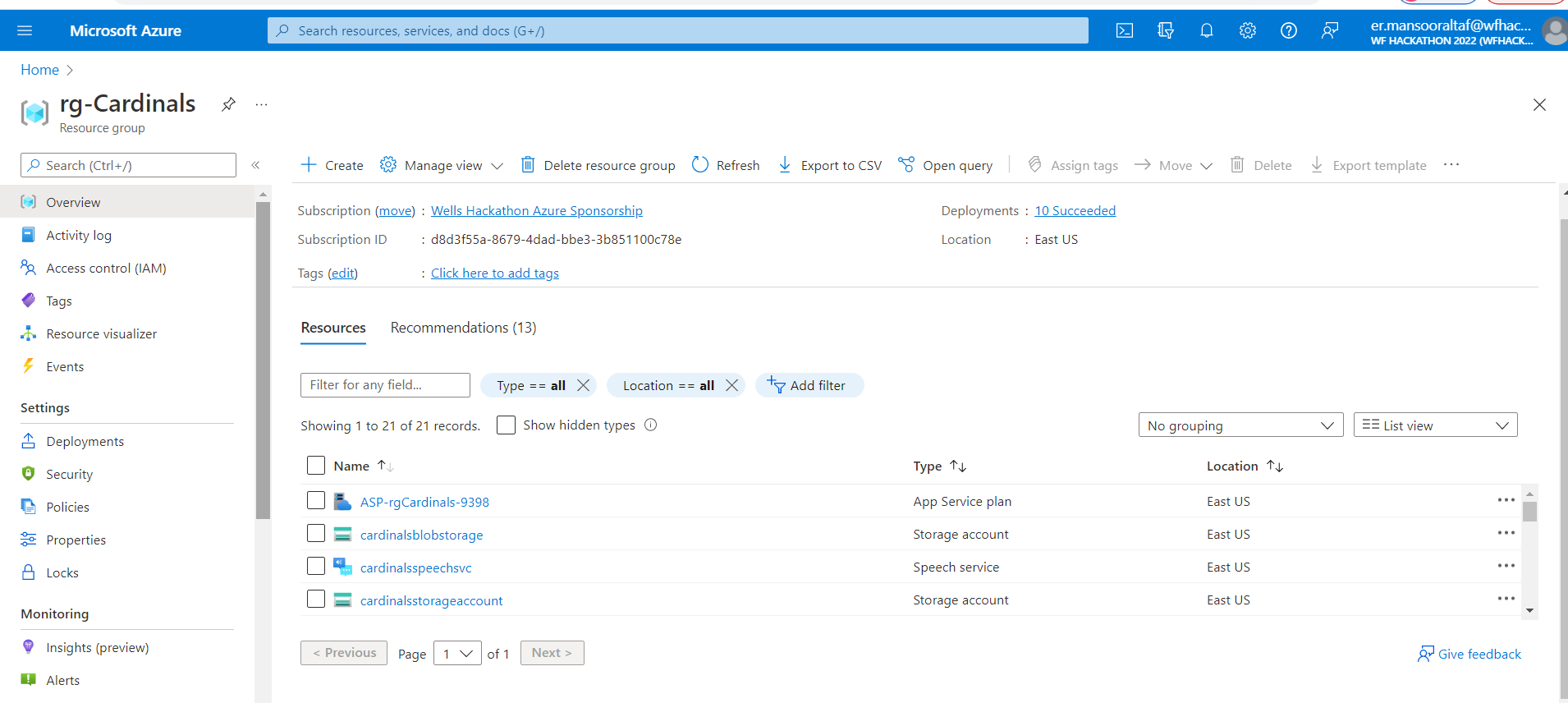
**Yuga Byte DB Table snapshot:**

Graphical user interface, text, application, email

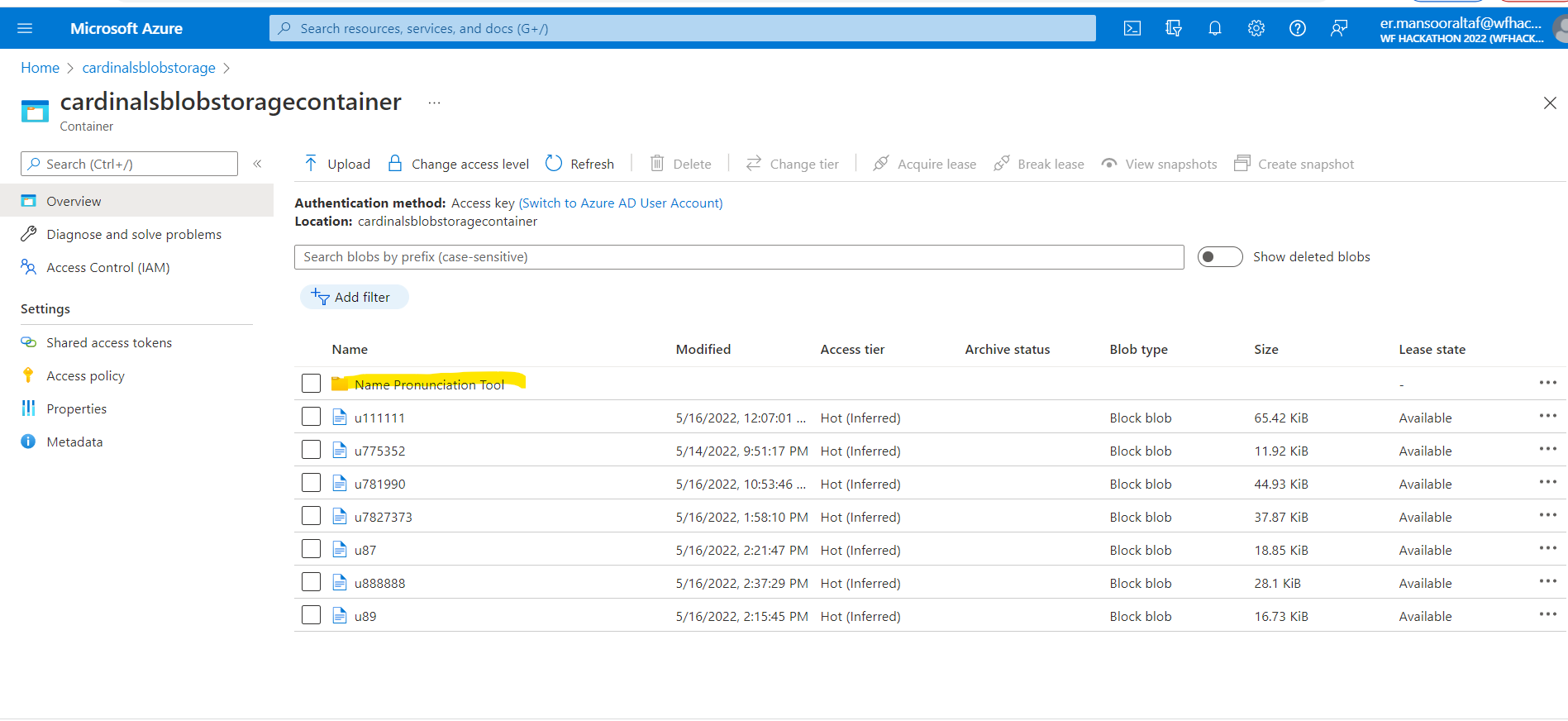
Description automatically generated

**Azure Snapshots:**

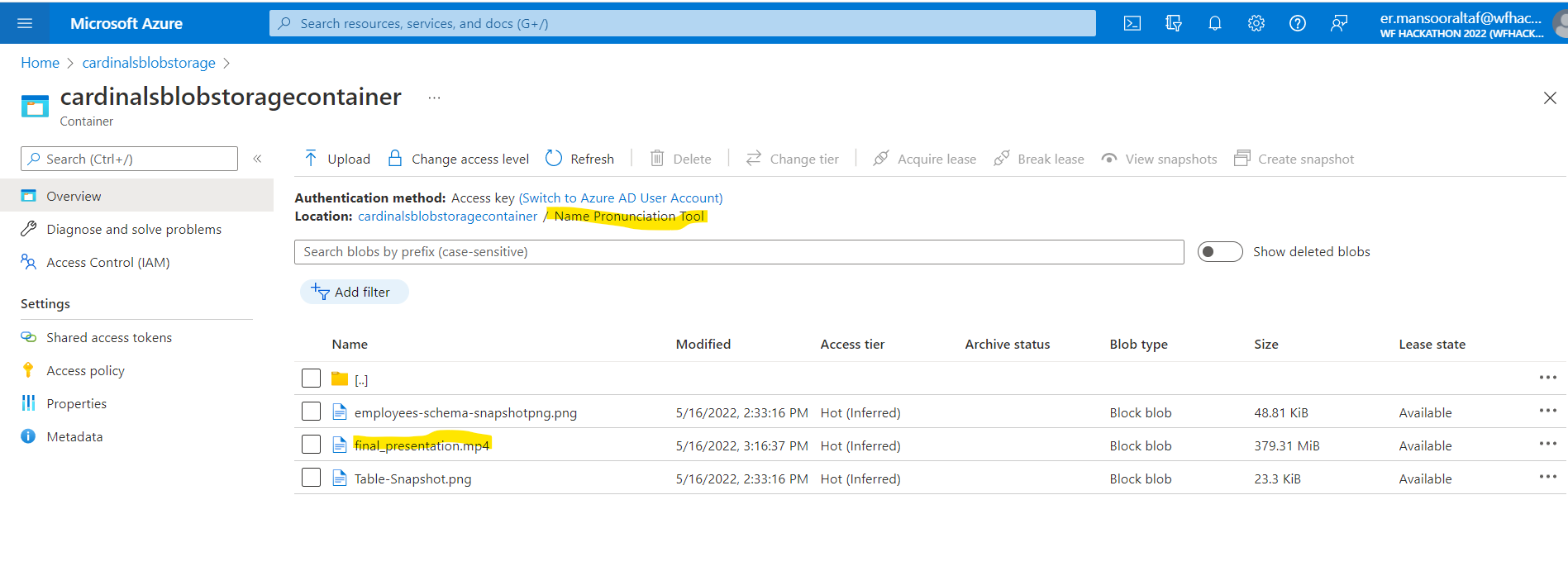
**Cardinals Resource Group:**

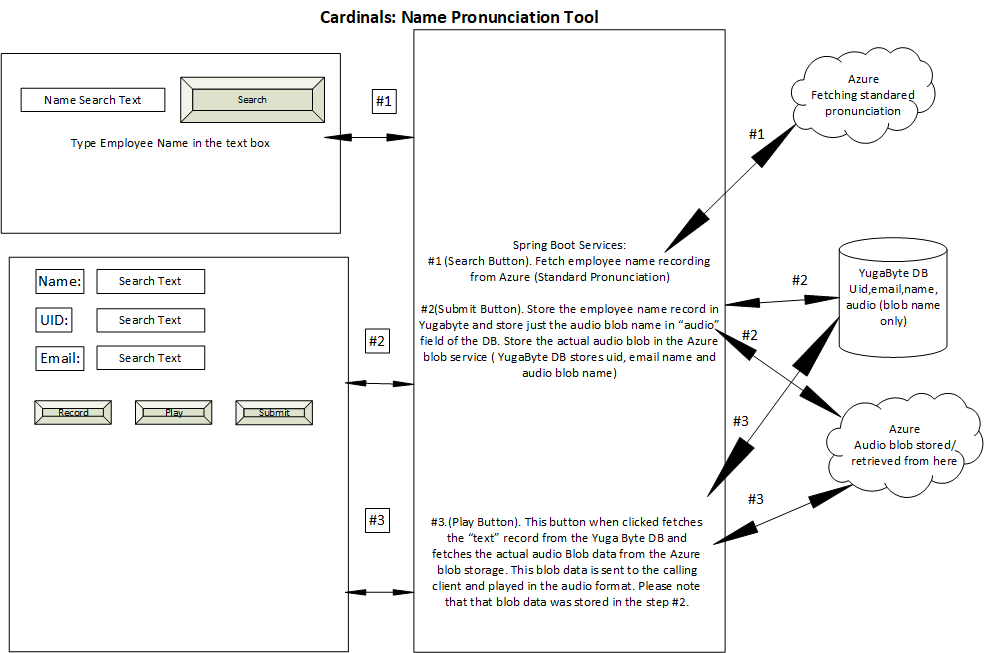
****

**Cardinals Blob Storage Container:**

****

**Name Pronunciation Tool Demo Location Snapshot:**

****



Thanks!

Cardinals Team.