AZ204 Quizmaster Deployment Challenge

This challenge is designed to give hands-on experience with most of the topics covered in the AZ204 exam. The areas covered are:

* App Services
* Blob Storage
* Azure Functions
* Identity Management
* Azure Key Vault
* MySQL Server

It does not include Services bus or Redis cache.

Let’s start with local deployment:

Prerequisites:

* IDE of your choice, but Visual Studio and/or VSCode are recommended
* MySQL Server (MySQL workbench is especially helpful as well)

There are two repositories that will need to be forked and cloned into your own repository and IDE.  
Frontend: <https://github.com/Kevinclane/az204quizmasterClient>

Backend: <https://github.com/Kevinclane/az204quizmasterAPI>

Open the frontend application in your IDE and then run “npm i” in the terminal (Try GitBash if PowerShell doesn’t work).

Run “ng serve” and ensure the homepage is loading.

Ensure you have an instance of MySQL server running.

Open the backend application in your IDE.

Create a file named localSecret.txt in the az204quizmasterAPI directory. It should be at the same level as the Controllers, Services, ect.

Add a connection string with your database information:   
Server="localhost";Port=3306;Database="{database}";UserID="root";Password="{password}";

***Ensure this file to your .gitignore - It should already be there, but double check***

Restore NuGet packages to install dependencies.

Run the application to ensure the database is connected properly.

You can also use the /ping endpoint to test that it’s running fine. You should get “Pong” as a response.

Shut down the application and then run “dotnet ef database update”

This should create all tables needed for your MySQL server. You can use a database inspecting tool such as MySQL Workbench to verify.

Now you should be ready to ingest some data:

Start the front and backend applications and open a browser to the frontend’s url (localhost:4200).

Click the upload icon in the top right corner of the homepage. Select a json file to upload, there is one provided in the az204quizmasterAPI/JsonIntakes folder.   
Confirm the request was successful by using the browser inspection tool and/or querying the database.

Now everything should be ready to go. Create a quiz to get started.

You can create your own JSON files with questions if desired. There is a short guide in the az204quizmasterAPI readme that explains the formatting.

Live Deployment:

Go to portal.azure.com and sign in. After all the resources are created and running, it should only charge the account around $12 a month. The most expensive part is the database at about $8. You can reduce the charges by shutting services down after you’re finished and it will cost almost nothing. Additionally, if you are using a new account, Azure provides some “free trial credit” and it could cost you nothing.   
  
Just as a reminder, this is a challenge, not a tutorial. A general path will be given to follow, but it is up to you to fully create and connect all of these services. If you get stuck there are blacked out sections in this document that may help. Simply highlight the blacked out section and change the highlight to white to reveal the answer. Let’s get into it!

You will need to create the following resources:

* Subscription
* Resource Group
* MySQL flexible server
  + Version 8.0
  + Burstable B1ms 1vCores, 2GiB RAM, 20GiB storage
  + MySQL authentication only
* Key Vault
* Storage Account
  + Primary Service: Azure Blob Storage
  + Access tier: Cool
* Function App
* App Service
  + Dotnet - v8.0
  + Windows
  + Enable Continuous Deployment via GitHub Actions
* Application Insights (optional but recommended for debugging)

Configure the blob storage account to host a static web page. Upload a compiled az204quizmasterClient index.html file. Don’t forget to update the apiUrl in the environment.ts file.

Create a new secret in Azure Key Vault named “ConnectionString” and use your database’s connection string and with the database named “quizmaster”. Your web app will need permissions to access this value.

Update the uri in the az204quizmasterAPI’s program.cs to hit your Azure Key Vault.

Run a command to use Entity Framework to create the needed tables into the database.

Add a function in Azure Functions to detect new files in the blob storage, stringify the contents, and send it to the /api/json endpoint.

Upload the same JSON file used in local deployment to the blob storage. This should populate the database with data.

Congratulations! Everything should be set up, connected, and ready to use. Don’t forget to delete all the created resources so you don’t get charged.