Bulk Exchange UserProfile App

Design By: Cuyler Quint, deanquint@gmail.com

High Level Overview

Design a proof of concept application enabling end users to persist basic profile information.

Requirements

1. Create a proof of concept for the following:
    - Create an application using the Django framework to take basic user profile information (First name, Surname, email address and phone number)
2. Users should be able to save their profile info
3. There should be validation to ensure that:
    - Email address is in the correct format
    - That duplicate email addresses and cell phone numbers cannot be entered
4. Create a GitHub project for this assignment
5. Automation of the provisioning of your app and data layers.
6. A readme file must be populated with your git project detailing your solution and how to run it. (edited)
7. Provide details of your high-level design that would meet the requirements outlined above.

Proposal 1

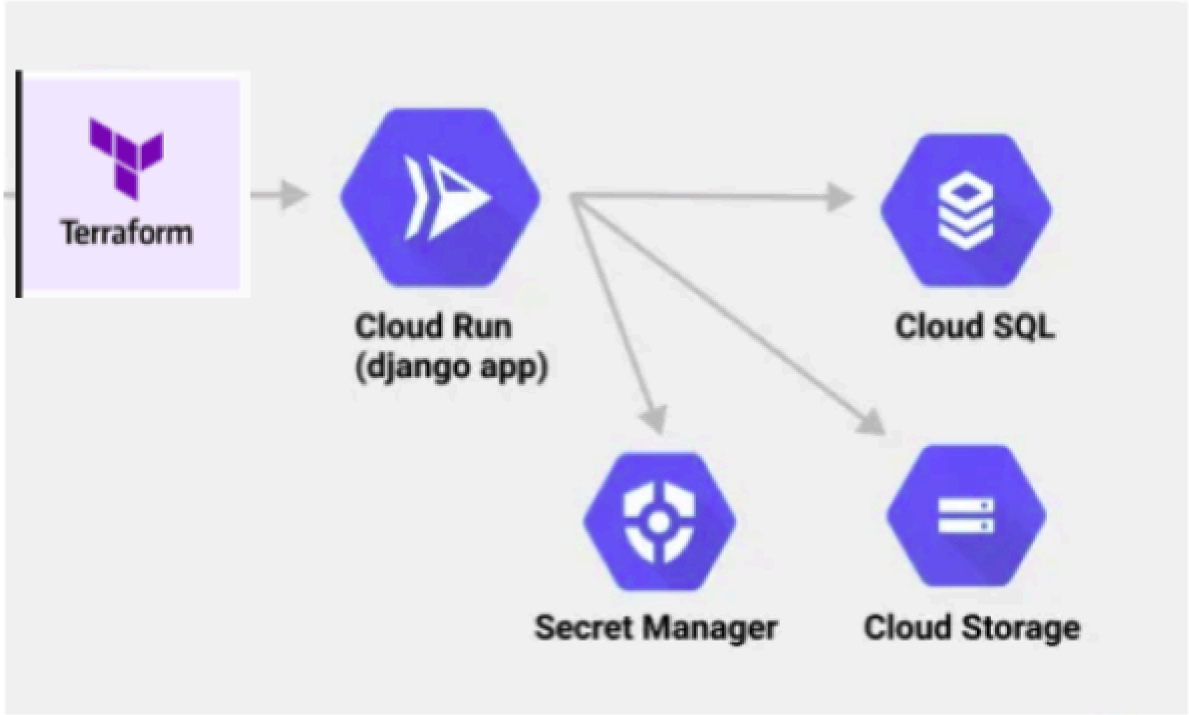
Tech Stack:

To deliver the requirements I would approach this profile application with a simple django/gunicorn server that utilizes the django rest framework for exposing and processing api requests.   To persist this data will use a postgres instance for user profile data.

For deploying this services we have multiple options of providers available:

- AWS
- Heroku
- Digital Ocean
- GCP

Personally I prefer GCP for its simplicity and low costs. To get setup in gcp I will build the infrastructure with terraform to do all of the provisioning handy work to get around setting up manually which allows for scalability as the application grows.   Below you can see a high level of application:



API/Model Design

I personally have done auth and profile views for numerous applications and I found the following setup is common practice when dealing with user profiles and email services. In order for these basic services we need the following Three Models:

UserProfile, AuthToken and an UpdateEmailToken.

UserProfile

- Stores the user information

AuthToken

- Stores the JWT Token for client authentication

UpdateEmailToken

- Stores a token to validate email change/updates

Endpoints:

Register

- Allows end user to create a profile and persist their information
- URL: **“/api/register/”**
- Method: POST
- Body:

```
{
        “email”: “john@gmail.com”,
        “first_name”: “john”,
        “last_name”: “doe”,
        “password”: “secret”,
        “phone_number”: “18005556666”
}
```

Login

- Allows the user to authenticate and returns JWT token for client to further make requests
- URL: **“/api/login/”**
- Method: POST
- Body:

```
{
        “email”: “john@gmail.com”,
        “password”: “secret”
}
```

Profile ModelViewSet

- Allows authenticated Users to fetch/update their account details
- URL: “api/profiles/<pk>”
- Method: GET, PATCH
- Body:

```
{
        “email”: “john@gmail.com”,
        “first_name”: “john”,
        “last_name”: “doe”
        “password”: “secret”,
        “phone_number”: “18005556666”
}
```

Next Steps/Improvements:

- Host a react Frontend to consume this API service
- Implement CI/CD within Github too kick off pipelines for pushes onto various branches
- Implement email service
- Implement More account APIs for updating email/password
- Add authentication to profile endpoints for updating Profiles