# Infrastructure Assessment

# *Check-in Application*

**Objective**

To demonstrate your ability in:

* *Utilising Infrastructure as Code for deploying to AWS*
* *Responding to a set of requirements*
* *Being pragmatic with deliverables given a tight deadline*
* *Understanding network architecture*

This assessment should take no longer than 4 hours to complete

**Background**

IdentityE2E has been identified by a client to migrate an existing on-premises service from a datacentre to the cloud. The client sees this migration as an avenue for reducing costs and increasing their service reliability as their existing service is suffering from outages due to hardware failures. They would also like to take this opportunity to increase their security posture due to several high-profile news items relating to information security. The client does not have experience working with cloud platforms and has approached you asking you implement a pilot solution.

The environment will be extended in the future as their services are migrated. The client has indicated that they would rather use AWS over a platform as a service (Heroku, AppEngine, etc) as this would facilitate their integration with other Cloud-based partners.

**The Problem**

Attached are two simple microservices which compose a *User* *Check-In* application. The Check-In application is formed of a frontend and a backend, both utilising Python Flask.

The frontend is a simple interface which allows a user check in, check out, or to display all known users and their state. The frontend forwards requests to the backend, which in turn needs to communicate with an AWS DynamoDB table ‘UserCheckin’ to persist changes and updates.

Graphical user interface, application

Description automatically generatedUsing infrastructure as code, implement a solution which will deploy the attached microservices to a functional state. The backend must be able to communicate with DynamoDB, and the front end must be able to communicate with the backend.

Note: The BACKEND\_URL environment variable can be overridden on the frontend application to tell the frontend where to find the backend. The DynamoDB table schema is also included for reference.

**Outputs**

A public Github repository containing your code and an accompanying README for deployment. You are encouraged to make comments on your chosen solution and your recommendations of future work, particularly how this could be productionised.

**Considerations**

* The client intends to migrate more applications in the future. How will the solution remain intuitive to new developers and scale to their needs?
* Encrypting resources at rest and in transit will help the client improve their security position.
* Enabling services to be self-recovering will reduce staff time spent on-call.
* How can the application be deployed in such a way to scale with demand?