System Operation

A user (patient/doctor/staff) can login via the UI
Doctors can manage their availability via the UI
Staff can manage doctors and appointments via the UI

Patients will be able to book an appointment through the UI, which shows the current available time slots.

When a patient attempts to book an appointment, it calls the backend service which checks the requested time slot to ensure a Doctor is available. If a Doctor is not available (i.e. the time slot has been filled / doctor has updated their availability), the system will return an error to the patient and reload the available time slot data.

When an appointment has been successfully scheduled. This will then create a once-off Schedule in AWS EventBridge as a trigger for the reminder. This pushes to SQS, which then is handled by a Lambda that calls SES to finally send the email.

Assumptions

- This is an entirely new platform for the medical clinic
- No business cases that prevent patients, doctors and staff from using the same interface
- This is a web application, not a phone application
- SMS as “notifications”
- This is not a tenanted application, just for a single clinic

Additional Questions

- What sort of notifications? SMS / Push / Web Notifications / App Notifications?
- TimeFrame for completion / go live?
- Are ongoing costs a consideration?
- Expected/Projected traffic amount?
- Is this for a chain of clinics or exactly one?
- Possibility for additional locations?

Rationale

- Email as notifications
    - More simple then sending SMS messages
    - If SMS was needed AWS SNS could be used, but it isn't really nice to send individual messages to individual recipients
    - Otherwise a third party like Twillio would be sufficient

- S3 + CloudFront
    - Easy deployment of the UI
    - CDN high availability
    - ReactJs/GraphQL
- ECS
    - Scalable and high availability
    - Easy to setup and use
    - NestJS (Typescript) / GraphQL
- RDS
    - Any relational database could be used, PgSQL/MariaDB/MySQL
    - Master / Slave will be on different Availability Zones
- AWS EventBridge
    - Simple CRON / Date interface for individual notifications
    - Auto-cleans up when single schedule is completed
- EventBridge / SQS / Lambda / SES
    - I made the decision to create an SQS queue that feeds the lambda. The queue allows batching of multiple messages to multiple consumers to speed up execution time and increase throughput