Software Engineer
Interview Challenge

Thank you for applying to become an integral part of the MK team. We are looking for bright minds, excited for the opportunity to build amazing software. This challenge helps us evaluate candidates on a comparable basis, as assessing candidates based upon resumes is often misleading and ineffective.

Your project will be reviewed for both functionality and source code. Complete as much as you can. You will be using the tools and infrastructure used in our production environments. You may ask intelligent questions if you run into issues. Questions should only be asked if Project specifics have been left out, or if you have sufficiently Googled and cannot find a good solution.

MK's key success measure for Software Engineers is the ability to provide business solutions using existing tools and new code in an efficient and sustainable manner. Efficiency is measured by tasks successfully completed, minus bugs, divided by time. Sustainability is measured as the cost to operate and improve software as business requirements change. Sustainability is achieved with well thought out, full featured, tested, and documented systems. Reusing existing code can be greatly improve efficiency, as long as sustainability is not negatively affected.

Project Outline:

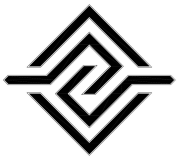
Build a website contact form. The form should contain fields for Name, Email and message at a minimum. Upon submit, the form should send the message to the email provided in the form, with the name as the email subject.

Project Requirements:

1. Set up a free account with Amazon Web Services (AWS) (https://aws.amazon.com/)
2. Using Material UI, Build a form (name, email, message, submit button). The page must be a static website, hosted in an S3 bucket.
3. On submit, the UI should call API Gateway to trigger a Lambda function.
4. The Lambda function should send an email using SES, and store the message in DynamoDB.
5. Push your code to a public GitHub repository.
6. Respond to this email with your application's URL, credentials and GitHub link.

Bonus Points:

1. Access to the form should be protected via a login screen. Set up a login page with AWS Cognito User Pools and Federated Identities.



2. Enable SSL (TLS 1.2) for your page.
3. Allow authentication via Facebook, Google or another third party provider.

Reference

Serverless:
https://blog.inkdrop.info/how-to-add-a-contact-form-to-a-static-website-with-aws-lambda-4ce1b14f4ed9
http://connorleech.info/blog/Tutorial-for-building-a-Web-Application-with-Amazon-S3-Lambda-DynamoDB-and-API-Gateway/
https://docs.aws.amazon.com/lambda/latest/dg/with-ddb.html
Material UI: https://material-ui.com/demos/text-fields/
S3 bucket: https://www.fullstackreact.com/articles/deploying-a-react-app-to-s3/