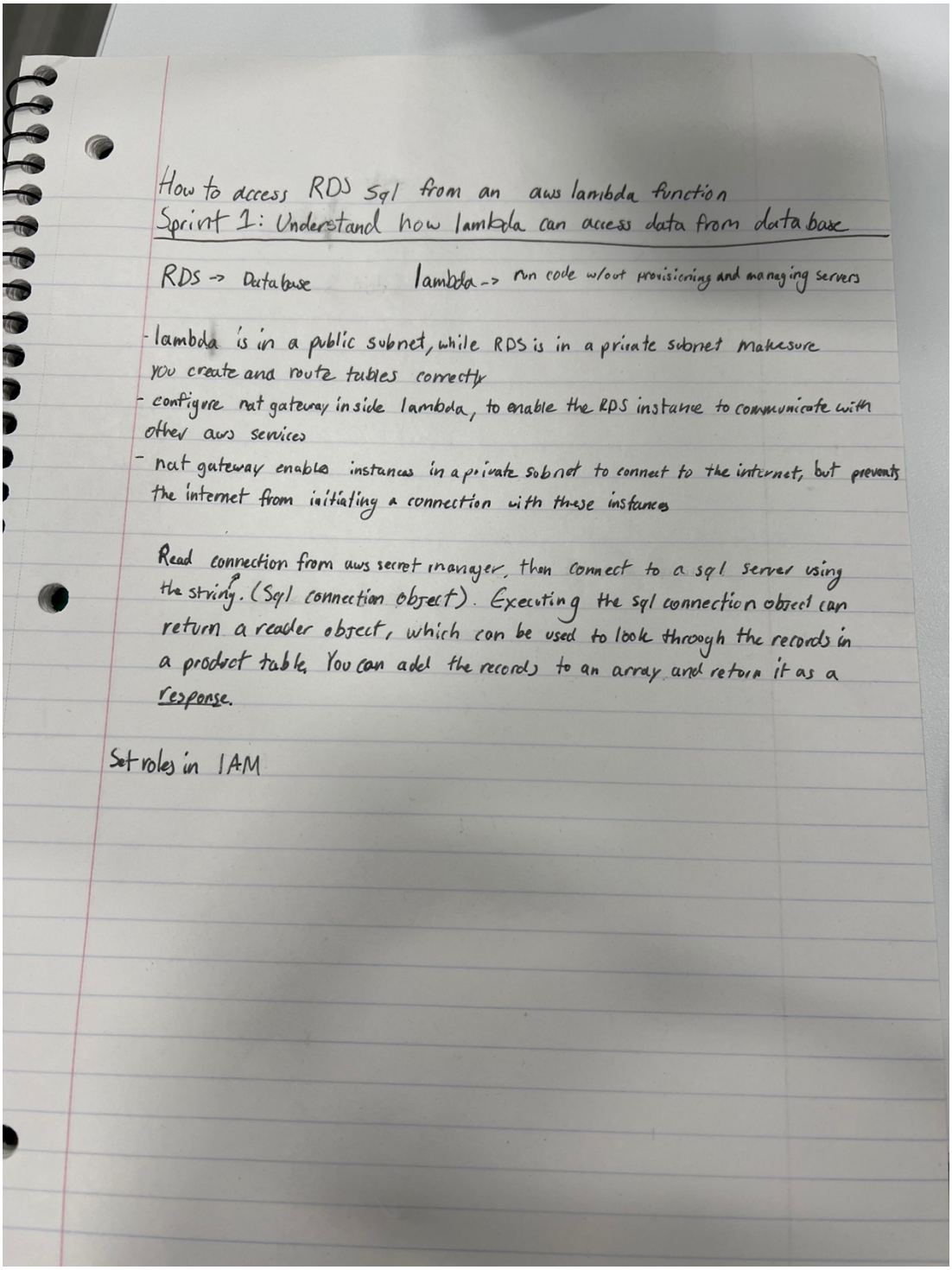
Task: How lambda interacts with the database



How to access RDS Sql from an aws lambda function

Sprint 1: Understand how lamkda can access data from database

RDS -> Database lambda -> run code w/out provisioning and managing servers

- lambda is in a public subnet, while RDS is in a private subnet makesure you create and route tables correctly
- configure nat gateway inside lambda, to enable the RDS instance to communicate with other aws services
- nat gateway enables instances in a private subnet to connect to the internet, but prevents the internet from initiating a connection with these instances

Read connection from aws secret manager, then connect to a sql server using the string. (Sql connection object). Executing the sql connection object can return a reader object, which can be used to look through the records in a product table. You can add the records to an array and return it as a response.

Set roles in IAM

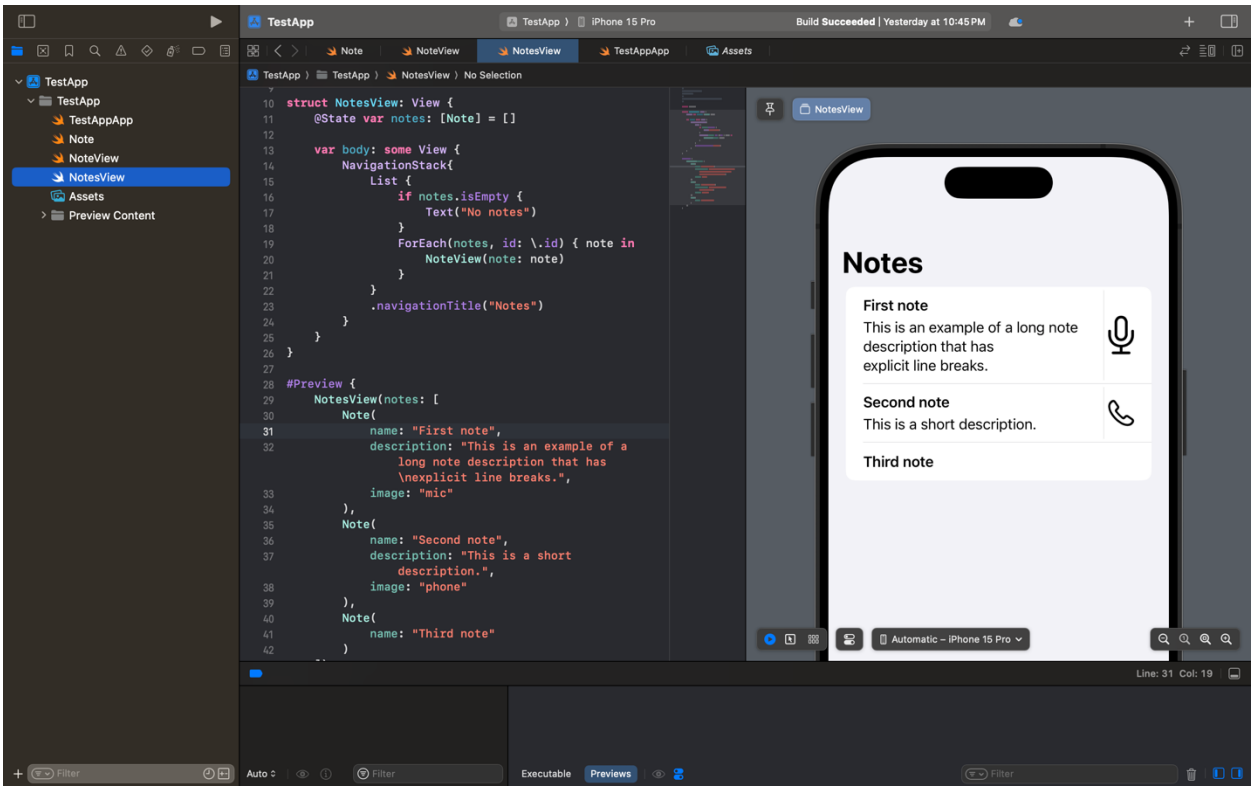
https://www.youtube.com/watch?v=LDGgAkSf8-Y

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-lambda-tutorial.html

Task: Install swift onto computer
Swift was already installed on my computer.

Task: Determine if swift will be viable for project or if android will be better

I followed the tutorial from aws at this link to create a mobile app
https://aws.amazon.com/getting-started/hands-on/build-ios-app-amplify/



I looked at different ways to implement a mobile app, the two ways I researched was to use swift using the vapor framework to connect to our amplify server. The other way was to build a react native app using the webview framework, I believe using this way would be easier to implement since we are already using reactjs for our front end. Doing it this way would eliminate swift from our project.

https://reactnative.dev/docs/integration-with-existing-apps
https://medium.com/swiftybeaver-blog/deployment-of-a-vapor-app-to-aws-ec2-f577eaa6c38c
https://curiosum.com/blog/dedicated-mobile-application-in-react-native-webview-guide
https://docs.amplify.aws/swift/start/getting-started/add-api/