Condo Daily Report 9-1-21

Wednesday

* Started integrating YAML scripts
  + Researched best way for module “plug and play” template model
    - Discovered the best way to approach this is a nested stack
      * Implemented VPC template
      * Implemented subnet template
        + Realized that if these are to be hardcoded then there will be CidrBlock collision

This is bad because if CidrBlock bounds overlap there may be IP collision

* + - * + Found a way around it by implementing a lambda function that talks to the subnets and VPC’s

It takes the API’s, and gets the used CidrBlocks

This is important because we can compare it to usable CidrBlocks which will allow auto scaling and easy deployment

* + - Thought about how we need these template files to be in S3 bucket to perform nested stack functions
      * Nested stack functions take YAML templates in a s3 bucket and pass them in as a “CloudFormation” object rather than the actual resource object
        + This allows for less coupling
        + This is also beneficial because when we have to fix lets say a vpc CidrBlock because we need more IP addresses we only have to change one file rather than a whole bunch of large files with VPC implementation
      * Found a way to complete this task
        + Was going to use an AWS “quickstart” to form a connection from Github to s3
* Quickstart
  + Get the quickstart repo and clone it to local computer
  + Take the repo and put it in an s3 bucket
  + Use cloud formation to open the s3 repo and target the template file located in the templates/ subdirectory
  + Let it run and look for rollback
    - If there aren’t any rollbacks you are good to go
  + If you look in the templates YAML file you will see in the outputs section that the webhooks API, and the SSH public key are both available
  + Go to outputs section in cloud formation
  + Take the given public key and use it to establish an SSH connection on Github
  + Take the webhooks link and go into the target repo
    - Go into settings, then webhooks section
      * Post the link and make sure you change the format to json because that is the structure of the file
        + If you keep it as the other structure, then you will fail the call with a 400 error you will be able to see in Github webhooks request section or in API gateway
  + This should complete the call portion
  + After doing this I was able to see the call being made anytime I pushed to the Github repository
    - The only trouble is the data was not transferred to the S3 bucket
      * Looked through log files in Cloudwatch
      * Found that the problem was happening near the last step in the process which was the lambda function that interacts with CodeBuild
      * Started debugging the lambda function
        + Found that the lambda function started having trouble at the codebuild\_client.start\_build portion of the lambda function

Line 160ish

* + - * + Called all the variables passed to print them in the logs
        + They printed in expected formats
        + Going to investigate how the start\_build method is structured tomorrow to make sure variables are being passed in correctly
* Wifi
  + Wifi went down temporarily today
  + Everyone in the company had to log back in
  + The password was generic, so it was time to change
  + Changed password to more complex password plus reversed from ordinary
  + Sent out instructions in case trouble connecting to wifi