Tuesday

* ACloudgur
  + Practiced some of the tests again
  + Reviewed some material about efs/ebs
* Tech meeting
  + Objs
    - Lambda
      * Get lambda to work with different connectors
        + Talked to kayce about woocommerce

He said that we would need to do a work agreement type deal with a customer who has woocommerce to see if its possible

* + - * In the example of me performing an api connection in fivetran, I didn’t have a delete statement and I only had one table
        + Kayce told me that he would like to see an example of a lambda function with both a delete and multiple tables
    - Training
      * Acloud guru
    - Tech support
      * Take all the tech issues for the day while kayce works on competitor analysis
    - Look into making a connector with some type of authentication
    - Cloudformation
      * Review the stack we have
        + Think about how we can improve with all of the new tools in aws
* Lambda
  + Implementation of a delete/ additional table
    - I had to think about who to decide what records to delete
      * I thought yes a person could specify a specific row they would not like, but that isn’t realistic
      * In data sets generally people don’t like the row if certain values are lower or higher
        + Outliers
      * Null values
    - I decided to make mine simple
      * My dataset that the api connected to was crypto
      * The threshold has nothing to do with real life strategy, but I decided to delete all rows whos crytpo value is less than 10 bucks
      * Ill add a code snipit to show how I did it
* lambda\_shiz = []
* not\_lambda\_shiz = []
* other\_shiz = []
* for t in timeline:
* # Remember the first id we encounter, which is the most recent
* if since\_id == None :
* since\_id = t["id"]
* # Add all tflLineStatus
* if float(t["priceUsd"]) > 10:
* lambda\_shiz.append({
* "linename": t["id"],
* "currentprice": t["priceUsd"],
* "changePercent24HR" : t["changePercent24Hr"]
* })
* elif float(t["priceUsd"]) <= 10 and t["id"] == "tron":
* other\_shiz.append({
* "linename": t["id"],
* "currentprice": t["priceUsd"],
* "changePercent24HR" : t["changePercent24Hr"]
* })
* else:
* not\_lambda\_shiz.append({
* "linename": t["id"],
* "currentprice": t["priceUsd"],
* "changePercent24HR" : t["changePercent24Hr"]
* })
  + - * Alright the important pieces are the if float and the else.
        + The elif in the middle is for the extra table
        + The if float price usd one is a conditional that says find the attribute in the json from the api that is correlated with price
        + Then cast the value to a float

This is important because it needs to be a float to be comparable to another number

* + - * + Also above you will notice I have 3 empty lists created outside the for loop

Lambda\_shiz

The primary table

Not\_lambda\_shiz

What rows to delete

Other\_shiz

Other table

These are all throwaway variable names and will be changed when going into production

* + - * + If the value is greater than 10 it gets appended to the main table
        + Else it gets appended to the delete list
* "state": {
* since\_id: since\_id
* },
* "schema" : {
* "lambda\_shiz" : {
* "primary\_key" : ["linename"]
* }
* },
* "insert": {
* "lambda\_shiz": lambda\_shiz,
* "other\_shiz": other\_shiz
* },
* "delete": {
* "not\_lambda\_shiz": not\_lambda\_shiz
* },
* "hasMore" : False
* }
  + - * Alright now this is the format that is required to pass into fivetran
      * It requires a state, schema, insert, and has more
        + It doesn’t require a delete
      * So the categories I want to focus on are the delete and the insert
      * As you can see the delete just passes in the list that we created earlier
        + Not\_lambda\_shiz
      * And the insert inserts the tables we create
        + Lambda\_shiz
        + Other\_shiz
    - After that I just return that structure
    - The structure has all the values in a json format which is a passable object to fivetran
* Call with john
  + I was asked to join the call about the woocommerce connector
    - The customer asked if it would be possible for the woocommerce connection to happen
    - I told them no unless he moves to a dedicated host and has root access to the server along with admin access to the mysql database
    - The sales person said we could try
      * I shut that down at that moment because the dev team wasn’t there on their side to make a user with access and perms
    - He is going to see about moving his shared host to a dedicated host
  + Going to talk to kayce about getting woocommerce api key for this particular client
* API practice
  + This is also doing some lambda stuff
    - It isn’t exactly lambda, but instead im doing it on my local computer doing python scripts
    - Wanted to do an instance where It was an api that needed some form of authorization
    - Tried google drive, but it has you use its own library
    - Then tried github
      * Did it two ways
        + Using the requests library
        + Using the pygithub library
* Acloud guru
  + Attempted the final exam but ran out of time because of the end of the day
    - Need some more work just wanted to see how far along I am!