1. Get dataset from kaggle
2. Clean dataset up on Jupiter using pandas
3. Upload dates into a folder in s3
4. Create a crawler in glue, include s3 bucket path, create iam role so bucket can to access s3 bucket, and create database where crawled data can be shown
5. Run the crawler and see the results in the database what tables have been crawled, edit schema, if needed
6. Create a job in glue. And create iam role for glue and add s3 policy and glue service role. Choose finding records to remove duplicates for ml or change schemea. For change scheama use s3 for data store and use parquet format, and target path is ur s3 bucket with parquet name
7. Run glue etc job and the new parquet dataset will appear on s3.
8. Run crawler again to see results for parquet in s3
9. Open Athena and choose s3bucket to run queries
10. Create redshift cluster called redshift-clusters-ahaz
11. Go to editor on redshift and connect to database named dev as default and awsuer for default
12. On the query connect it to glue database which is connected to s3 bucket instead of copying it directly from s3
13. Write command on redshift cluster create external schema fire_schema FROM data catalog database 'ahazdatabaseglue' iam_role 'arn:aws:iam::287673164100:role/AhazRedShift' region 'us-west-1'
14. Open up quick sight and sign up for it and connect to s3 bucket X
15. Download tableu and make sure cluster is open for external access
16. Open tablue put in redshift address port number and connect to dB
17. Create Lambda function, runtime in python, add trigger as s3, and add permissions for lambda to access glue console
18. Write python code in lambda funciton to start crawler and add event then deploy

```
import json
import boto3
client=boto3.client('glue')
def lambda_handler(event, context):
    print('————————————————')
    response =
client.start_crawler(Name='AhazS3crawler')
    # TODO implement
    print(json.dumps(response, indent=4))
```

19. Upload document into s3 then check cloud watch for logs
20. Go to aws sns and create a topic
21. Then go ahead and create a subscription and put the arn from topic to subscription
22. Add code into aws lambda to access arn from topic
23. Attach sns aws to lambda trigger, every time a file is uploaded the log is sent to email successful or not.