PODCAST SUMMARIZER

Abhishek Shah

R·I·T

ROCHESTER INSTITUTE OF TECHNOLOGY

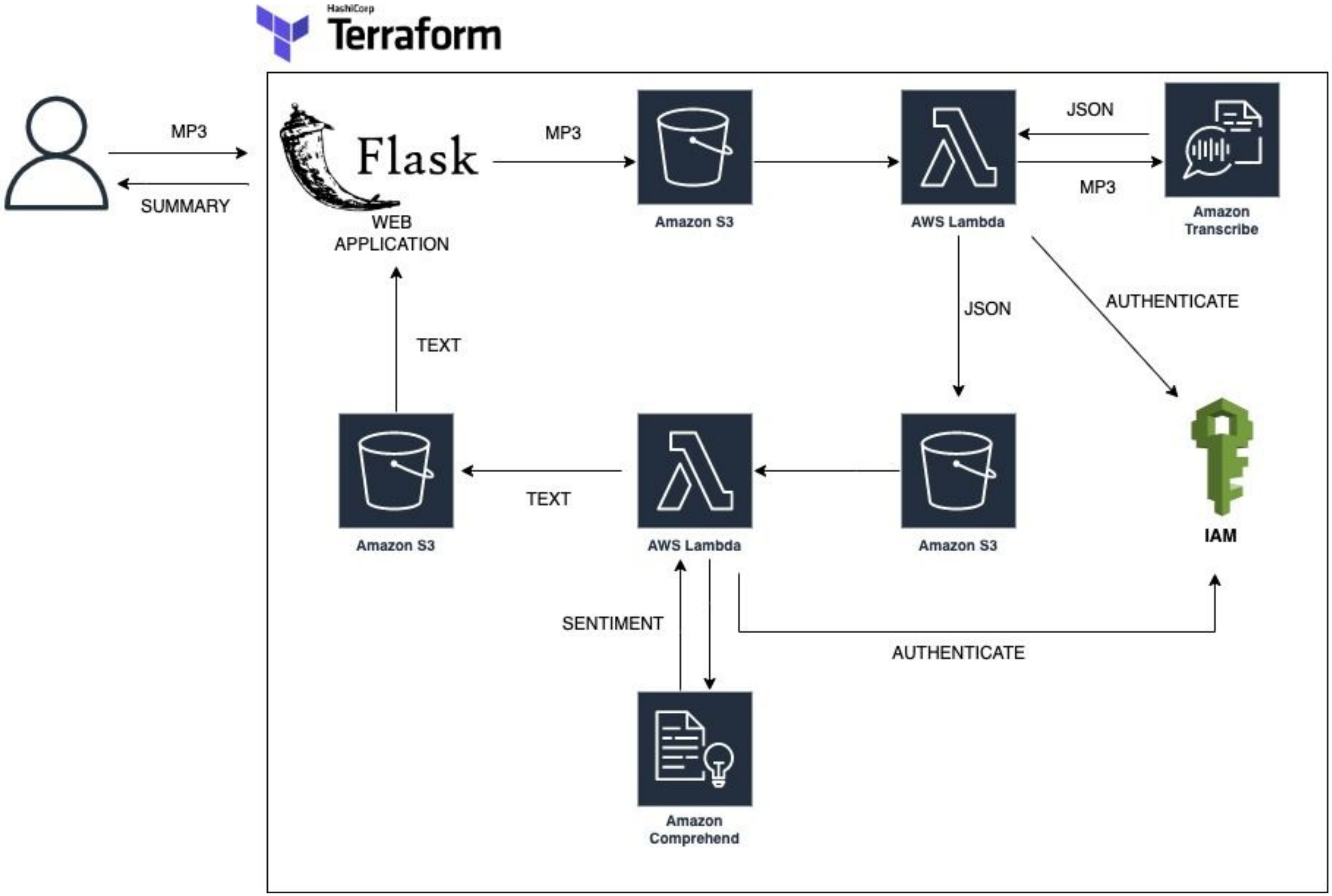
OVERVIEW

- ❏ Developed and fine-tuned an extractive summarization model that will automatically generate summaries of audio podcasts in approximately 4 to 5 sentences
- ❏ Implemented a sentiment analysis script to detect the sentiment of the summarized text
- ❏ Designed a web app using Flask for publishing the summarized text along with its sentiment

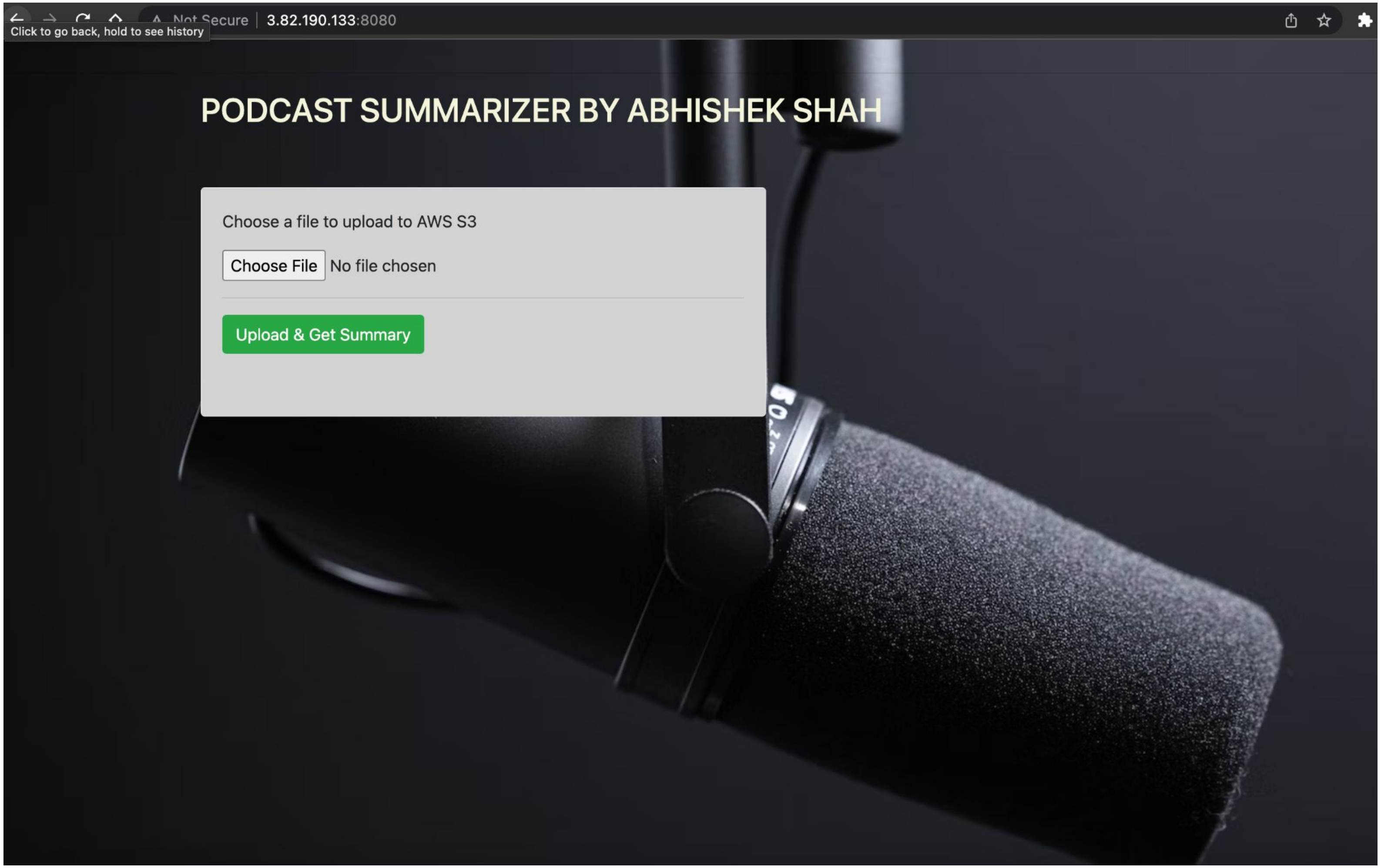
AWS TECHNOLOGIES

- S3
- LAMBDA
- TRANSCRIBE
- TERRAFORM
- EC2
- IAM
- COMPREHEND

HIGH LEVEL ARCHITECTURE



FRONTEND - USER INTERFACE



PODCAST SUMMARIZER BY ABHISHEK SHAH

Choose a file to upload to AWS S3

FRONTEND - FLASK APPLICATION

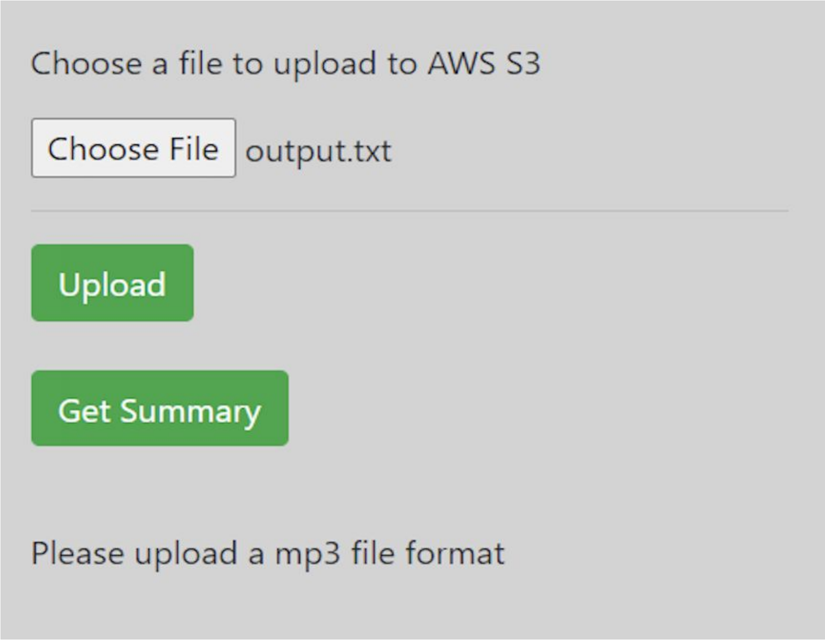
❑ Our Flask application runs within an EC2 instance

❑ Packages that we downloaded to create our shared AMI image - include Python3, pip3, Flask, gunicorn3, Boto3

❑ Our application uploads mp3 file to S3 and gets summarized text and sentiment from S3 bucket that stores the output file

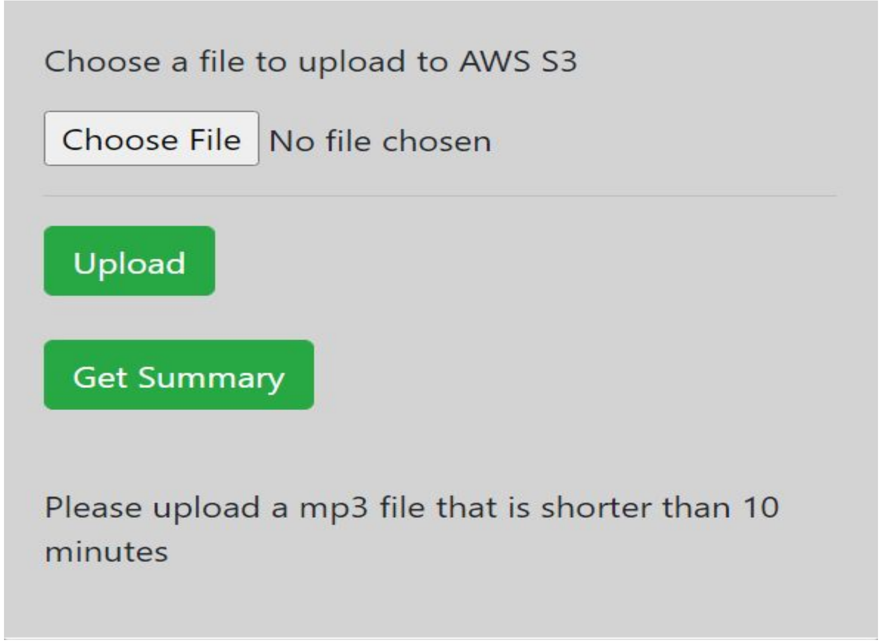
❑ Handles routes such as upload podcast MP3 files and retrieve their summary/sentiment

FRONTEND - VALIDATION



Please upload a mp3 file format

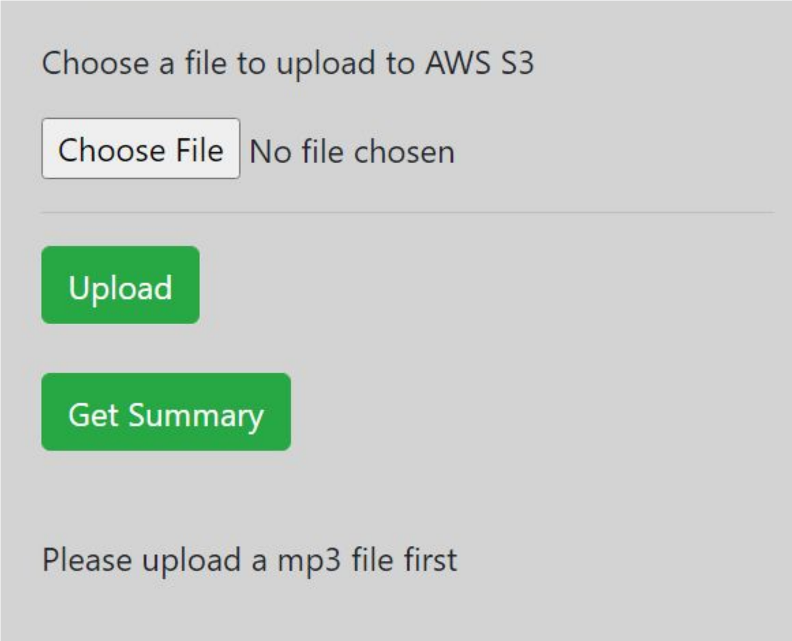
Incorrect file type



Please upload a mp3 file that is shorter than 10 minutes

Uploaded MP3 longer than 10 mins

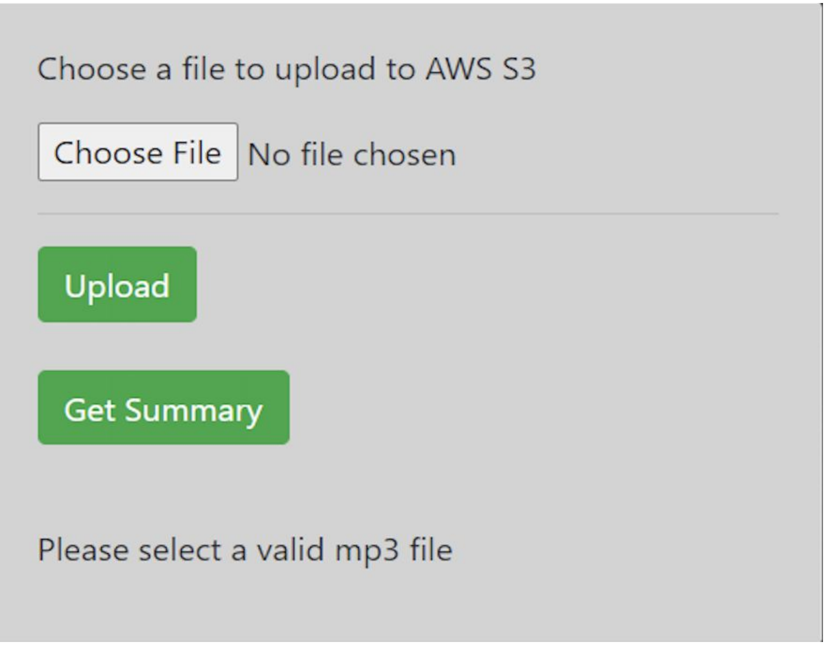
FRONTEND - VALIDATION



Choose a file to upload to AWS S3

Please upload a mp3 file first

Clicking on Get Summary without uploading a file first

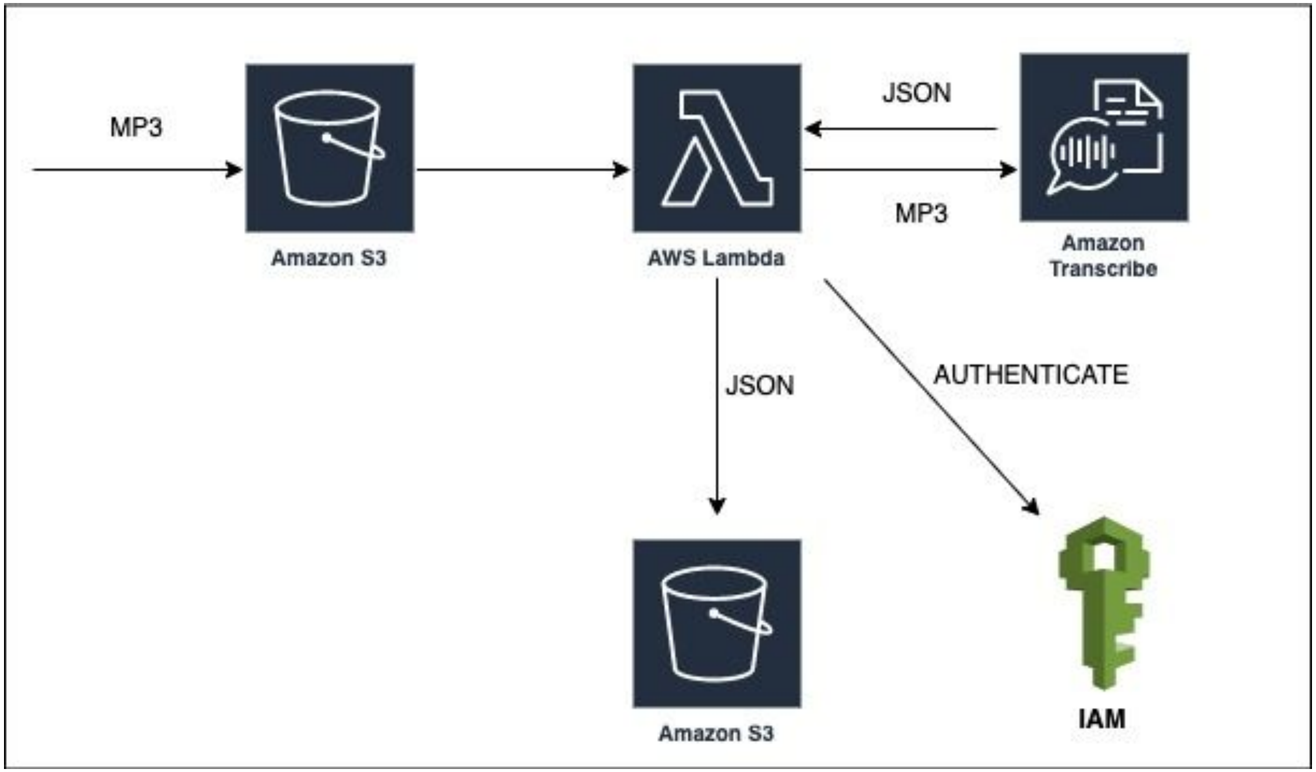


Choose a file to upload to AWS S3

Please select a valid mp3 file

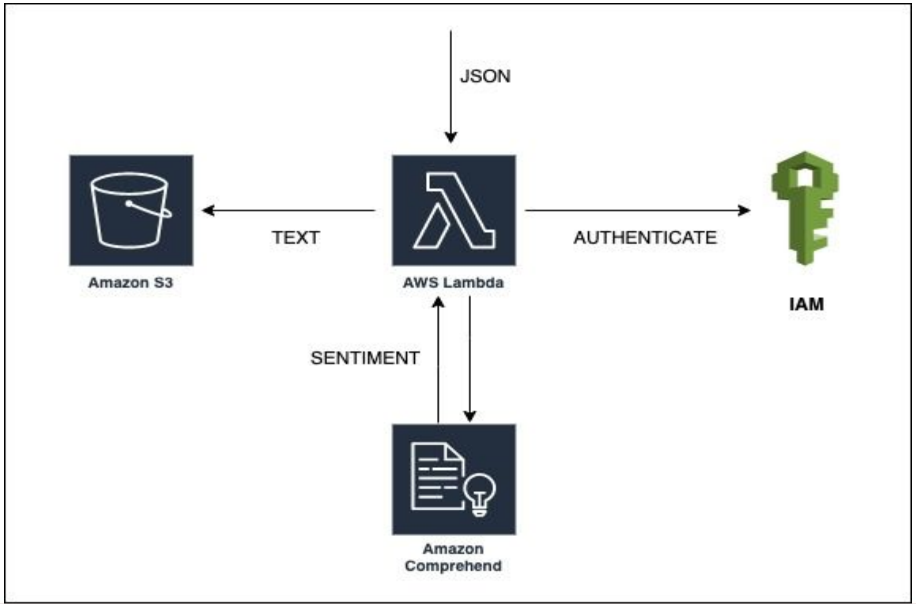
Clicking the Upload button without choosing a file

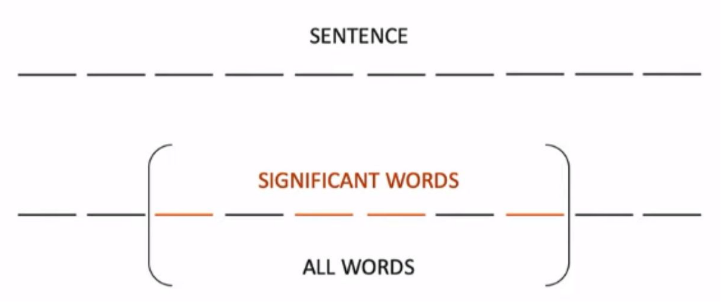
TRANSCRIBE LAMBDA



SUMMARY LAMBDA

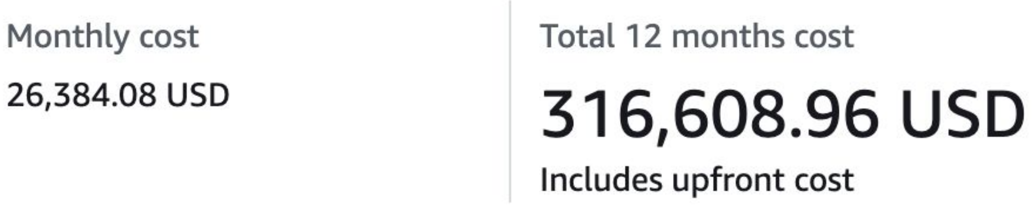
- ❏ NLTK:
  - ❏ Tokenizing & stemming text
  - ❏ Get a list of stopwords to pass to Luhn Summarizer
- ❏ Luhn summarizer:
  - ❏ Get list of most significant words via word frequency
  - ❏ Sentence ranking
    - ❏ Score=(n^2) /m (n: number of significant words, m: max window span)
- ❏ AWS comprehend is called via boto3 to determine summary’s sentiment





ESTIMATED MONTHLY COSTS

- **Assumptions:**
  - 10,000 users/month
  - 2 podcasts/user per day
  - 600,000 podcasts/month
  - Each podcast ~ 10mins
  - Each podcast ~1200-1500 words



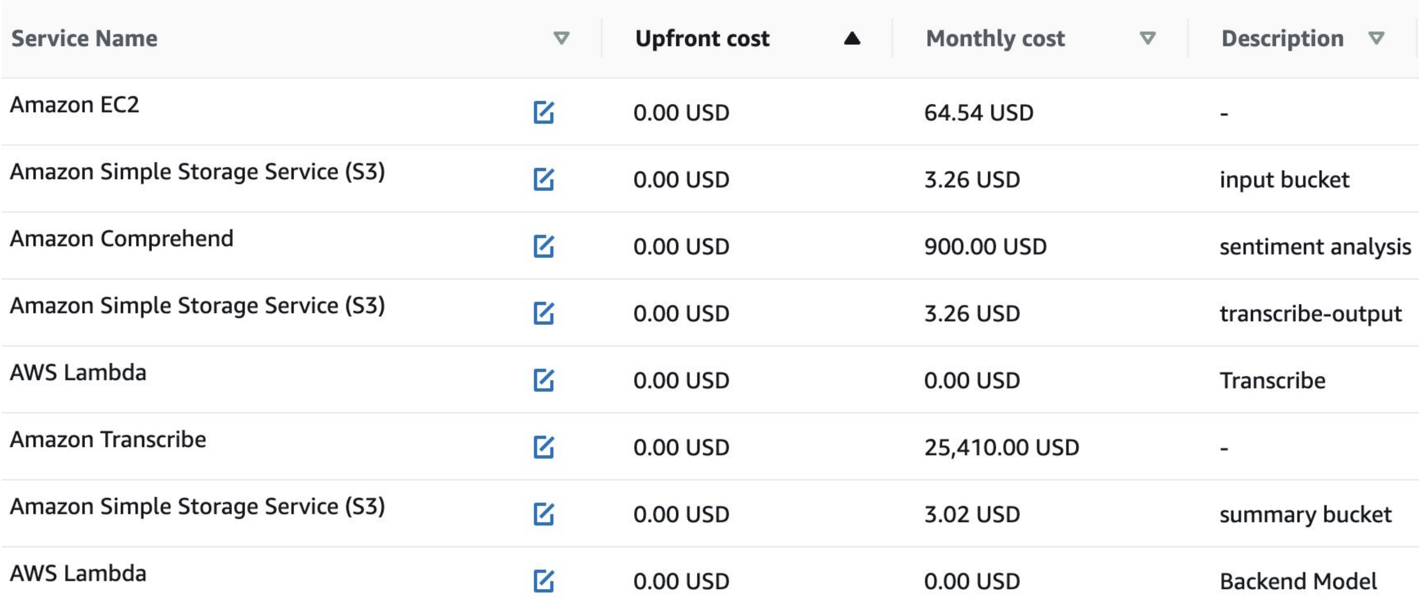
Monthly cost

26,384.08 USD

Total 12 months cost

316,608.96 USD

Includes upfront cost



Service Name	Upfront cost	Monthly cost	Description
Amazon EC2	0.00 USD	64.54 USD	-
Amazon Simple Storage Service (S3)	0.00 USD	3.26 USD	input bucket
Amazon Comprehend	0.00 USD	900.00 USD	sentiment analysis
Amazon Simple Storage Service (S3)	0.00 USD	3.26 USD	transcribe-output
AWS Lambda	0.00 USD	0.00 USD	Transcribe
Amazon Transcribe	0.00 USD	25,410.00 USD	-
Amazon Simple Storage Service (S3)	0.00 USD	3.02 USD	summary bucket
AWS Lambda	0.00 USD	0.00 USD	Backend Model

CONCLUSION

- **Recommendations/Findings**
  - Effective Podcasts Management
  - Marketing
  - Tag Generation
  - Content Generation
  - Personalization