

Twitter is one of the biggest social media networks on the internet. Conversations are always rolling, be they about poliical topics impacting everybody to minor updates in niche communities you personally care about. All your friends and your favorite celebrities are on it. If you're like me, sometimes you may find it overwhelming. You may not have the time to participate in all the action. To solve this challenge, our team at Duke has created this Cloud-Powered tool to always keep you engaged in the conversation.

Checkout our [github page](#) for a deeper look into our project.

Checkout the [front end to our tool](#).

Or see what our [twitter bot](#) is up to. Follow it if you're into wrestling!

To run tool, simply select a trending topic from the list of trending topics and hit "Submit".

To build this tool, we utilized the services provided on the AWS platform. All decisions made during development were subject to the following constraints:

- The tool should be up to date with trending topics and thus needs to select trending topics everyday, with the machine learning model training everyday as well.
- We need to minimize running costs as much as possible, preferably to less that a dollar a day.
- The infrastructure specification should be explicit to allow easy debugging and consistent behavior.

We thus use Infrastructure as code (IAC) to delpoy the cloud infrastucture. To do this, we used the excellent AWS CDK, which allows all infrastructure to written as Python objects. These Python scripts can be executed to create yaml templates for AWS Cloudformation, which is then used to actually deploy the infrastructure.

We use AWS EventBridge to schedule triggers daily. Upon a trigger, we execute a lambda function to pull trending topics from the Twitter API and save them to an S3 bucket. A few hours later, another trigger calls a batch schedule job. This job spins up cheap AWS Spot instances to scrape tweets from each topic within the past 24 hours and save them to S3. It also trains a simple n-gram Natural Language Processing Machine Learning model to "learn" the text from the past 24 hours. We dockerize all our codes to pass them to the Lambda functions and the spot instances. A Google Cloud Run instance acts as the server for our front end, calling which triggers a model inference to generate a brand new tweet that you can use to post to your handle.

Contributors:

- [Anna Dai](#)
- [Deekshita Saikia](#)
- [Moritz Wilksch](#)
- Satvik Kishore