﻿ **Columbia University in the City of New York**

AI and OR at Scale on the Cloud

Assignment 5

Group 2: Harsh Mehta (hsm2148) Anunay Sanganal (avs2160) Skand Upmanyu (su2236)

﻿In this assignment you'll be deploying your model as a service on AWS.

If you didn’t succeed in training a model and saving it with tensorflow save, you can use this

one for this assignment https://aiops-2020-public.s3.us-east-2.amazonaws.com/model.tar.gz.

Otherwise, use your own.

**1. Sagemaker inference**

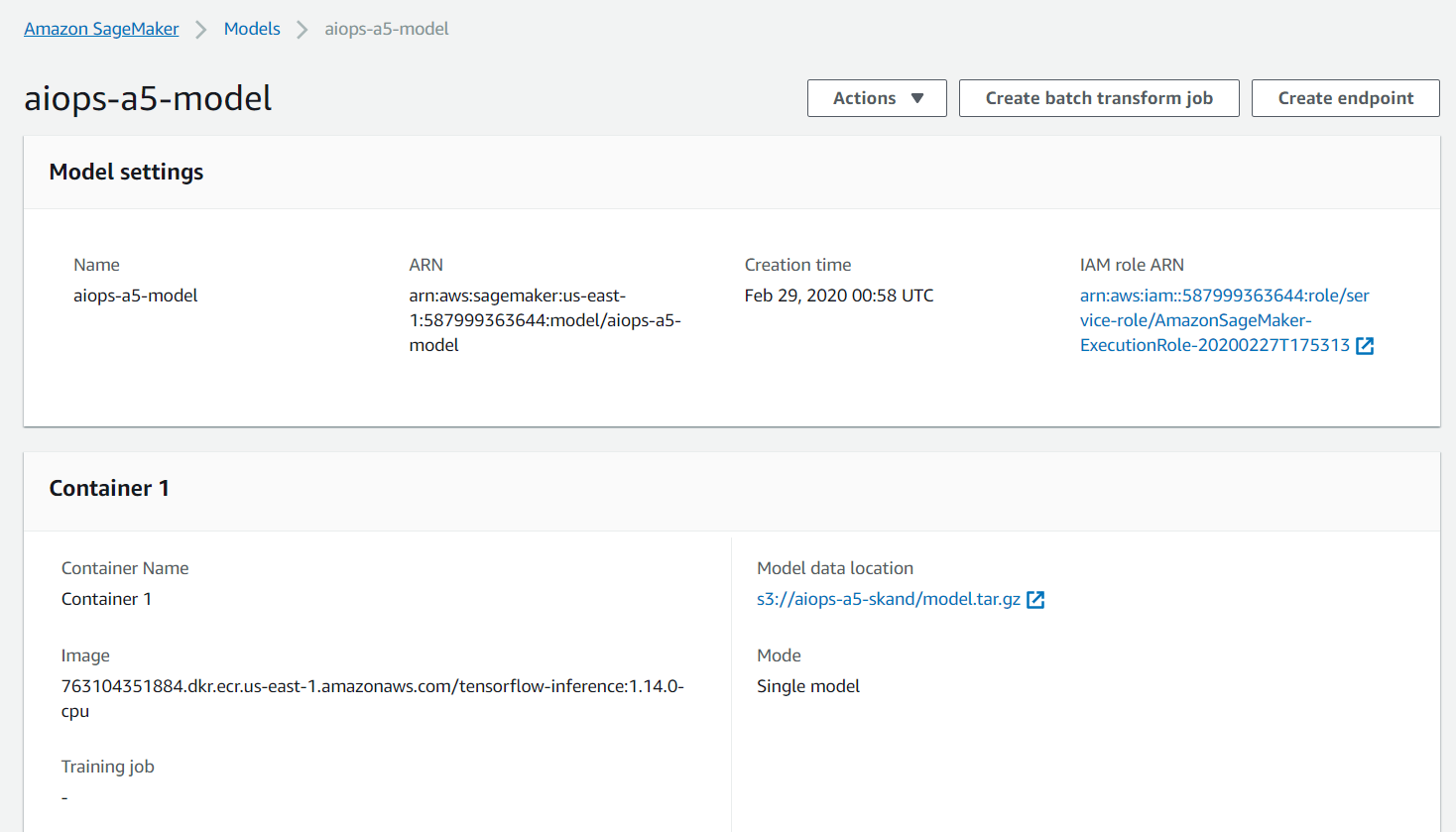
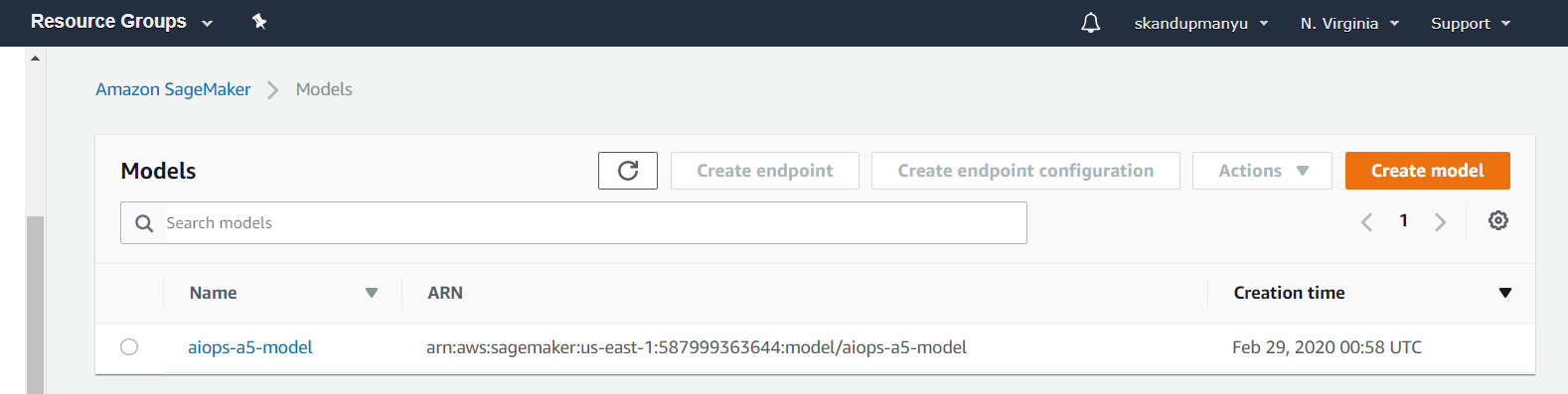
Following the example in class, deploy your model as a SageMaker inference endpoint.

Make sure to edit the model in the endpoint configuration to select the cheapest instance:

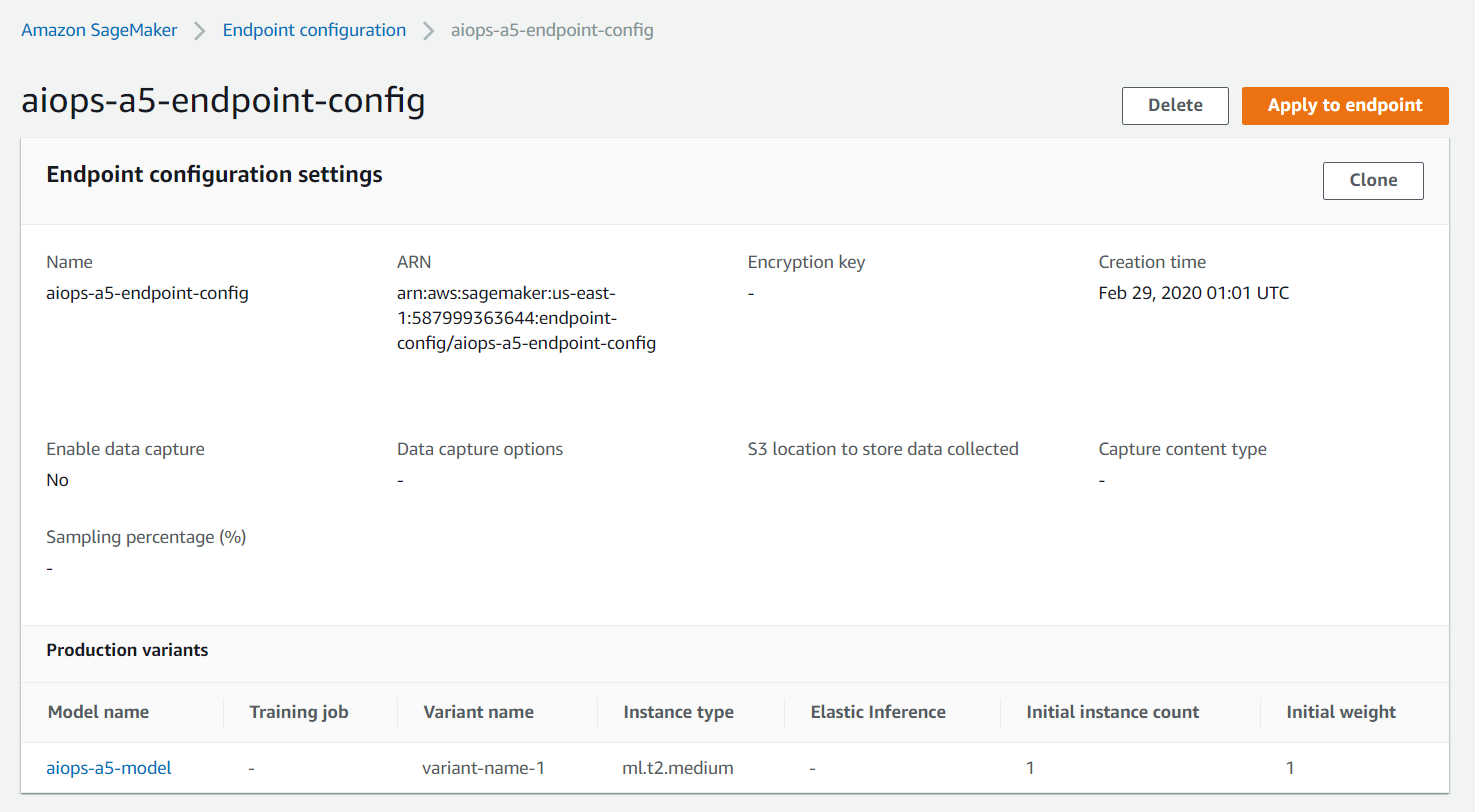
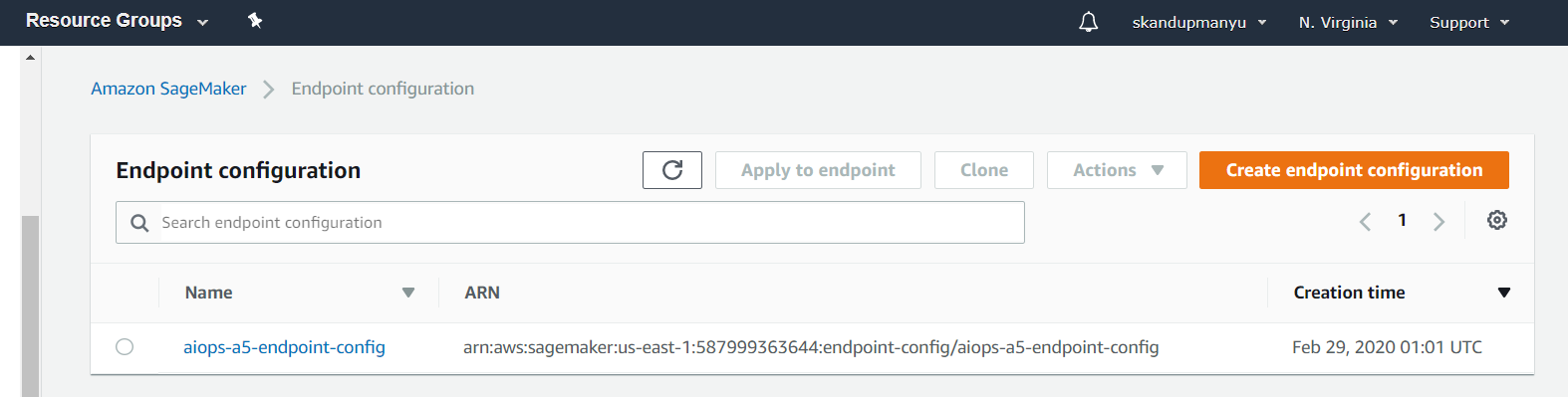
“mk.t2.medium” When creating a “Model” use this image:

763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference:1.14.0-cpu

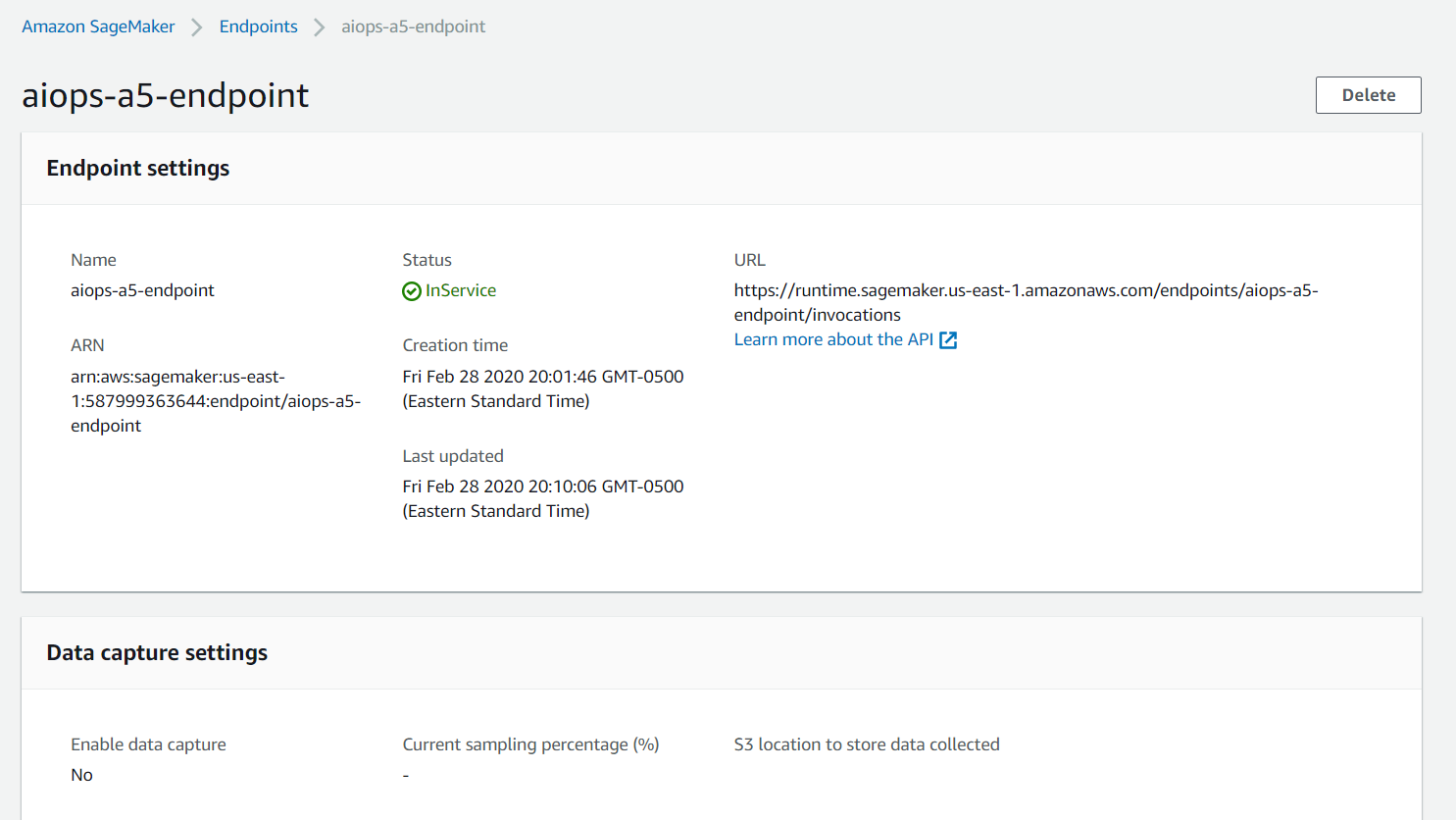
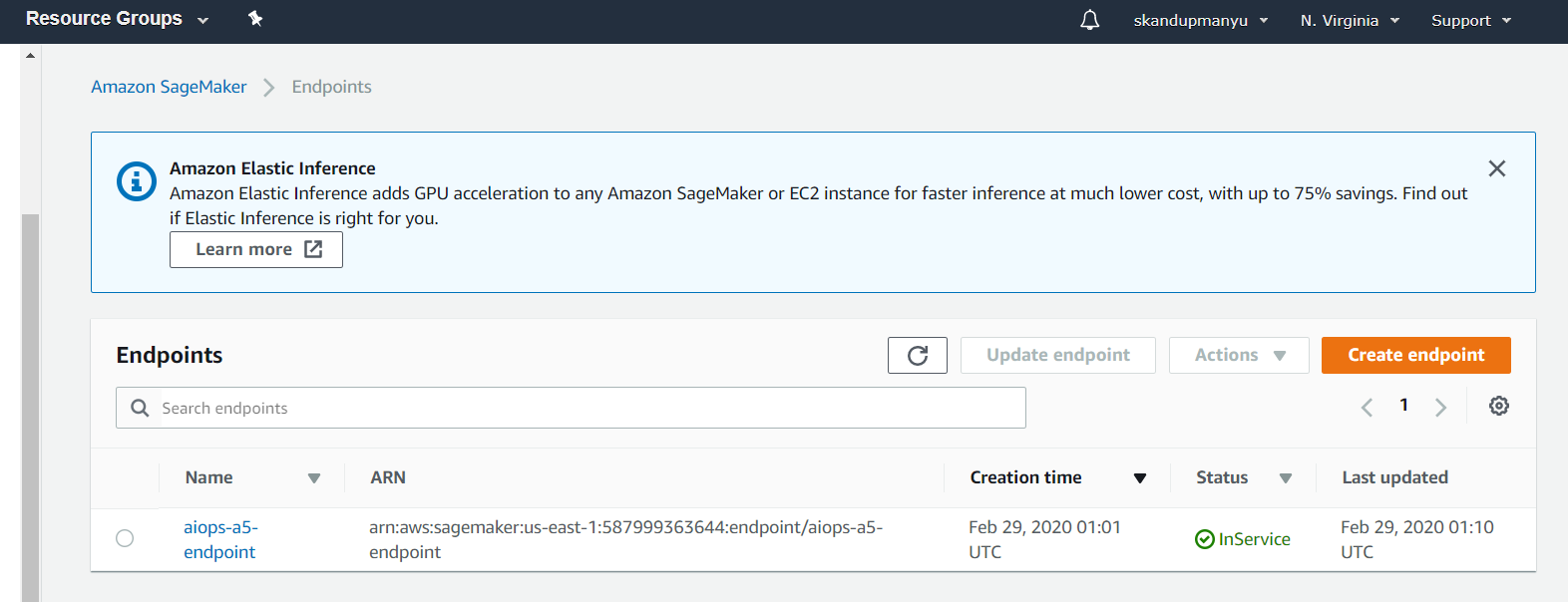
Creating model:



Creating endpoint config:



Creating endpoint:



**2. Lambda functions**

Following the example in class, build a lambda function that performs:

• Pre processing using your code from HW3

• Model inference using your SageMaker endpoint

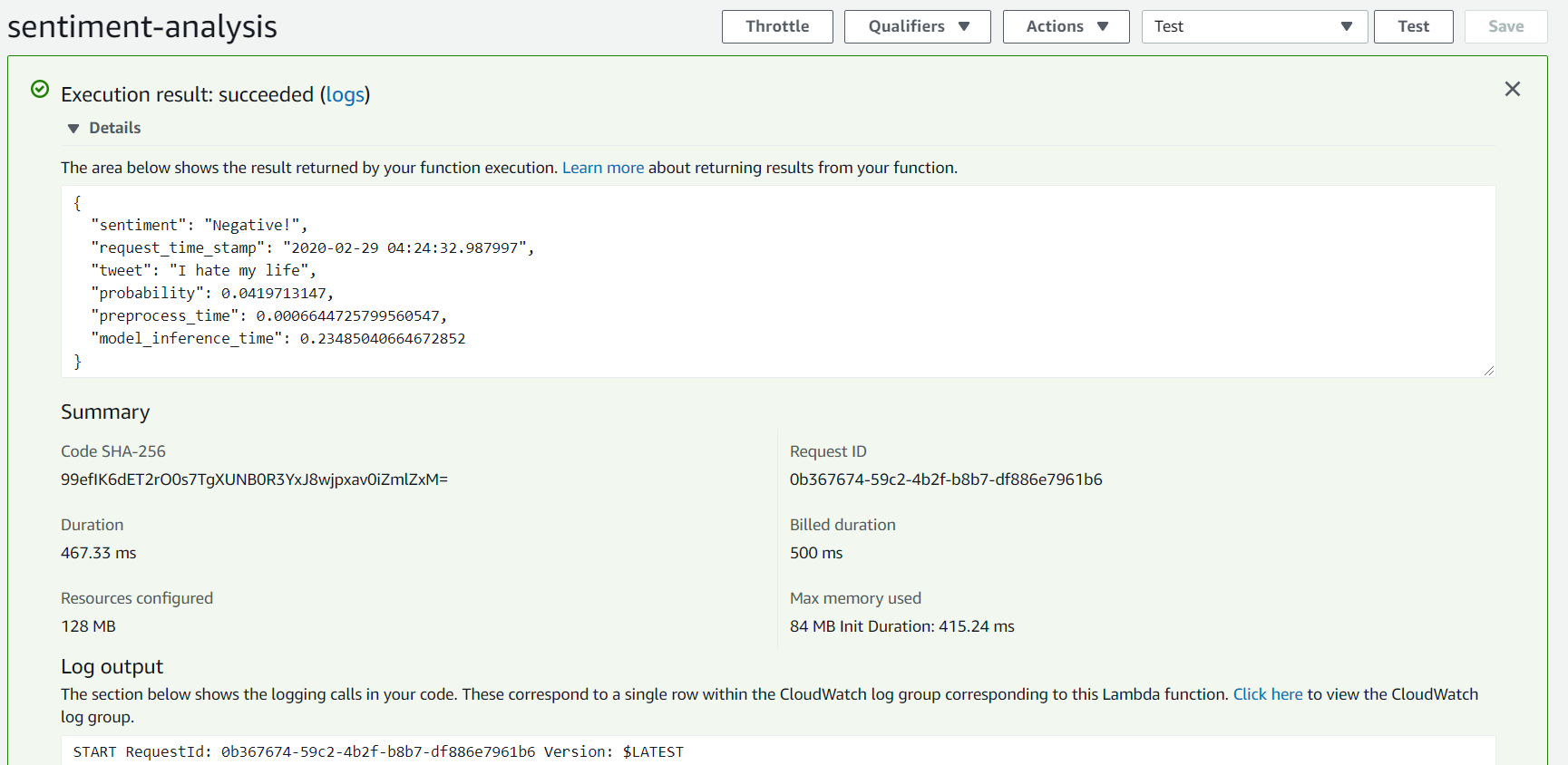
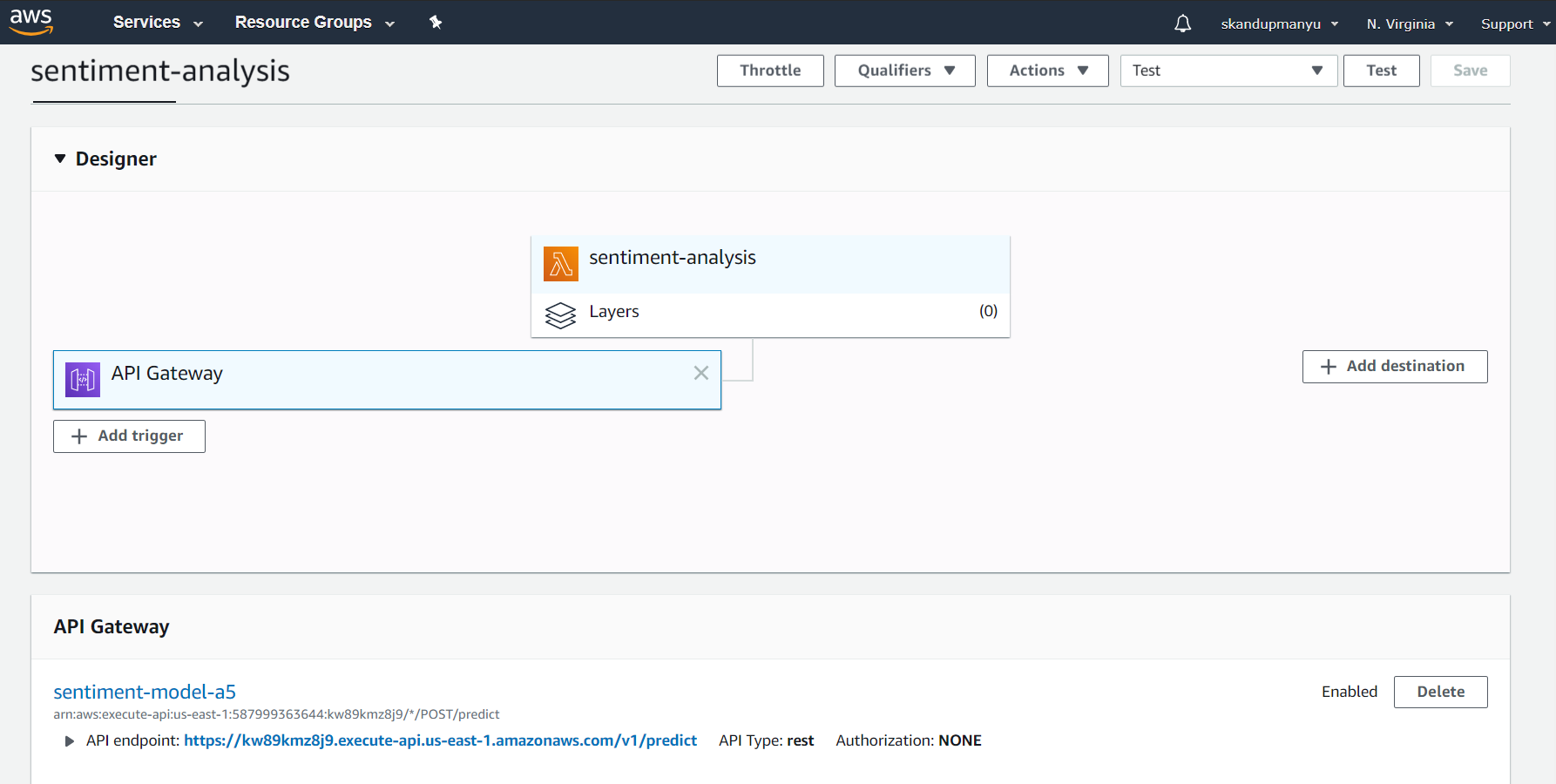
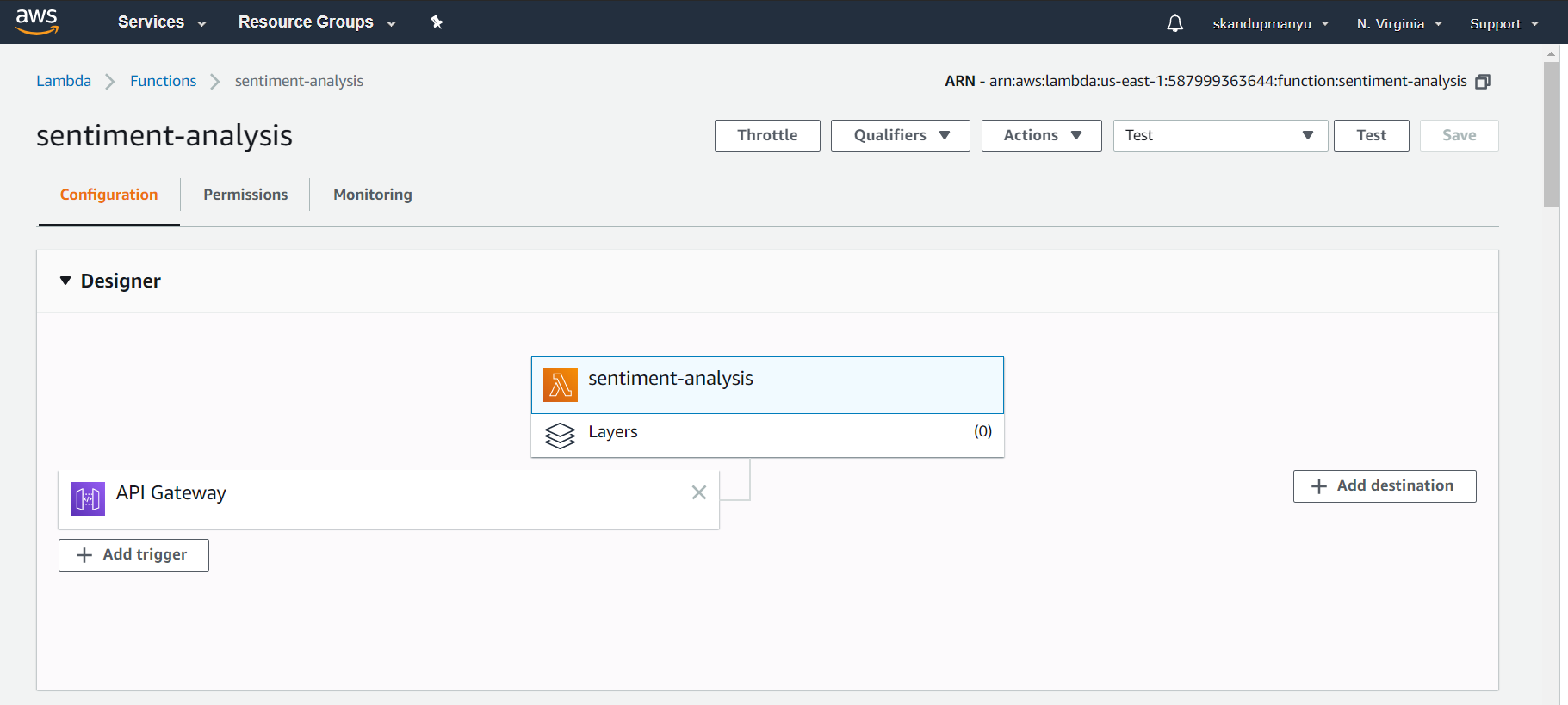
• Post processing using a logic demonstrated in class

Your lambda function should take a JSON input with a “tweet” key and produce a JSON output

with a “sentiment” key and a value that can either be “positive” or “negative” based on the

model prediction.

Creating lambda:



**3. Payload logging**

Modify your lambda function to implement payload logging.

After the post processing, your lambda should be logging a JSON object to a bucket in S3.

This object should have the following items:

• Date and time of the request

• Tweet

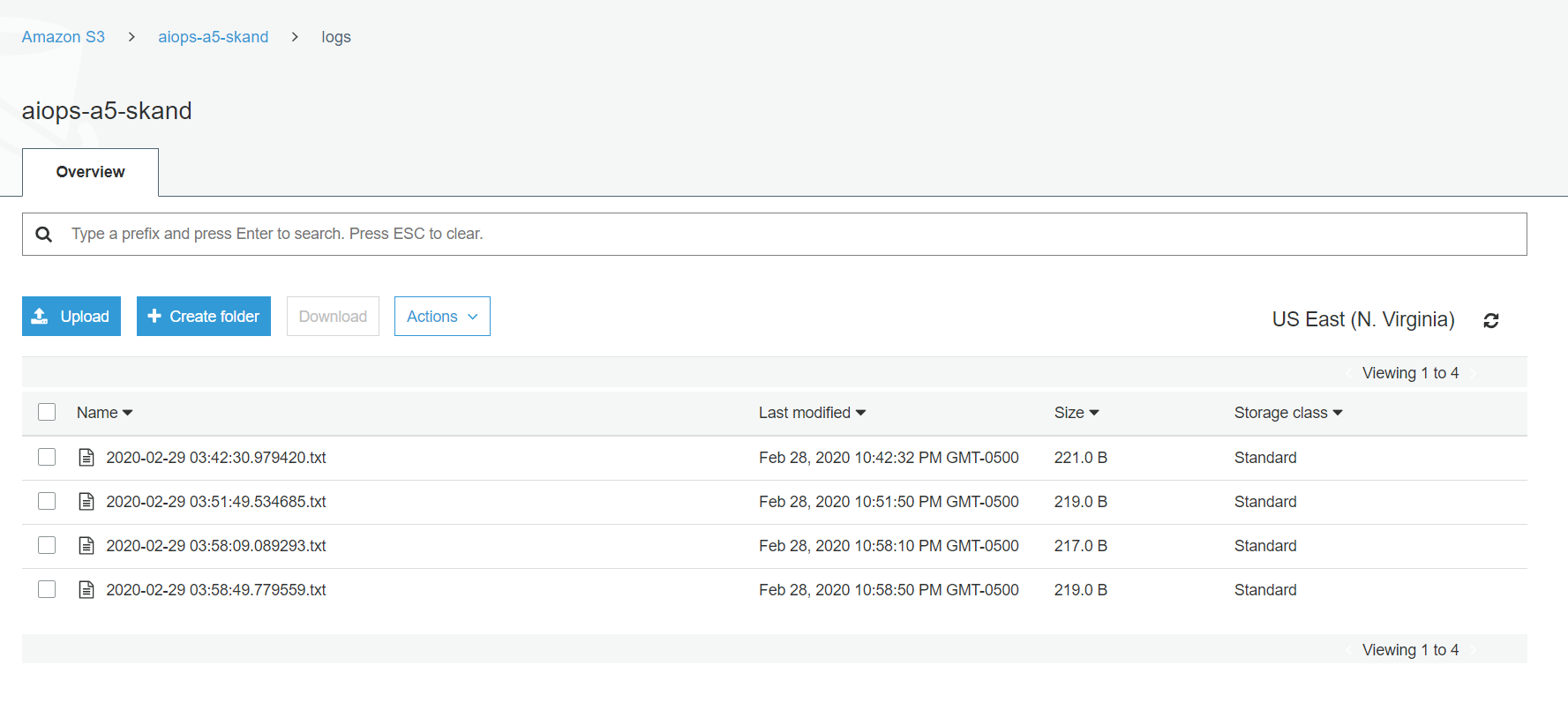
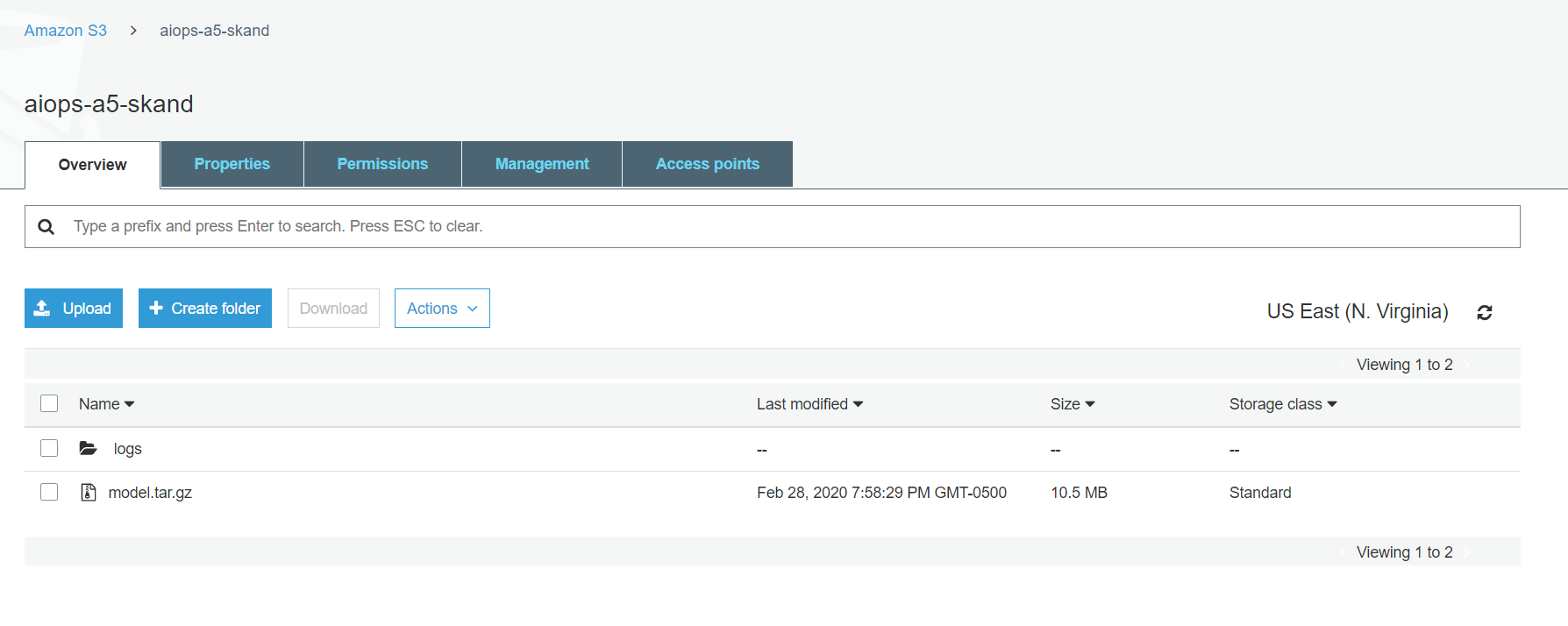
• Sentiment

• Probability from the model

• Pre processing time

• Model inference time

Each request should create a unique JSON object in your payload S3 directory.



**4. REST API**

Following the example in class, create an API Gateway to expose your lambda function.

The gateway should implement a “/predict” resource with a “POST” request method.

Deploy it under a “v1” stage.

The following kind of request:

curl -X POST https://<your\_endpoint>/v1/predict --header

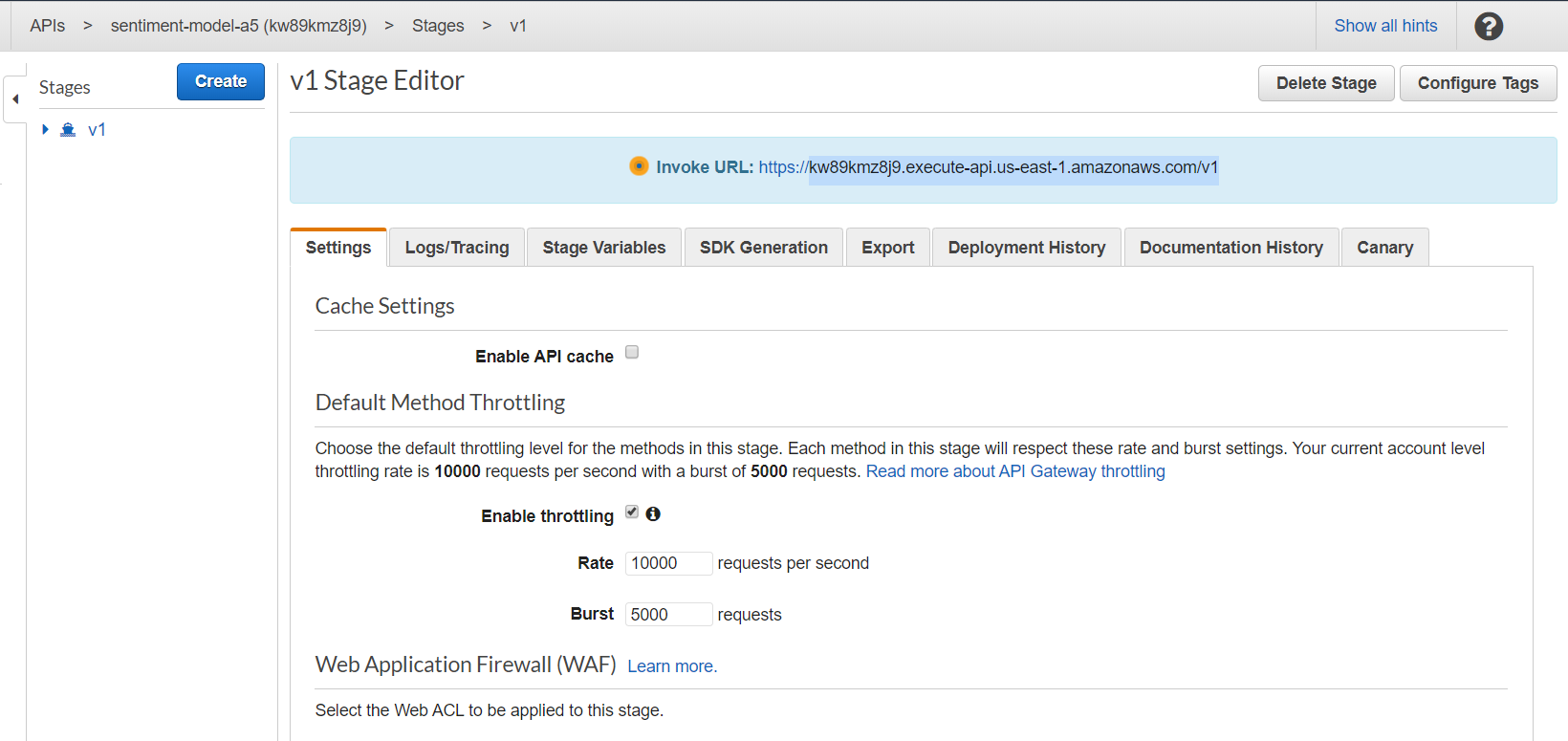
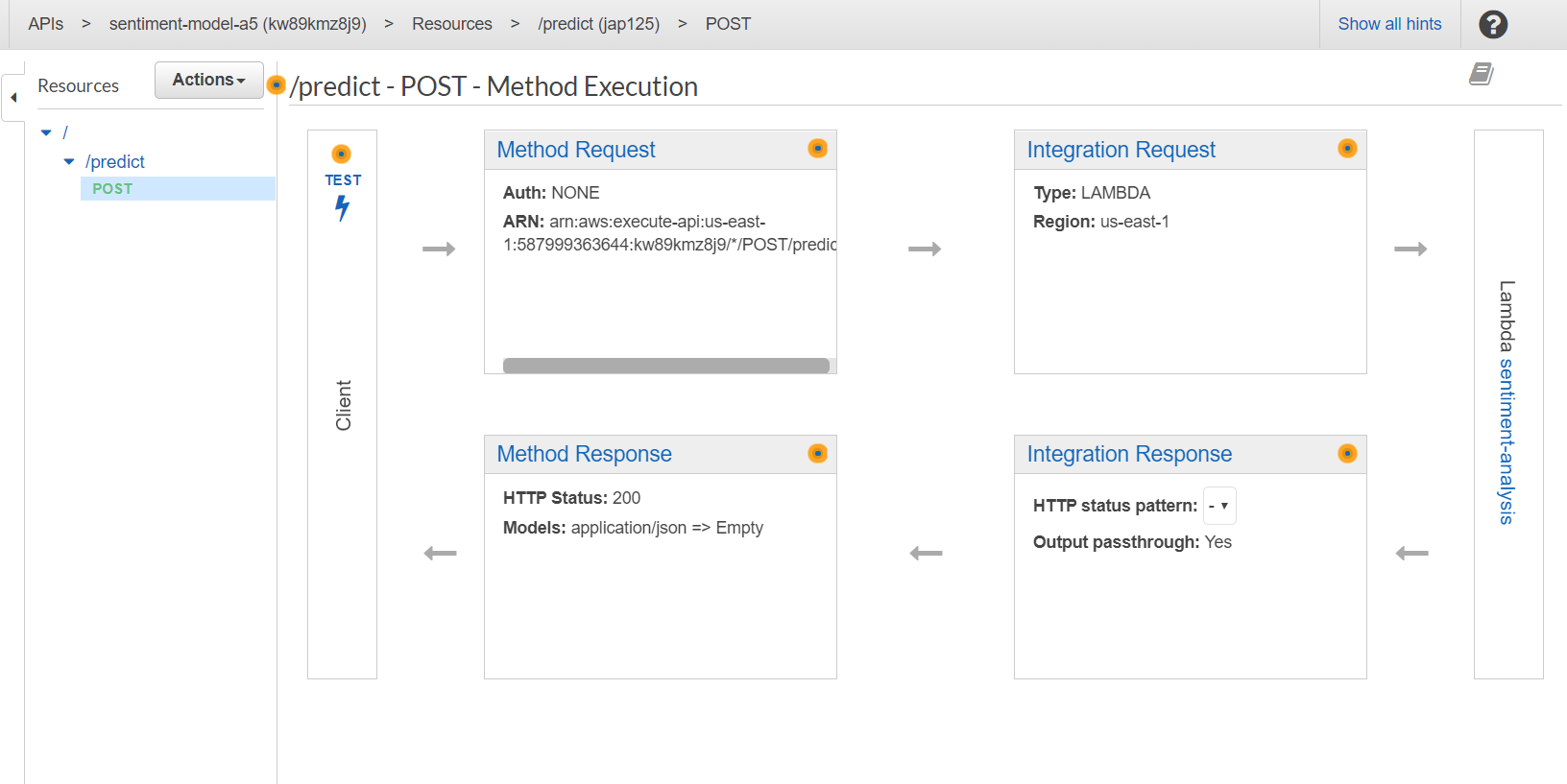
"Content-Type:application/json" --data '{"tweet": "I love

apple"}'

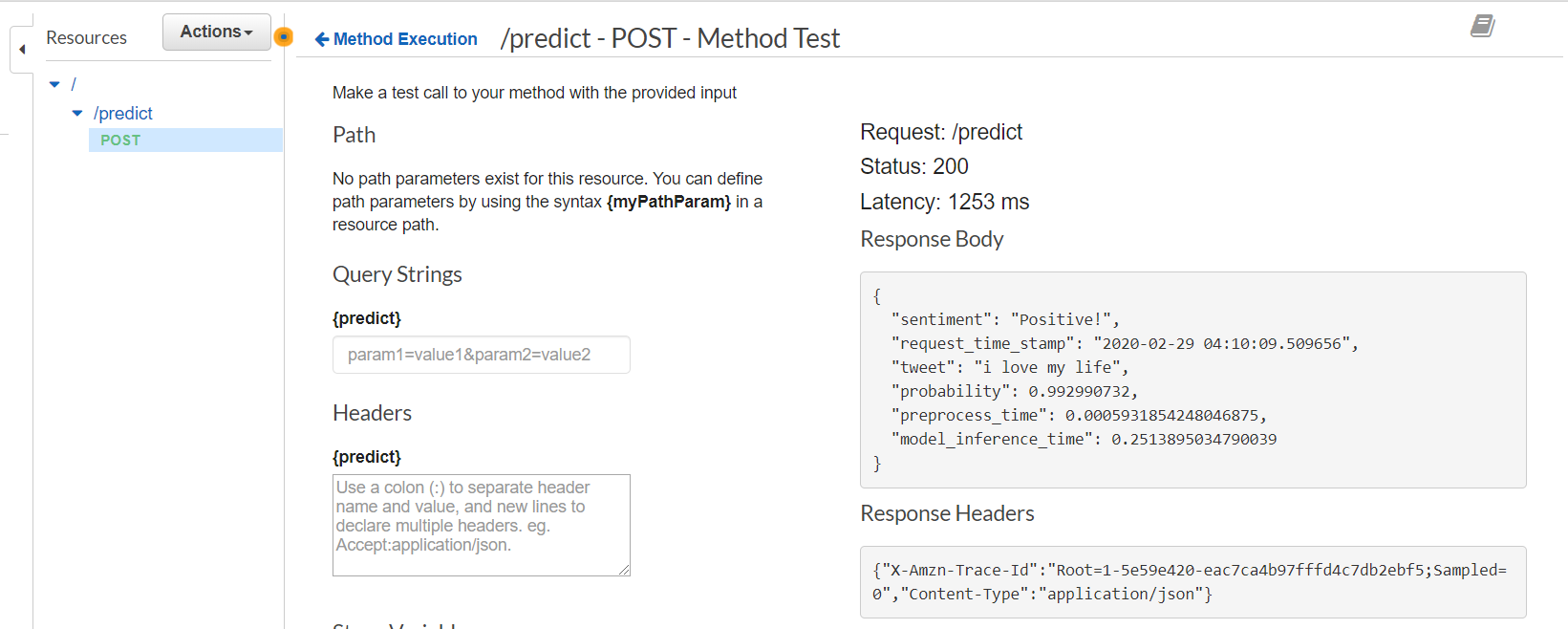
Should return something like:

{"sentiment": "positive"}

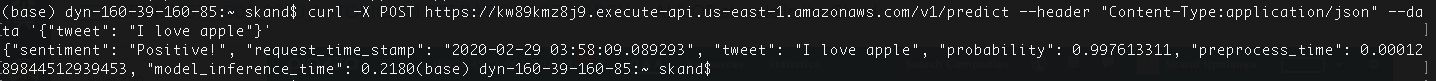
Creating API gateway predict resource:



Test API Gateway:



Test API Gateway from local:



API Gateway Link: <https://kw89kmz8j9.execute-api.us-east-1.amazonaws.com/v1/predict>

Github Link: <https://github.com/harsh1495/AI-Ops-A5>