DEVELOPMENT AND HOSTING OF ML MODELS USING S3, LAMBDA, EC2, AND API GATEWAY

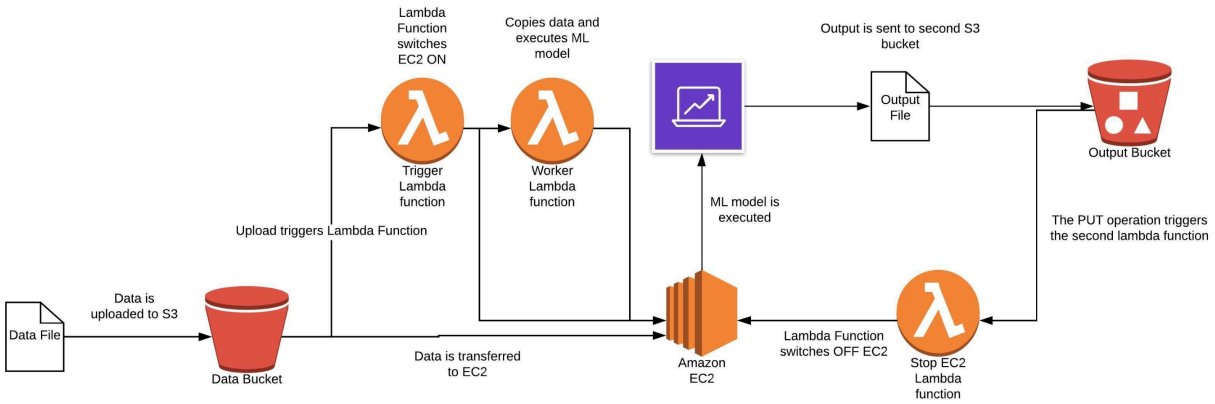
Name:Amit Gupta

Proposal: There are 2 parts to this proposal.

Part 1: An automated workflow to continuously train machine learning models when new data comes in. EC2 is turned on during training, and turned off when the learning has completed.

Part 2: Make the model accessible via the API gateway.

Part 1("Compute as Glue"): Visualization of automatic workflow. The below diagram has been borrowed from https://towardsdatascience.com/automating-machine-learning-models-on-aws-bfa183fe4065. The author has provided some starter code on github, but doesn't discuss how all the blocks are wired up, and has not provided the machine learning model used. My implementation will fill all these gaps.



Steps:

1) Data posted to input S3 will trigger a Lambda function ("Trigger Lambda") which will turn on the EC2 instance.
2) "Trigger Lambda" will invoke "Worker Lambda" that copies the data and code from S3 to EC2 instance
3) "EC2" instance executes the ML model and writes the model to output S3.
4) The output S3 triggers a Lambda function ("Stop Ec2 Lambda") to stop the EC2 instance.

Part 2 (“Compute as Backend”): Once the model is trained in Part1, now the challenge is to make it available to end users to access it and run predictions.

We will use the following 2 building blocks to achieve the Part 2 goals.

Steps:

1) Lambda function will host the trained machine learning model and return predictions. The lambda function will have “Lambda Layers” that will be zip files of the needed python packages and machine learning model python files.

2) The API gateway will provide a secure access via an API key and will have a query usage plan. The “predict” resource will be the lambda function, and we will use “POST” as an action.

3) Users will be able to get predictions using the REST API protocol specifying the API gateway end point and the “predict” resource. (api-end-point-url/production/predict/).