Deployment Plan

After you write your model training script and testing/ Inference, you can deploy them to automate the process using the following approach.

A) Automate the training process (with Amazon SageMaker Batch Transform):
when we need to train or retrain the model on a schedule or when some new data comes in then our architect will be batch transform.

1) Batch transform :

- The process will start with some event trigger and that event trigger can be something that gets uploaded to an s3 bucket or it can be a cron job that tells you the time of day/week/month you want to run this, all that you can specify with your cron job.
- This event trigger starts a lambda function, the lambda function will let you execute your code and connect with that S3 bucket. That s3 bucket will connect with Amazon SageMaker.
- Then SageMaker will start the batch transform job and run your model training script (written inside SageMaker or EC2) and then the newly trained model goes back to s3

2) How to make it happen:

- Set 2 event triggers.
- Load the data into an s3 bucket to initiate the batch transform.
- Now the 1st event is triggered, this will start the execution of the lambda function to start the SageMaker notebook instance and run your model training script.
- After the model training is completed, the newly trained model is saved inside the s3 bucket.
- As soon as the model is dropped inside s3, the 2nd event is triggered, and this will start the execution of another lambda function to turn off the SageMaker notebook instance. This step is not necessary but important in cost-cutting because it avoids the unnecessary running of SageMaker notebook instance which reduces the bill amount.

Note: 1. Here we can use both i.e., either SageMaker notebook instance or EC2 instance
2. Batch Transform (above pipeline) can also be used to automate the inference process.

B) Automate the Inference process (Deploy a Model on Amazon SageMaker Endpoint):
SageMaker endpoint is best for cases of online inferencing when you have data that is coming in from the internet and you need to serve prediction responses in real-time.

- The **endpoint** can be multiple ec2 instances. In the case of multiple ec2 instances, there are over different availability zones in order to increase your high availability when you specify model deploy in the sage maker Python SDK which a single line of code.
- Run that code, and this will manage your endpoints that are multiple ec2 instances.
- each ec2 instance has a web server and your model artifact and able to respond to requests so each instance is going to be serving prediction responses that are handled by a load balancer.
- The load balancer is taking care of health checks making sure that your instances are up. All of those instances are sitting behind your model endpoint.
- Your model endpoint is a restful API. SageMaker is automatically creating a restful API for your specific model for every model that you deploy on SageMaker.
- Here we can use a lambda function to connect between two points i.e., your API gateway and AWS services.

Note: Turn off the endpoints when not in use using the lambda function for cost-cutting