Write UP

1. Endpoint name deployed = pytorch-inference-2022-01-25-20-17-35-400
2. **2. Write a justification of why you choose your type of EC2?**

The EC2 instance chosen was m5. large with EBS only storage, 2GB vCPU, 8GB memory and up to 10GB EBS optimized available. It has an amazon machine image of AWS Deep learning (Linux 2). This AMI was chosen because it has sufficient memory and CPU capacity for the training Job performed and to ensure cutting down unnecessary cost since the task is a moderate one.

3. **Write about the EC2 code and how it compares to code from step 1?**

The EC2 code was ran using m5.large instance with an EBS storage optimizable to 10GB. The model training was done via the command line. The EC2 uses command line interface and has more technicalities than sagemaker. Sagemaker's interface is more friendly and easier to use. With EC2 one has to be very careful while inputting on the command line due to the technicalities involved.

EC2 has a slower startup compared to sagemaker although it is less expensive.

4. **Write about your Lambda function in your final writeup**

Answer 3: The lambda functions enables a model and its inferences to be accessed by API's and other programs, so it's a crucial part of production. The lambda function created in the project was used to invoke an endpoint built on the trained model to perform a test. The function script from lambda_function.py was copied to code source tab and deployed. Thereafter a test was carried out and it returned a status code of 200 which implied it was successful as seen below.

Test Event Name
test-123

Response


```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "COntent-Type-In": "<__main__.LambdaContext object at 0x7f0b4fda44a8>",
  "body": "[[-4.4106926918029785, -3.5302228927612305, 0.32907092571258545, -0.13970650732517242, -
1.4143508672714233, -3.2415733337402344, -0.5279808640480042, 1.0270906686782837, -3.9141042232513428, -
0.8055006265640259, -1.490047812461853, -0.908812940120697, 0.028274334967136383, 1.9345248937606812, -
2.8035812377929688, -0.8355246782302856, -3.1118252277374268, -1.3321032524108887, -2.7802841663360596,
2.009772539138794, -1.320298194885254, 1.6258431673049927, -2.886414051055908, -0.9023059606552124, -
3.895293951034546, -6.250190258026123, -2.967088460922241, -2.345414161682129, -0.1939409077167511, -
0.9348788261413574, -2.160616636276245, -1.5843970775604248, -5.257950305938721, -1.075924038887024, -
1.370546817779541, -3.33560848236084, -2.762471914291382, -1.5191233158111572, 0.053386081010103226, -
2.15029239654541, -1.786787986755371, -0.6009842157363892, 1.377799153327942, -2.2184433937072754,
1.0787429809570312, -4.905442714691162, -1.155449390411377, 0.16727134585380554, -2.8572463989257812, -
0.8475739359855652, -2.7476534843444824, -1.6223269701004028, -4.335642337799072, -1.151687741279602, -
4.476847171783447, -2.306262969970703, -2.130222797393799, -3.2529916763305664, -1.1515498161315918, -
0.6160231232643127, -3.841362953186035, -2.8342039585113525, -3.2431254386901855, -5.888490676879883, -
```

2.2695233821868896, -5.742074966430664, 0.41386380791664124, -3.5433013439178467, 0.009719409048557281, -0.9010305404663086, 0.6887349486351013, -4.08201265335083, -2.64021372795105, -6.072409629821777, -1.2267177104949951, -1.2024897336959839, -2.9783072471618652, -0.1823846846818924, -2.5847110748291016, -2.221667766571045, 0.5917548537254333, -4.069221496582031, 0.8626347780227661, -1.2698026895523071, -2.6204395294189453, -4.323845863342285, 1.1156407594680786, -7.1868767738342285, -0.2213197648525238, -0.42465370893478394, -3.8655755519866943, -2.6154978275299072, -3.635685920715332, -5.108319282531738, -3.243997097015381, 0.005724083632230759, -3.6497442722320557, -0.22129812836647034, -5.058711528778076, -2.7986159324645996, -2.595886707305908, -0.4035526216030121, -1.6680855751037598, -2.813999891281128, -3.7970163822174072, -5.104593753814697, -1.6005299091339111, -0.5218651294708252, -0.2849648594856262, 1.195355772972107, -0.4082486033439636, 0.3989885449409485, -3.020618438720703, -3.8559648990631104, 0.10875650495290756, 2.2255947589874268, -2.09409236907959, 0.1753983348608017, -2.4133806228637695, 1.3981733322143555, -0.15331529080867767, -2.0865538120269775, -2.990349054336548, -4.692201137542725, -5.727423191070557, -2.116166830062866, -0.4005201756954193, -1.182114601135254, -4.613989353179932, -3.6380248069763184, -3.732269525527954, -2.092519760131836, -2.950108766555786]]"
}

5. **Write about whether you think your AWS workspace is secure and whether you think there are any vulnerabilities**

   The security regarding operations in AWS workspace is dependent on the users. AWS has provided enough security systems in the cloud for user to employ it's use.
   Access are not just granted on it's own, it has to be permitted in IAM using the role of the services that requires a more priviledged access. There are different policies that spells us the necessary rights. E.g fullaccess or specific access

6. **Write about your configuration of concurrency and auto-scaling**

   In order not to use up the unreserved concurrence type completely, I set a reserved concurrency type of 400 for the lambda function created to ensure enough a wider gap for execution . The target value for the auto scaling configuration was set to 80, so that once the cpu utilization has gone up to 80%, the server machines/cpu can scale up to meet increased demand.