Initial Setup

I decided to create an instance of type ml.t2.medium. Considering the cost, computing power, and speed of launching for each instance type it seemed like the most inexpensive. Since this is a project time if not an issue so my main priority was to minimize cost and ml.t2.medium seems to have enough computing power for the training job and deployment.

EC2 Setup

I select an AMI for your EC2 instance "Amazon Deep Learning AMI" so that I can use its libraries. Considering the cost, computing power, and speed of launching for each instance type it seemed like the most inexpensive. Since this is a project time if not an issue so my main priority was to minimize cost and t2.micro seems to have enough computing power for the training job and deployment.

Write about the EC2 code

The code I worked within Step 1,train_and_deploy-solution.ipynb, was specifically written to work in Sagemaker. For example, the Sagemaker python package allows us to deploy models that were saved in s3 using the model location variable. To train the same model on an EC2 instance, we used the adjusted code in ec2train1.py. So that we could upload the data locally using PyTorch's data loader and the os python package to locate the data in the local directory. The adjusted code in ec2train1.py is very similar to the code in train_and_deploy-solution.ipynb and hpo.py. But there are a few differences between the modules used - some modules can only be used in SageMaker. Much of the EC2 training code has also been adapted from the functions defined in the hpo.py starter script.

Setting up a Lambda function

The endpoint_name variable can be found in the "Inference > Endpoints" section of Sagemaker. InvokeEndpoint is a service from Amazon SageMaker Runtime. Note that you can invoke the endpoint directly using the invoke_endpoint() API, passing the endpoint name, the content type, and the payload. After you deploy a model into production using Amazon SageMaker hosting services, your client applications use this API to get inferences from the model hosted at the specified endpoint. Finally, the return value is a message that contains data from the event is received as input.

Return value

[[-8.827474594116211, -5.315217018127441, -2.679171562194824, -2.224059581756592,
-8.012042999267578, -7.082286357879639,
-2.7181594371795654, -4.142338275909424, -5.428321838378906, -2.010838508605957,
-0.04754798859357834, -4.79606819152832,
-4.357832431793213, 0.6782151460647583, -6.203027725219727, -6.342438697814941,
-10.945418357849121, -3.3885092735290527,
-4.914143085479736, 0.6404876708984375, -2.828984260559082, 0.26214462518692017,
-8.489106178283691, -6.780303001403809,
-5.9718475341796875, -13.093317031860352, -5.9652180671691895, -6.049808025360107,
-5.262589454650879, -3.7049248218536377,

-3.3212335109710693, -5.331852436065674, -8.226842880249023, -6.198317527770996,
-7.514371871948242, -10.509113311767578,
-6.895763874053955, -3.931474208831787, -3.3469533920288086, -6.706569671630859,
-3.9635751247406006, -7.024862289428711,
-0.07414942979812622, -4.264065265655518, -3.816535711288452, -10.30774974822998,
-2.630918025970459, -2.759493112564087,
-2.2937357425689697, -4.658539772033691, -4.682111740112305, -10.929527282714844,
-8.06988525390625, -7.820631504058838,
-7.401383876800537, -0.43766000866889954, -6.064032554626465, -6.9791059494018555,
-3.9207050800323486, -3.1747307777404785,
-6.377753734588623, -7.651472091674805, -8.05556869506836, -10.289271354675293,
-5.524887561798096, -9.332978248596191,
-1.4454807043075562, -3.685203790664673, -3.1378536224365234, -1.5135645866394043,
-0.9677587151527405, -5.030588626861572,
-5.2426629066467285, -7.746395111083984, -6.42454195022583, -5.5126447677612305,
-8.413322448730469, -2.715717077255249,
-9.795592308044434, -3.44254732131958, -0.6094019412994385, -8.094404220581055,
-0.18943068385124207, -3.2422235012054443,
-7.136514186859131, -4.103004455566406, -2.975022792816162, -5.085076808929443,
-3.8406848907470703, -2.8601725101470947,
-9.486451148986816, -6.364745140075684, -6.801111698150635, -7.015524387359619,
-6.769260883331299, -3.2887113094329834,
-6.933145523071289, -5.284571170806885, -7.844706058502197, -8.488814353942871,
-10.949792861938477, -2.0617775917053223,
-5.24845027923584, -7.220120429992676, -8.459930419921875, -7.099983215332031,
-4.450875759124756, -1.5618680715560913,
-2.440831184387207, -2.58955979347229, -2.381957530975342, -3.2262468338012695,
-8.432437896728516, -8.009161949157715,
-7.912553787231445, -2.3951950073242188, -6.87275505065918, -1.5069741010665894,
-7.6189045906066895, -0.29479655623435974,
-3.2912397384643555, -4.502758502960205, -8.972395896911621, -5.0272440910339355,
-7.618856430053711, -7.297399520874023,
-6.193563461303711, -0.7575976252555847, -5.8397064208984375, -8.79106616973877,
-8.613133430480957, -2.079134941101074,
-8.879327774047852]]

Test result

Test Event Name
Event-project

Response

```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "COntent-Type-In": "<__main__.LambdaContext object at 0x7f131ed6db20>",
  "body": "[[-8.827474594116211, -5.315217018127441, -2.679171562194824,
-2.224059581756592, -8.012042999267578, -7.082286357879639, -2.7181594371795654,
-4.142338275909424, -5.428321838378906, -2.010838508605957, -0.04754798859357834,
```

-4.79606819152832, -4.357832431793213, 0.6782151460647583, -6.203027725219727,
-6.342438697814941, -10.945418357849121, -3.3885092735290527, -4.914143085479736,
0.6404876708984375, -2.828984260559082, 0.26214462518692017, -8.489106178283691,
-6.780303001403809, -5.9718475341796875, -13.093317031860352, -5.9652180671691895,
-6.049808025360107, -5.262589454650879, -3.7049248218536377, -3.3212335109710693,
-5.331852436065674, -8.226842880249023, -6.198317527770996, -7.514371871948242,
-10.509113311767578, -6.895763874053955, -3.931474208831787, -3.3469533920288086,
-6.706569671630859, -3.9635751247406006, -7.024862289428711, -0.07414942979812622,
-4.264065265655518, -3.816535711288452, -10.30774974822998, -2.630918025970459,
-2.759493112564087, -2.2937357425689697, -4.658539772033691, -4.682111740112305,
-10.929527282714844, -8.06988525390625, -7.820631504058838, -7.401383876800537,
-0.43766000866889954, -6.064032554626465, -6.9791059494018555, -3.9207050800323486,
-3.1747307777404785, -6.377753734588623, -7.651472091674805, -8.05556869506836,
-10.289271354675293, -5.524887561798096, -9.332978248596191, -1.4454807043075562,
-3.685203790664673, -3.1378536224365234, -1.5135645866394043, -0.9677587151527405,
-5.030588626861572, -5.2426629066467285, -7.746395111083984, -6.42454195022583,
-5.5126447677612305, -8.413322448730469, -2.715717077255249, -9.795592308044434,
-3.44254732131958, -0.6094019412994385, -8.094404220581055, -0.18943068385124207,
-3.2422235012054443, -7.136514186859131, -4.103004455566406, -2.975022792816162,
-5.085076808929443, -3.8406848907470703, -2.8601725101470947, -9.486451148986816,
-6.364745140075684, -6.801111698150635, -7.015524387359619, -6.769260883331299,
-3.2887113094329834, -6.933145523071289, -5.284571170806885, -7.844706058502197,
-8.488814353942871, -10.949792861938477, -2.0617775917053223, -5.24845027923584,
-7.220120429992676, -8.459930419921875, -7.099983215332031, -4.450875759124756,
-1.5618680715560913, -2.440831184387207, -2.58955979347229, -2.381957530975342,
-3.2262468338012695, -8.432437896728516, -8.009161949157715, -7.912553787231445,
-2.3951950073242188, -6.87275505065918, -1.5069741010665894, -7.6189045906066895,
-0.29479655623435974, -3.2912397384643555, -4.502758502960205, -8.972395896911621,
-5.0272440910339355, -7.618856430053711, -7.297399520874023, -6.193563461303711,
-0.7575976252555847, -5.8397064208984375, -8.79106616973877, -8.613133430480957,
-2.079134941101074, -8.879327774047852]]"
}

Function Logs
START RequestId: 640ad78a-0d5c-425f-94e3-c1e149825149 Version: $LATEST
Context::: <__main__.LambdaContext object at 0x7f131ed6db20>
EventType:: <class 'dict'>
END RequestId: 640ad78a-0d5c-425f-94e3-c1e149825149
REPORT RequestId: 640ad78a-0d5c-425f-94e3-c1e149825149 Duration: 984.39 ms Billed Duration: 985 ms Memory Size: 128 MB Max Memory Used: 69 MB

Request ID
640ad78a-0d5c-425f-94e3-c1e149825149

Lambda function security

Note that roles that have "FullAccess" policies attached may be too permissive and may lead to problems. Roles that are old or inactive may lead to vulnerabilities because they may belong to people who are no longer working on the project and who may not be careful about ensuring the project's success. You might also find roles for functions that the project is no longer using, which should probably be deleted.

Concurrency and auto-scaling

I use 5 for your reserved concurrency, and 3 for your provisioned concurrency. Remember that your provisioned concurrency always needs to be less than your reserved concurrency. Choosing a high number for provisioned concurrency is suitable for very high-traffic projects because it will turn on instances that can be used by my Lambda function at any time. However, provisioned concurrency is expensive, so I wanted to keep this number low. Choosing lower numbers for either or both types of concurrency would be more suitable for lower-traffic projects such as this one.