Step1:

1. Write a short justification of why you chose the instance type you did.

Ans: The chosen type of notebook instance is 'ml.t3.medium', which provides a balance between computational power and cost-effectiveness.

Step2:

1. Write a justification of why you chose the instance type you did.

Ans: The chosen type of ec2 instance is ‘t2.micro’ with ‘spot instance’ to reduce costs for this non-time-sensitive exercise.

1. Write at least one paragraph about the differences between the code in ec2train1.py and the code you used in Step 1.

Ans: The script ec2train1.py contains functions that are defined in the hpo.py starter script. Although ec2train1.py is a Python script rather than a Jupyter notebook like in Step1, making it slightly less readable, it primarily utilizes PyTorch and other widely-used Python packages. This enhances its versatility for applications that do not rely on AWS.

On the other hand, the code in Step 1 streamlines access to numerous AWS services, including S3, training jobs, hyperparameter tuning jobs, endpoints, and inference capabilities. In essence, the ec2 approach provides developers with increased flexibility and transferability, while the Step1 approach offers a more convenient development path for machine learning projects that need to run on AWS.

Step3:

1. Write at least 1 paragraph describing how this function is written and how it works.

Ans: This function, lambda\_handler(event, context), is an AWS Lambda function written in Python that serves as an interface for invoking a SageMaker endpoint. It accepts an input event and context object, processes the event data, and sends it to the SageMaker endpoint for inference. The endpoint returns a result, which the function then formats and returns as a JSON response.

When the Lambda function is triggered, it first prints the context and event type for debugging purposes. It then initializes a SageMaker Runtime client, which is used to interact with the SageMaker endpoint. The endpoint name is defined earlier in the script as endpoint\_Name.

The function proceeds to invoke the SageMaker endpoint using the invoke\_endpoint method, passing the necessary parameters such as EndpointName, ContentType, Accept, and Body. The ContentType and Accept parameters specify that the input and output data are in JSON format. The Body parameter contains the input event data, which is converted into a JSON string using json.dumps(bs).

Upon receiving the response from the SageMaker endpoint, the function extracts the result from the Body field, decodes it from UTF-8, and loads it into a JSON object named sss. Finally, the function constructs and returns a dictionary containing the response's status code, headers, content type, and body, which includes the JSON-formatted inference result.

Step4:

1. Write at least 1 paragraph describing how this function is written and how it works.

Ans: Response

{

"statusCode": 200,

"headers": {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

},

"type-result": "<class 'str'>",

"COntent-Type-In": "<\_\_main\_\_.LambdaContext object at 0x7ffb54cd5b50>",

"body": "[[-4.915011405944824, -2.197737216949463, -3.5556468963623047, -0.24649950861930847, -5.9260783195495605, -4.401652812957764, -2.0319101810455322, -2.212007761001587, -3.1056225299835205, -1.0332725048065186, 0.3646068274974823, -3.2227976322174072, -2.9902937412261963, -0.3180422782897949, -3.570253849029541, -0.298353374004364, -2.4708292484283447, -2.0397722721099854, -1.2035408020019531, -3.2853050231933594, -3.4228086471557617, -3.198397636413574, -7.84968900680542, -3.3118298053741455, -3.8887388706207275, -3.840667486190796, -3.7362899780273438, -2.2427568435668945, -5.2785115242004395, -2.490705966949463, -2.6749937534332275, -5.789021968841553, -1.0231953859329224, -2.840050220489502, -5.10482120513916, -2.8297760486602783, -2.6188888549804688, -4.994965076446533, -0.981313169002533, -2.0196168422698975, -2.672938585281372, -2.7775468826293945, -0.18550577759742737, -2.1512608528137207, -1.7047051191329956, -7.1473541259765625, -1.621359944343567, -1.4228898286819458, -2.2380738258361816, -1.3564914464950562, -3.5988612174987793, -6.5004353523254395, -4.902629375457764, -1.5115056037902832, -3.56086802482605, -1.4489376544952393, -7.148298263549805, -5.100677013397217, -2.2645492553710938, -2.783297061920166, -5.456346035003662, -4.656990051269531, -5.530662536621094, -6.964460849761963, -3.746718168258667, -3.8927884101867676, 0.8492692708969116, -6.814851760864258, -3.544127941131592, -2.7454819679260254, 0.5358855128288269, -3.2353174686431885, -2.80914044380188, -3.442049026489258, -2.600576639175415, -3.2324559688568115, -2.213721513748169, -2.9604969024658203, -4.61943244934082, -2.858325242996216, -1.0430690050125122, -4.748679161071777, -1.8204225301742554, -0.40656137466430664, -4.983902454376221, -6.527281761169434, -1.2874844074249268, -3.559001922607422, -0.9316497445106506, -3.07651686668396, -6.045462131500244, -4.585297107696533, -5.904694557189941, -0.6258265972137451, -2.2173542976379395, -1.0649325847625732, -1.9005670547485352, -2.569733142852783, -4.702241897583008, -4.5916900634765625, -5.511369705200195, -3.055222511291504, -5.803351402282715, -4.022275924682617, -3.5061473846435547, -5.149112224578857, -3.867997646331787, -2.1442720890045166, -2.2105531692504883, -1.9344937801361084, -2.04313588142395, -3.199120044708252, -2.4205374717712402, -4.544490814208984, -4.862744331359863, -1.4813787937164307, -5.578724384307861, -3.114680528640747, -1.978859782218933, -0.9581792950630188, -0.5196277499198914, -3.5855765342712402, -3.4046640396118164, -1.3520874977111816, -3.9899556636810303, -4.355140209197998, -4.000913619995117, -0.7444117665290833, -3.6996448040008545, -7.552887439727783, -2.1805167198181152, -2.90285587310791, -4.851222515106201]]"

}

1. Write at least 1 paragraph describing how this function is written and how it works.

Ans: Since the Lambda Function is granted with “AmazonSageMakerFullAccess”, following are some of the concerns:

1. Unauthorized access: The policy grants access to all SageMaker resources, which may lead to unauthorized access to sensitive data and resources if the Lambda function is compromised.
2. Accidental modification or deletion: With full access to SageMaker resources, the Lambda function could inadvertently modify or delete critical resources, such as endpoints, training jobs, or models, which could disrupt the operation of your machine learning workflows.
3. Data leakage: Full access to SageMaker resources means the Lambda function can read and write data to and from S3 buckets and other storage services. This increases the risk of data leakage, either accidentally or through a malicious actor gaining control of the function.
4. Privilege escalation: Granting the "AmazonSageMakerFullAccess" policy to a Lambda function may allow an attacker to escalate privileges by creating new resources or modifying existing ones, such as creating new SageMaker notebook instances with elevated permissions.
5. Increased attack surface: By allowing a Lambda function to access and manage all SageMaker resources, you increase the attack surface that an attacker could exploit.

To mitigate these risks, it is recommended to follow the principle of least privilege when granting permissions to AWS resources. This means assigning only the minimum necessary permissions to the Lambda function to accomplish its intended purpose. Consider creating custom IAM policies that provide access to specific SageMaker resources and actions, instead of using the "AmazonSageMakerFullAccess" managed policy.

Step5

1. Write about the choices you made in the setup of concurrency and auto-scaling, and why you made each of those choices.

Ans: I have set up two reserved and two provisioned concurrency for the Lambda function, as a means of exercise. To ensure compatibility, I have also defined the maximum instance count for autoscaling the endpoint as two. Additionally, I have configured the "Target value", "Scale in cool down", and "Scale out cool down" as 30. While these values have been set to showcase my ability to make adjustments, it is important to note that the actual justification for these values should be determined based on usage data collected in future case studies.