How to choose SageMaker instance type? Which type?

ml.m5.4xlarge. It is to ensure that it can support data size. It is also recommended to use ml.t3.medium. This is the default instance type for CPU-based SageMaker images, and is available as part of the AWS Free Tier.

Which type is used for EC2 instance? Why?

t2.micro Linux/UNIX Spot Instance-hour in US East (N. Virginia) in VPC Zone #12

1.281 Hrs

It is available as part of AWS Free Tier.

Differences between the code in ec2train1.py and the code you used in Step 1?

Delete all sagemaker dependencies.

Include torch dependencies.

No creation of estimator with specific resource type and count details.

Hpo.py is no more an entry input in ec2train1.py

How lambda is written and how it works?

Lambda is Amazon’s serverless compute service. Lambda can execute code without specifying underlying infrastructure, like hardware specifications, the operating system, or the maintenance of standard libraries. This service is ideal for small tasks that are frequently repeated.

Lambda functions can be developed in Python through a Python package called Boto3. Boto3 is the AWS SDK to interact and manage AWS services, including Lambda.

It starts by executing the handler function. It invokes the endpoint, at the same time, accepts a json file as input, already got from event parameter. We test a Lambda function through the console via a test event.

Is my AWS workspace secure or there are some vulnerabilities?

Yes, it is secure. This is ensured by enabling a secure access to different permissions. This is by enabling or not one policy to a given role. However, it is recommended to do a careful security audit regularly, every month or more, to make sure that the account is safe.

What kind of concurrency is used? How I set up the auto-scaling? Why?

Concurrency enables Lambda functions to handle a larger number of simultaneous requests.

I chosed the provisioned concurrency. This argued by the fact that when the number of open requests increases, Application Auto Scaling increases provisioned concurrency in large steps until it reaches the configured maximum. The function continues to scale on standard concurrency until utilization starts to drop. When utilization is consistently low, Application Auto Scaling decreases provisioned concurrency in smaller periodic steps.