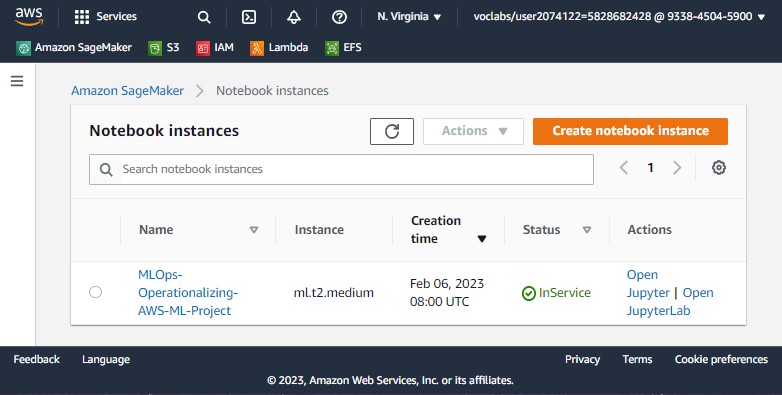
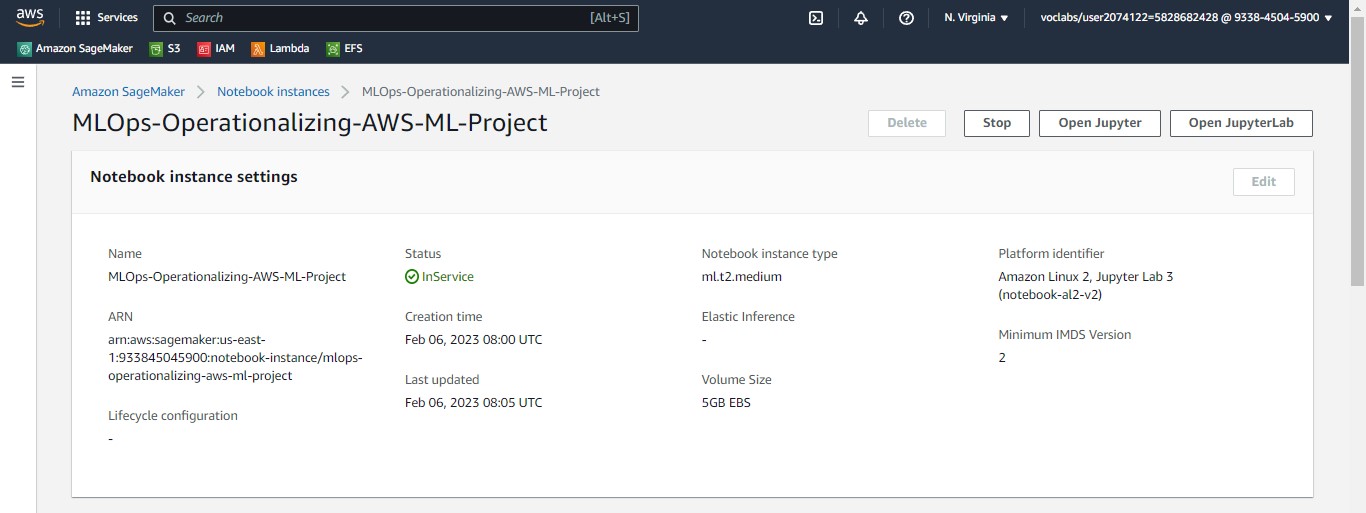
**Operationalizing an AWS ML Project**

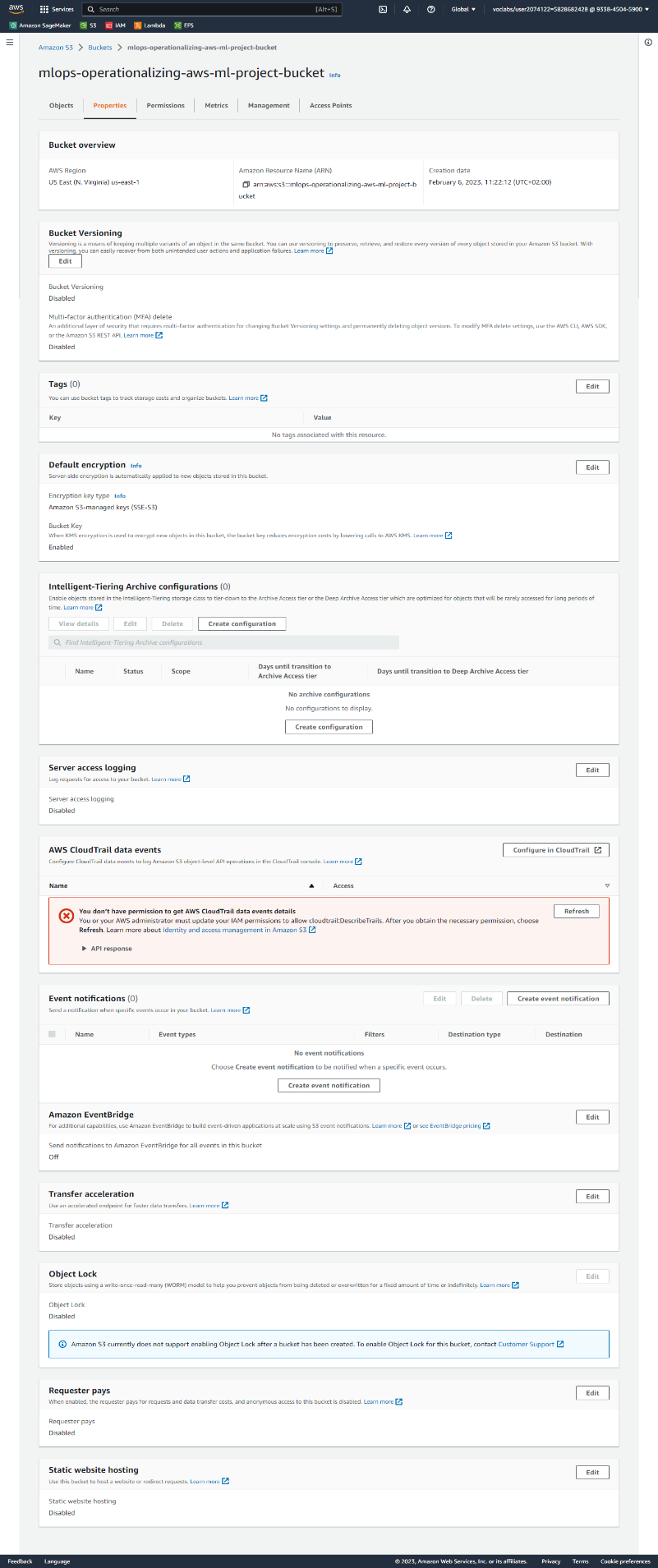
# Step 1: AWS Sagemaker

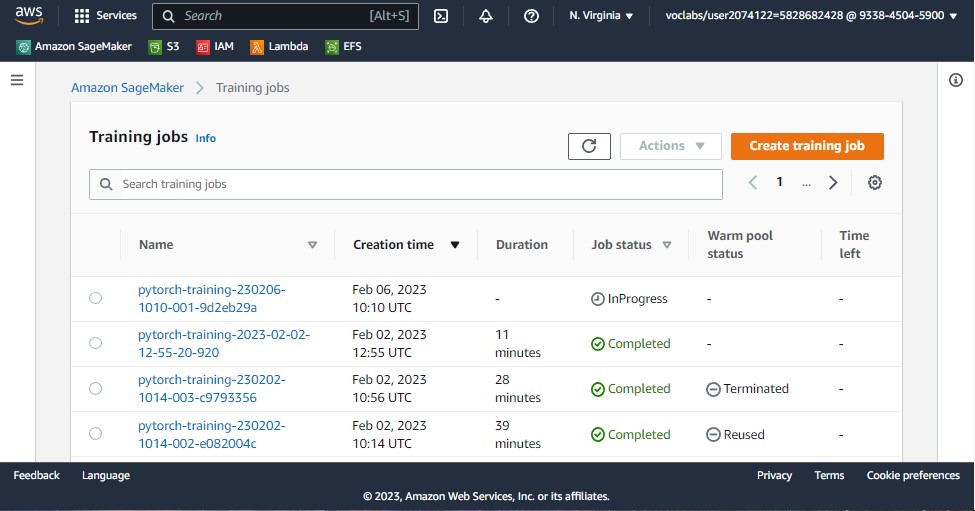
**Write about the Sagemaker instance you created, including a justification of why you chose the instance type you did.**

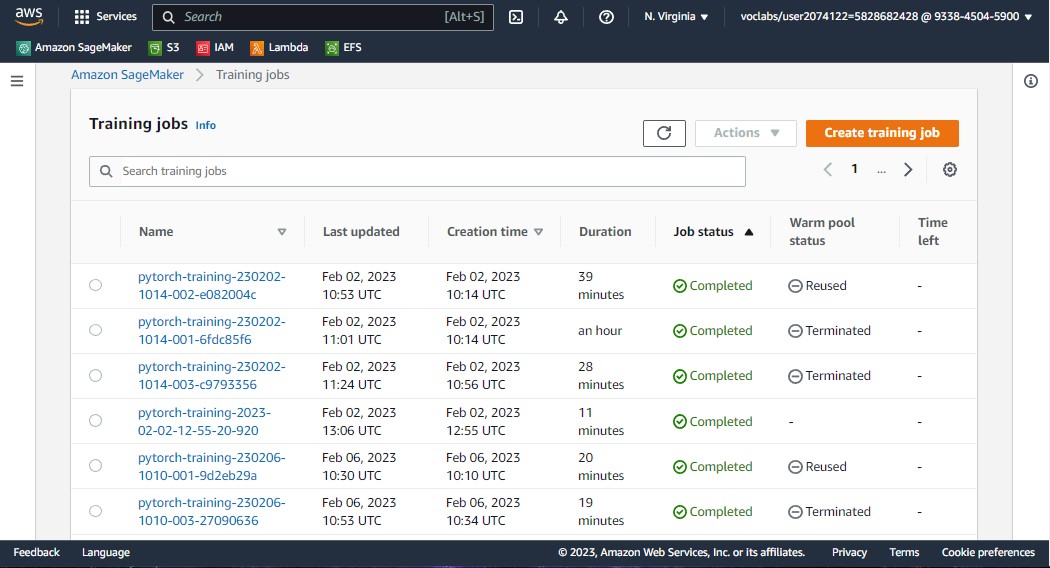
* Chosen [ml.t2.medium]:
  + Cost saving.
  + The size of my project doesn’t exceed 5 GB EBS.
  + It comes with specs 2 vCPUs, 4GB main memory, no GPU.
  + Amazon Linux 2, Jupyter Lab 3(notebook-al2-v2)
  + Estimated training time was about 20 mins which is reasonable.
  + Fast launch.

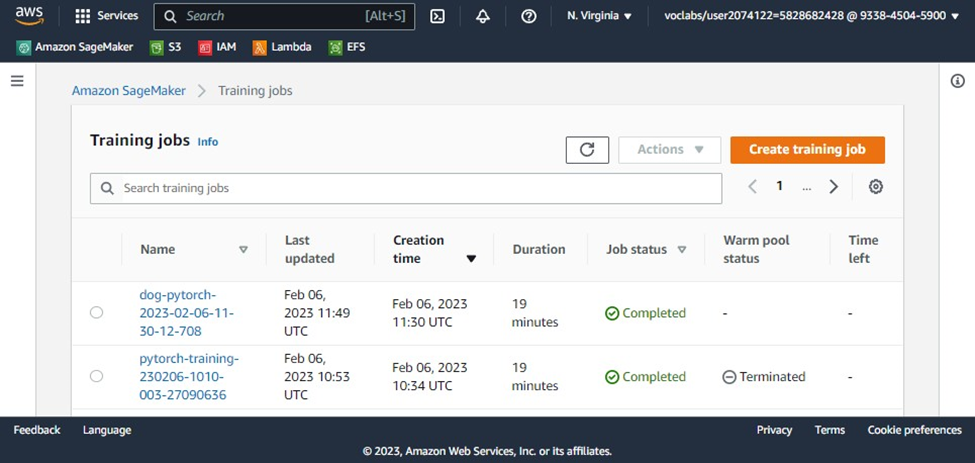


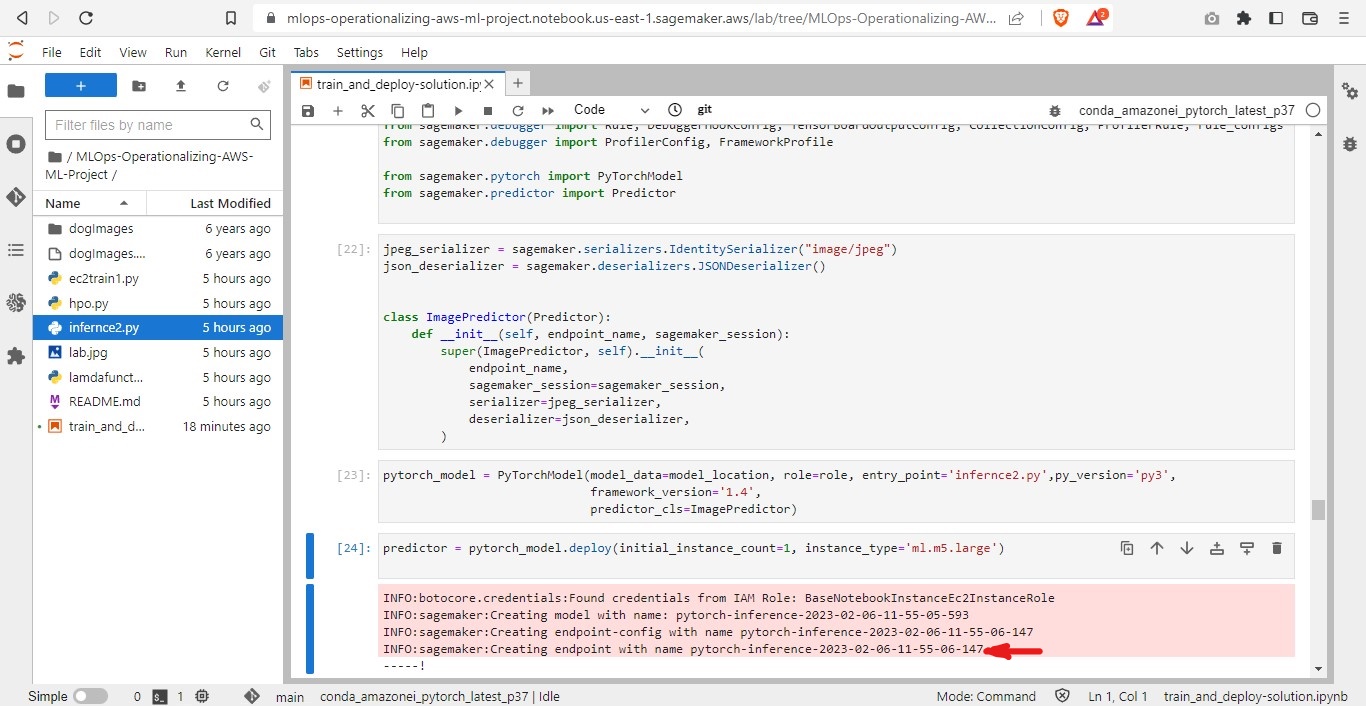


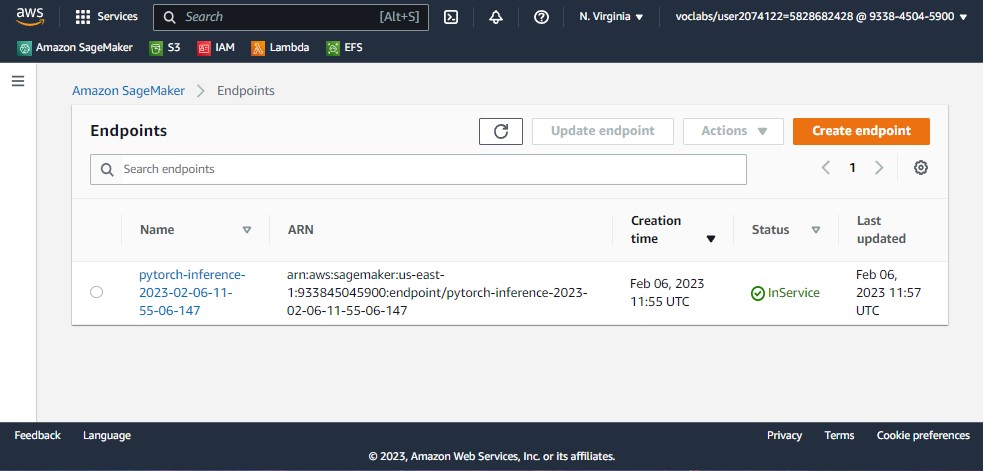
**S3 Bucket:**

**Training Job (in Progress):**

**Training Job (Completed):**

**Training Job (Multi-Instance):**

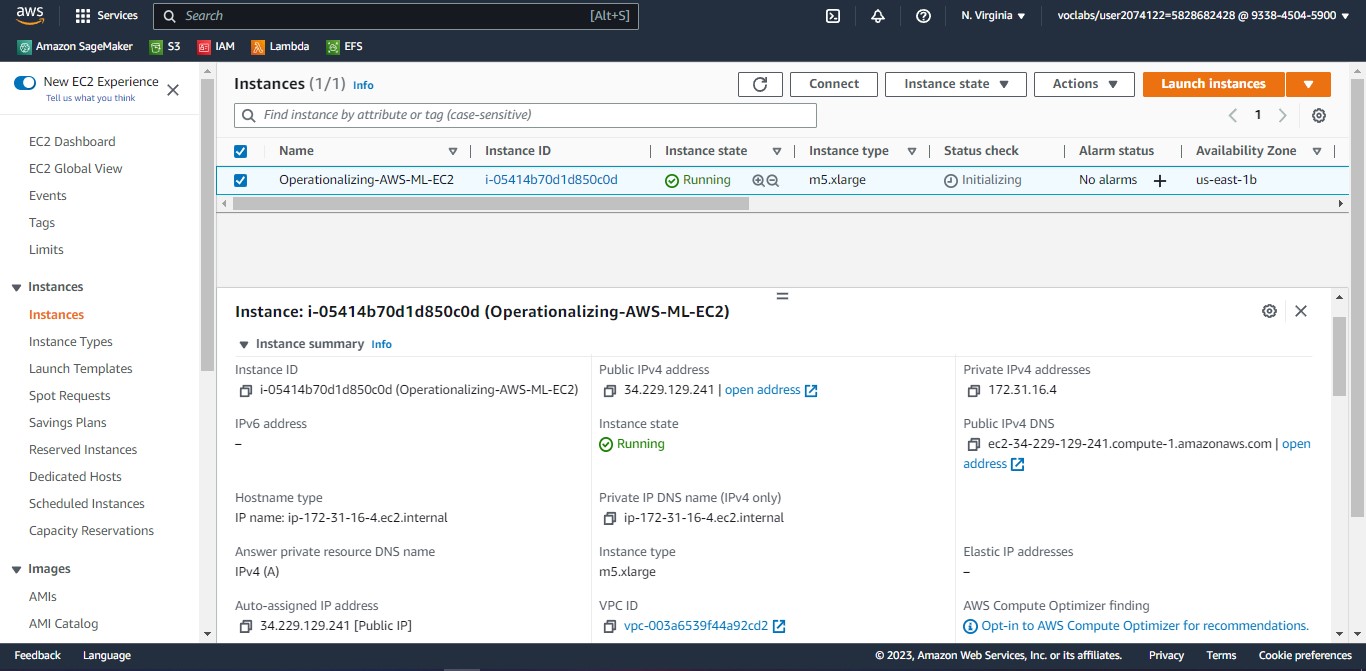
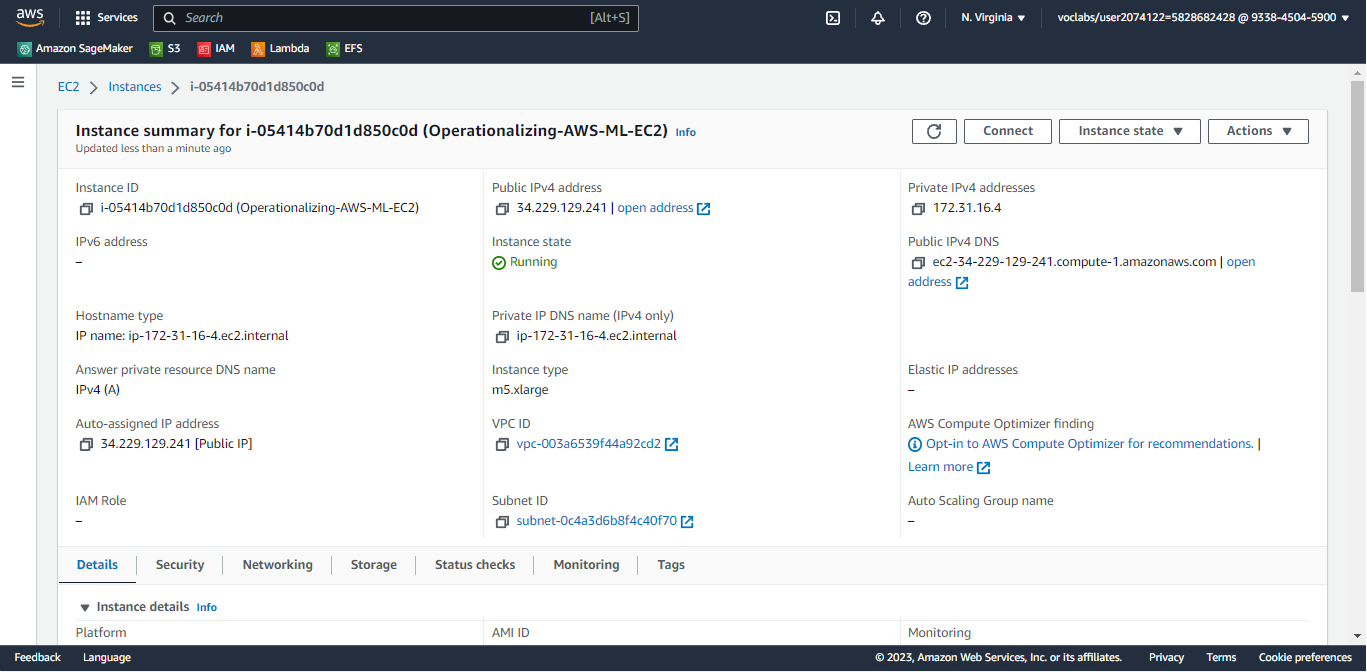
**Endpoint deployment:**



# Step 2: AWS EC2 Workspace

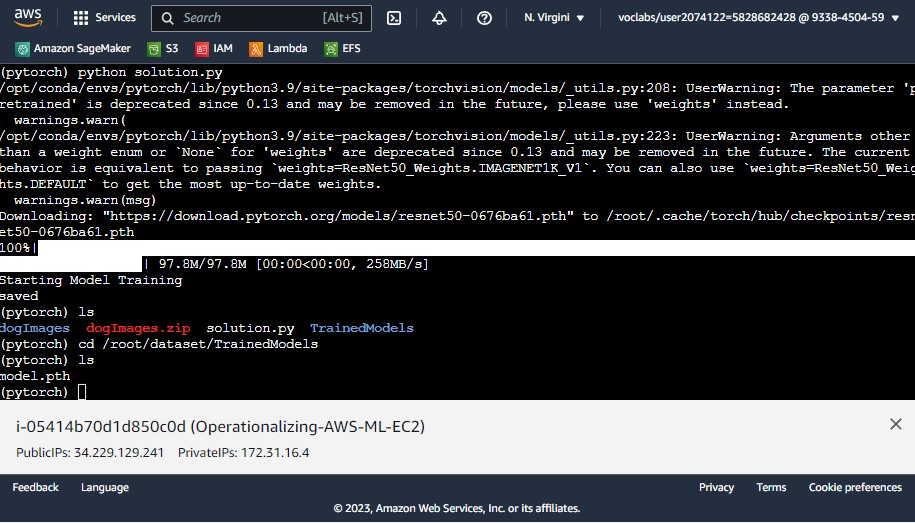
**Write about the EC2 you created, write a justification of why you chose the instance type**.

* Storage Instance [m5.xlarge] with 4 vCPU and 16 Memory (GiB).
* EBS-Only Instance Storage(GB) with Up to 10 Network Bandwidth (Gbps) and Up to 4,750 EBS Bandwidth (Mbps).
* Suited for computer vision classification workloads, but must be careful as the EBS storage cost increase with time instance is on.
* With [Deep Learning AMI GPU PyTorch 1.13.1 (Amazon Linux 2) 20230201]



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

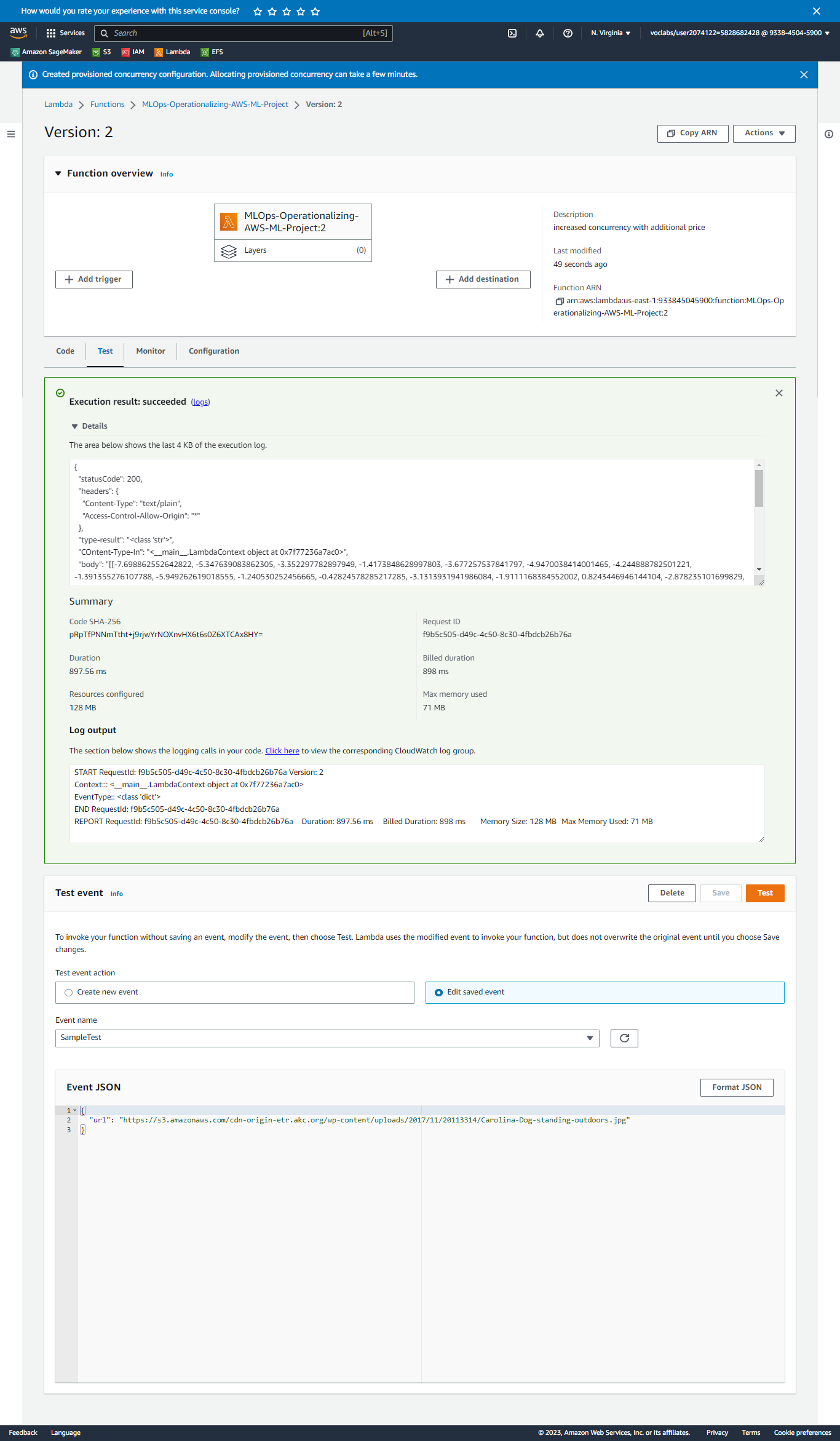
* it is self-containing script which trains the model on any machine using local file system to access the data, while Sagemaker scripts required methods to access S3 dataset. Using EC2 instances is similar to usage of local machine.
  + storage for data is on local path is different (the training job is local)
  + model is also stored in a folder locally.
  + computation infrastructure used is local with not submission to another instance environment for execution.



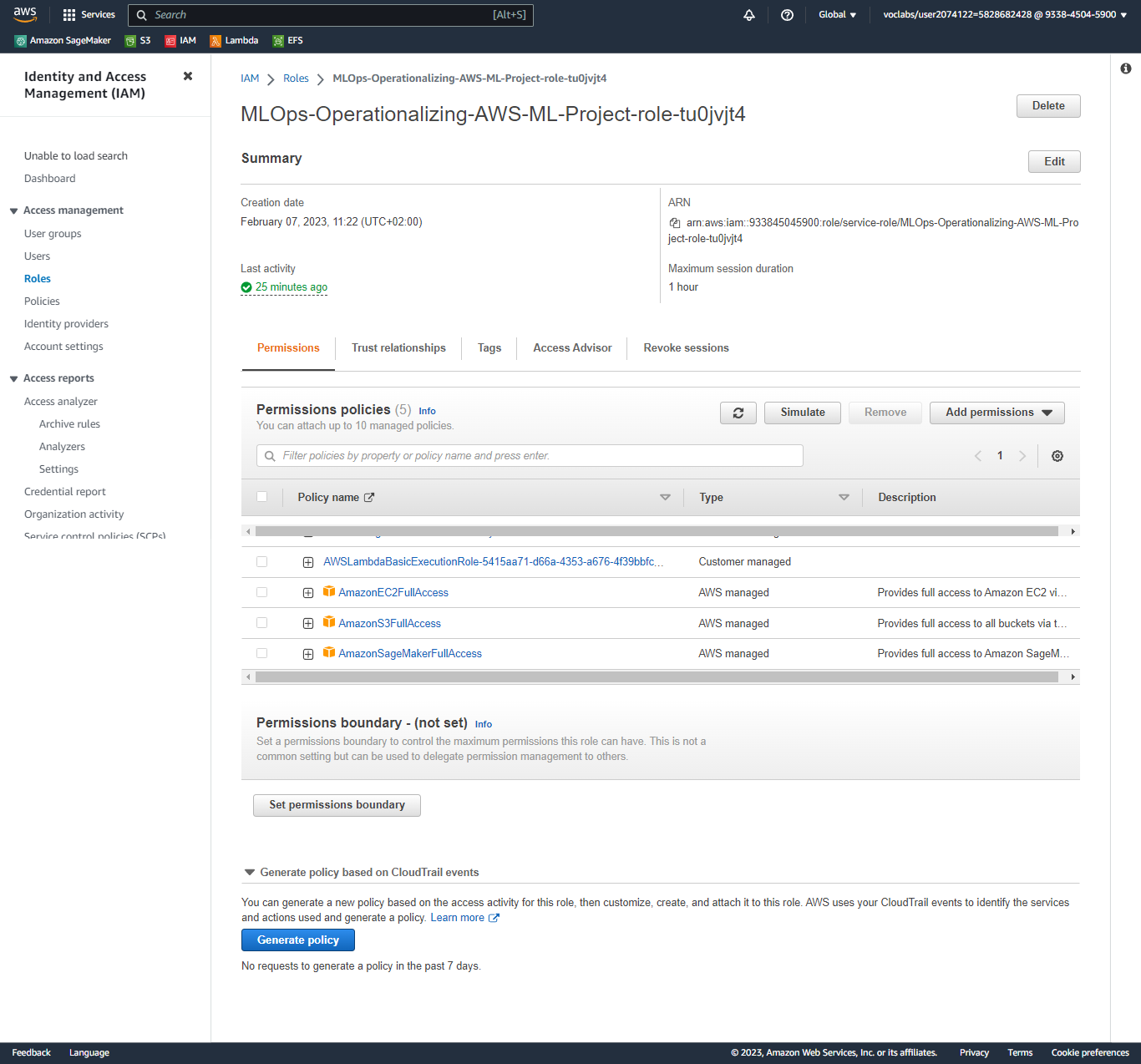
# Step 3: Lambda Function

**Describe how the function is written and how it works**

* This lambda functions invokes the deployed endpoint created from the first step. it accepts the URL of an image and then uses endpoint inference services to execute the operations and return the results in JSON format for breed identification.



**Attach a security policy to your Lambda function so it can access your Sagemaker endpoint**.

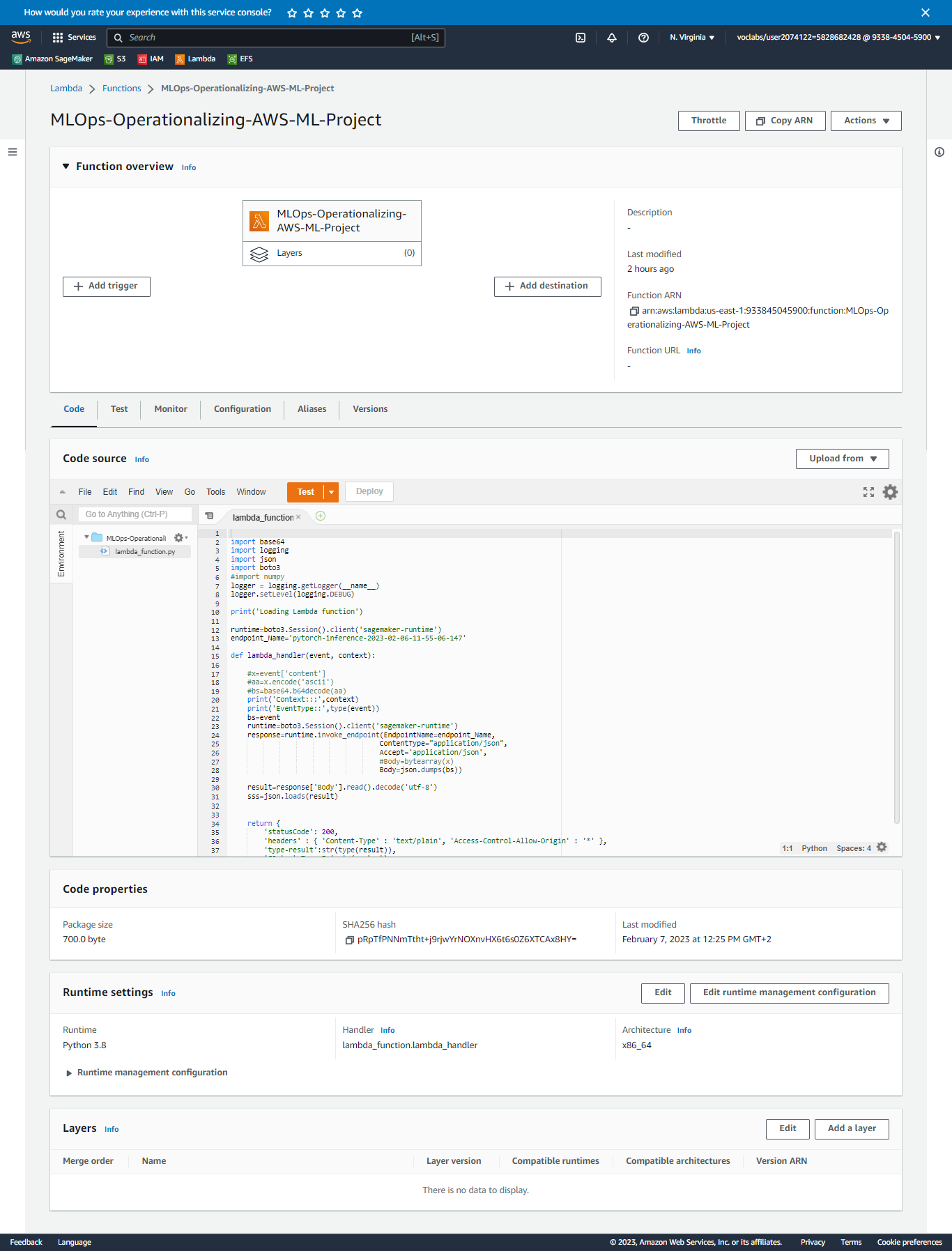


**Test your Lambda function, and add the result of your test to your writeup**

**Test Response Result:**

**7.698862552642822, -5.347639083862305, -3.352297782897949, -1.4173848628997803, -3.677257537841797, -4.9470038414001465, -4.244888782501221, -1.391355276107788, -5.949262619018555, -1.240530252456665, -0.42824578285217285, -3.1313931941986084, -1.9111168384552002, 0.8243446946144104, -2.878235101699829, -4.649499416351318, -6.185570240020752, -4.9580864906311035, -4.424695014953613, 0.41801193356513977, -2.9353291988372803, -2.7232508659362793, -7.775493621826172, -7.063922882080078, -7.322050094604492, -7.940590858459473, -4.056085109710693, -7.363180160522461, -5.64165735244751, -3.111562728881836, -4.576196670532227, -5.910926818847656, -6.551362037658691, -5.1530656814575195, -7.405170917510986, -7.548816204071045, -6.143508434295654, -5.465755939483643, -1.5719480514526367, -5.441058158874512, -6.1667304039001465, -3.9466495513916016, 0.4702502191066742, -4.881587028503418, -1.9118139743804932, -12.494391441345215, -3.2319061756134033, -1.2295335531234741, -4.9419426918029785, -2.914275884628296, -4.3234686851501465, -7.038097381591797, -5.984295845031738, -2.81803035736084, -6.6579909324646, -3.4884591102600098, -8.707923889160156, -7.949629783630371, -3.3004395961761475, -3.240016460418701, -5.740108013153076, -7.424511432647705, -7.692864894866943, -7.383284568786621, -5.518505096435547, -6.879883289337158, 0.0034765787422657013, -7.727499485015869, -4.429338455200195, -2.8595452308654785, -1.6434866189956665, -4.072175025939941, -4.924370288848877, -5.589024066925049, -6.112706661224365, -2.314589023590088, -7.842854022979736, -3.007693290710449, -6.61555290222168, -6.093110084533691, -2.6826772689819336, -7.160601615905762, 0.6826741695404053, -1.168666958808899, -7.320164203643799, -5.28354549407959, -2.049327850341797, -7.931869983673096, -3.4631261825561523, -1.7312875986099243, -7.897009372711182, -6.111726760864258, -5.259680271148682, -8.255675315856934, -6.626457214355469, -3.025027275085449, -4.766608715057373, -3.9160499572753906, -6.909079074859619, -5.580389022827148, -8.45504379272461, -3.9216482639312744, -4.523221492767334, -4.135554313659668, -5.4708075523376465, -8.630441665649414, -3.0037379264831543, -1.521710991859436, -2.8514323234558105, -1.0108898878097534, -2.2446963787078857, -2.4315717220306396, -4.832242488861084, -6.192027568817139, -7.673048496246338, -2.0159521102905273, -8.957904815673828, -0.31093719601631165, -4.665895462036133, 0.1316596418619156, -0.8865324854850769, -4.975542068481445, -3.7051947116851807, -4.13137149810791, -8.761499404907227, -7.0732550621032715, -1.6587605476379395, -0.0707918107509613, -5.985174655914307, -5.818277359008789, -5.7232465744018555, -1.8361643552780151, -6.063624858856201]]**

**Take a screenshot of your Lambda setup and add it to your solution archive.**

****

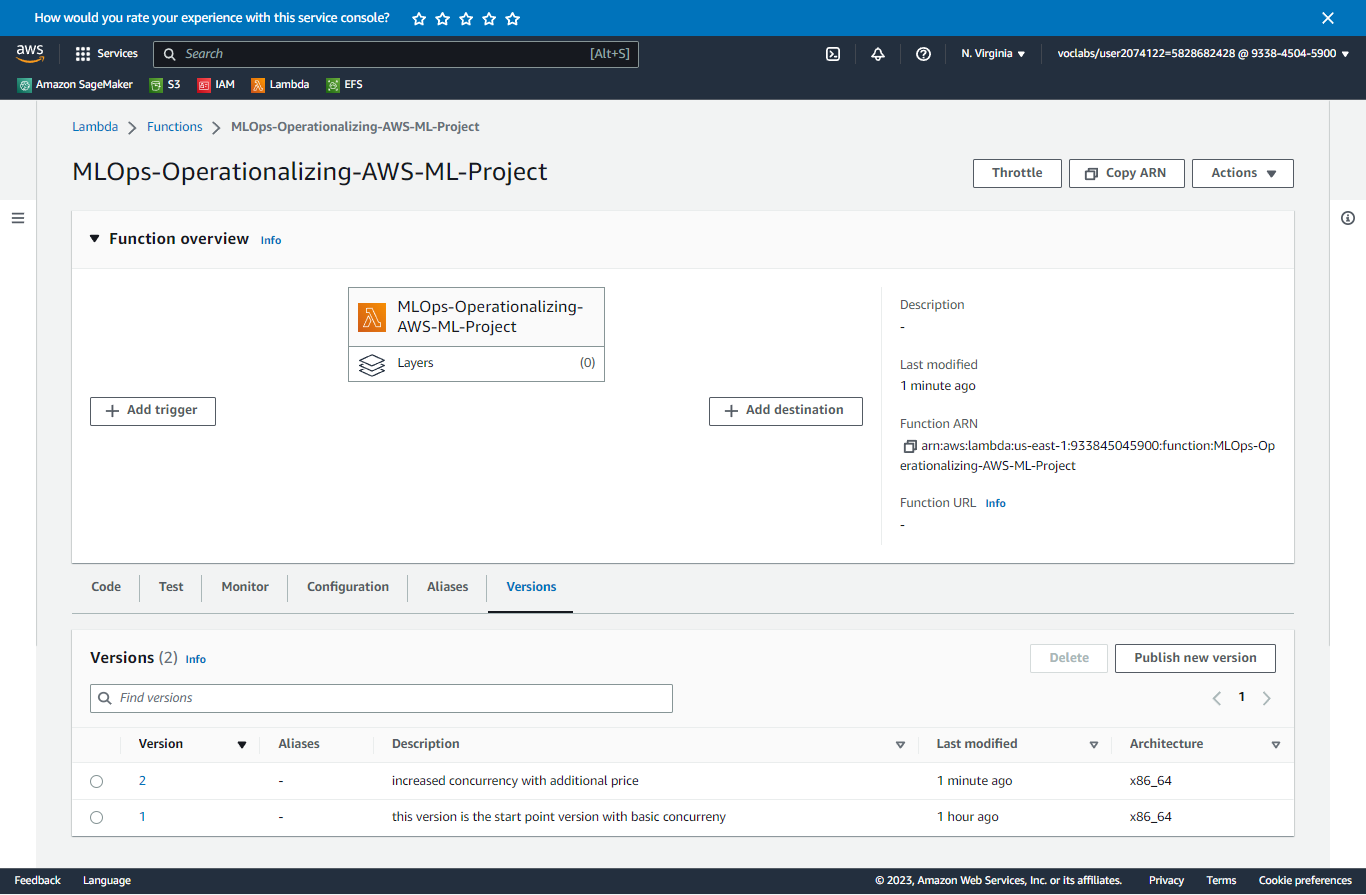
**Write about whether you think your AWS workspace is secure, and whether you think there are any vulnerabilities.**



* insecure since there are vulnerabilities that the endpoint is exposed to from external request calls. Granting full access to lambda function can result unexpected errors. So it should be monitored or limited.

# Step 5: Concurrency and auto-scaling

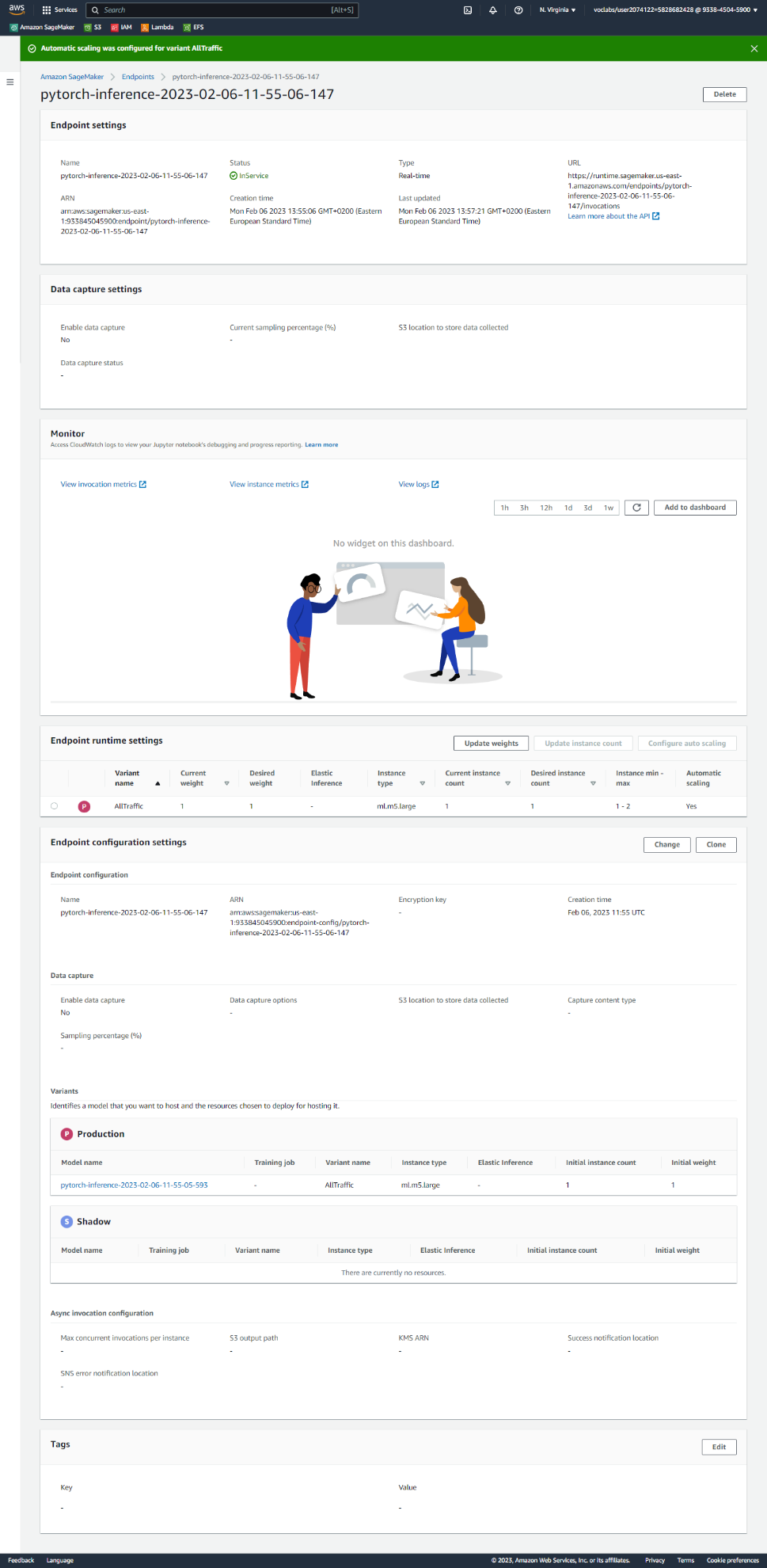
**Set up concurrency for your Lambda function**

**Version:**

**Concurrency:**



**Set up auto-scaling for your deployed endpoint**

****

**Write about your configuration of concurrency and auto-scaling. Make sure to mention what kind of concurrency you set up, how you set up auto-scaling, and why you made your decisions.**

* **Concurrency:** decided based on assumed usage that there are only two request which may get overlapped and extra request can be queried.
* **Autoscaling:** was turned on for scaling to the maximum of 2 instance which should typically start during the peak traffic (30 simultaneous requests) for min of 30 seconds and cool down in 30 seconds.