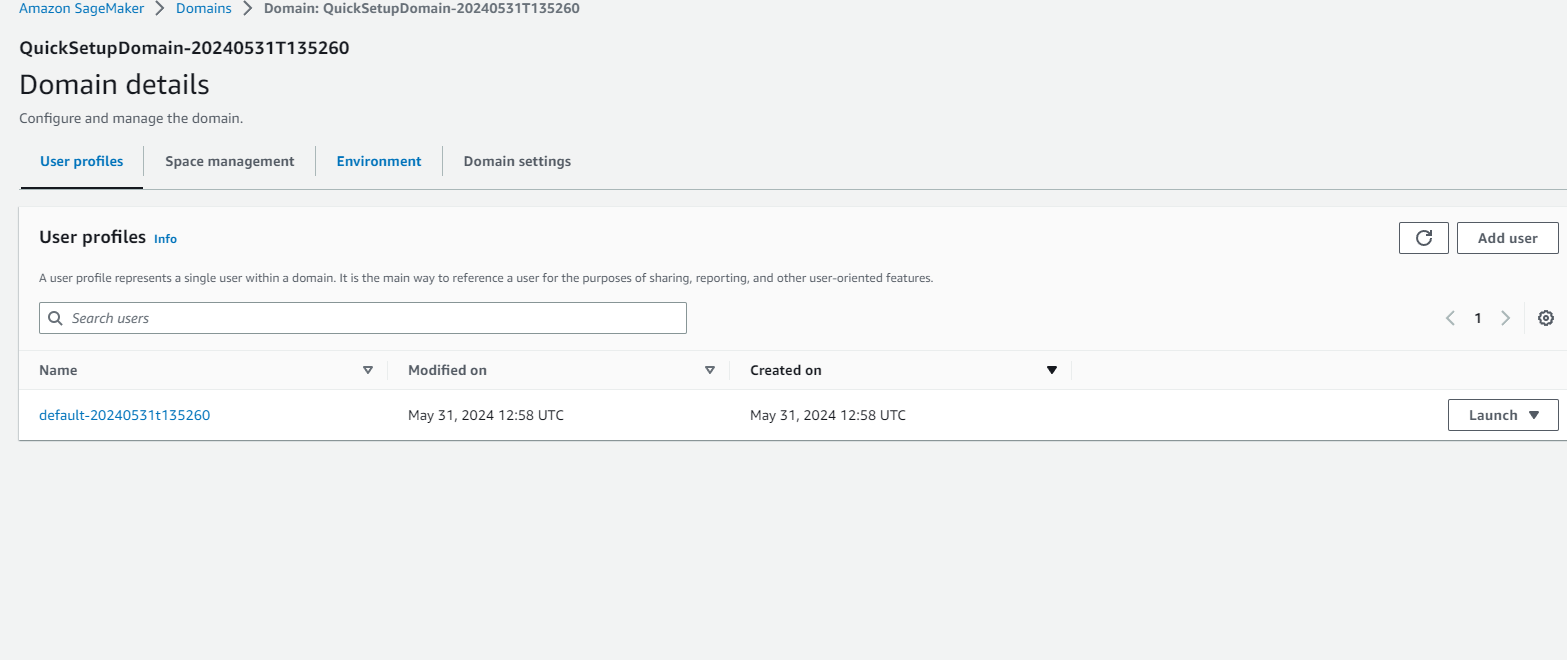
# Project: Operationalizing an AWS ML Project

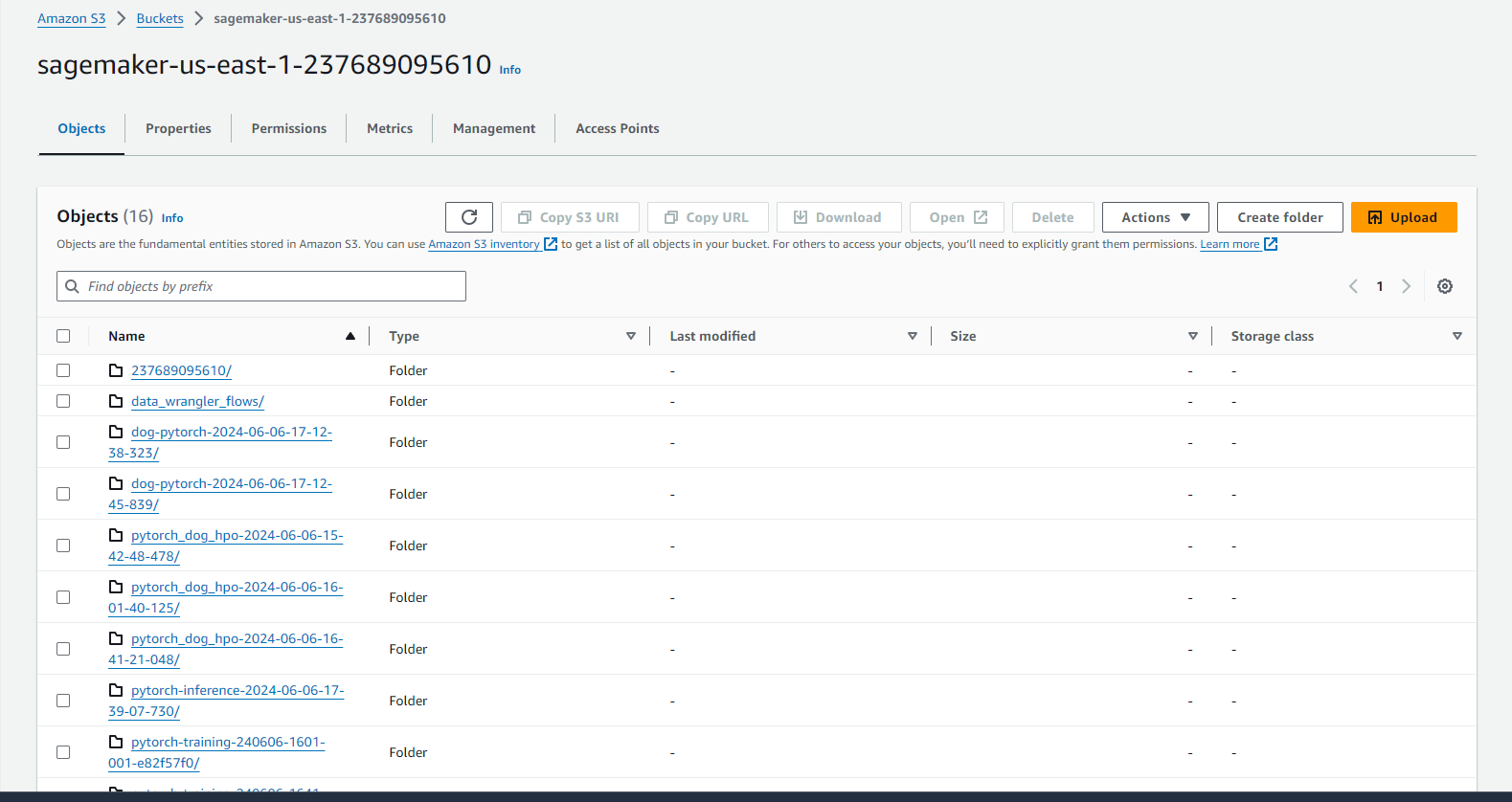
### Step 1: Training and deployment on Sagemaker

I choosed the default quick setup domain which has all the jupyter notebooks and configurations set up , i have used simple fast launch instances as they are cost effective, i have updated the cells with my s3 bucket name and also created the multi-instance training estimator and have also checked for the best hyper parameters

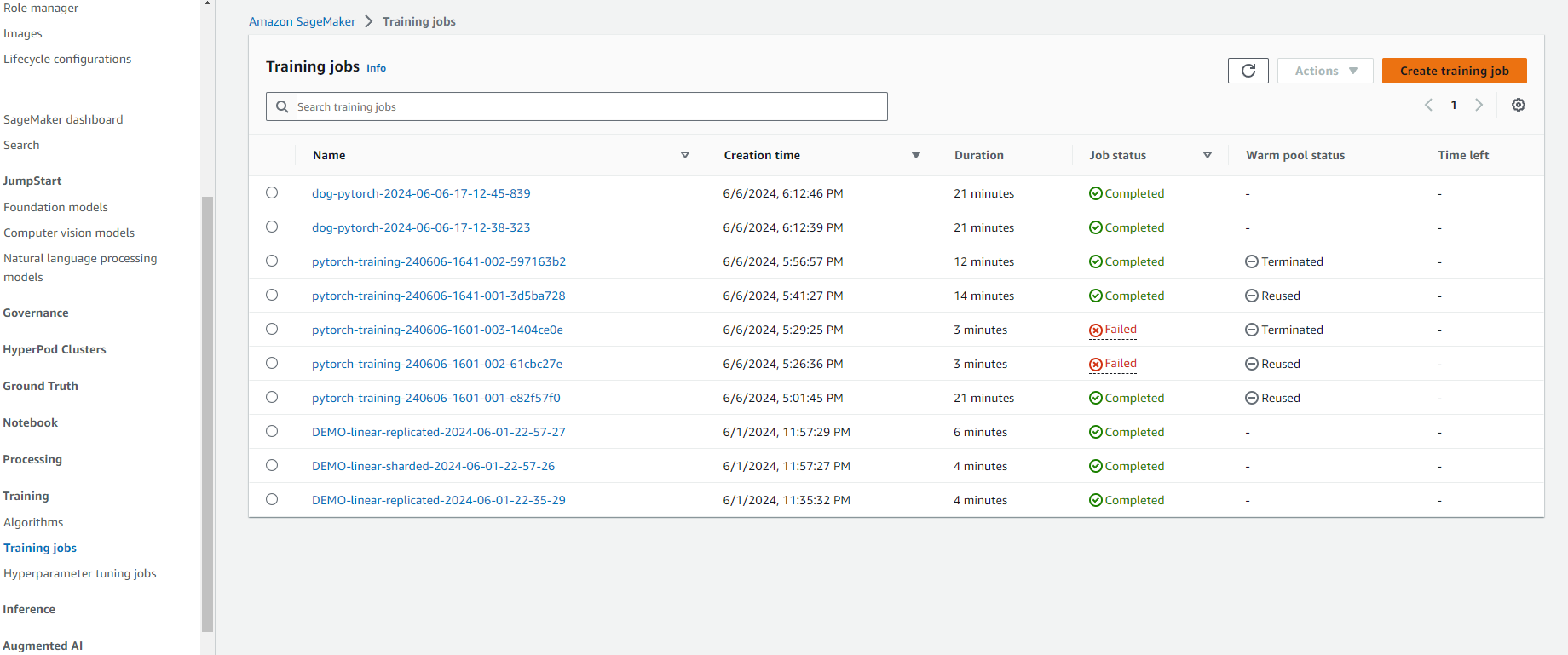
AWS sagemaker setup



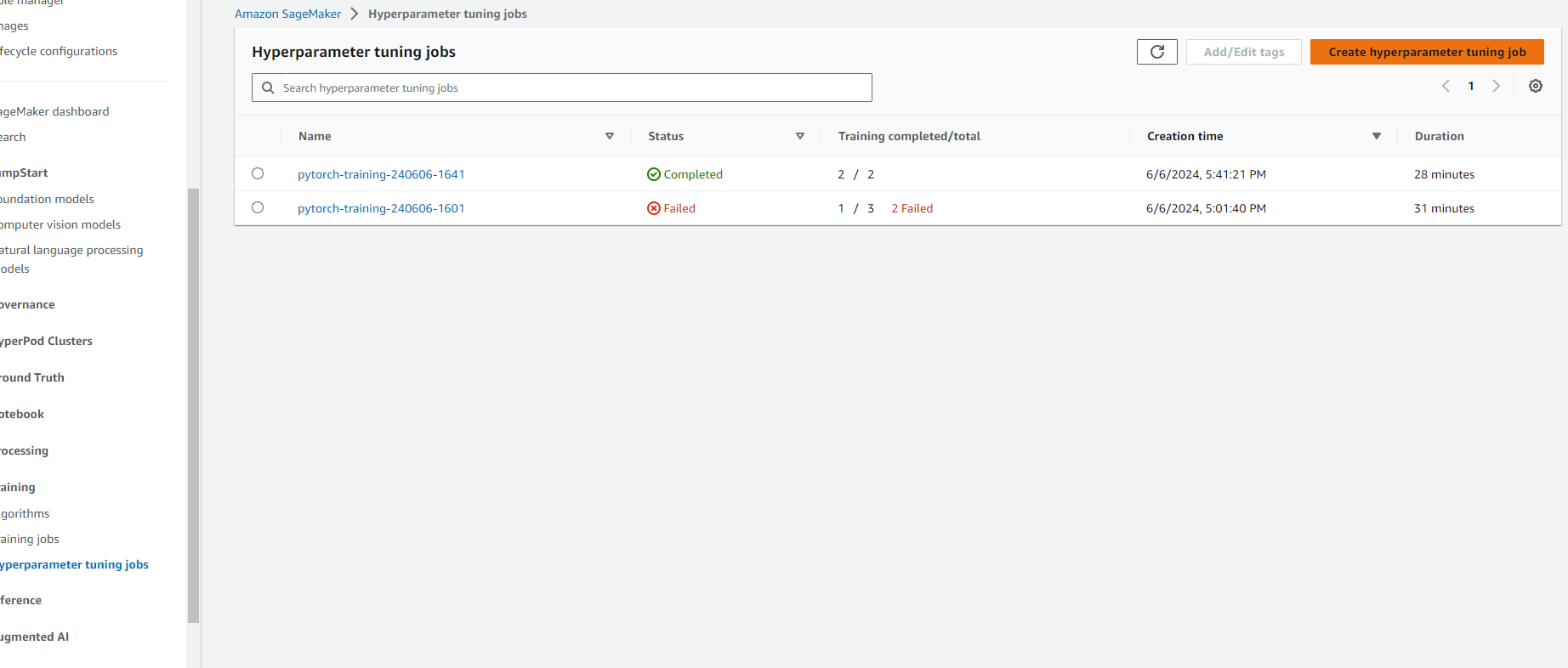
S3 Bucket



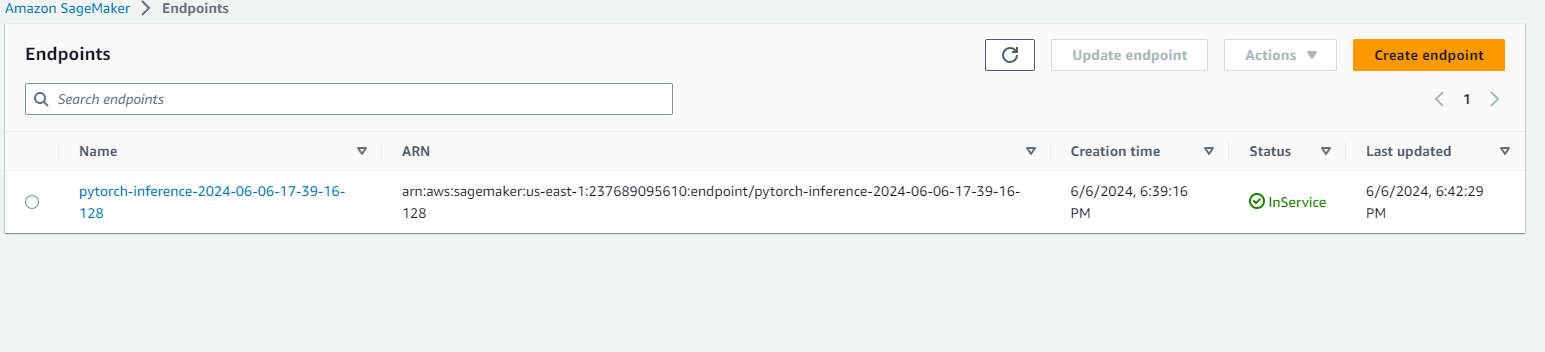
Training Jobs Completed:



Training Job hyperparamters completed



Endpoints in service



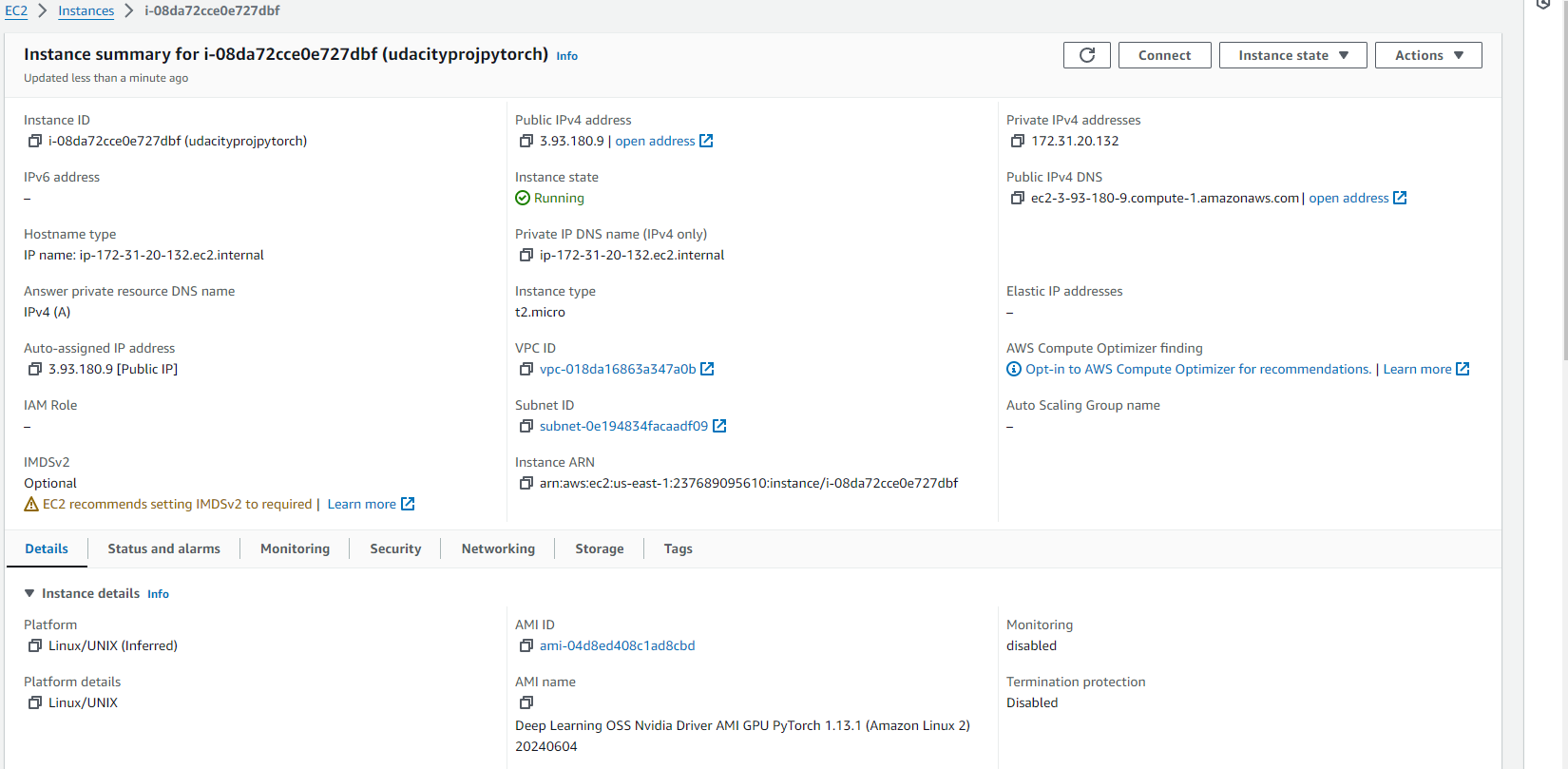
### 

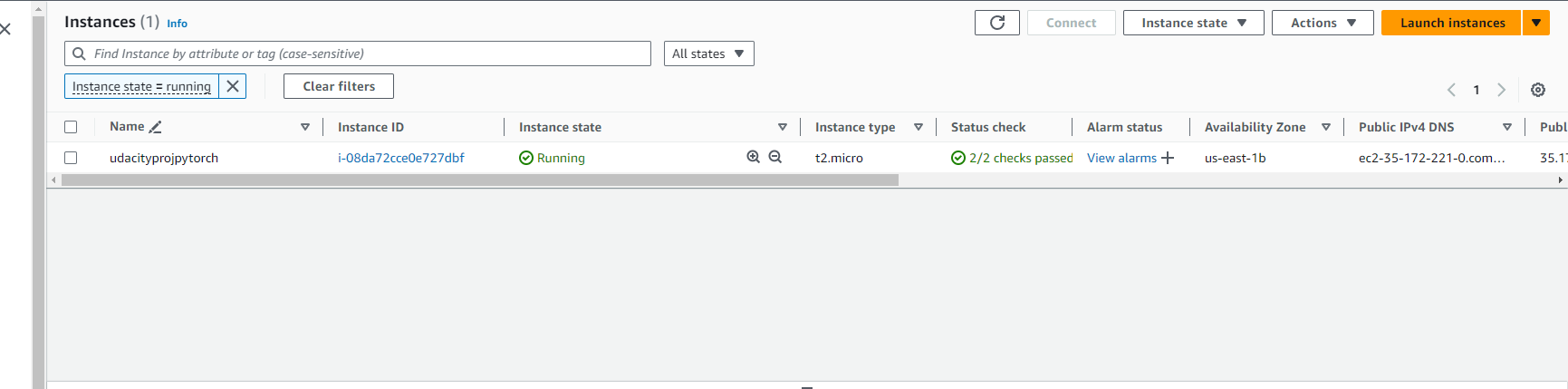
### Step 2: EC2 Training

Write about ec2 instance created and justification why you choose that instance?

Instance i choose was Deep Learning OSS Nvidia Driver AMI GPU PyTorch 1.13.1 (Amazon Linux 2) 20240604 because this instance has configurations and enough space for running torch library , EBS storage was chosen and that was set to 45GB just so that all libraries could be installed

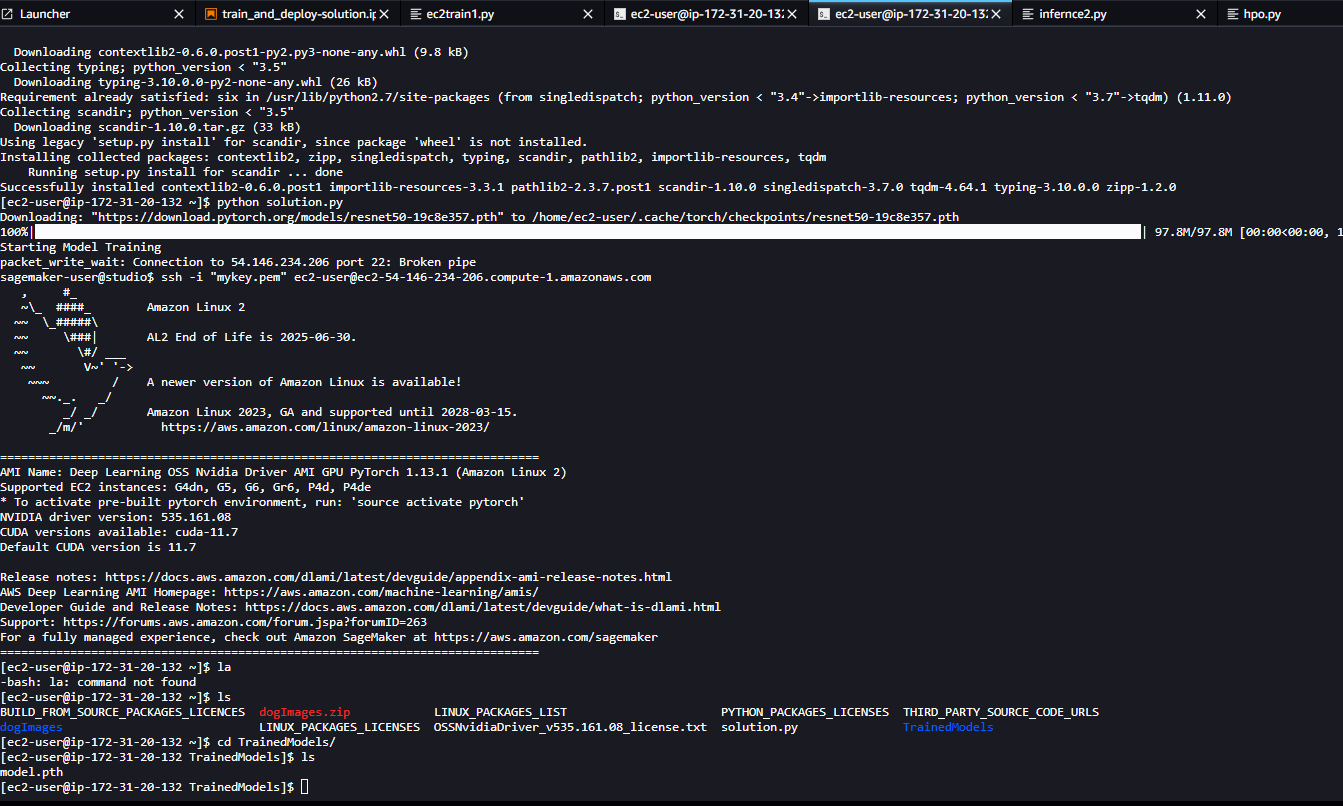
#### EC2 Setup





Write at least a paragraph about the difference between ec2train1.py and code used in step 1

This script trains the model using local system to get accessibility to data whereas sagemaker needs access to s3 , with ec2 instance it is like a local machine where you have your own storage and own computation structure, you can easily configure and install simple libraries and then run the script , i installed the numpy and torch library in the ec2 instance and then run the python solution file which was provided , after successful training of model it creates the model.pth which can be used afterwards

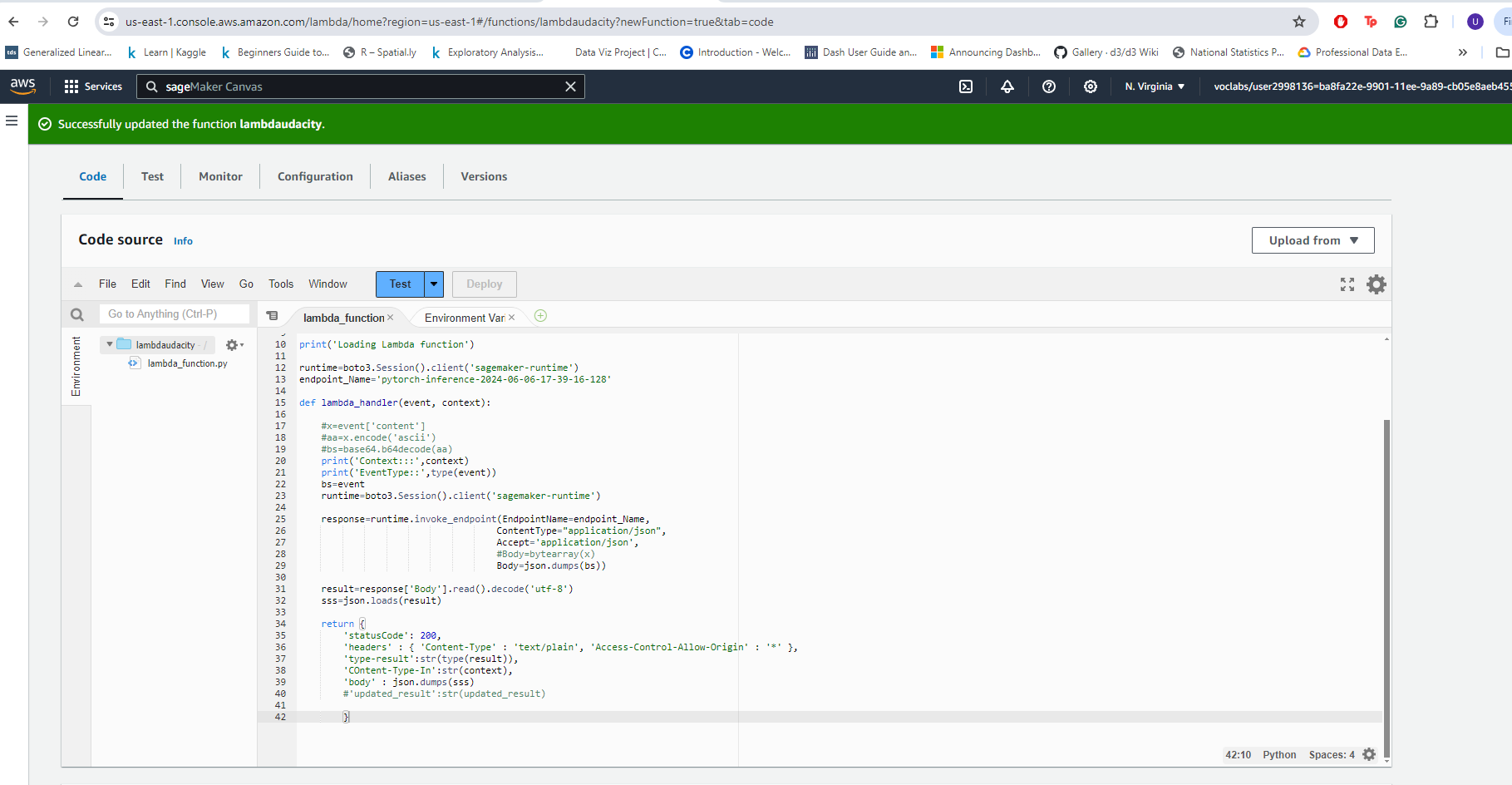


### Step 3: Lambda function setup and Step 4 attaching security to lambda function

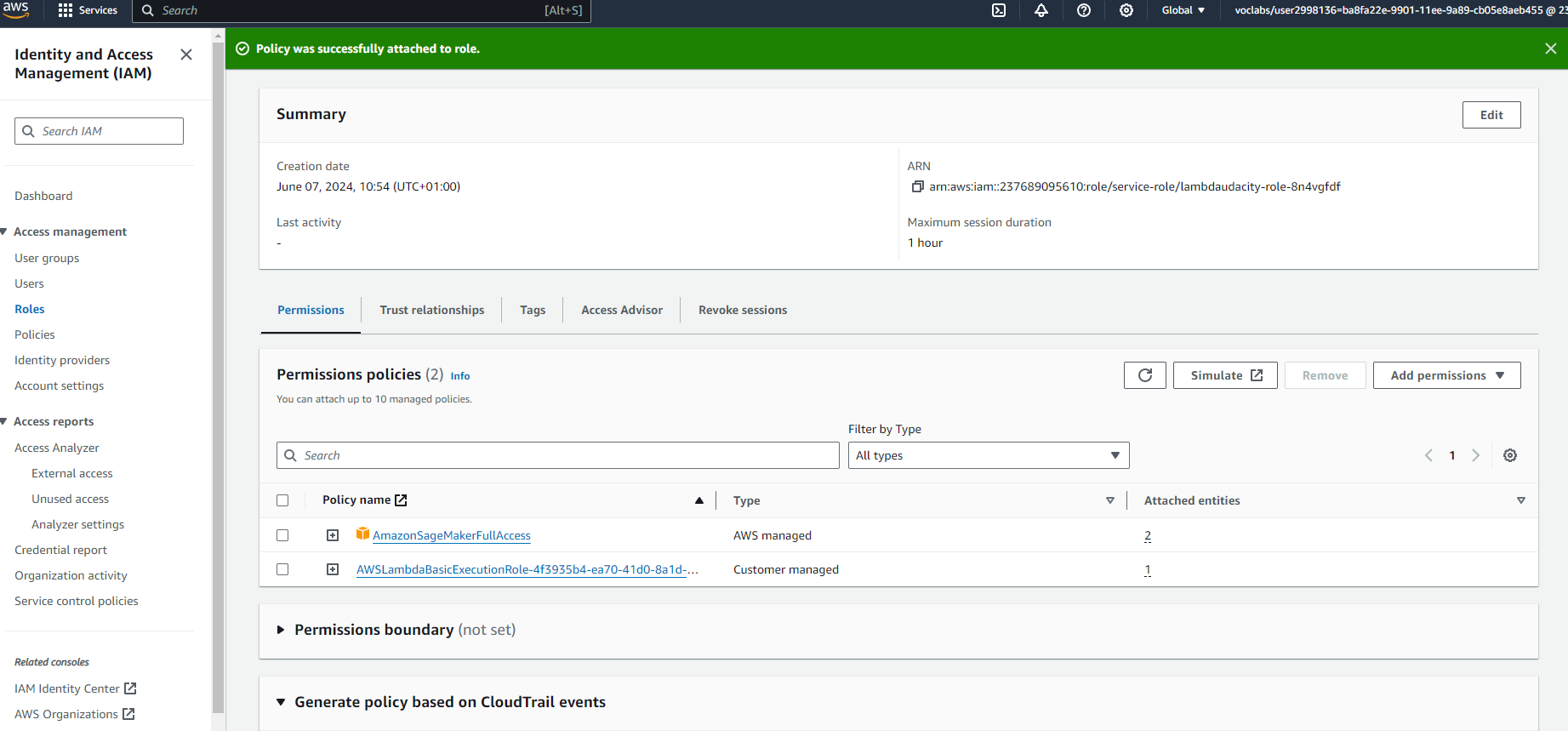
Describe how the function is written and how it works

This Lambda function invokes the deployed endpoint and creates a connection with it , it sends the url provided in the test and gets a response back , but before that i had to attach the aws sagemaker full execution role policy to this lambda function , the test was successful as you can also see in the screenshot provided

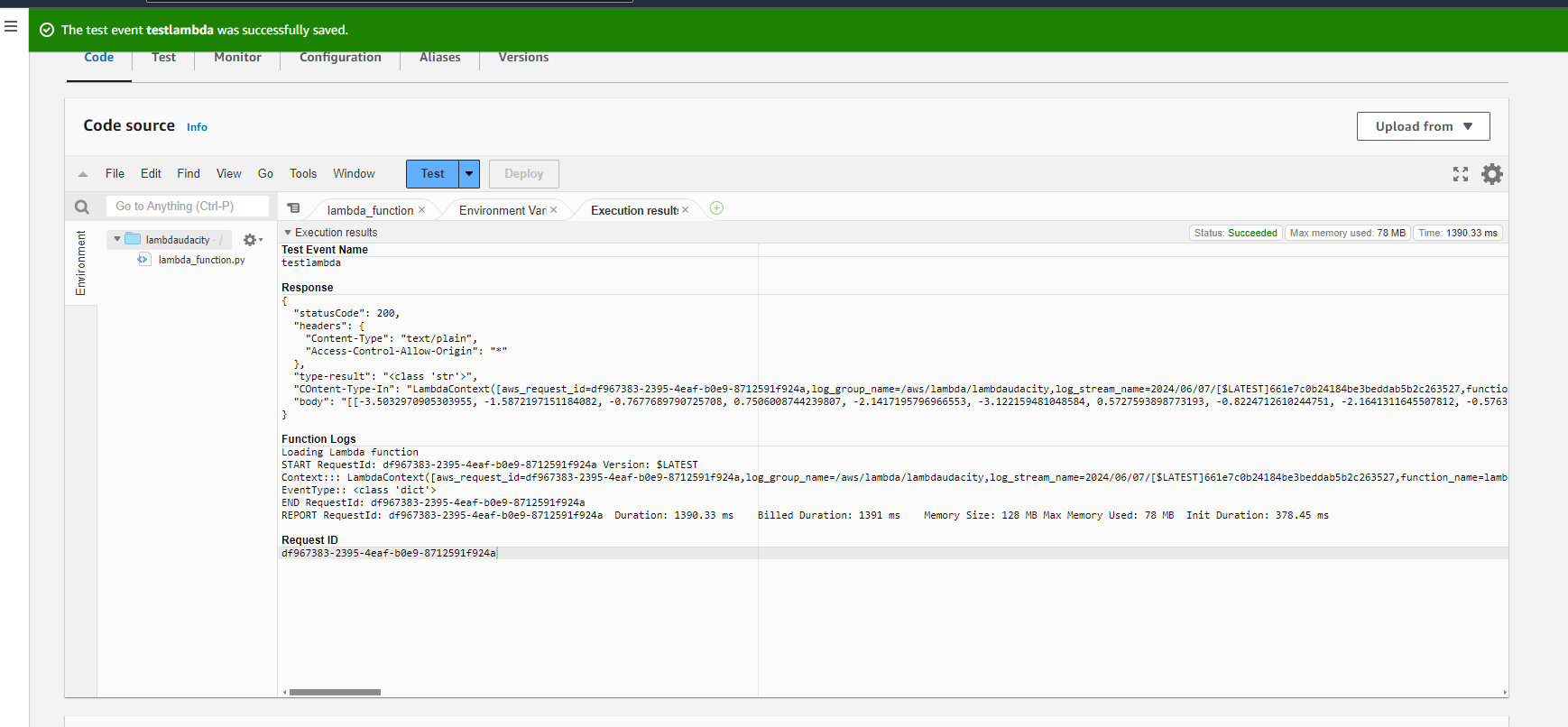
Deploying and writing the lambda function



Attaching the policy to the lambda function



Checking the response back from the test function

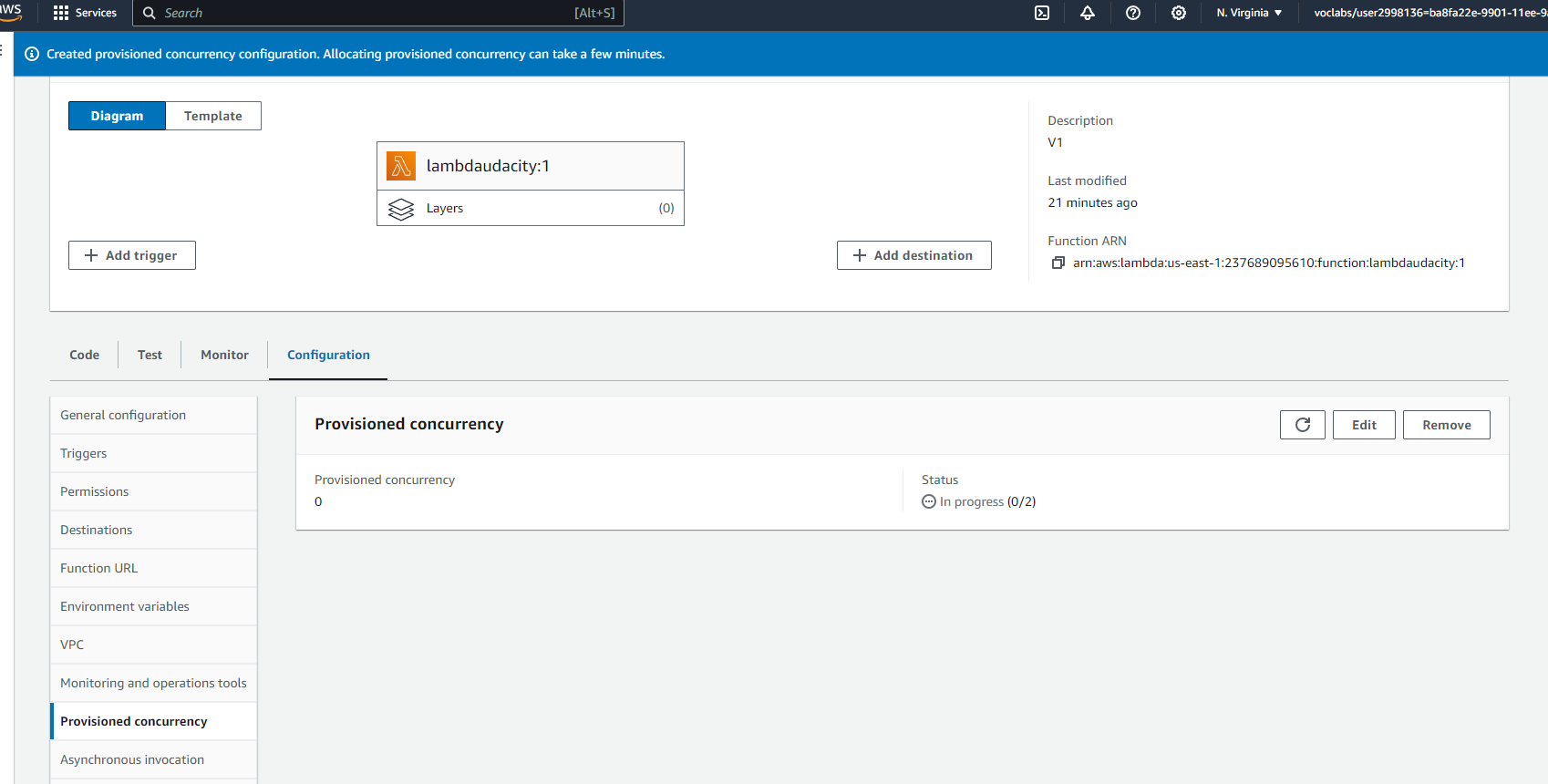


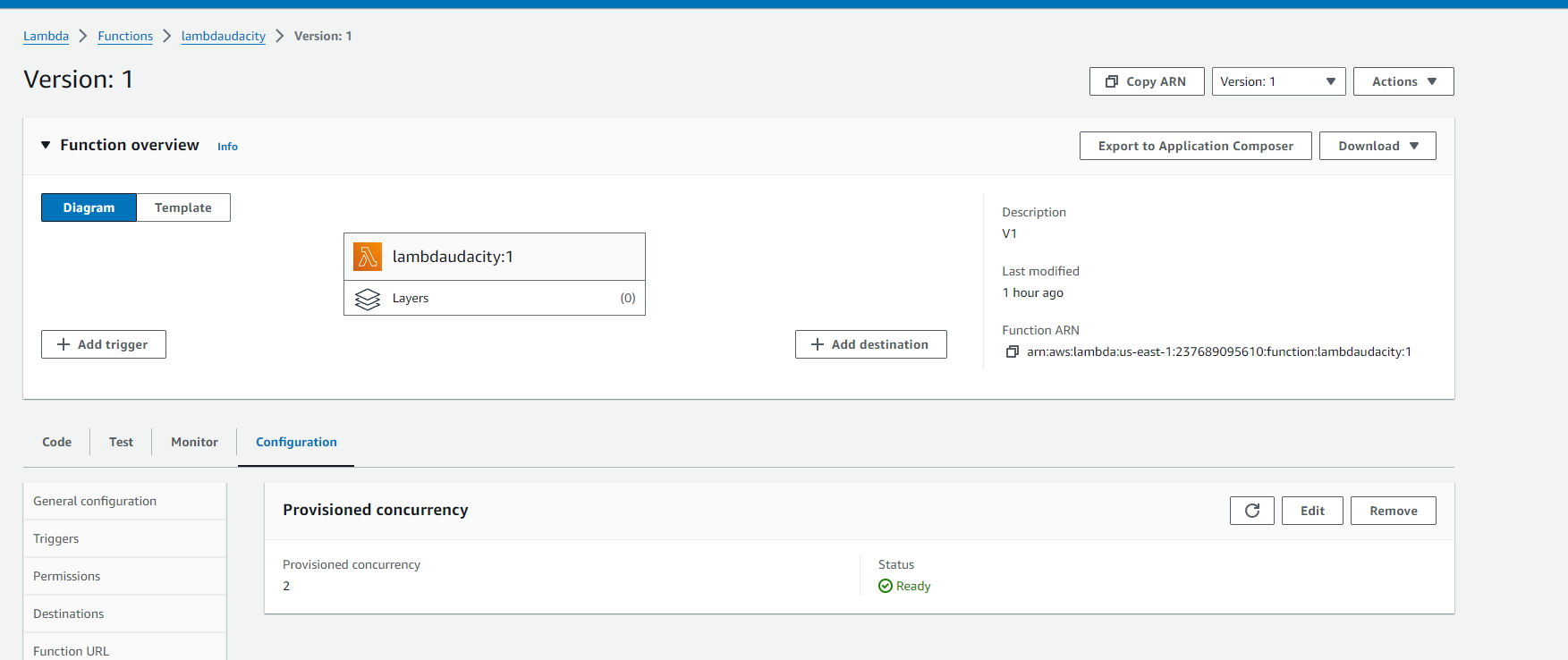
Testing my Lambda function and the response from it

[-3.5032970905303955, -1.5872197151184082, -0.7677689790725708, 0.7506008744239807, -2.1417195796966553, -3.122159481048584, 0.5727593898773193, -0.8224712610244751, -2.1641311645507812, -0.5763108730316162, -0.8086022734642029, -2.389901876449585, -1.9422714710235596, 1.100341796875, -0.26965034008026123, 0.21998783946037292, -4.448609828948975, -1.4109994173049927, -2.8092644214630127, 1.148894190788269, -1.8793400526046753, 0.2365783154964447, -4.0021281242370605, -3.8213226795196533, -2.092172384262085, -5.77842378616333, -1.631203532218933, -0.36555516719818115, -3.2245607376098633, -0.5889348387718201, -0.9163044095039368, -1.3189921379089355, -3.1707143783569336, -0.8363109827041626, -4.637266159057617, -3.038100481033325, -2.3919320106506348, -1.856014370918274, 0.7525979280471802, -2.055586814880371, -0.2888779938220978, -0.931816816329956, 1.2027690410614014, -0.7588621973991394, -2.080308675765991, -4.121342658996582, -0.23925864696502686, -0.4919027090072632, -1.1479555368423462, -0.9333341121673584, -1.6911085844039917, -2.2167131900787354, -2.755300998687744, -1.782326340675354, -2.2286133766174316, -1.389290690422058, -2.8503918647766113, -3.0832958221435547, -0.8418097496032715, 0.2474696785211563, -3.468285083770752, -3.046025276184082, -3.208275318145752, -2.329455852508545, -3.292487621307373, -2.7571160793304443, 0.29264193773269653, -2.2366702556610107, 0.6742424964904785, -0.44905605912208557, -0.5664674639701843, -2.5348920822143555, -1.873576045036316, -1.817552924156189, -2.528771162033081, -1.2557942867279053, -2.237632989883423, 0.4063810110092163, -2.647940158843994, -1.7562816143035889, 0.5188231468200684, -3.0768613815307617, -0.5471802353858948, -0.9216118454933167, -2.7565505504608154, -2.093254804611206, -1.695647954940796, -1.180632472038269, -0.8637321591377258, -0.33544591069221497, -2.9250197410583496, -3.5262932777404785, -3.9739744663238525, -2.7770373821258545, -2.3573548793792725, -1.3743940591812134, -2.9559221267700195, -2.738215446472168, -1.6649876832962036, -2.1590394973754883, -5.241294860839844, -1.4712973833084106, -0.40772566199302673, -4.421698093414307, -2.7951135635375977, -2.847907304763794, -4.077500343322754, -0.6380893588066101, -2.5278351306915283, 0.6058333516120911, -1.5183559656143188, -1.4645717144012451, -1.57234787940979, -1.371683955192566, -2.6448655128479004, -1.3635883331298828, -3.314955949783325, -0.9221130013465881, -2.977008104324341, -1.1008497476577759, -0.4431454539299011, -1.3449079990386963, -4.170413494110107, -1.8815847635269165, -5.411243915557861, -2.5093414783477783, -2.258676052093506, -1.4860609769821167, -1.7014336585998535, -1.3456860780715942, -4.2330427169799805, -2.9709153175354004, -3.000408172607422]]"

### Step 5 Concurrency and autoscalling

Setup of concurrency on my lambda function after defining the version





Setting autoscalling for the endpoint

