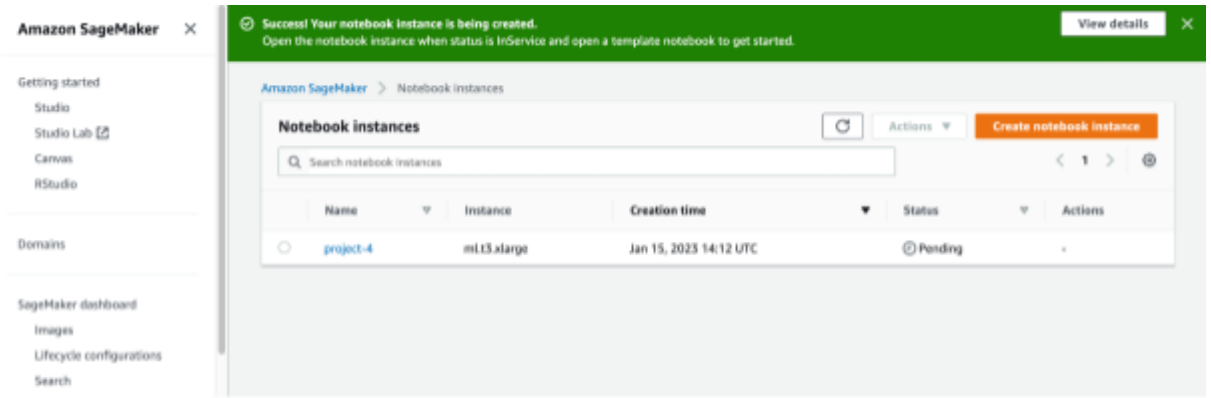
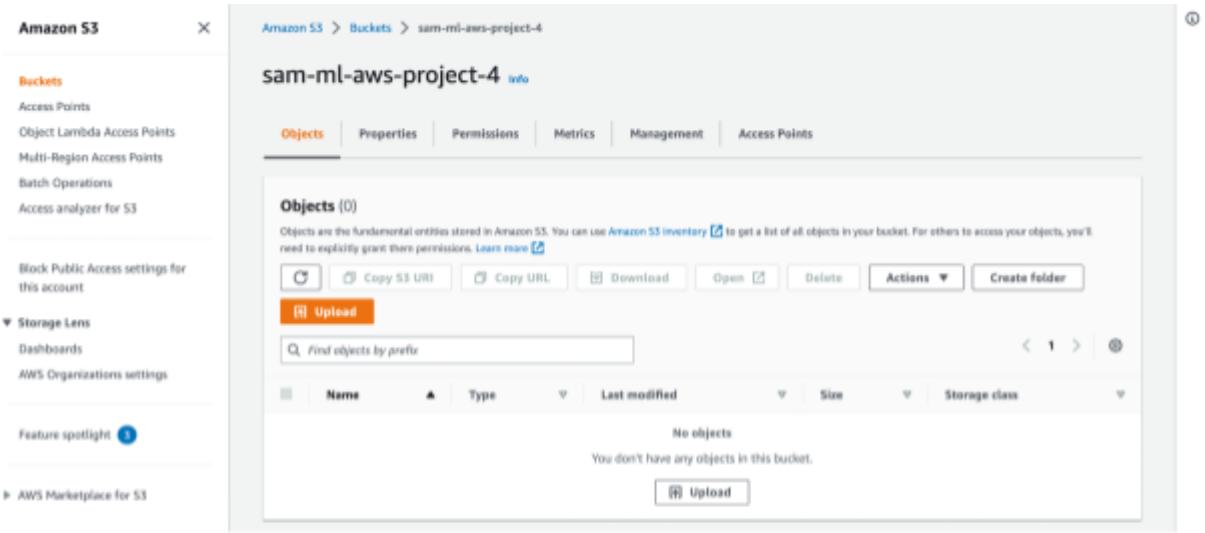
# ML with AWS Project 4

## Initial setup, training and deployment




Notebook instance

I've considered the instance types here: [https://aws.amazon.com/sagemaker/pricing/](https://aws.amazon.com/sagemaker/pricing/) and gone for the ml.t3.xlarge. With a vCPU of 4 and 16 GiB memory, it offers better performance than the standard, free tier ml.t3.medium while not breaking the bank (my credits for this project) at $0.20 per hour. It seems a reasonable starting point for this project.




S3 bucket

When I switch to multi-instance training (5 instances), 5 streams show up in Log.

# EC2 Setup

At first, I selected a p3.2xlarge instance, an accelerated computing option, as recommended here: https://docs.aws.amazon.com/dlami/latest/devguide/gpu.html for deep learning applications, costing $3.825 per hour, but my account would not allow it. I've instead gone for a m3.2xlarge instance costing an affordable $0.532 per hour, but should have the resources I need to complete the project step.


Evidence of saved model on EC2

Here are some key differences between the train_and_deploy-solution.ipynb notebook and ec2train1.py:

- The notebook imports and uses boto3 and sagemaker, the AWS and Sagemaker SDKs for Python. These allow interaction with AWS services and other ML capabilities, such as profiling the contents of the model, which are not in ec2train1.
- The notebook uses other .py files as entry points, whereas ec2train1 contains everything in one file
- The notebook goes through stages of tuning a model to find the best hyperparameters, training the model with those hyperparameters and then deploying that model to an endpoint. ec2train1 trains a model with predefined hyperparameters.

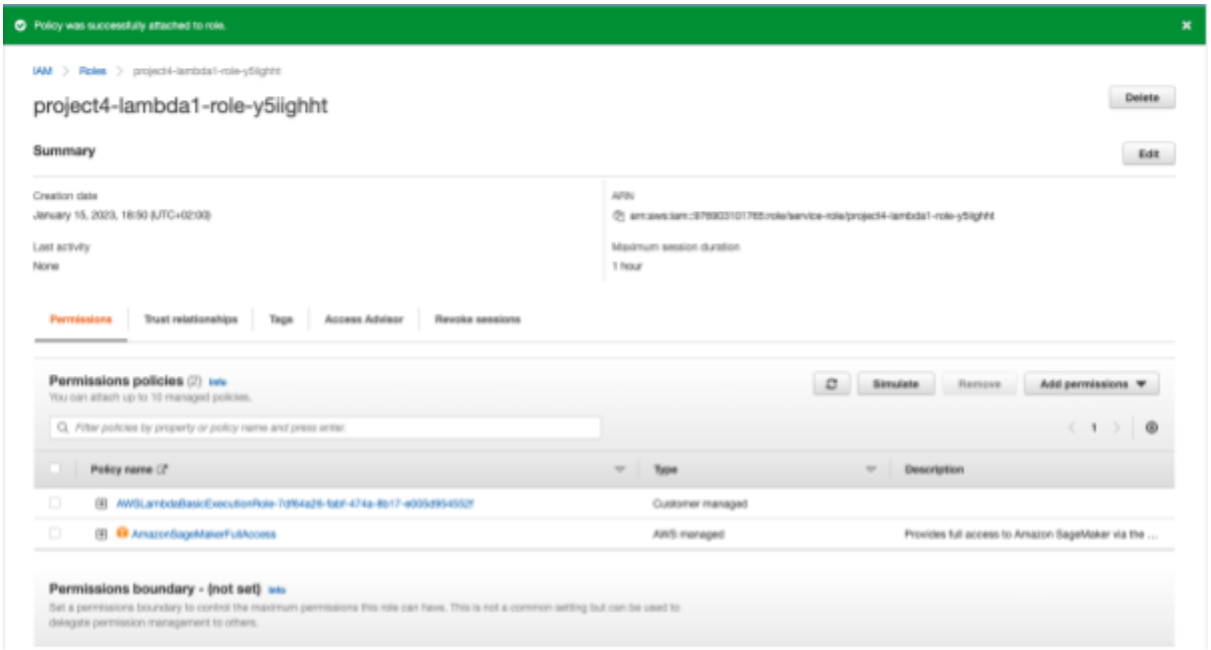
# Lambda function

Some of the key sections of the lambda function include:

- Importing the necessary libraries.
- Calling the Sagemaker API.
- Declaring the endpoint the function will interact with.
- Stating 'event' as an argument - the json inputs the function will receive.
- Invoking the endpoint - to get inferences from the model hosted here. Inferences are made based on the Body of this method, which is set to the event argument mentioned above.
- The result of this invoke method are declared and transformed into a data type that can be returned by the lambda function

# Security and testing

## Lambda for this project



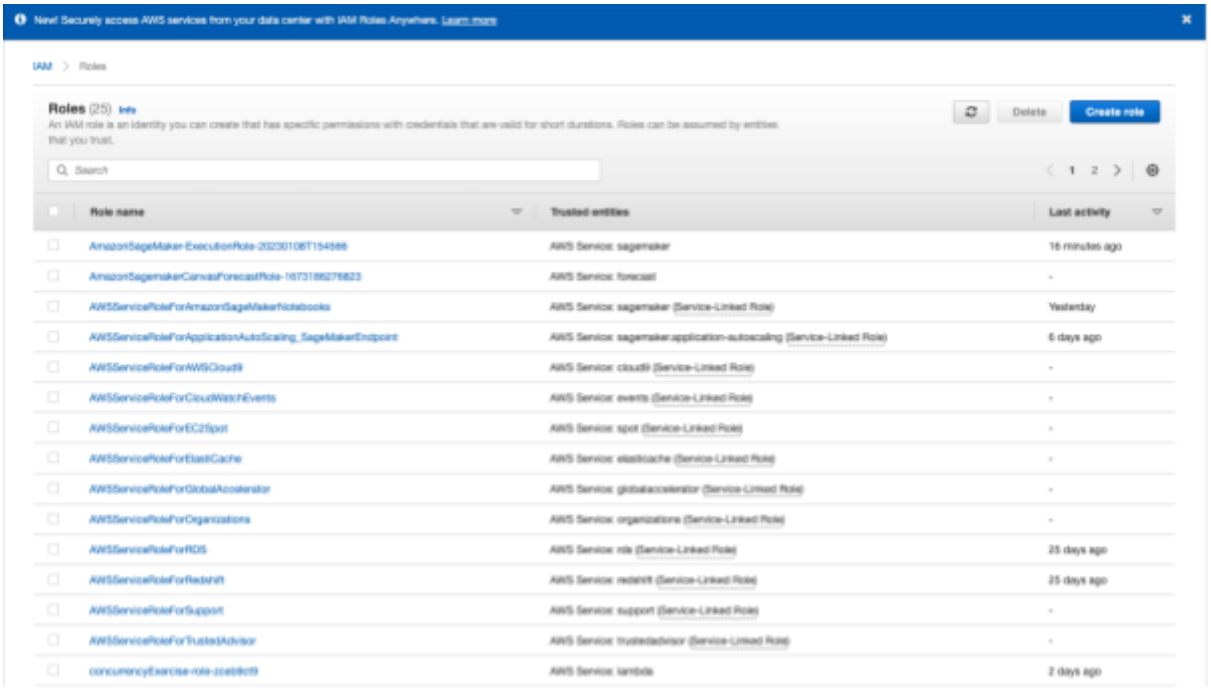

IAM role for lambda function


Successful test of lambda function

Result: [-8.603170394897461, -3.095884323120117, -7.02620792388916, 0.42671892046928406, -1.3242584466934204, -4.687857627868652, -6.269331455230713, -4.580907821655273, -9.891451835632324, 0.04751767963171005, -0.6612728834152222, -1.9501152038574219, -2.6280150413513184, -1.8509739637374878, -8.470413208007812, -5.328761577606201, -4.4046630859375, -2.523554801940918, -6.9708571434021, 0.3848723769187927, -0.698431134223938, 1.4331912994384766, -7.036735534667969, -10.118927955627441, -16.699235916137695, -11.394137382507324, -7.528539657592773, -14.564789772033691, -2.814797878265381, -5.508481502532959, -3.9747602939605713, -6.643276691436768, -8.956212997436523, -8.12283706665039,

-10.439419746398926, -5.174689292907715, -7.166004657745361,
-3.9198856353759766, -4.788877487182617, -9.020564079284668, -7.705960750579834,
-4.586427688598633, 1.0017045736312866, -6.83819580078125, -0.33990639448165894,
-11.302521705627441, -5.059659481048584, -1.34934401512146, -4.737830638885498,
-3.6282691955566406, 0.4330903887748718, -10.938536643981934,
-11.787221908569336, -1.9773625135421753, -9.982752799987793, -5.843520164489746,
-6.388765811920166, -8.687368392944336, -12.003975868225098, -5.46987771987915,
-11.055335998535156, -9.883378028869629, -7.709234237670898, -6.718817710876465,
-7.614677906036377, -9.490221977233887, 1.3321764469146729, -4.195179462432861,
-5.185770034790039, -7.993285179138184, 2.2226386070251465, -10.763575553894043,
-8.972954750061035, -10.71225643157959, -8.995817184448242, -2.8376572132110596,
-13.105772018432617, -5.465895175933838, -2.7797839641571045,
-8.585526466369629, -3.3453352451324463, -10.988717079162598,
-2.759714126586914, 0.6196799278259277, -9.313034057617188, -8.514826774597168,
-10.734067916870117, -10.8993501663208, -6.454702377319336, -2.6461448669433594,
-7.453824043273926, -1.9680378437042236, -14.967737197875977,
-6.899577617645264, -4.628106117248535, -3.748720407485962, -7.610393524169922,
0.949254035949707, -7.395874500274658, -6.737936973571777, -12.266091346740723,
-7.610177993774414, -8.500252723693848, -9.205257415771484, -4.634819984436035,
-2.431992769241333, -6.6676483154296875, 0.09716781228780746,
0.6686152219772339, -0.817338228225708, -4.460118293762207, -1.586832880973816,
-9.055063247680664, -7.609857082366943, -5.943742752075195, -1.8907426595687866,
-3.535428762435913, -1.3547980785369873, -7.84410285949707, -3.156365156173706,
-4.725563049316406, -9.838225364685059, -4.593583583831787, -6.295639514923096,
-11.550751686096191, -2.797152519226074, -4.660189151763916, -0.9294584393501282,
-4.738776683807373, -12.123159408569336, -8.88110065460205, -2.5190694332122803,
-7.847152233123779]

## Security considerations for this account



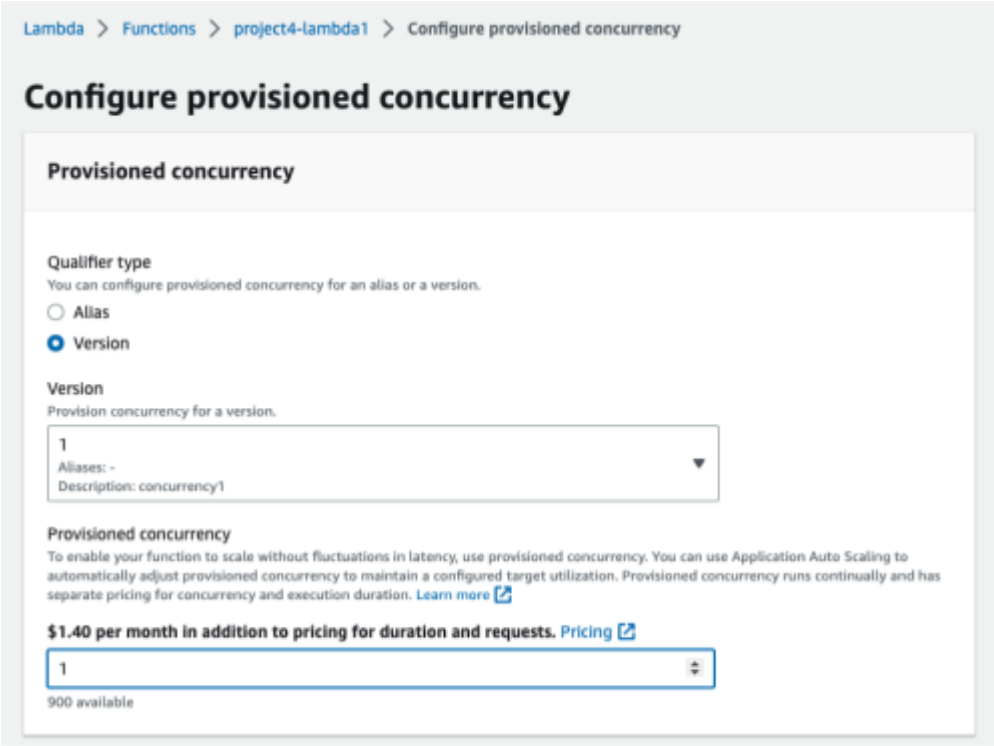

All IAM Roles in this account

There are many roles defined in this account. This could be quite hard to keep track of. Many do not have recent activity. These could be deleted. The Sagemaker execution role has been given full access by default to allow for easy operation for this project. This should typically be limited based on the specific user's needs in a production environment.

We could consider other security measures such as setting up a Virtual Private Cloud (VPC) and whitelisting certain IP addresses that we know should have access to using this account.

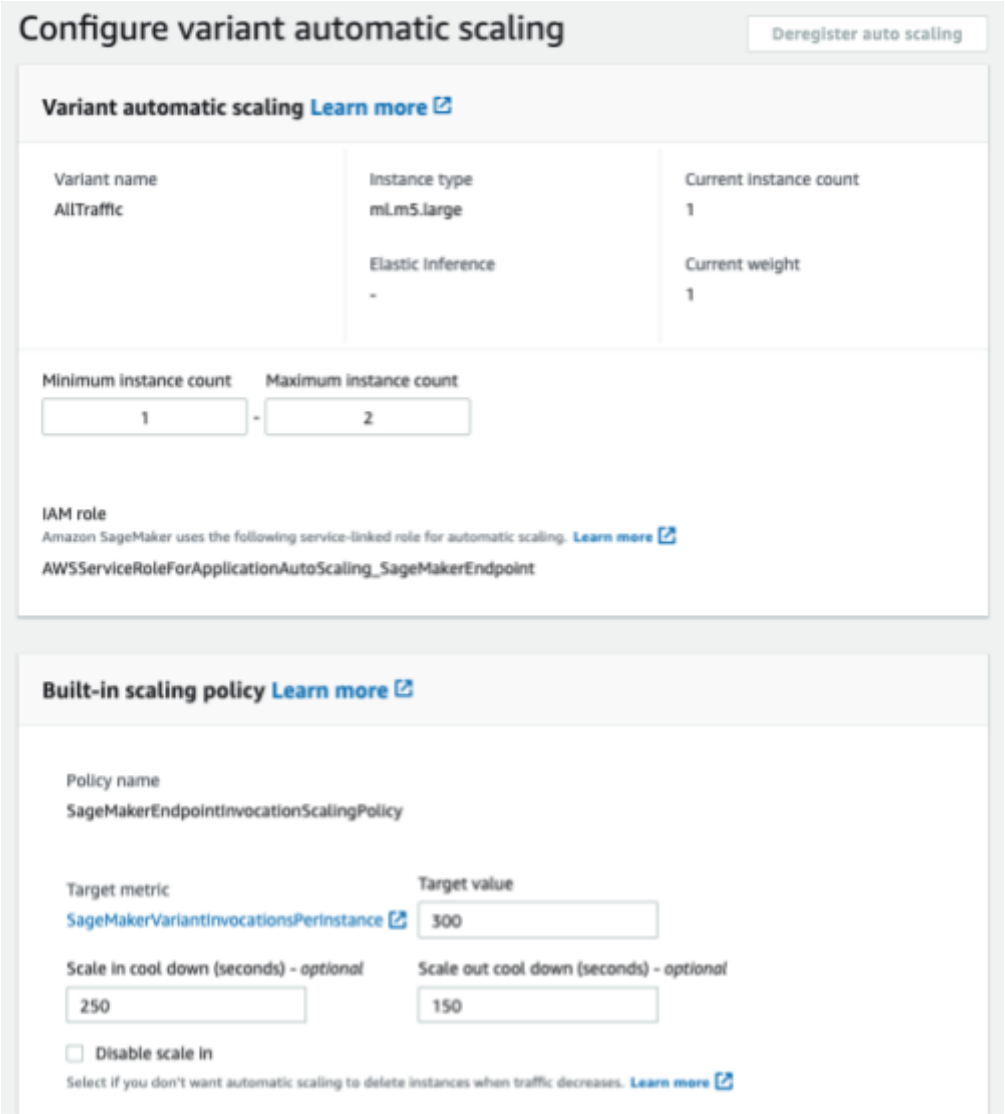
# Concurrency and auto-scaling

Concurrency allows for the lambda function to process multiple requests at once. I don't expect high traffic but, for the sake of the project demo, I've gone for the more expensive provisioned concurrency which makes use of already-initialised instances so is always ready to respond to high traffic.




My provisioned concurrency

I've catered for more high throughput and low latency with auto-scaling. This enables the endpoint to respond to multiple requests. I've added one extra instance (a max of 2). I've followed [this article]() and set the target value of the scaling policy to 300, based on an arbitrary, but low Max RPS of 10 and the recommended safety factor 0.5.




My auto-scaling