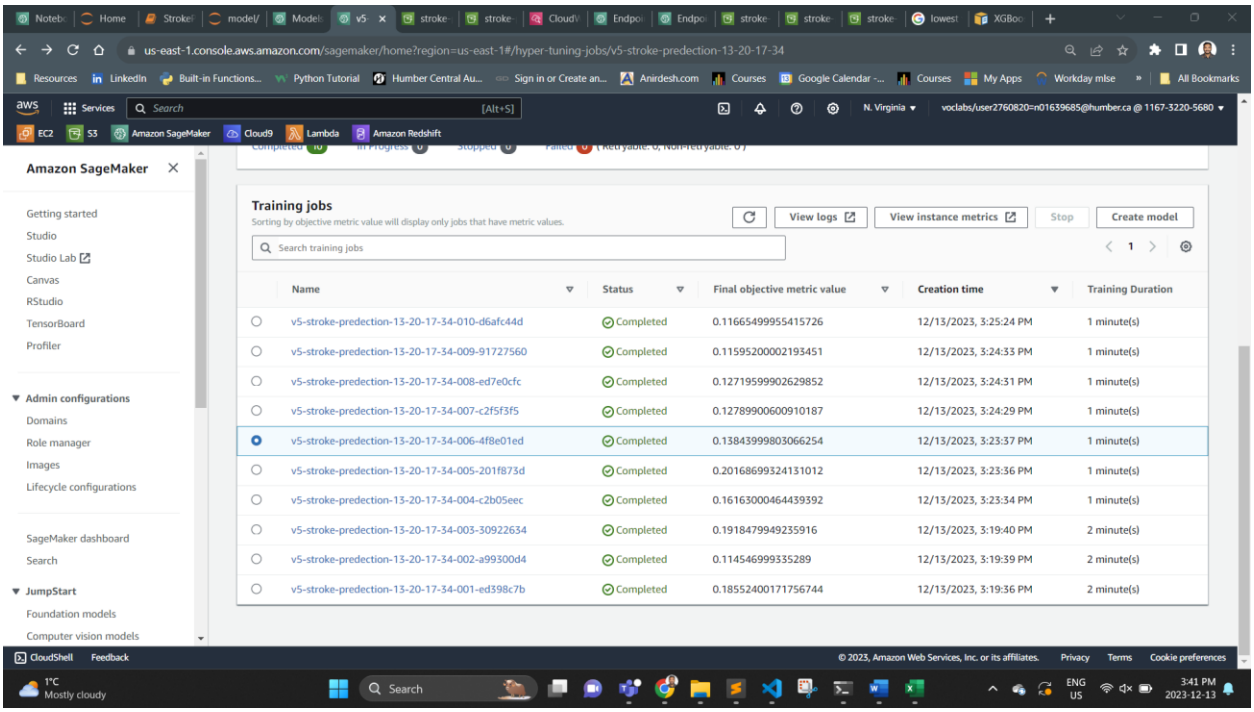
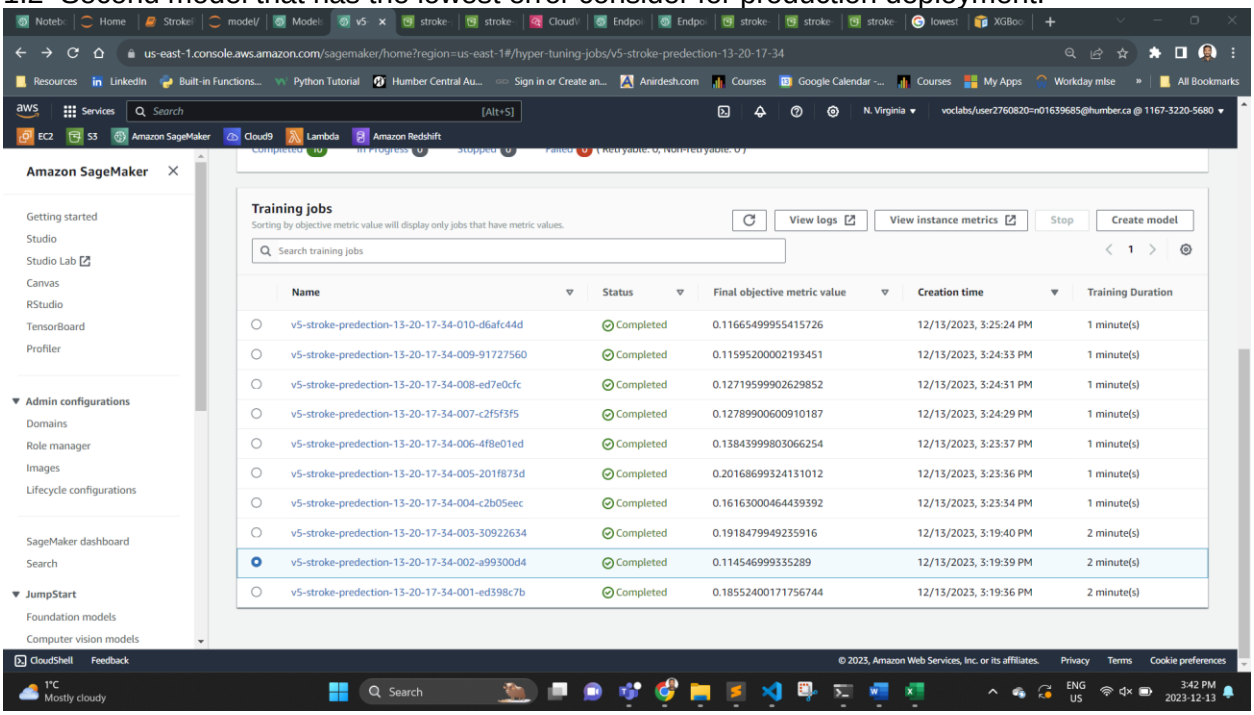
Final Project Report
Machine Learning Cloud Comp - AIGC-5003-0NA

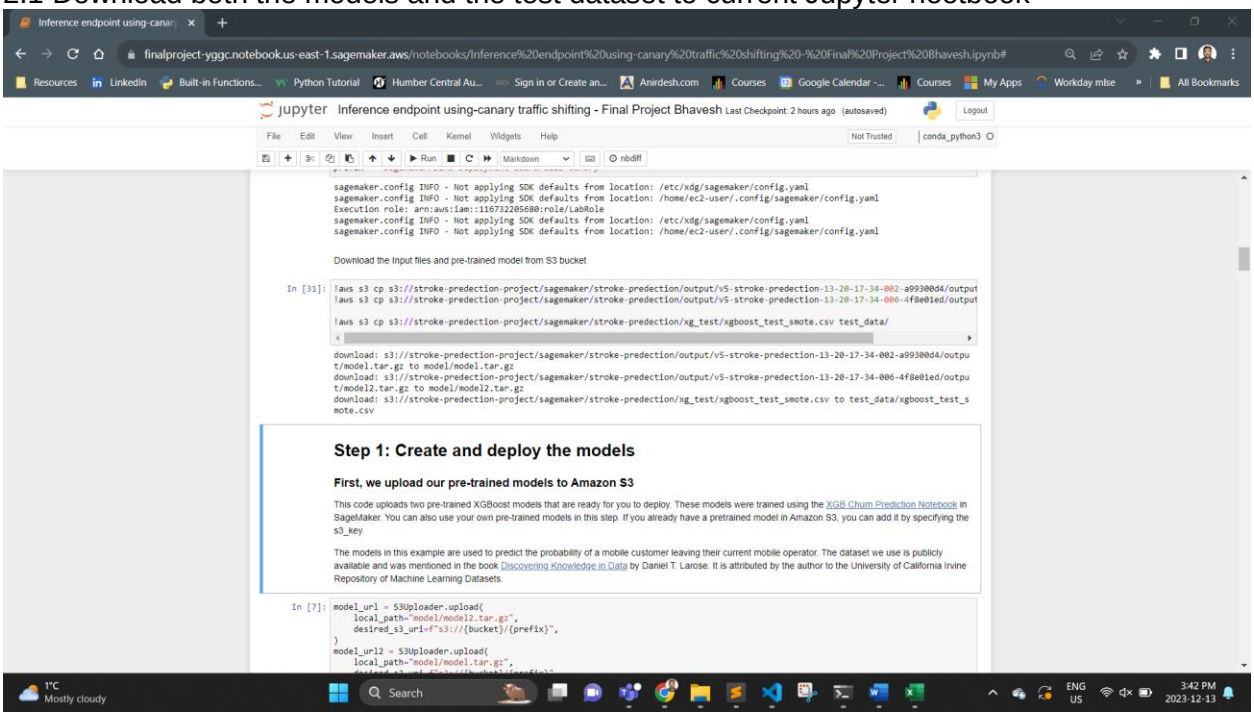
1.1 Check for the trained model from the last assessment and pick for adding it to production.



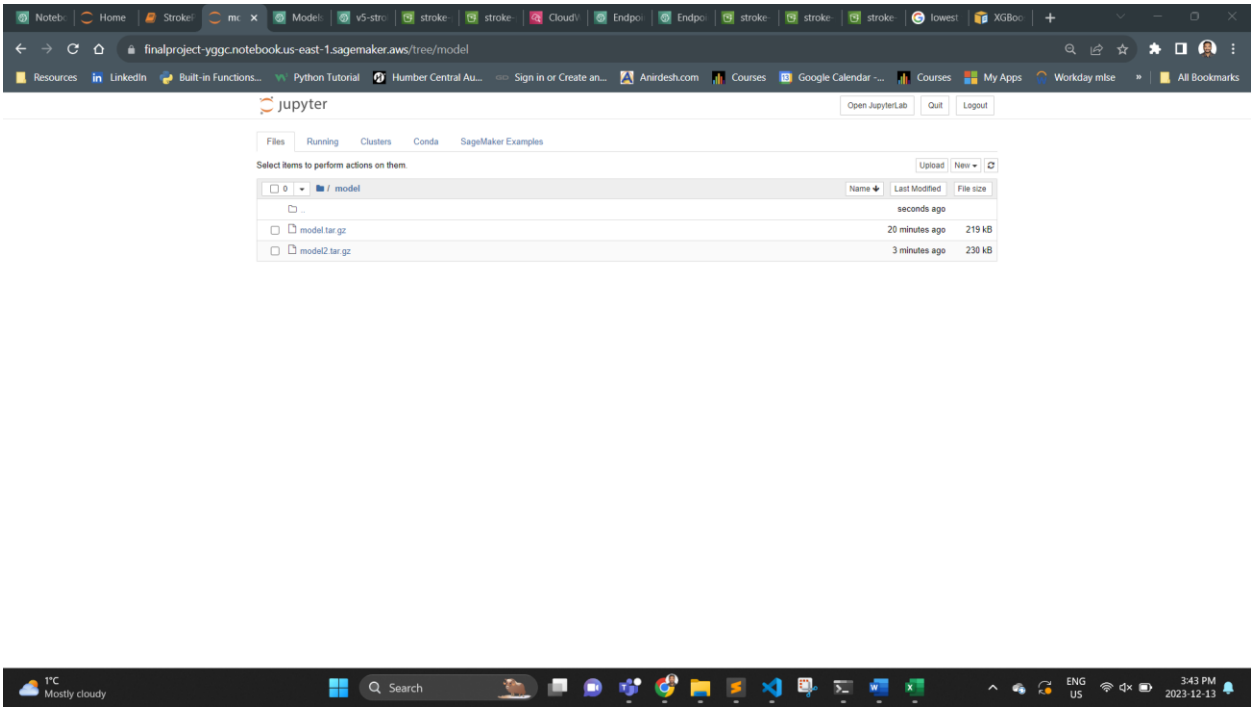
1.2 Second model that has the lowest error consider for production deployment.



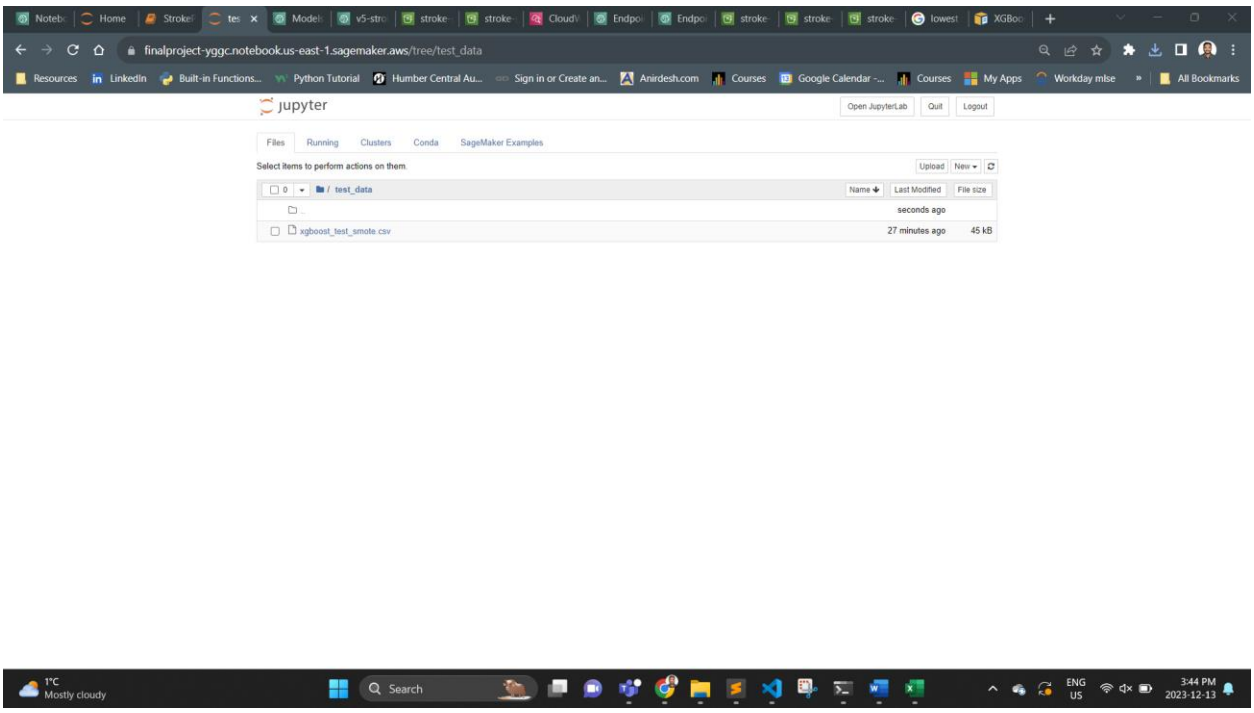
2.1 Download both the models and the test dataset to current Jupyter nootbook



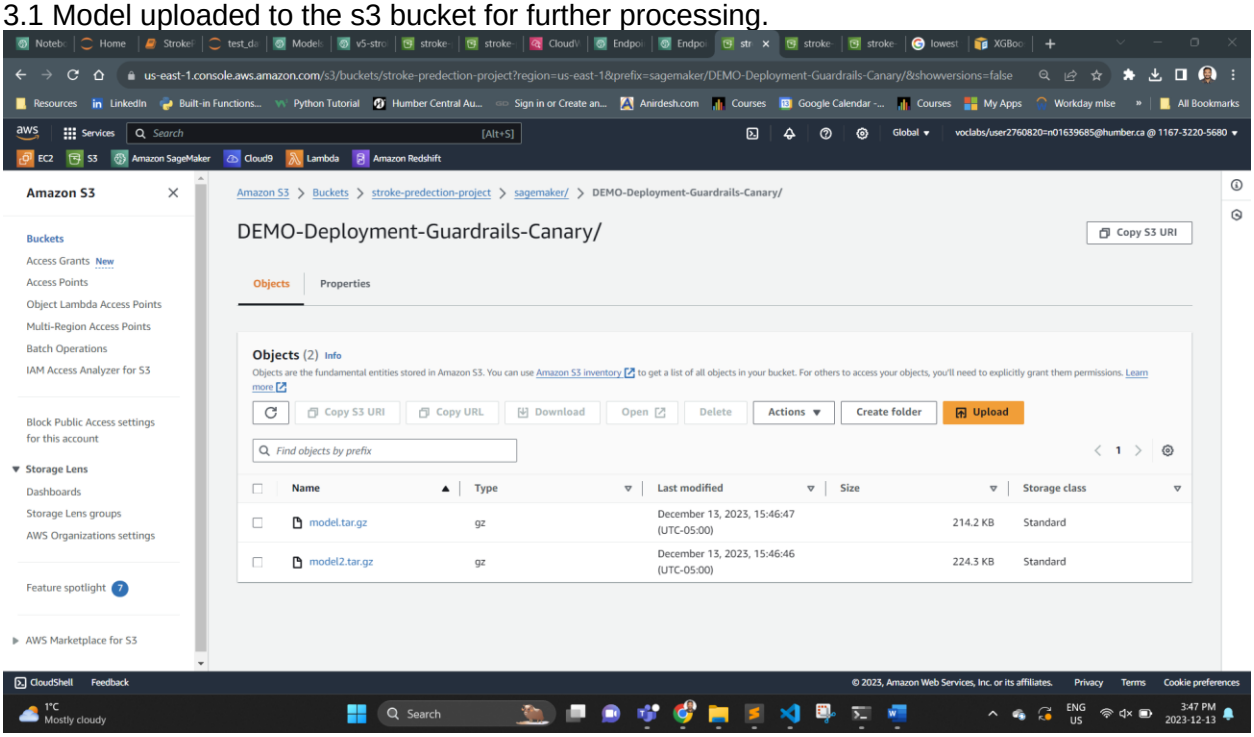
2.2 Models downloaded to the local folder.



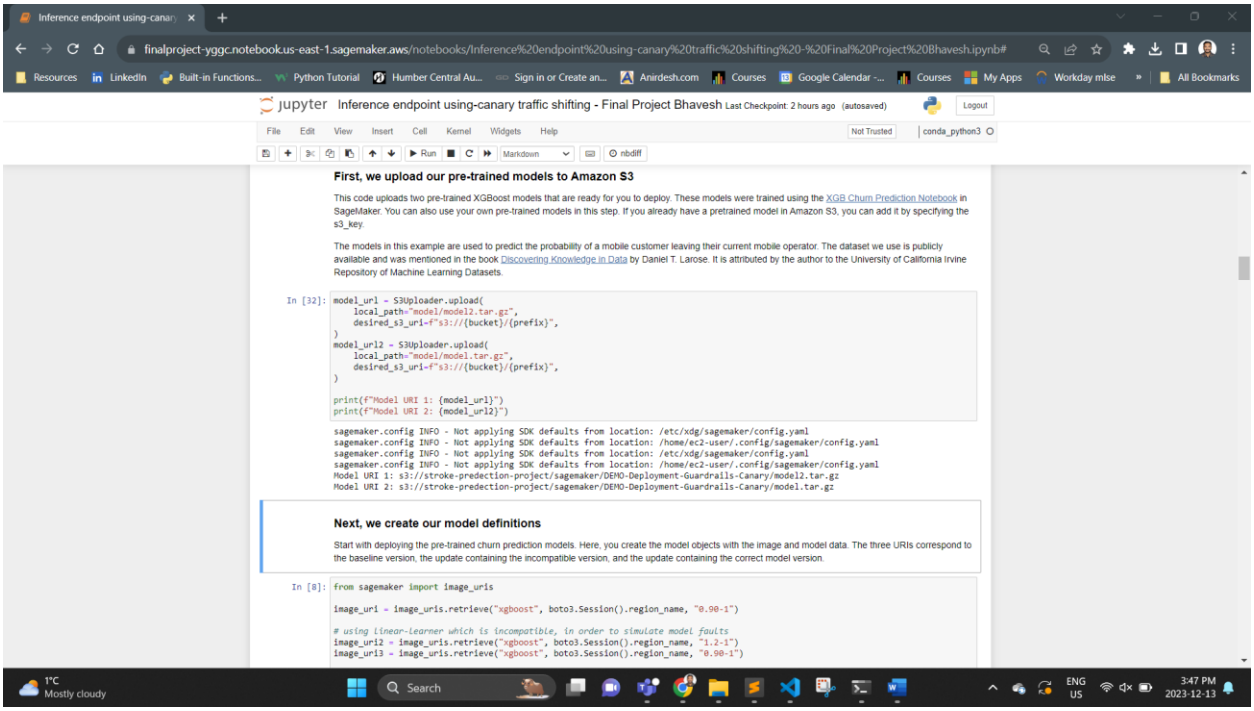
2.3 Test data downloaded to the local folder.



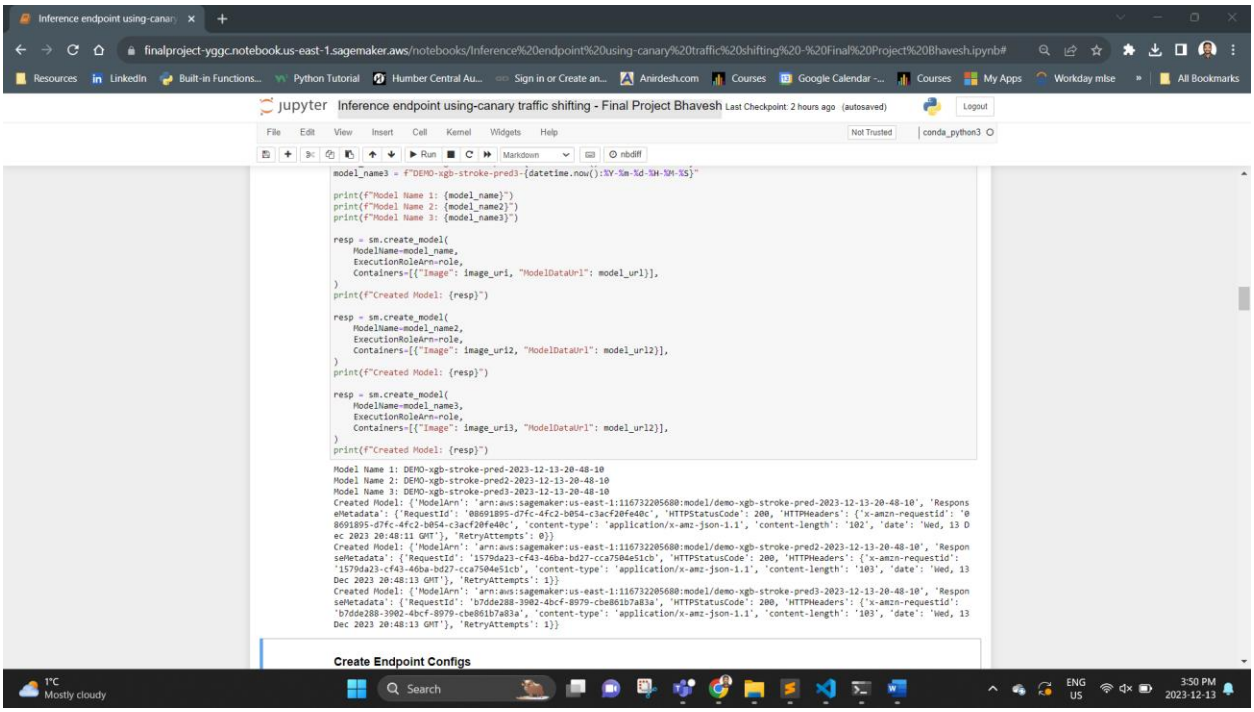
3.1 Model uploaded to the s3 bucket for further processing.



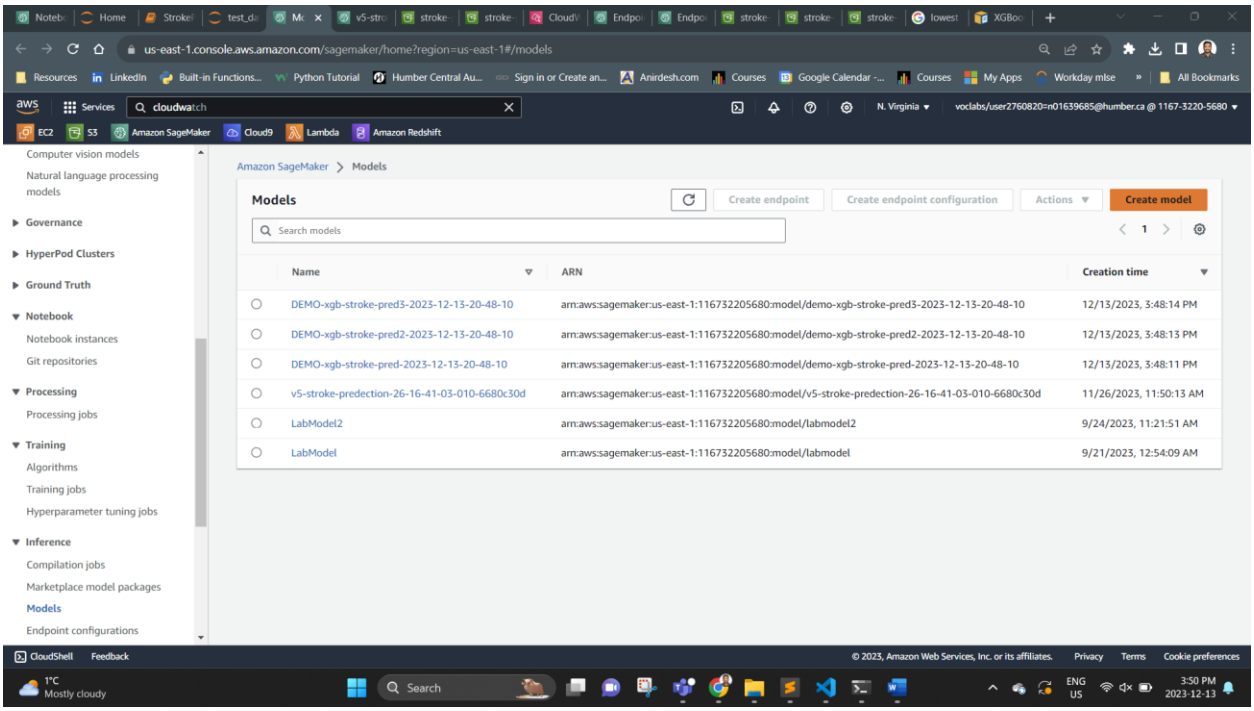
3.2 Save model tar.gz file to s3 bucket.



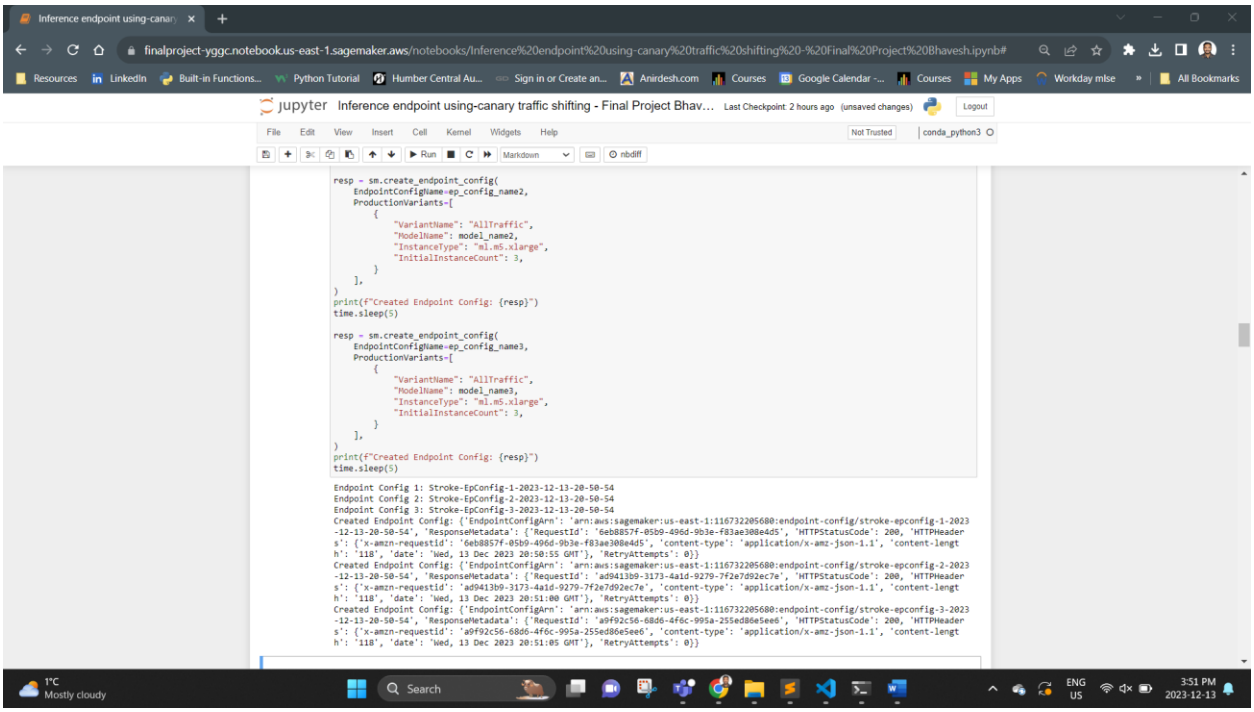
4.1 Model Creation with different versions of Xgboost.



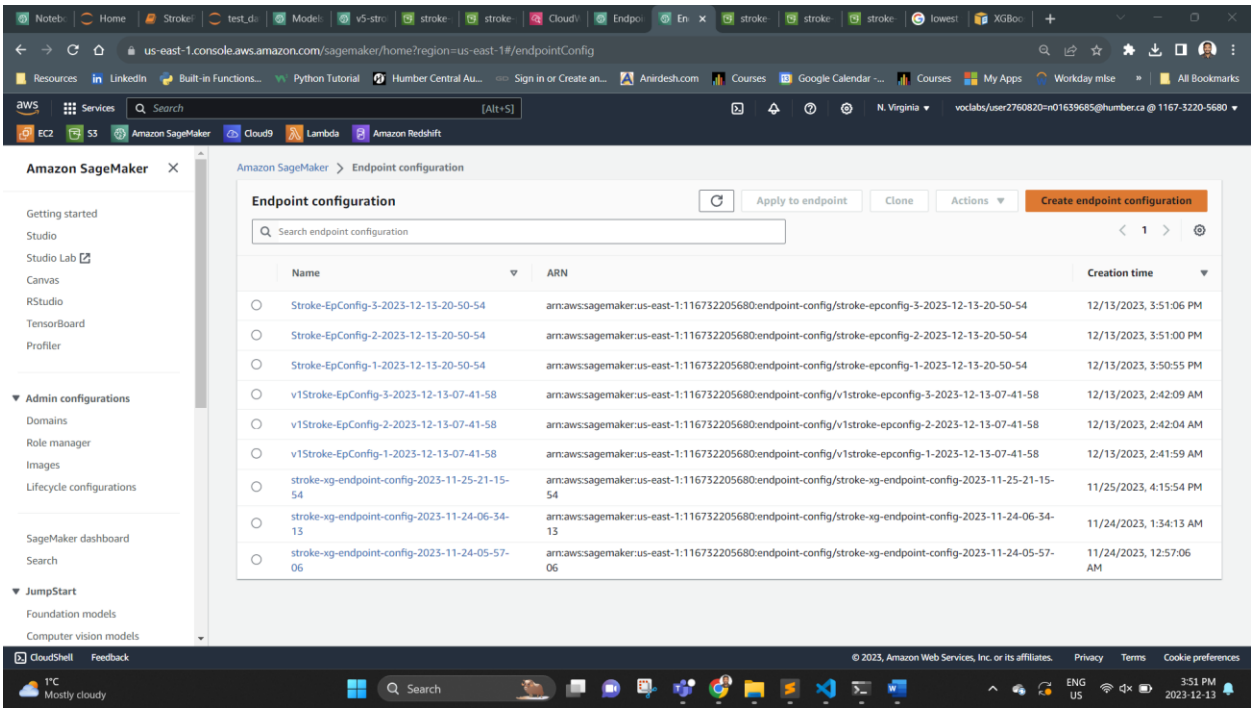
4.2 Model Created in SageMaker.



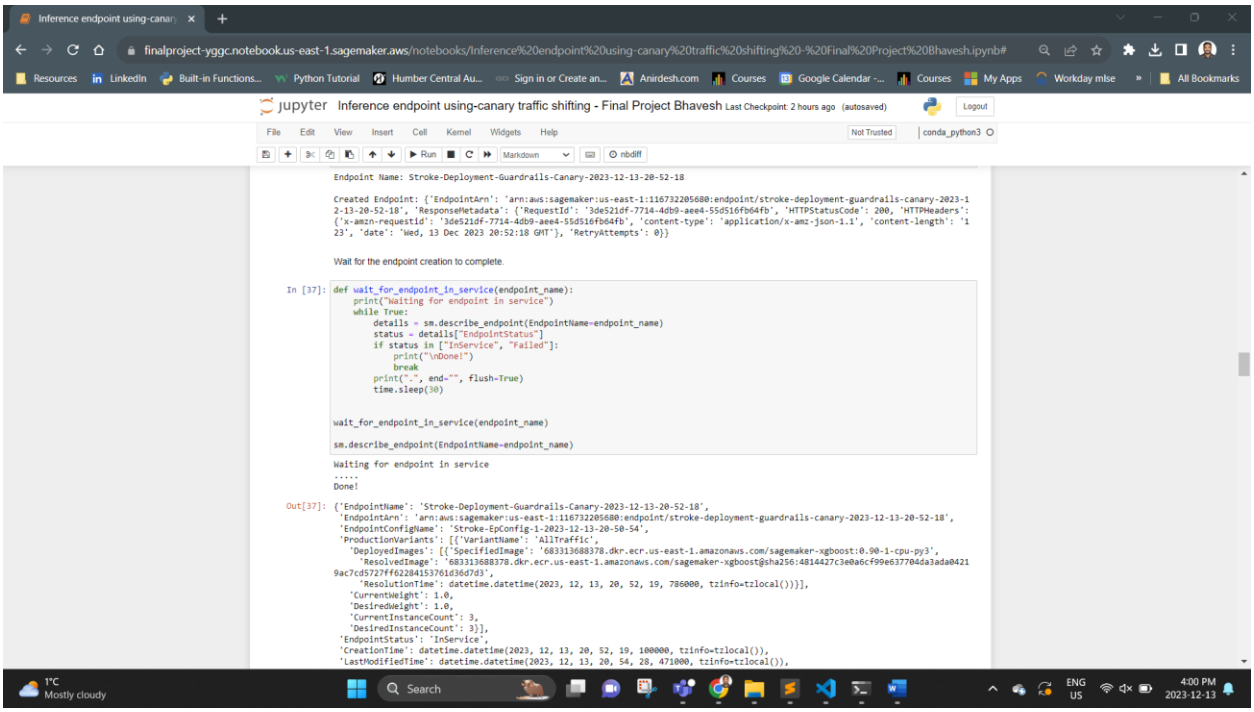
5.1 Endpoint Configuration with code



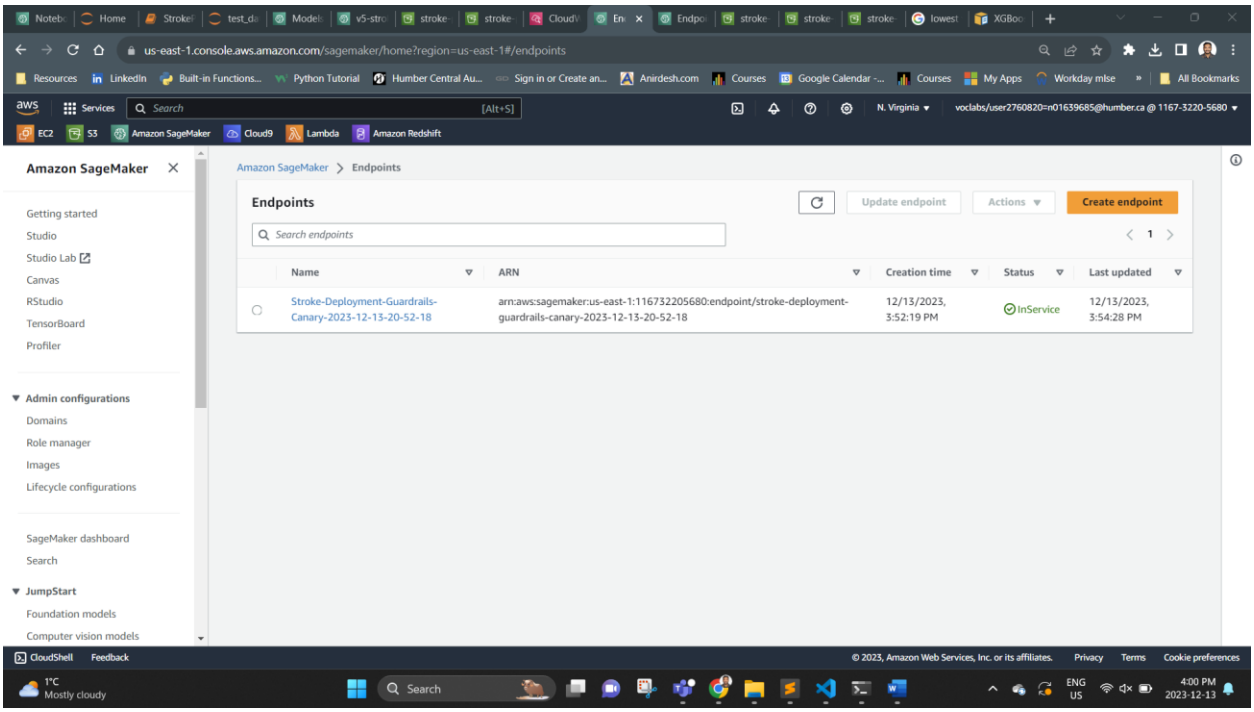
5.2 Endpoint Configuration reflected in SageMaker.



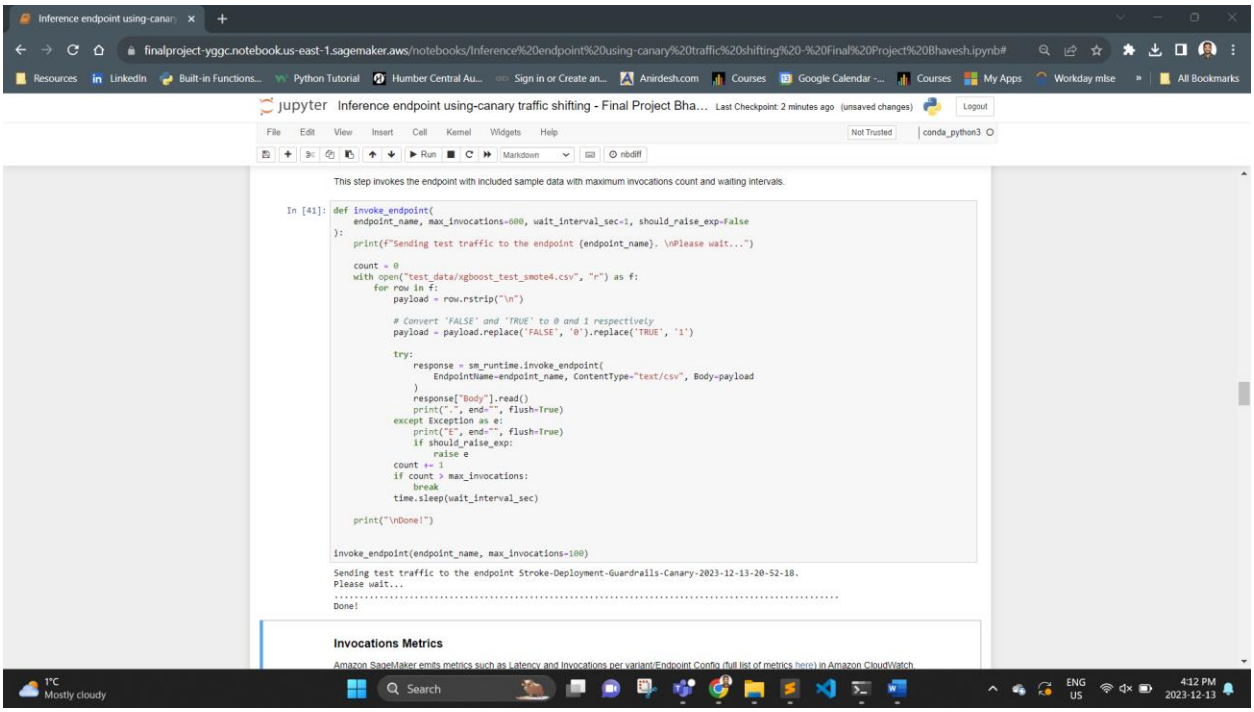
6.1 Creating an Endpoint from the configuration file.



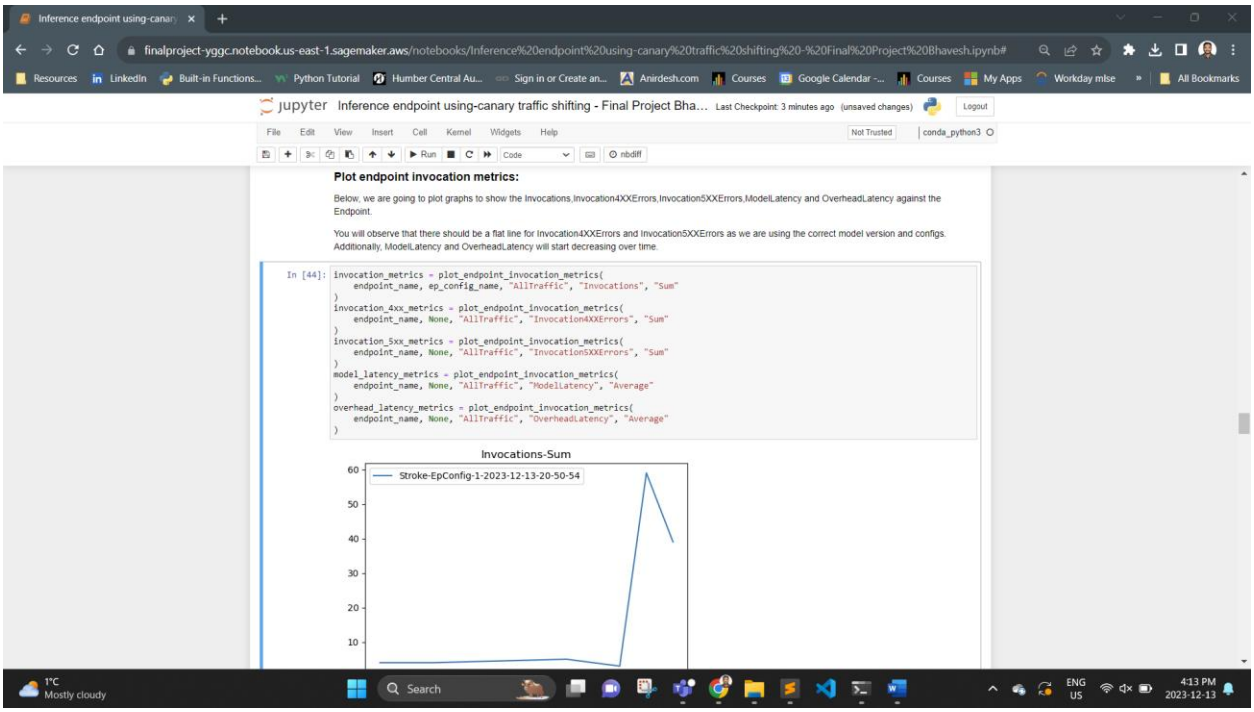
6.3 Endpoint creation confirmation in SageMaker.



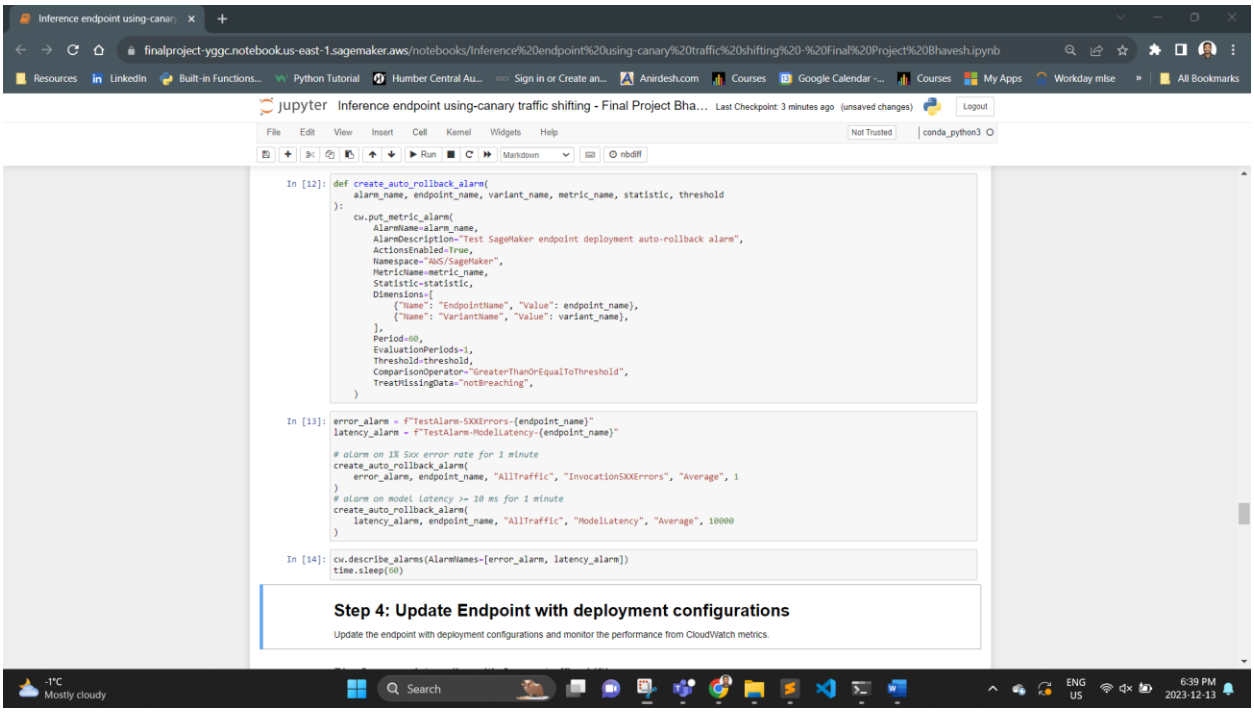
7. Successfully invoked the endpoint for the production model 1st time.



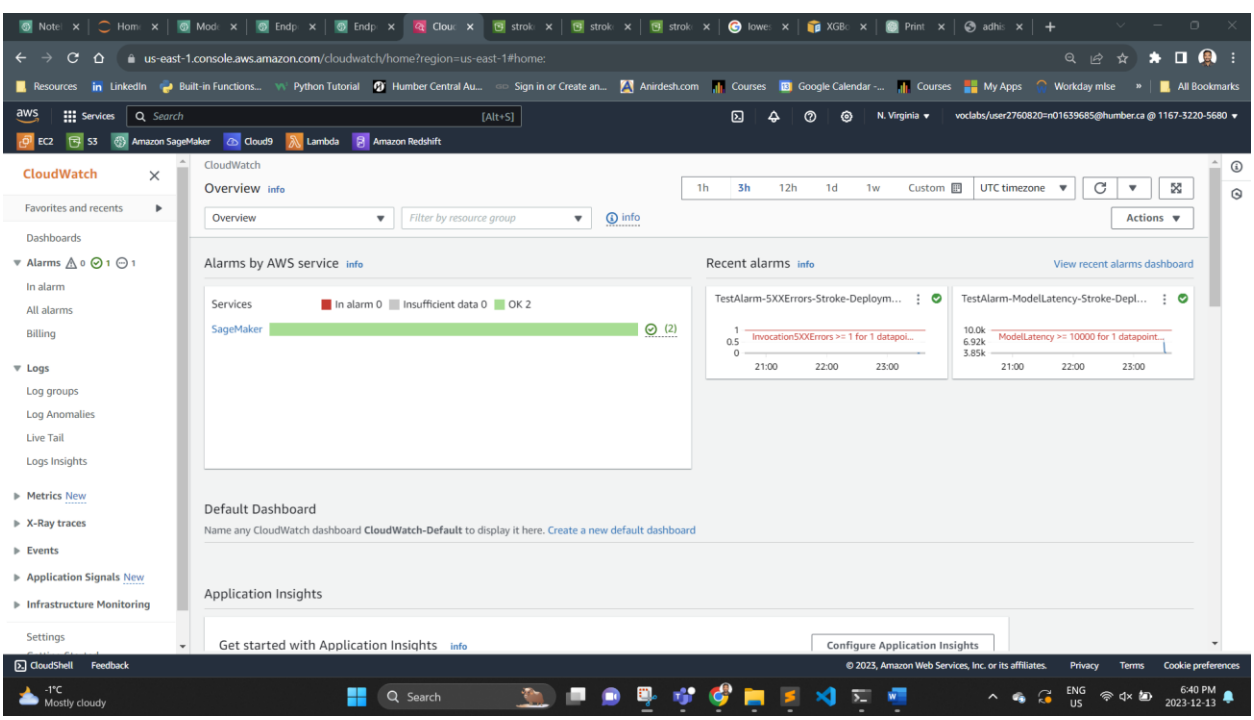
8. Plot the invocation matrix from the endpoint.



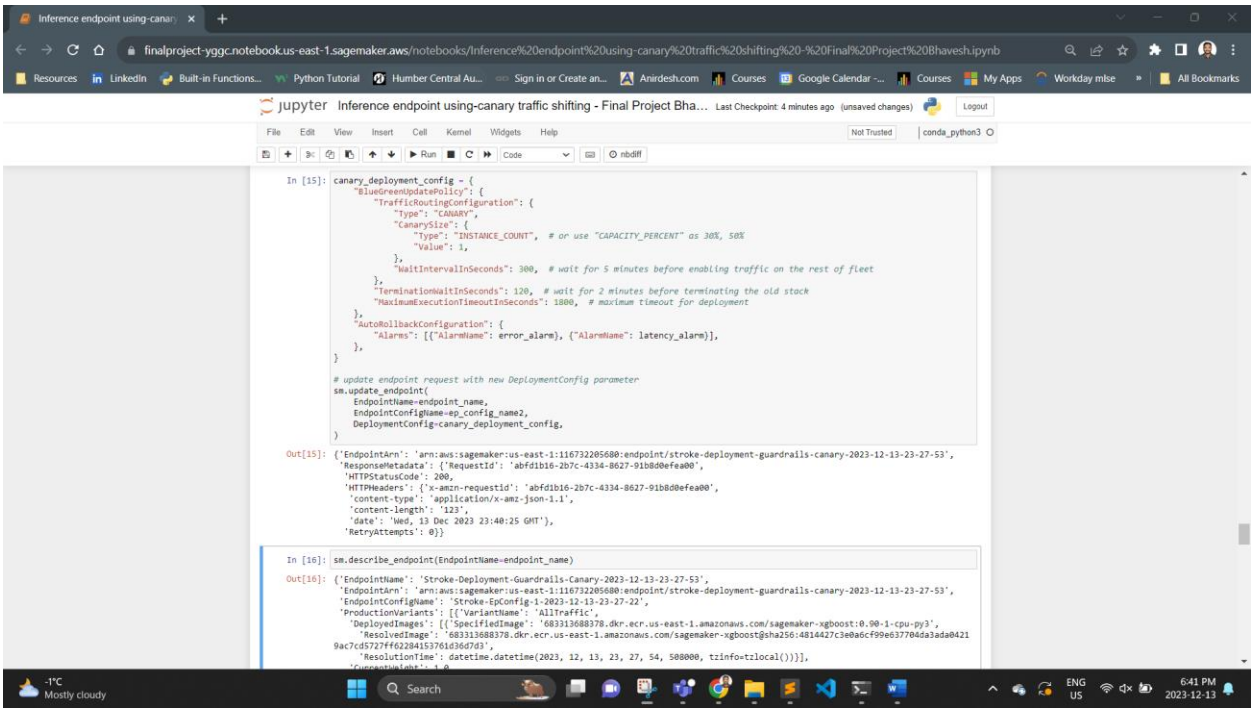
9.1 Set Cloudwatch Alarm for AutoRollback and update the endpoint with new endpoint config to make it fail.



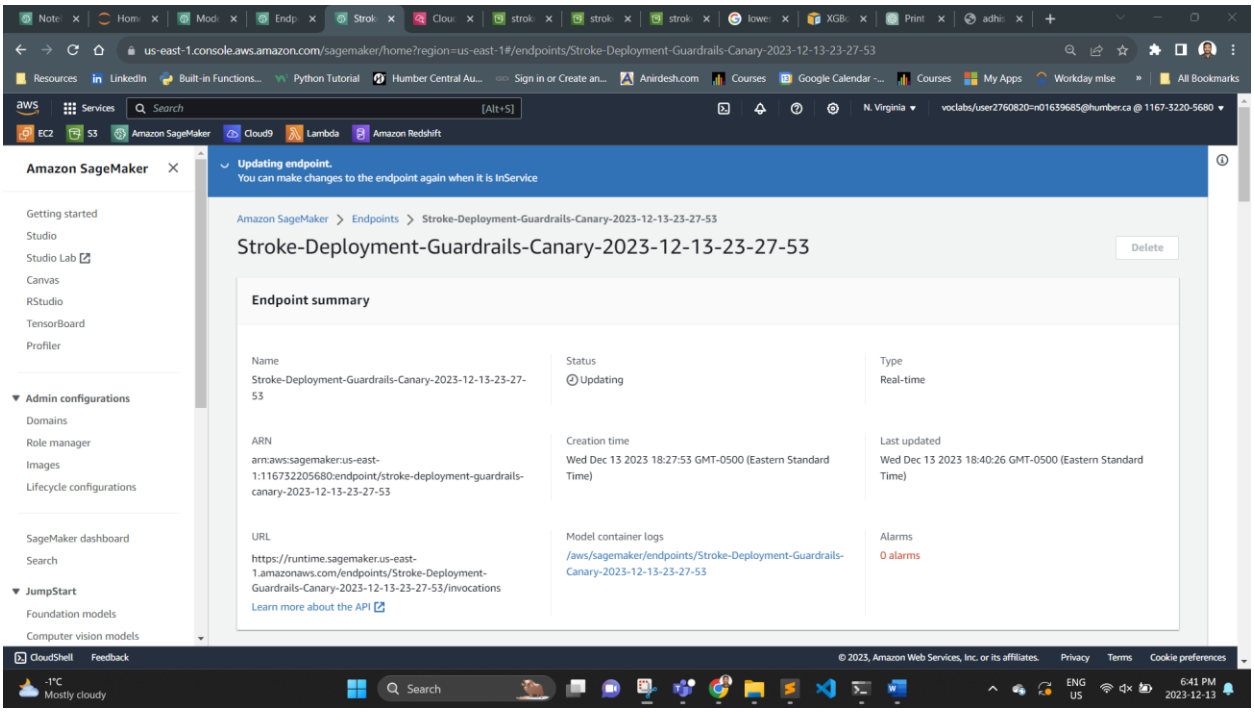
9.2 Check the configurations in Cloudwatch.



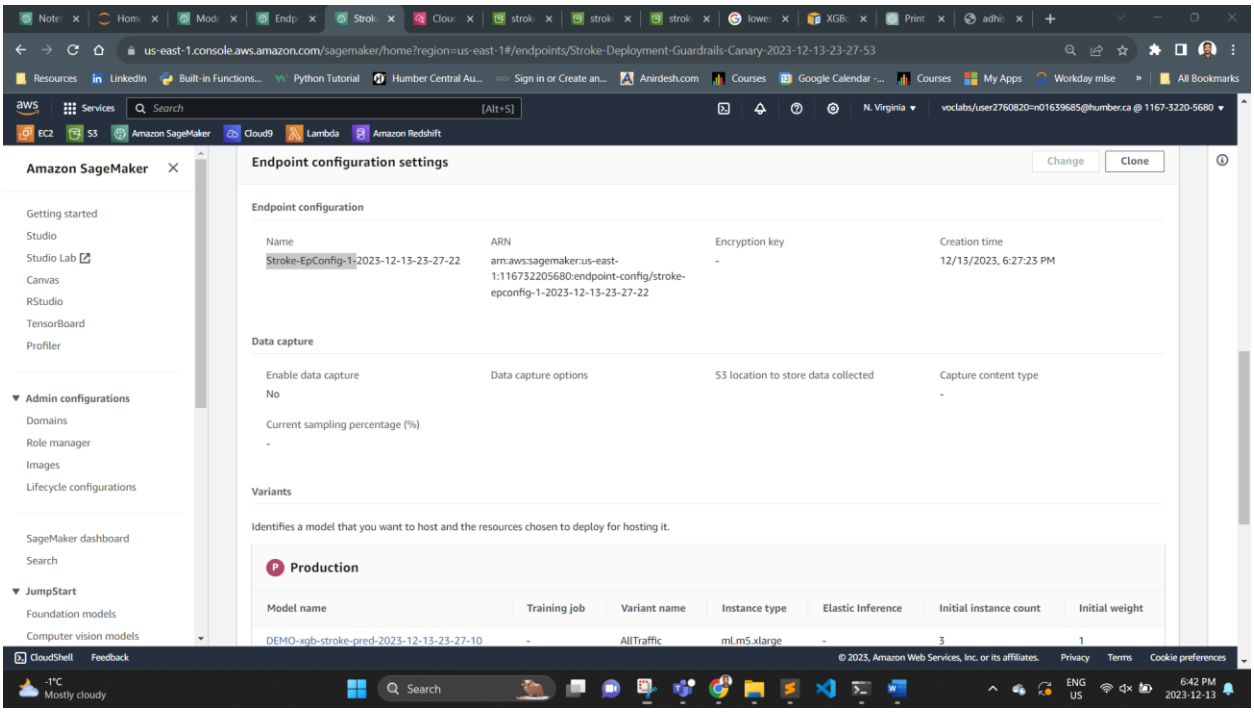
10.1 Config for updating the endpoint with the new model.



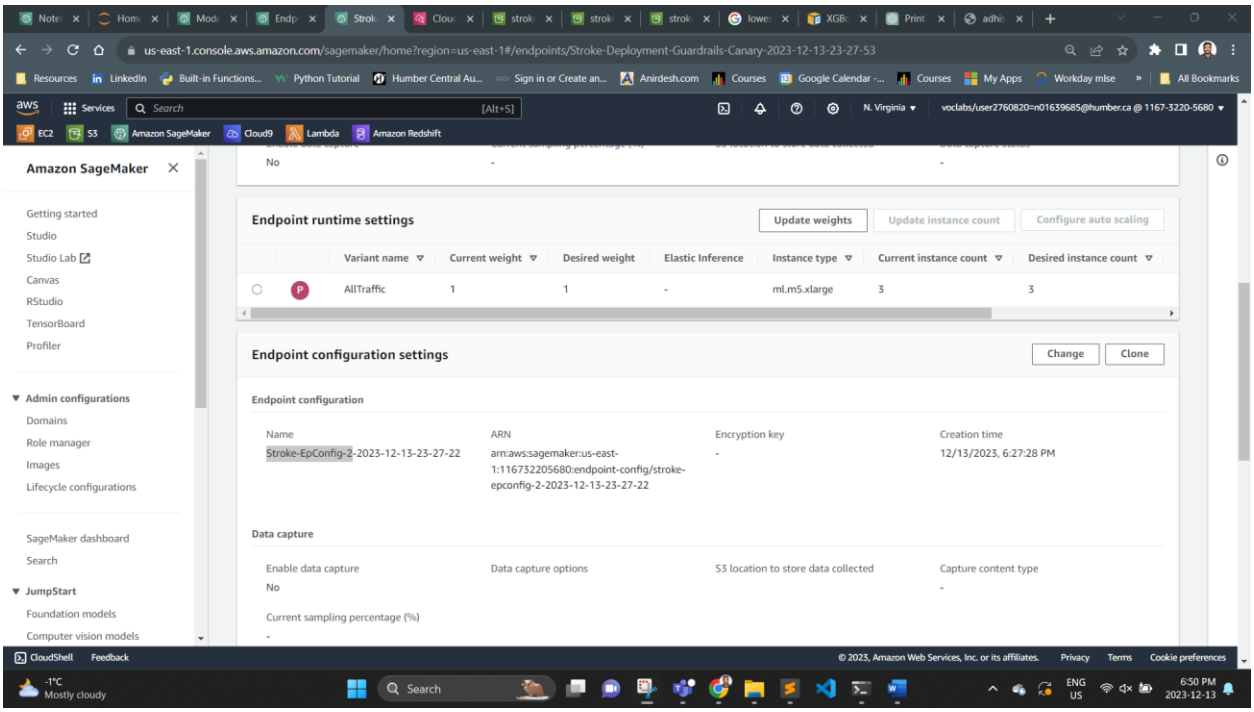
10.2 Endpoint went on Updating mode as it is applying new configurations.



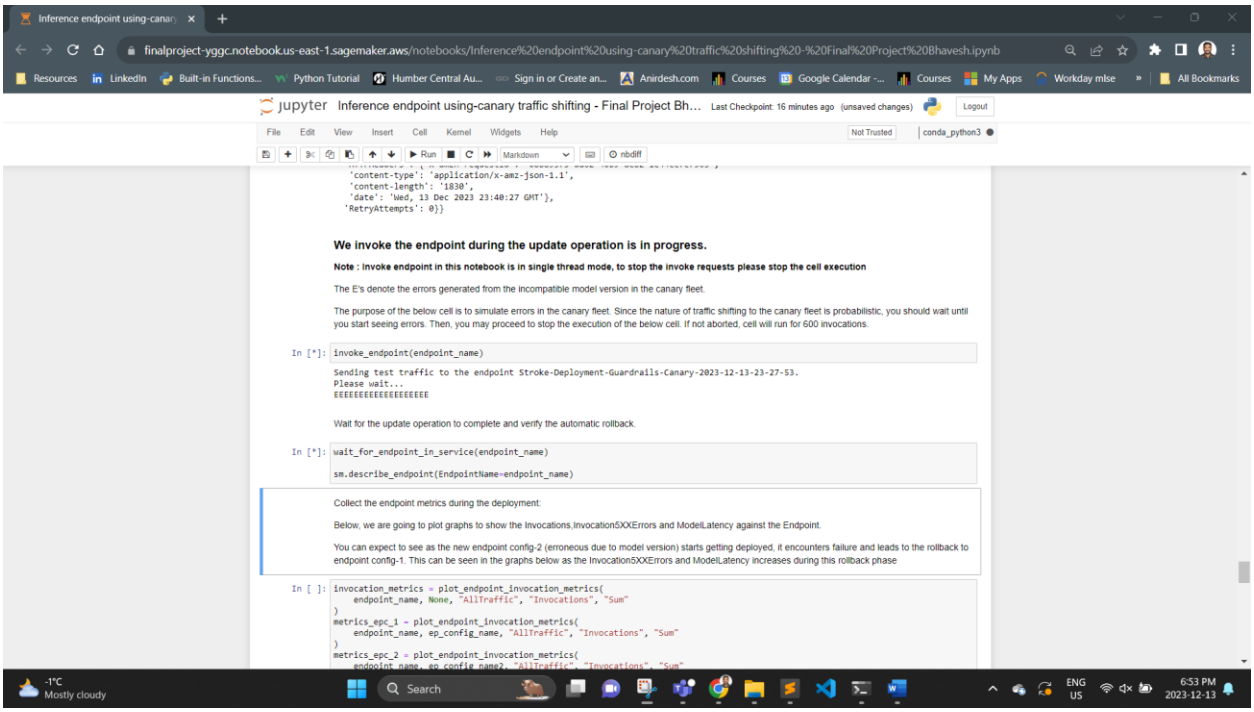
10.3 Before the endpoint was using endpoint config 1.



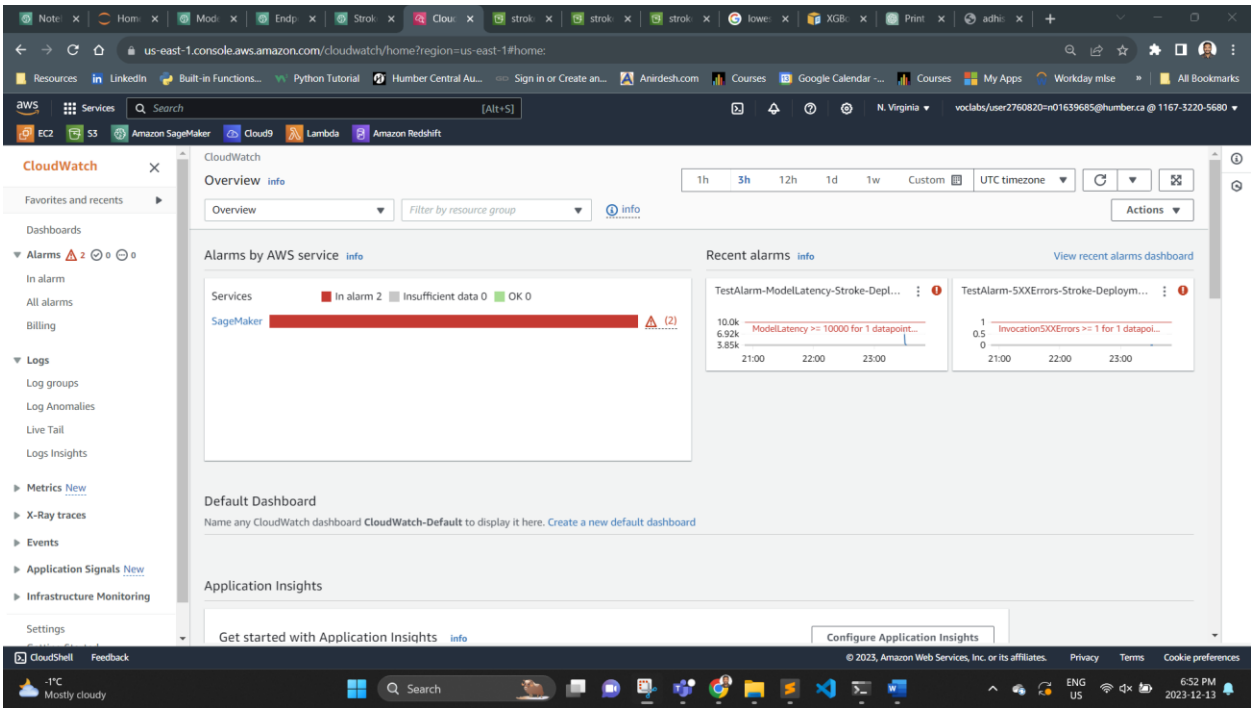
10.4 After the update it would switch to endpoint config 2.



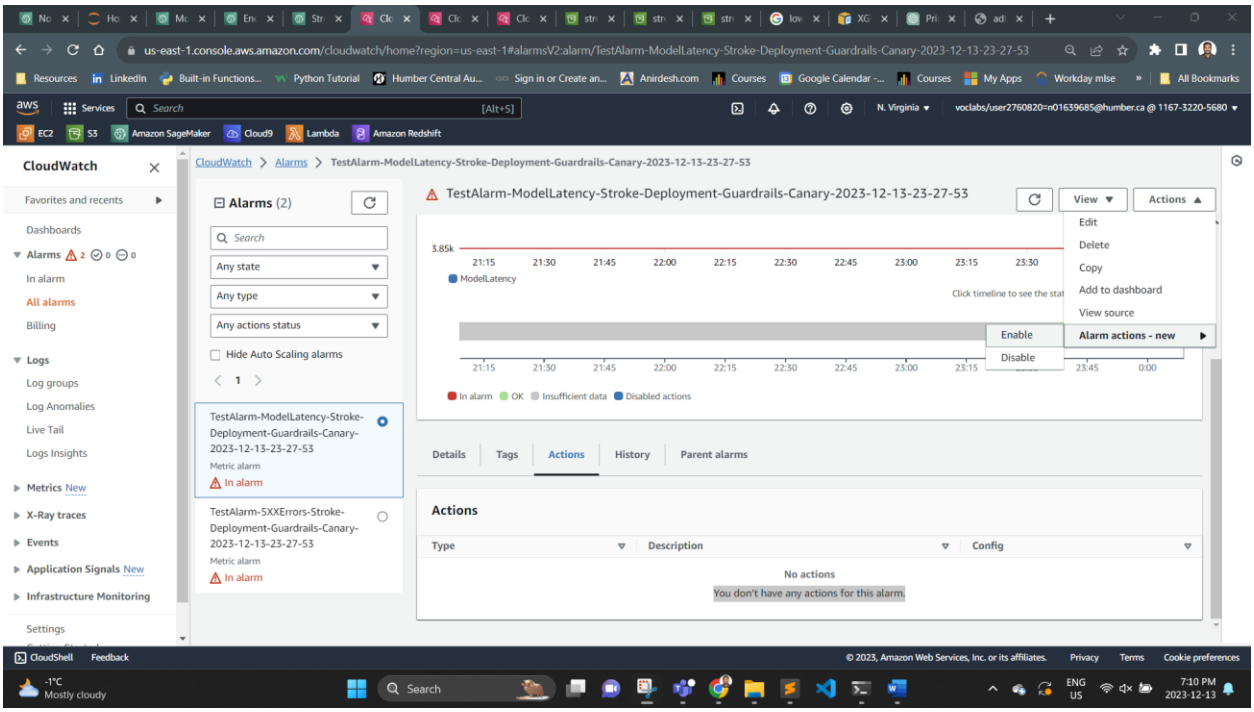
11.1 Error on the second model execution.



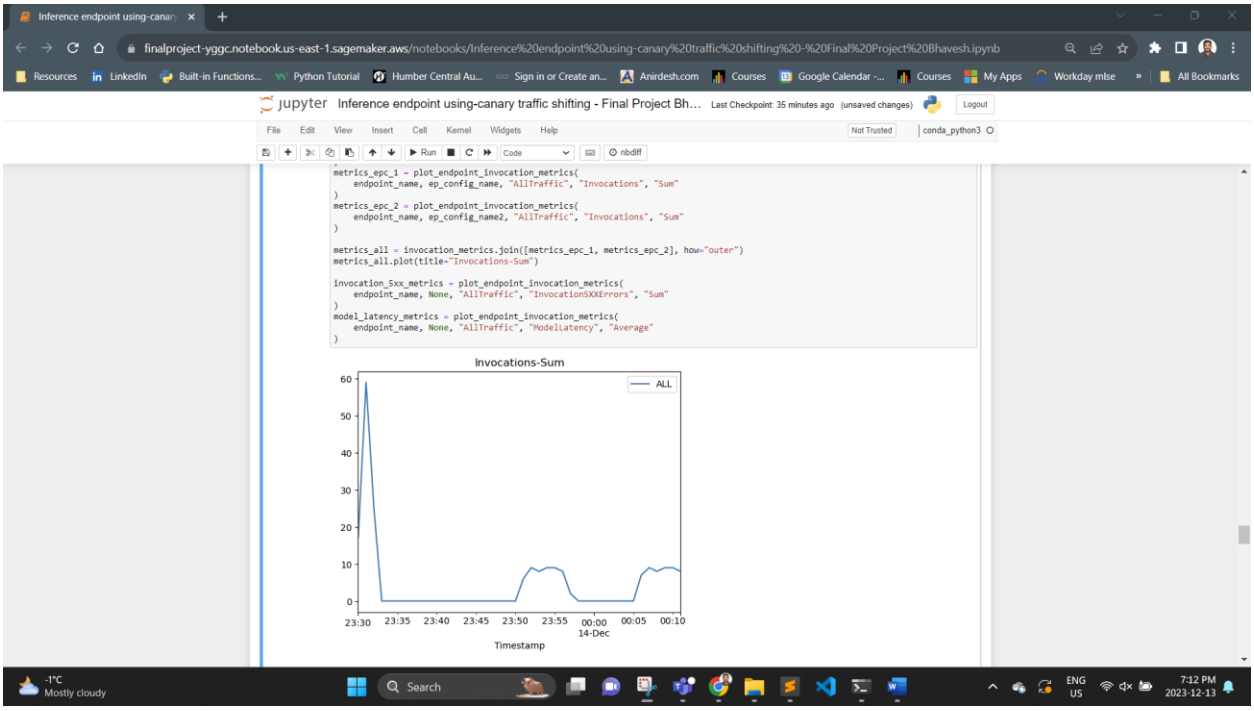
11.2 Ring an alarm that there was a error was occurred due to server version incompatibility.



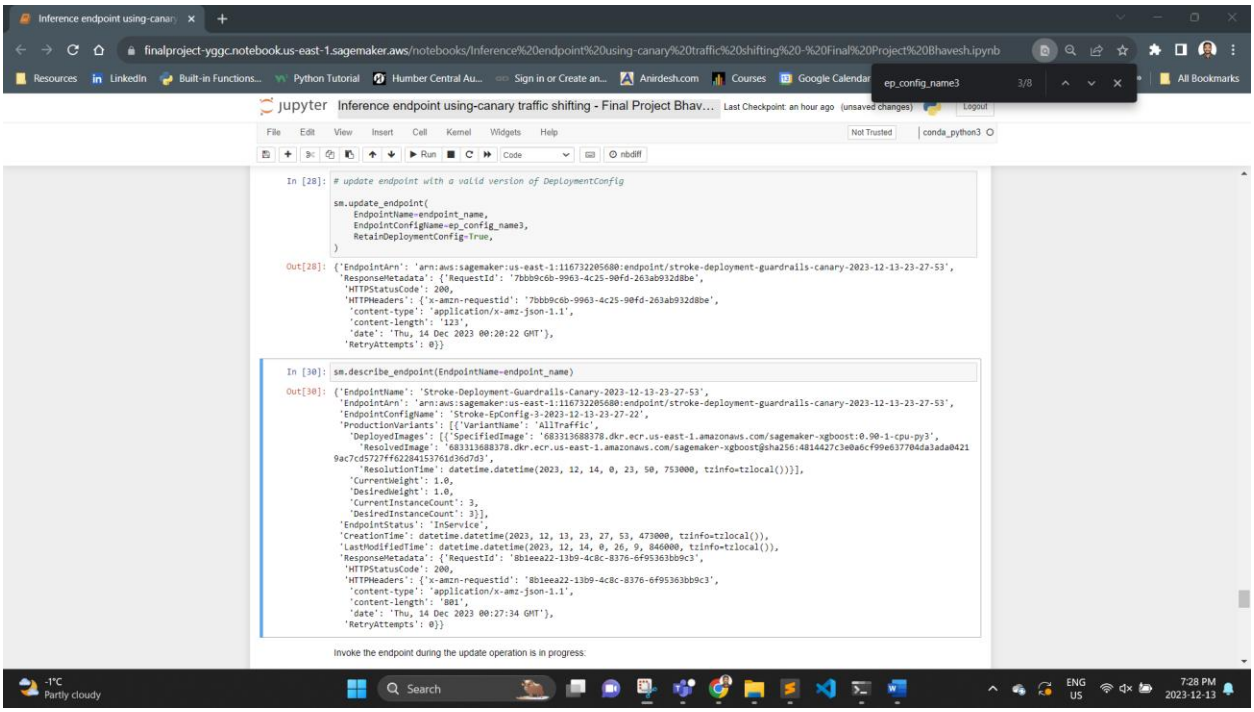
11.3 There is a permission issue that is the reason that there are no actions that can be performed on the alarm. USING VOCLABS.



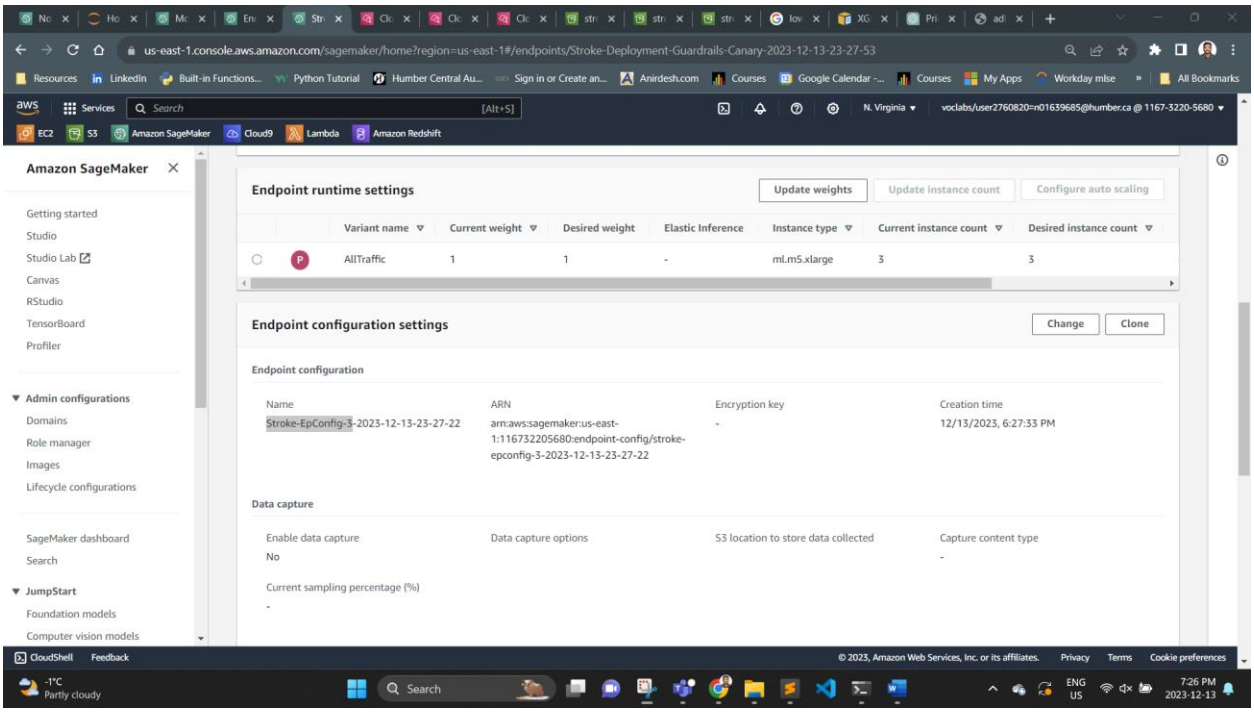
11.4 Check the matrix.



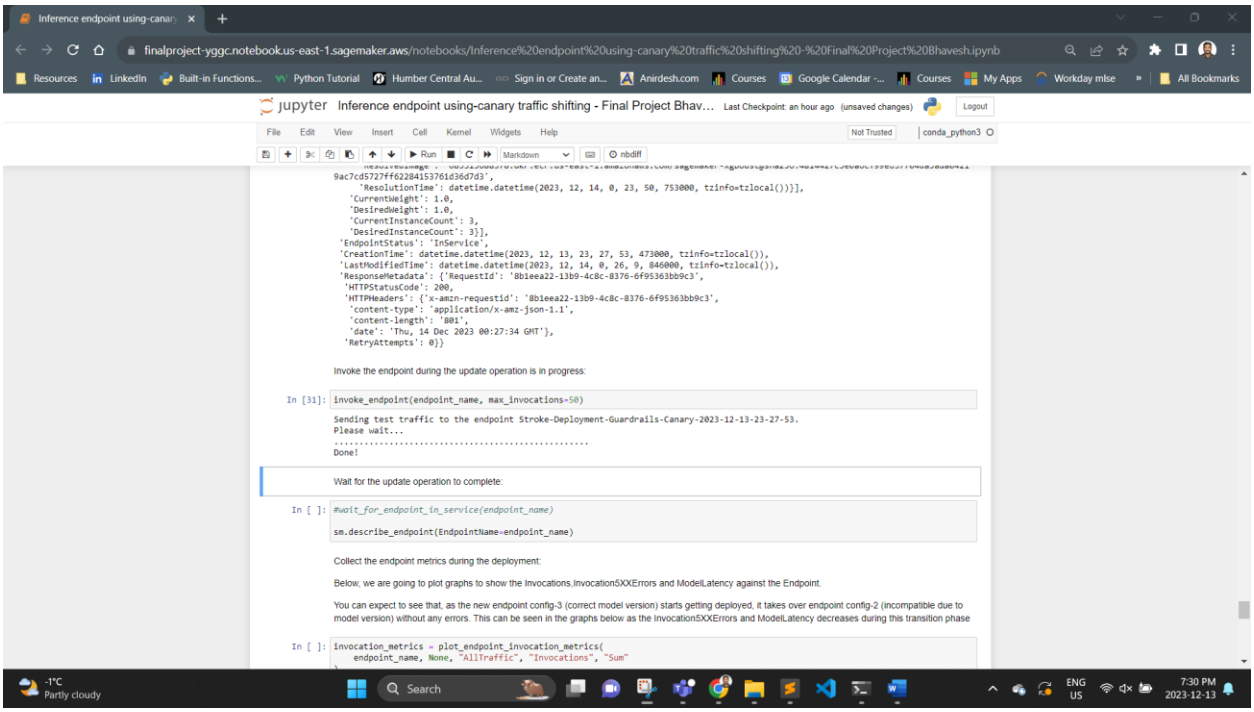
12.1 Update the endpoint with endpoint config 3.



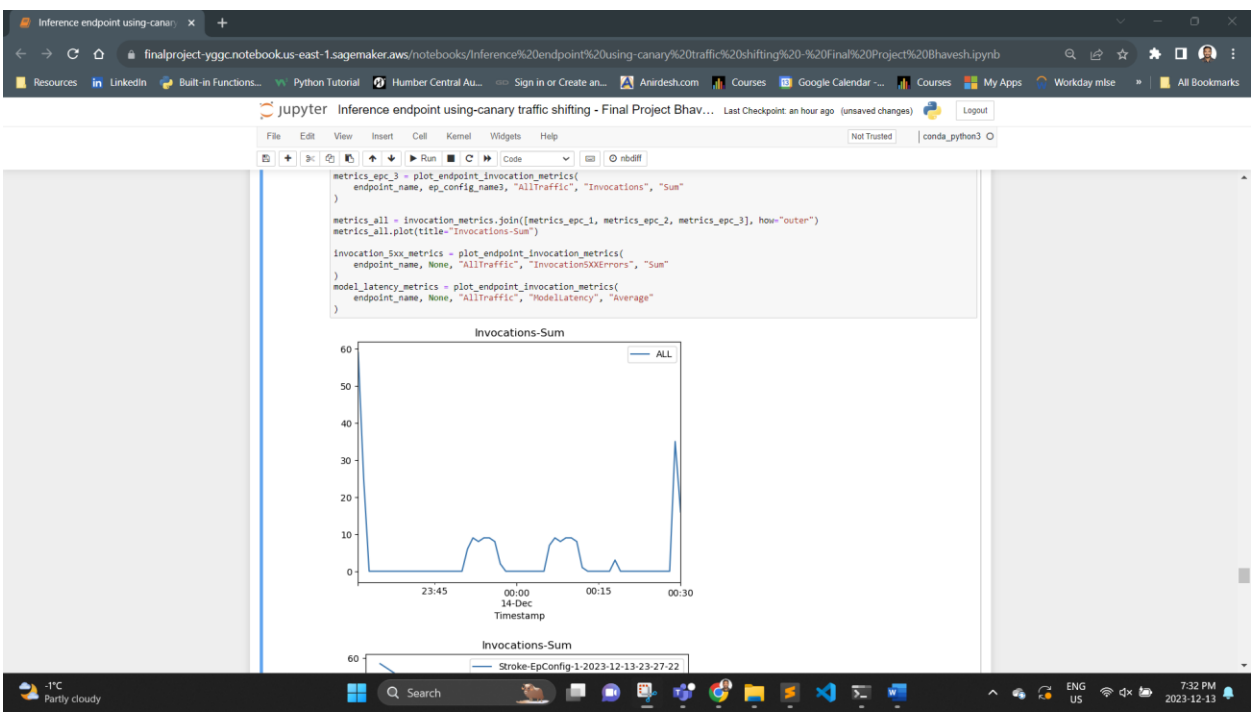
12.2 Verify the update of the endpoint config 3 in SageMaker.



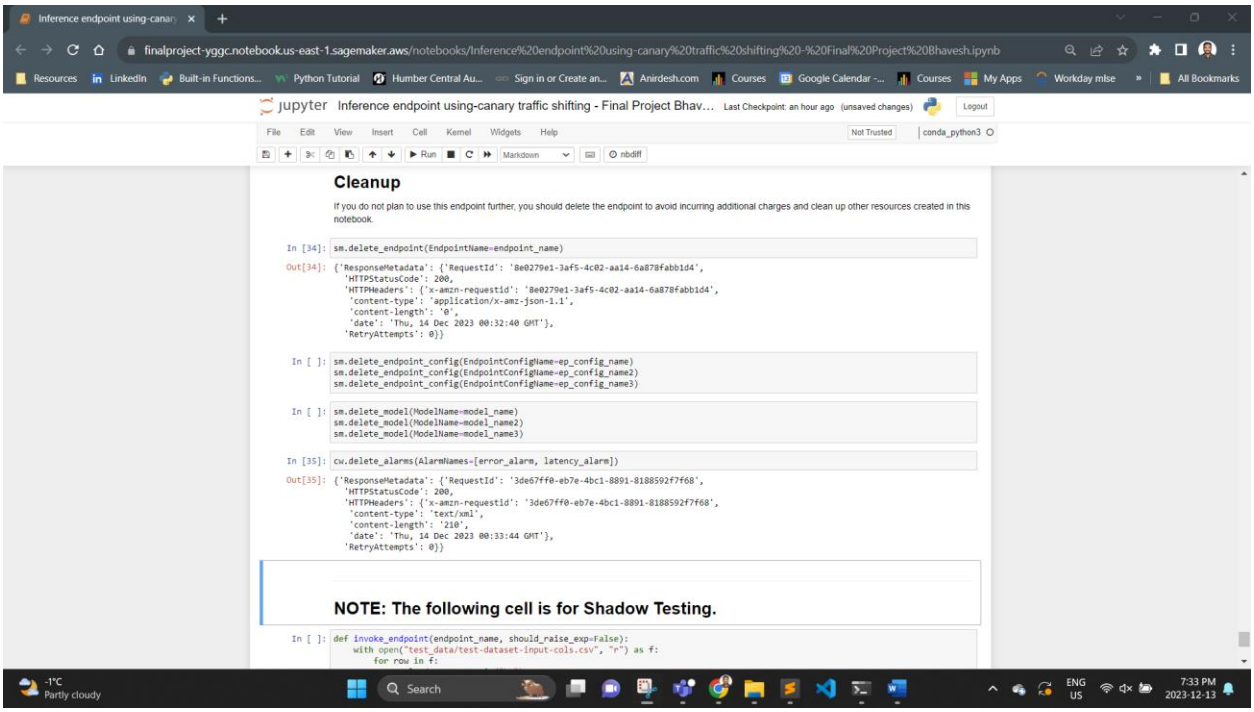
12.3 Invoke the endpoint configuration 3 and check the server is taking request without any error.



12.4 Endpoint Matrix when the endpoint is in service.



13. Delete the endpoint and the alarm that was created for serving the request and model testing.



Cleanup

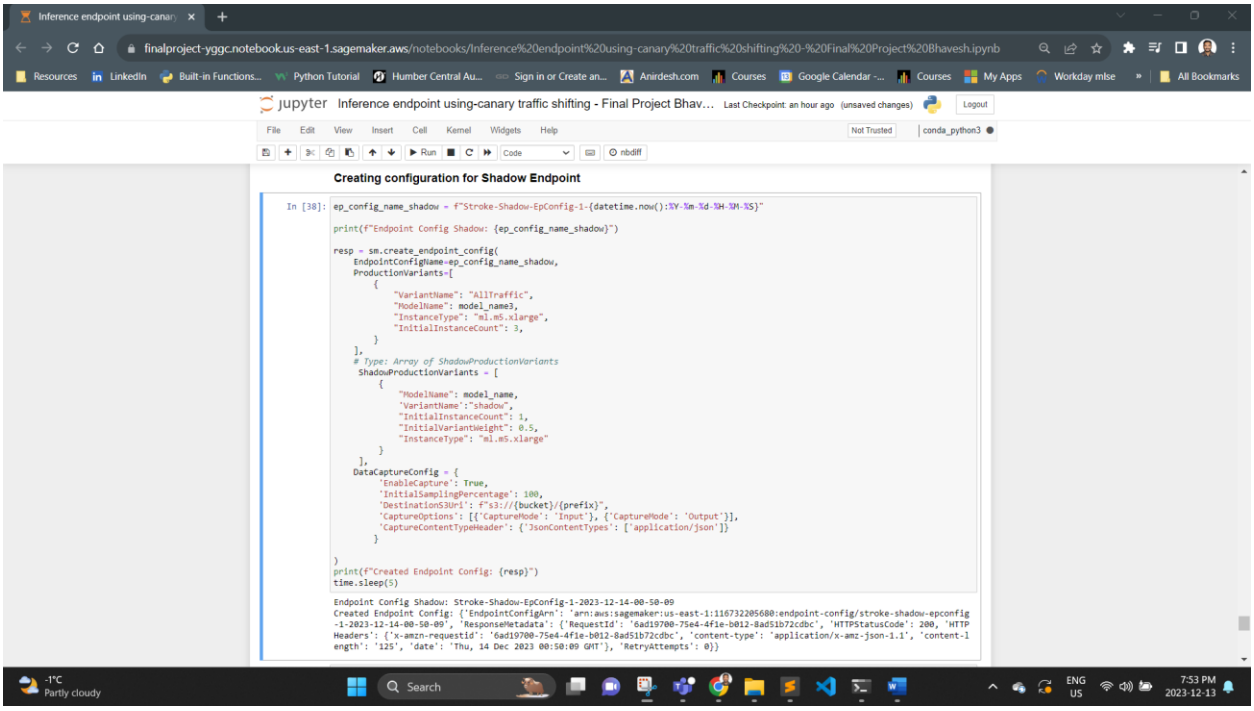
If you do not plan to use this endpoint further, you should delete the endpoint to avoid incurring additional charges and clean up other resources created in this notebook.

```
In [34]: sm.delete_endpoint(EndpointName=endpoint_name)
Out[34]: {'ResponseMetadata': {'RequestId': '8e0279e1-3af5-4c02-aa14-6a878fabb1d4',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '8e0279e1-3af5-4c02-aa14-6a878fabb1d4',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '0',
   'date': 'Thu, 14 Dec 2023 00:32:40 GMT'},
  'RetryAttempts': 0}}

In [ ]: sm.delete_endpoint_config(EndpointConfigName=ep_config_name)
        sm.delete_endpoint_config(EndpointConfigName=ep_config_name2)
        sm.delete_endpoint_config(EndpointConfigName=ep_config_name3)

In [ ]: sm.delete_model(ModelName=model_name)
        sm.delete_model(ModelName=model_name2)
        sm.delete_model(ModelName=model_name3)

In [35]: cw.delete_alarms(AlarmNames=[error_alarm, latency_alarm])
Out[35]: {'ResponseMetadata': {'RequestId': '3de67ff0-eb7e-4bc1-8891-8188592f7f68',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '3de67ff0-eb7e-4bc1-8891-8188592f7f68',
   'content-type': 'text/xml',
   'content-length': '210',
   'date': 'Thu, 14 Dec 2023 00:33:44 GMT'},
  'RetryAttempts': 0}}
```

NOTE: The following cell is for Shadow Testing.

```
In [ ]: def invoke_endpoint(endpoint_name, should_raise_exp=False):
            with open("test_data/test-dataset-input-cols.csv", "r") as f:
                for row in f:
```

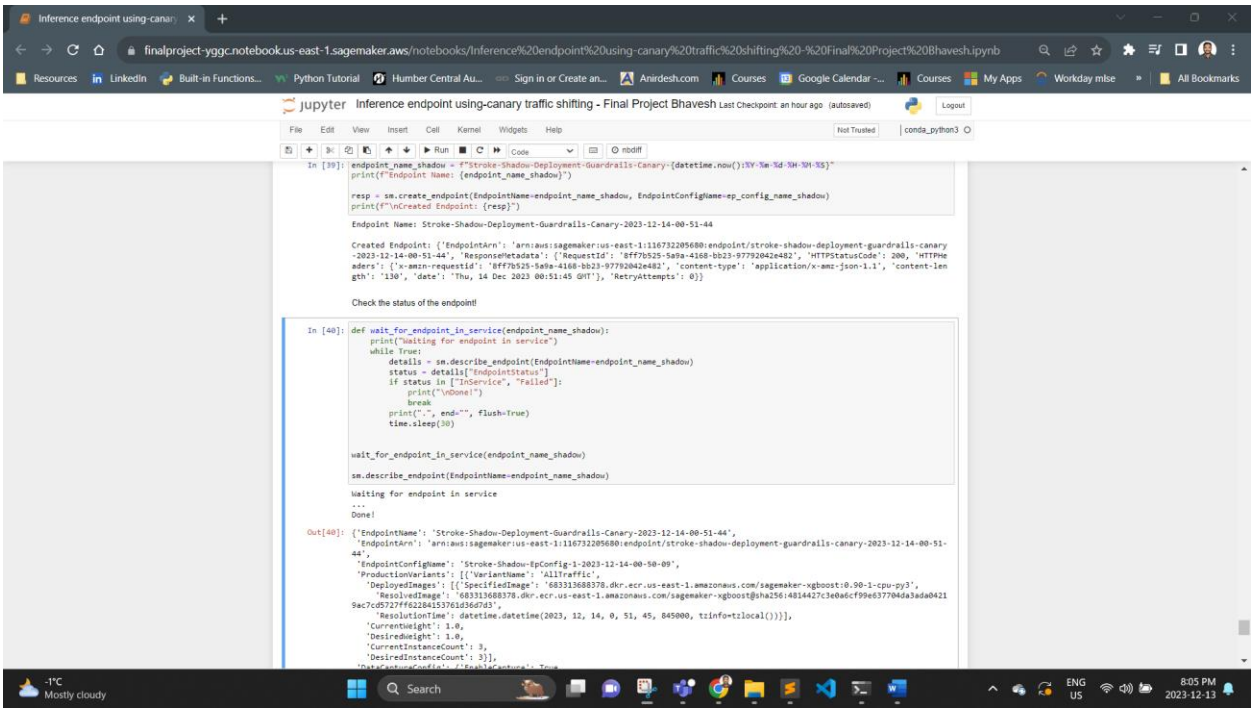
14. Shadow Testing Endpoint Configuration settings.



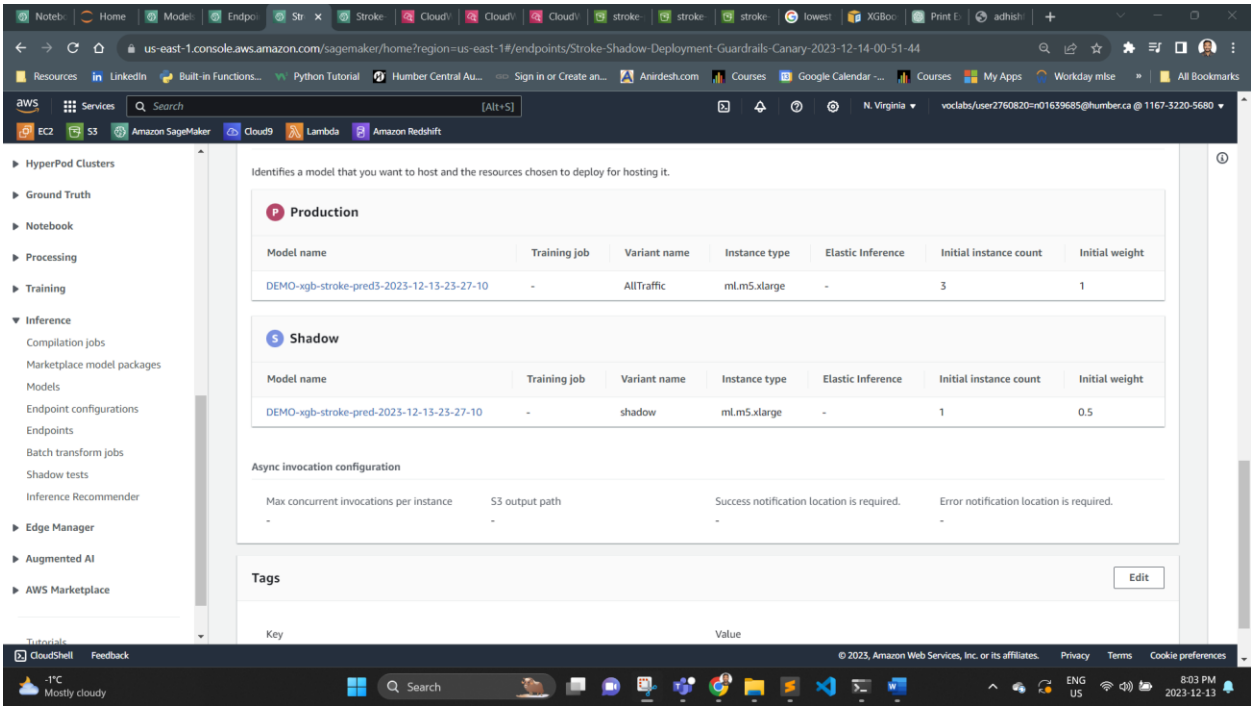
Creating configuration for Shadow Endpoint

```
In [38]: ep_config_name_shadow = f"Stroke-Shadow-EpConfig-1-{datetime.now():%Y-%m-%d-%H-%M-%S}"

         print(f"Endpoint Config Shadow: {ep_config_name_shadow}")

         resp = sm.create_endpoint_config(
             EndpointConfigName=ep_config_name_shadow,
             ProductionVariants=[
                 {
                     "VariantName": "AllTraffic",
                     "ModelName": model_name3,
                     "InstanceType": "ml.m5.xlarge",
                     "InitialInstanceCount": 3,
                 }
             ],
             # Type: Array of ShadowProductionVariants
              ShadowProductionVariants = [
                  {
                      "ModelName": model_name,
                      'VariantName':"shadow",
                      "InitialInstanceCount": 1,
                      "InitialVariantWeight": 0.5,
                      "InstanceType": "ml.m5.xlarge"
                  }
              ],
             DataCaptureConfig = {
                  'EnableCapture': True,
                  'InitialSamplingPercentage': 100,
                  'DestinationS3Uri': f"s3://{bucket}/{prefix}",
                  'CaptureOptions': [{'CaptureMode': 'Input'}, {'CaptureMode': 'Output'}],
                  'CaptureContentTypeHeader': {'JsonContentTypes': ['application/json']}
                 }

         )
         print(f"Created Endpoint Config: {resp}")
         time.sleep(5)

         Endpoint Config Shadow: Stroke-Shadow-EpConfig-1-2023-12-14-00-50-09
         Created Endpoint Config: {'EndpointConfigArn': 'arn:aws:sagemaker:us-east-1:116732205680:endpoint-config/stroke-shadow-epconfig
         -1-2023-12-14-00-50-09', 'ResponseMetadata': {'RequestId': '6ad19700-75e4-4f1e-b012-8ad51b72cdbc', 'HTTPStatusCode': 200, 'HTTP
         Headers': {'x-amzn-requestid': '6ad19700-75e4-4f1e-b012-8ad51b72cdbc', 'content-type': 'application/x-amz-json-1.1', 'content-l
         ength': '125', 'date': 'Thu, 14 Dec 2023 00:50:09 GMT'}, 'RetryAttempts': 0}}
```

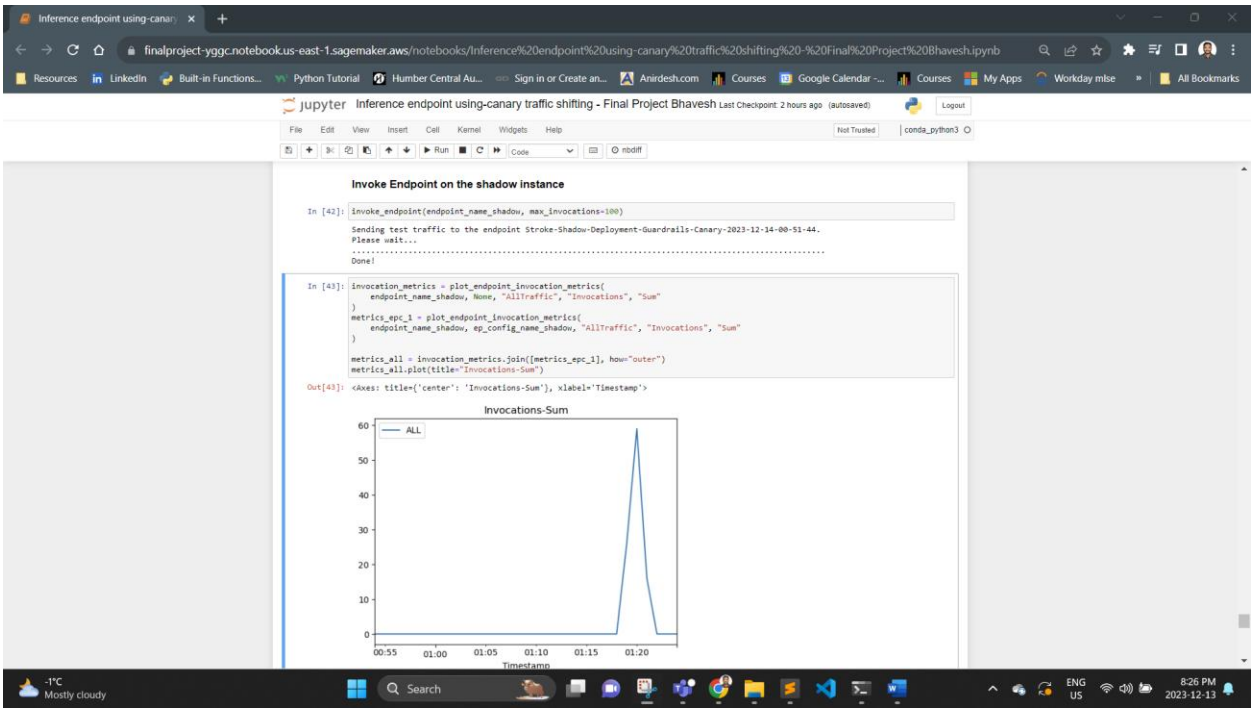
14.2 Creating an endpoint from the shadow configurations.



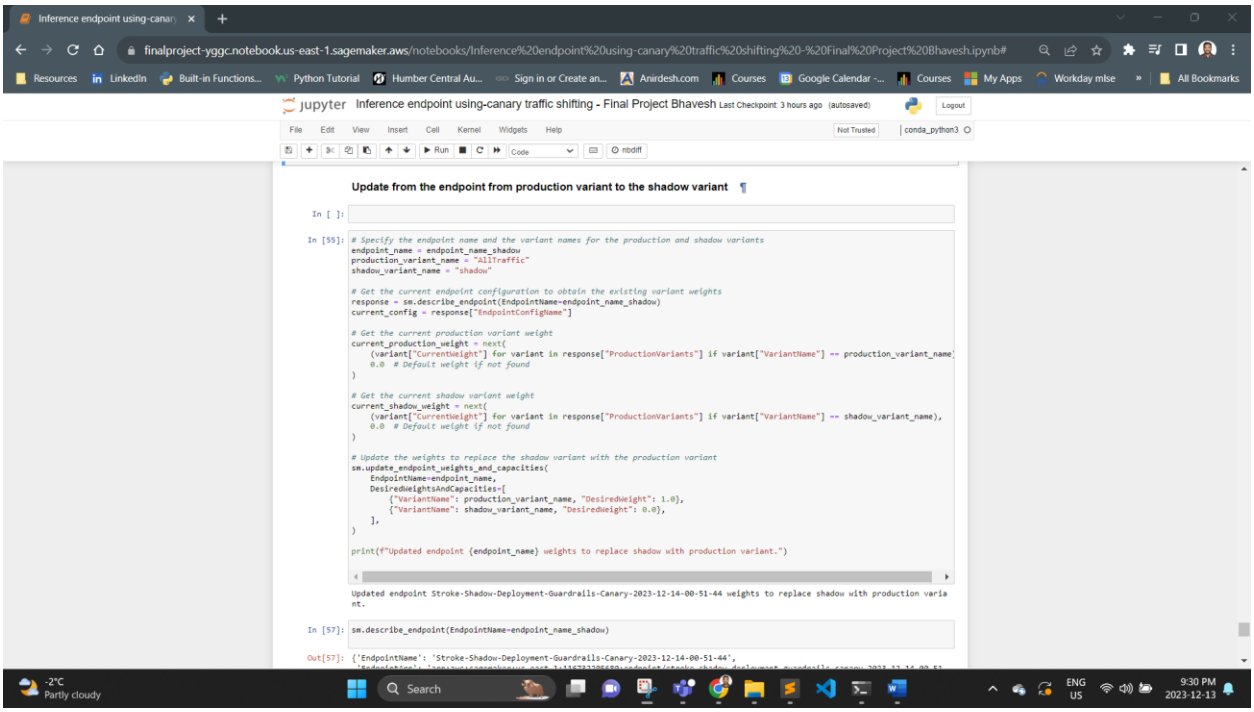
14.3 Verifying that the Production and the Shadow variants have been created inside the endpoint.



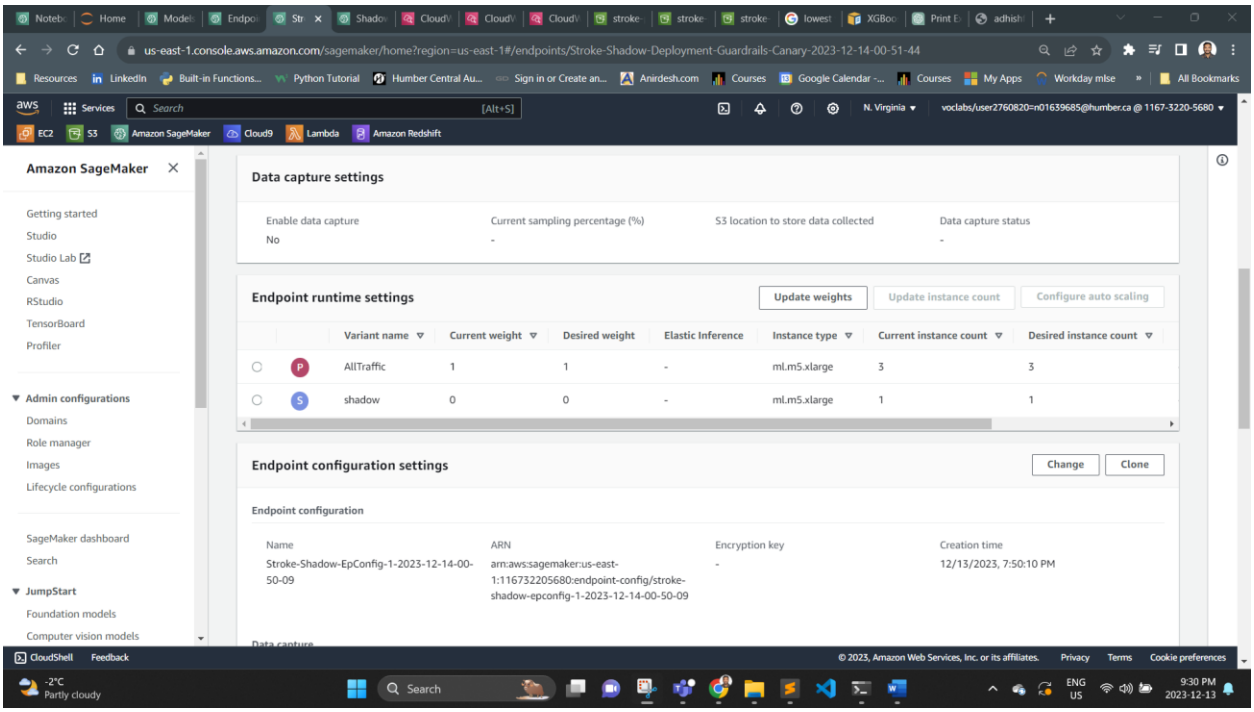
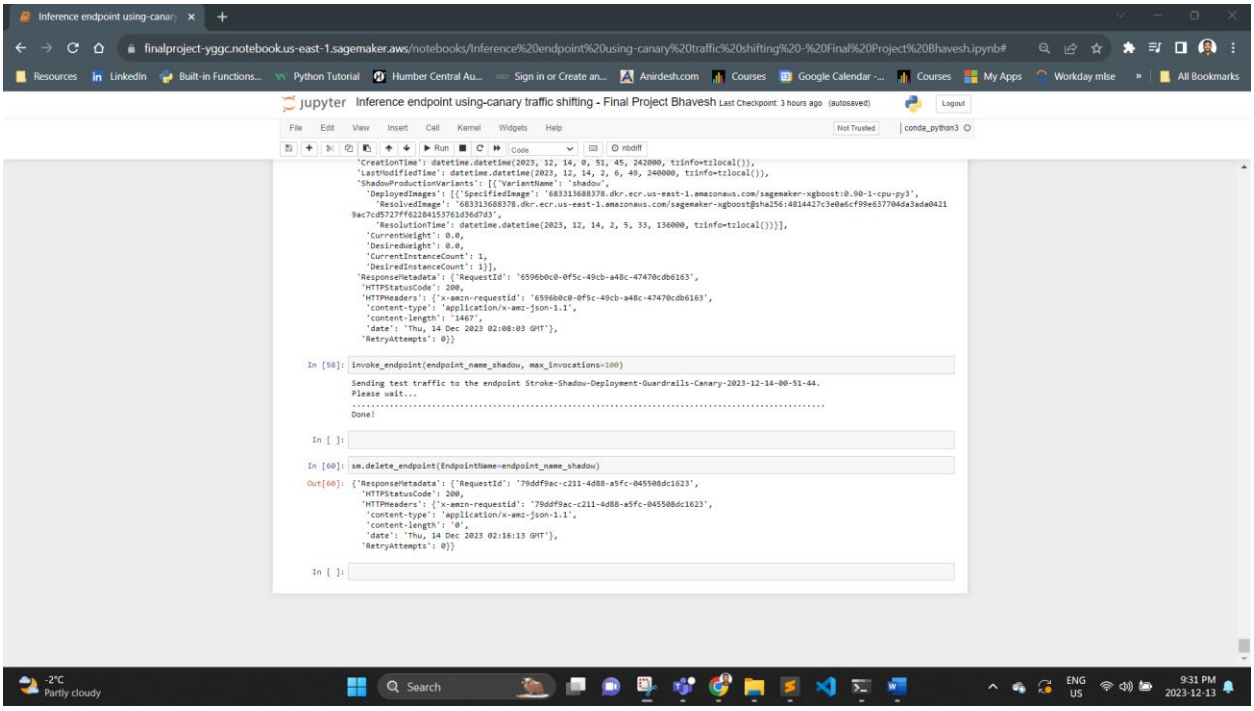
14.4 Invoke Shadow endpoint and check its matrix.



15.1 Update from Production Variant to Shadow Variant.



15.2 Check the update status in the Endpoint inside SageMaker.



16. Invoke the updated model and check if it can handle the request and then delete the endpoint.