Uploading the two different Models, one is with better accuracy and other having the worst accuracy.

Model 1: Better one
Model 2: Worst one

```
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
Execution role: arn:aws:iam::015532774890:role/LabRole
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
```

[2]:
```
model_url = S3Uploader.upload(
    local_path="model/model1.tar.gz",
    desired_s3_uri=f"s3://{bucket}/{prefix}",
)
model_url2 = S3Uploader.upload(
    local_path="model/model2.tar.gz",
    desired_s3_uri=f"s3://{bucket}/{prefix}",
)

print(f"Model URI 1: {model_url}")
print(f"Model URI 2: {model_url2}")
```

```
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
Model URI 1: s3://sagemaker-us-east-1-015532774890/sagemaker/DEMO-Deployment-Guardrails-Canary/model1.tar.gz
Model URI 2: s3://sagemaker-us-east-1-015532774890/sagemaker/DEMO-Deployment-Guardrails-Canary/model2.tar.gz
```

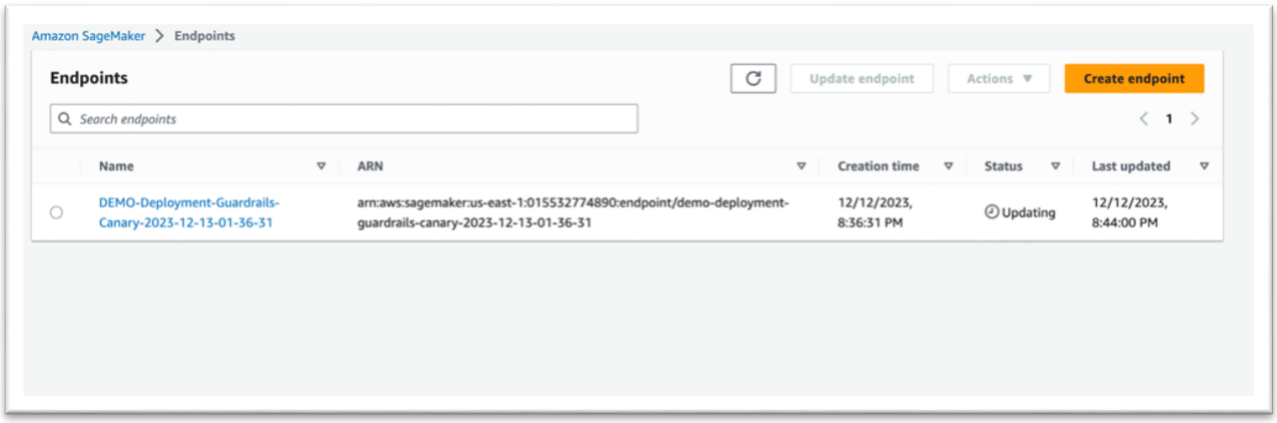
[3]:
```
from sagemaker import image_uris

image_uri = image_uris.retrieve("xgboost", boto3.Session().region_name, "0.90-2")

# using newer version of XGBoost which is incompatible, in order to simulate model faults
image_uri2 = image_uris.retrieve("xgboost", boto3.Session().region_name, "0.90-1")
image_uri3 = image_uris.retrieve("xgboost", boto3.Session().region_name, "1.7-1")
```

Invoking the First model with good accuracy, this one having no errors

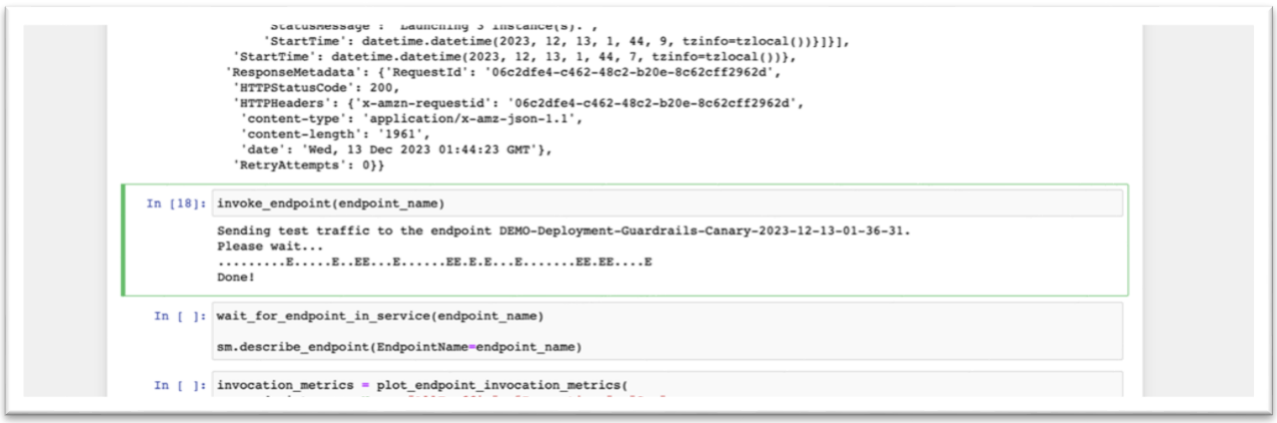
[8]:
```
def invoke_endpoint(
    endpoint_name, max_invocations=600, wait_interval_sec=1, should_raise_exp=False
):
    print(f"Sending test traffic to the endpoint {endpoint_name}. \nPlease wait...")

    count = 0
    with open("test_data/test-dataset.csv", "r") as f:
        for row in f:
            payload = row.rstrip("\n")
            try:
                response = sm_runtime.invoke_endpoint(
                    EndpointName=endpoint_name, ContentType="text/csv", Body=payload
                )
                response["Body"].read()
                print(".", end="", flush=True)
            except Exception as e:
                print("E", end="", flush=True)
                if should_raise_exp:
                    raise e
            count += 1
            if count > max_invocations:
                break
            time.sleep(wait_interval_sec)

    print("\nDone!")


invoke_endpoint(endpoint_name, max_invocations=100)
```

```
Sending test traffic to the endpoint DEMO-Deployment-Guardrails-Canary-2023-12-13-01-36-31.
Please wait...
..................................................
Done!
```

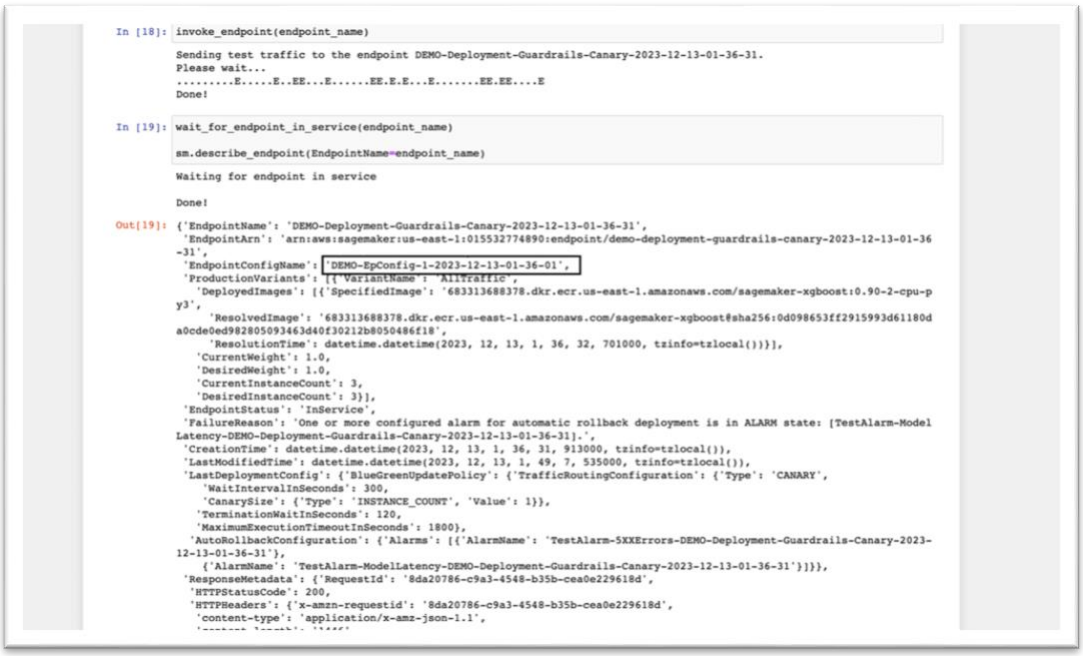
Updating the endpoint



While uploading the endpoint. Try to invoke the endpoint, this will return the error and stop updating and rollback to the first endpoint and not allow to deploy



Highlighted box shows that still the endpoint config is in 1 itself



Now try to invoke the endpoint config3 which is model3 with better model with correct version of the algorithm.

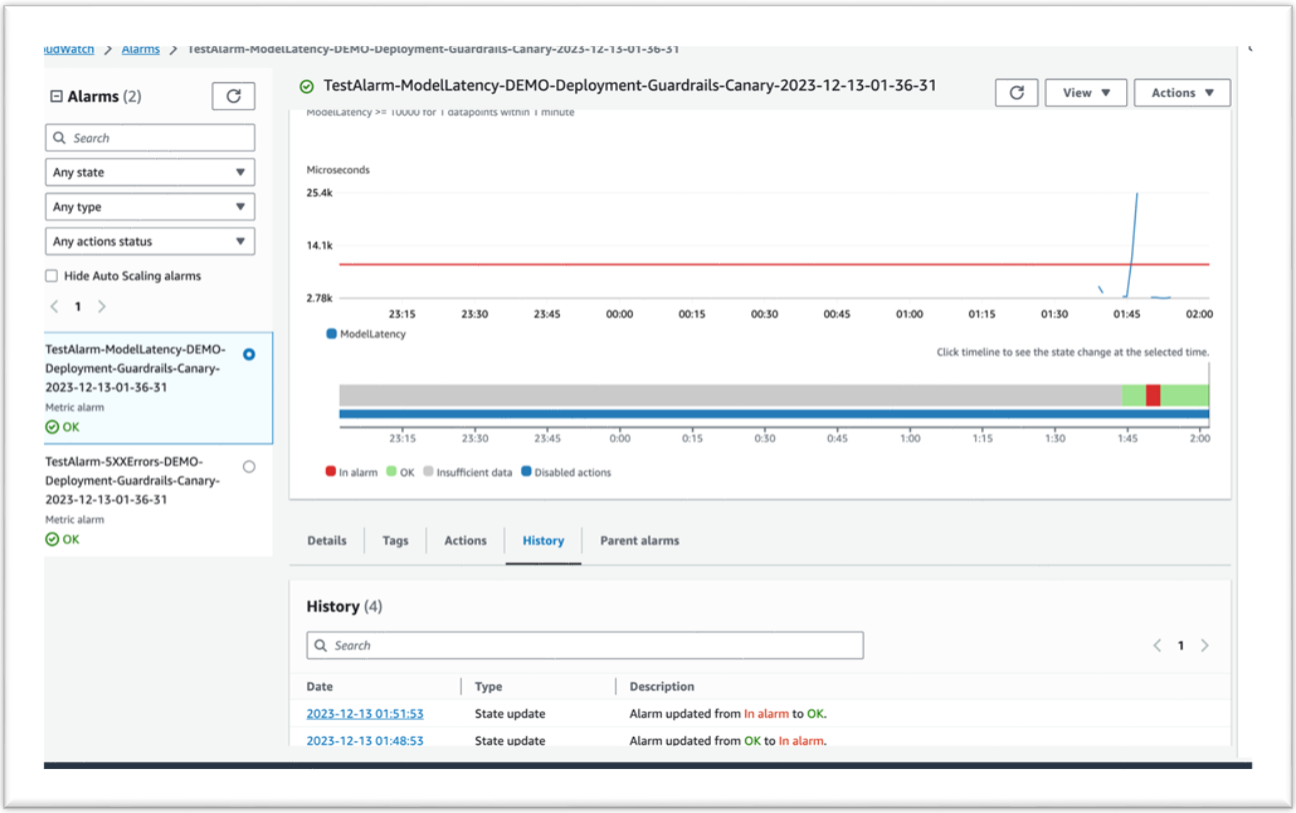
Note: Model 1 is better model with version is different

```
        'RetryAttempts': 0}}

[24]: invoke_endpoint(endpoint_name, max_invocations=500)

      Sending test traffic to the endpoint DEMO-Deployment-Guardrails-Canary-2023-12-13-01-36-31.
      Please wait...
      ..........................................................
      Done!
```

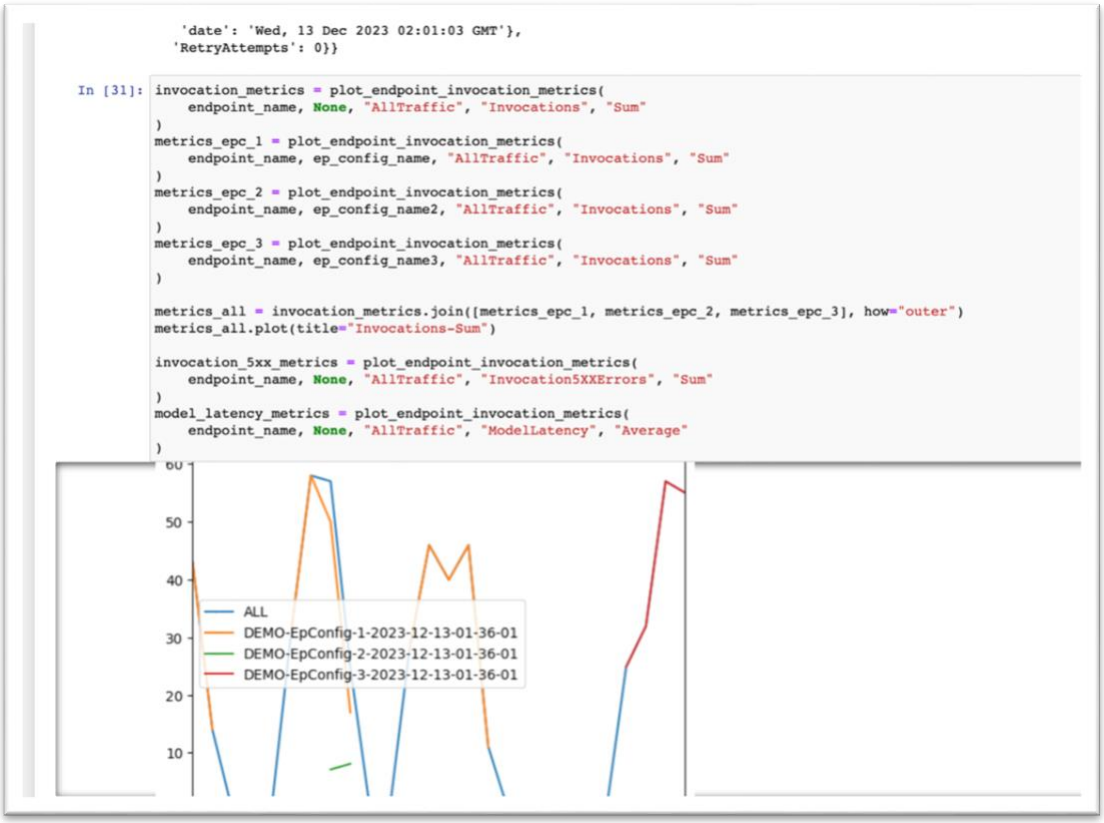
Highlighted box shows now its changed to endpoint config3. Now its success.

```
[24]: invoke_endpoint(endpoint_name, max_invocations=500)

      Sending test traffic to the endpoint DEMO-Deployment-Guardrails-Canary-2023-12-13-01-36-31.
      Please wait...
      ..........................................................
      Done!

[25]: wait_for_endpoint_in_service(endpoint_name)

      sm.describe_endpoint(EndpointName=endpoint_name)

      Waiting for endpoint in service
      ............
      Done!

t[25]: {'EndpointName': 'DEMO-Deployment-Guardrails-Canary-2023-12-13-01-36-31',
        'EndpointArn': 'arn:aws:sagemaker:us-east-1:015532774890:endpoint/demo-deployment-guardrails-canary-2023-12-13-01-36
      -31',
        'EndpointConfigName': 'DEMO-EpConfig-3-2023-12-13-01-36-01',
        'ProductionVariants': [{'VariantName': 'AllTraffic',
          'DeployedImages': [{'SpecifiedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.7-1',
            'ResolvedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost@sha256:6c8543f40a8de1ee00692fde
      82e4de4f9533d76ea0fbac28dee941c33418abd6',
            'ResolutionTime': datetime.datetime(2023, 12, 13, 1, 50, 30, 798000, tzinfo=tzlocal())}],
          'CurrentWeight': 1.0,
          'DesiredWeight': 1.0,
          'CurrentInstanceCount': 3,
          'DesiredInstanceCount': 3}],
        'EndpointStatus': 'InService',
        'CreationTime': datetime.datetime(2023, 12, 13, 1, 36, 31, 913000, tzinfo=tzlocal()),
        'LastModifiedTime': datetime.datetime(2023, 12, 13, 2, 0, 43, 370000, tzinfo=tzlocal()),
        'LastDeploymentConfig': {'BlueGreenUpdatePolicy': {'TrafficRoutingConfiguration': {'Type': 'CANARY',
          'WaitIntervalInSeconds': 300,
          'CanarySize': {'Type': 'INSTANCE_COUNT', 'Value': 1}},
          'TerminationWaitInSeconds': 120,
          'MaximumExecutionTimeoutInSeconds': 1800},
         'AutoRollbackConfiguration': {'Alarms': [{'AlarmName': 'TestAlarm-5XXErrors-DEMO-Deployment-Guardrails-Canary-2023-
      12-13-01-36-31'},
          {'AlarmName': 'TestAlarm-ModelLatency-DEMO-Deployment-Guardrails-Canary-2023-12-13-01-36-31'}]}},
        'ResponseMetadata': {'RequestId': '957020eb-6a76-490e-90ce-b1aa9298dfcf',
         'HTTPStatusCode': 200,
         'HTTPHeaders': {'x-amzn-requestid': '957020eb-6a76-490e-90ce-b1aa9298dfcf',
          'content-type': 'application/x-amz-json-1.1',
          'content-length': '1256',
          'date': 'Wed, 13 Dec 2023 02:01:03 GMT'},
         'RetryAttempts': 0}}
```

CloudWatch Alarm console



Showing the metrics



Shadow Testing

Create the new endpoint for the shadow test and create two variant names and created the shadow endpoint configuration response as well

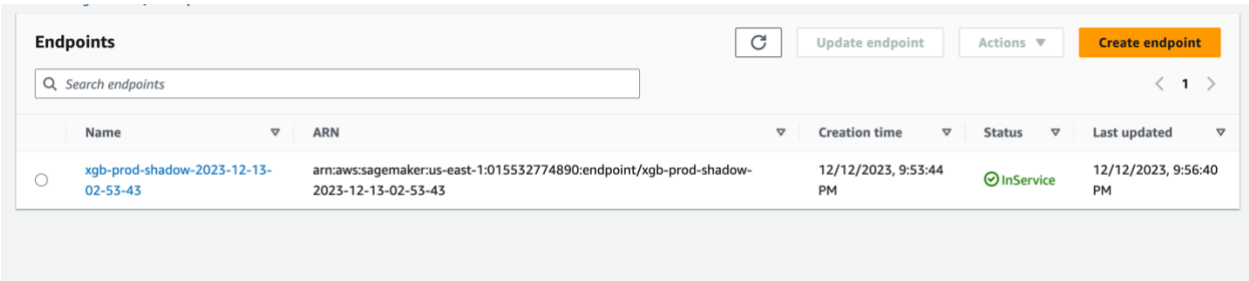
```
In [40]: shadow_ep_config_name = f"Shadow-EpConfig-{datetime.now():%Y-%m-%d-%H-%M-%S}"
production_variant_name = "production"
shadow_variant_name = "shadow"

create_shadow_endpoint_config_response = sm.create_endpoint_config(
    EndpointConfigName=shadow_ep_config_name,
    ProductionVariants=[
        {
            "VariantName": production_variant_name,
            "ModelName": model_name,
            "InstanceType": "ml.m5.xlarge",
            "InitialInstanceCount": 2,
            "InitialVariantWeight": 1,
        }
    ],
    ShadowProductionVariants=[
        {
            "VariantName": shadow_variant_name,
            "ModelName": model_name3,
            "InstanceType": "ml.m5.xlarge",
            "InitialInstanceCount": 1,
            "InitialVariantWeight": 0.5,
        }
    ],
)
print(f"Created EndpointConfig: {create_shadow_endpoint_config_response['EndpointConfigArn']}")

Created EndpointConfig: arn:aws:sagemaker:us-east-1:015532774890:endpoint-config/shadow-epconfig-2023-12-13-02-52-20
```

After creating the endpoint and use the describe to get the detailed information about various resources, such as training jobs, endpoints, models, notebooks, and other components within the SageMaker environment

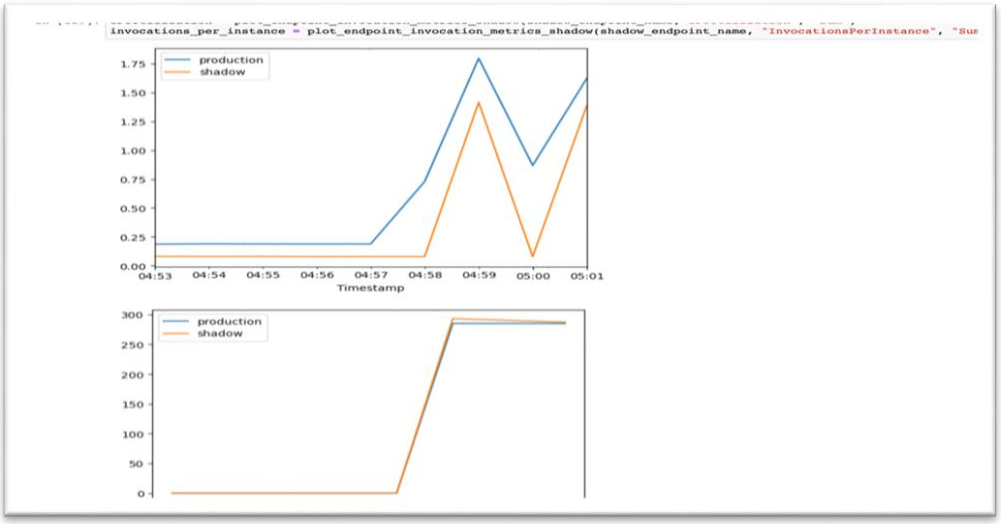
```
In [41]: shadow_endpoint_name = f"xgb-prod-shadow-{datetime.now():%Y-%m-%d-%H-%M-%S}"
create_endpoint_api_response = sm.create_endpoint(
    EndpointName=shadow_endpoint_name,
    EndpointConfigName=shadow_ep_config_name,
)

In [42]: sm.describe_endpoint(EndpointName=shadow_endpoint_name)

Out[42]: {'EndpointName': 'xgb-prod-shadow-2023-12-13-02-53-43',
 'EndpointArn': 'arn:aws:sagemaker:us-east-1:015532774890:endpoint/xgb-prod-shadow-2023-12-13-02-53-43',
 'EndpointConfigName': 'Shadow-EpConfig-2023-12-13-02-52-20',
 'ProductionVariants': [{'VariantName': 'production',
   'DeployedImages': [{'SpecifiedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:0.90-2-cpu-p
y3',
     'ResolvedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost@sha256:0d098653ff2915993d61180d
a0cde0ed982805093463d40f30212b8050486f18',
     'ResolutionTime': datetime.datetime(2023, 12, 13, 2, 53, 44, 859000, tzinfo=tzlocal())}],
   'CurrentWeight': 1.0,
   'DesiredWeight': 1.0,
   'CurrentInstanceCount': 2,
   'DesiredInstanceCount': 2}],
 'EndpointStatus': 'InService',
 'CreationTime': datetime.datetime(2023, 12, 13, 2, 53, 44, 16000, tzinfo=tzlocal()),
 'LastModifiedTime': datetime.datetime(2023, 12, 13, 2, 56, 40, 728000, tzinfo=tzlocal()),
 'ShadowProductionVariants': [{'VariantName': 'shadow',
   'DeployedImages': [{'SpecifiedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.7-1',
     'ResolvedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost@sha256:6c8543f40a8delee00692fde
82e4de4f9533d76ea0fbac28dee941c33418abd6',
     'ResolutionTime': datetime.datetime(2023, 12, 13, 2, 53, 44, 928000, tzinfo=tzlocal())}],
   'CurrentWeight': 0.5,
   'DesiredWeight': 0.5,
   'CurrentInstanceCount': 1,
   'DesiredInstanceCount': 1}],
 'ResponseMetadata': {'RequestId': '2eeff623-39d2-49a2-897f-7b4298da3fa4',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '2eeff623-39d2-49a2-897f-7b4298da3fa4',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '1199',
   'date': 'Wed, 13 Dec 2023 03:08:43 GMT'},
  'RetryAttempts': 0}}
```

Invoking the Shadow endpoint

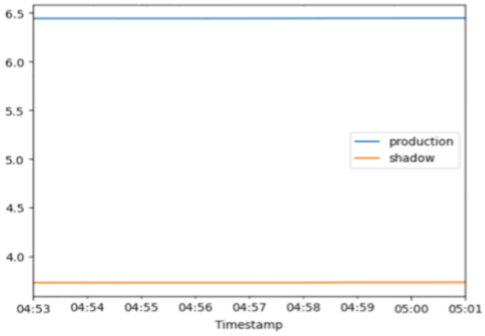
```
In [127]: def invoke_endpoint(endpoint_name, should_raise_exp=False):
              with open("test_data/test-dataset.csv", "r") as f:
                  for row in f:
                      payload = row.rstrip("\n")
                      try:
                          for i in range(10):  # send the same payload 10 times for testing purpose
                              response = sm_runtime.invoke_endpoint(
                                  EndpointName=endpoint_name, ContentType="text/csv", Body=payload
                              )
                      except Exception as e:
                          print("E", end="", flush=True)
                          if should_raise_exp:
                              raise e


          invoke_endpoint(shadow_endpoint_name)
```



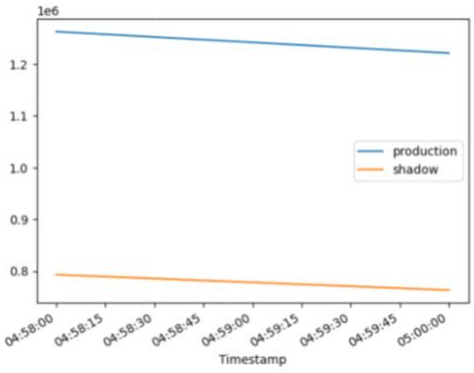
Now following image to show the different metrics and showing that the model3 is better than the model 1 and changing the production variant with shadow variant and removing the shadow variant :



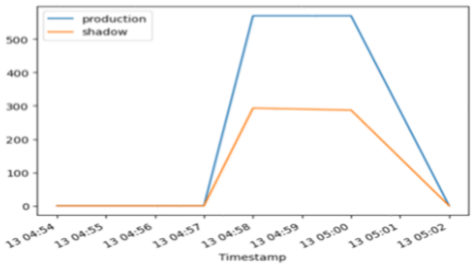
```
In [130]: memoryUtilization = plot_endpoint_invocation_metrics_shadow(shadow_endpoint_name, "MemoryUtilization", "Sum")
```




```
In [132]: modelLatency=plot_endpoint_invocation_metrics_shadow(shadow_endpoint_name,"ModelLatency","Sum")
```



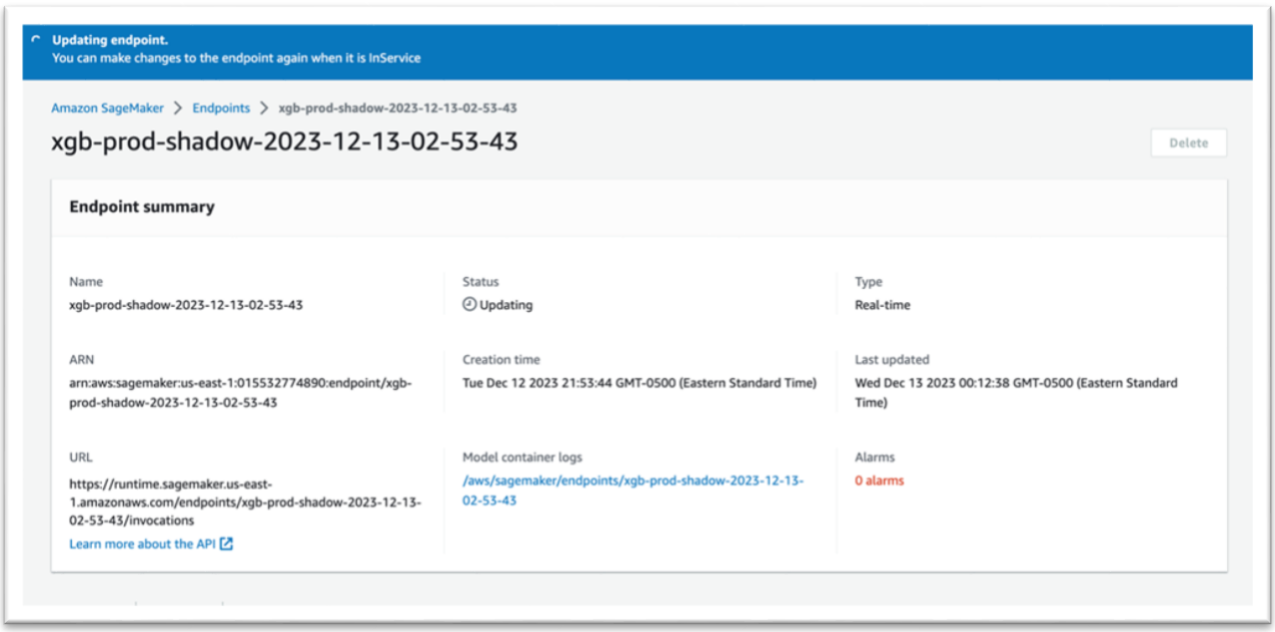

```
invocations_per_instance = plot_endpoint_invocation_metrics_shadow(
    shadow_endpoint_name, "InvocationsPerInstance", "Sum"
)
```



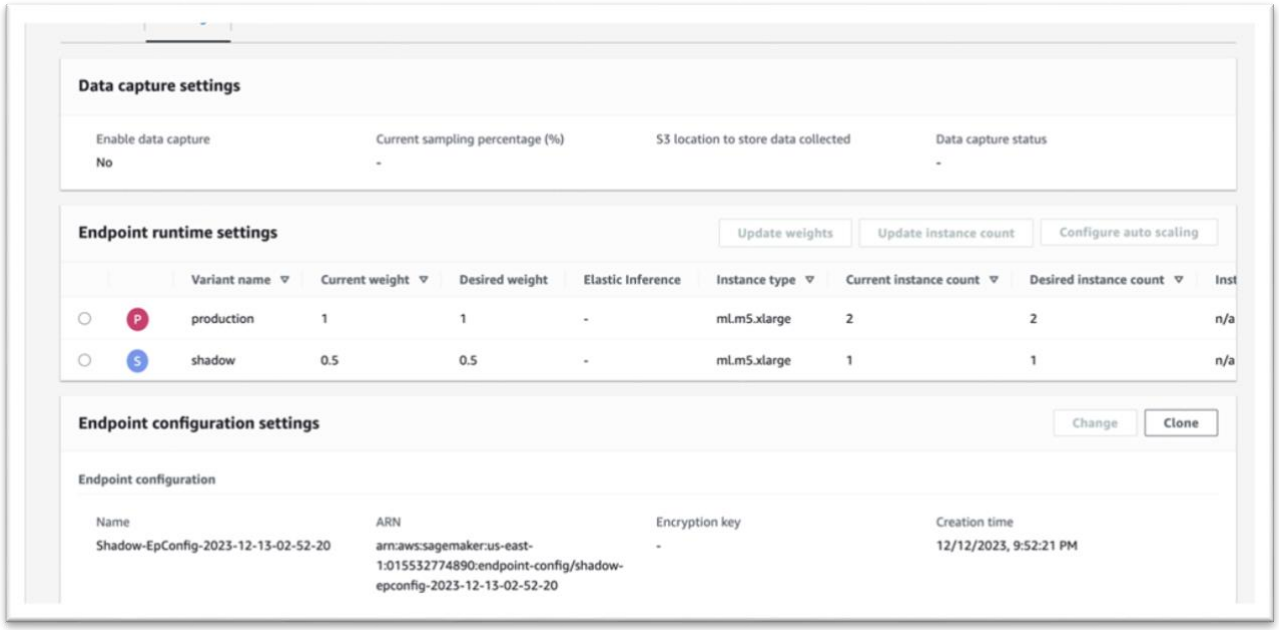
Now updating the production with the shadow variant

```
In [134]: promote_ep_config_name = f"PromoteShadow-EpConfig-{datetime.now():%Y-%m-%d-%H-%M-%S}"

create_endpoint_config_response = sm.create_endpoint_config(
    EndpointConfigName=promote_ep_config_name,
    ProductionVariants=[
        {
            "VariantName": shadow_variant_name,
            "ModelName": model_name3,
            "InstanceType": "ml.m5.xlarge",
            "InitialInstanceCount": 2,
            "InitialVariantWeight": 1.0,
        }
    ],
)
print(f"Created EndpointConfig: {create_endpoint_config_response['EndpointConfigArn']}")

Created EndpointConfig: arn:aws:sagemaker:us-east-1:015532774890:endpoint-config/promoteshadow-epconfig-2023-12-13-05
-10-16

In [135]: update_endpoint_api_response = sm.update_endpoint(
    EndpointName=shadow_endpoint_name,
    EndpointConfigName=promote_ep_config_name,
)

wait_for_endpoint_in_service(shadow_endpoint_name)

sm.describe_endpoint(EndpointName=shadow_endpoint_name)

Waiting for endpoint in service
.....
Done!

Out[135]: {'EndpointName': 'xgb-prod-shadow-2023-12-13-02-53-43',
 'EndpointArn': 'arn:aws:sagemaker:us-east-1:015532774890:endpoint/xgb-prod-shadow-2023-12-13-02-53-43',
 'EndpointConfigName': 'PromoteShadow-EpConfig-2023-12-13-05-10-16',
 'ProductionVariants': [{'VariantName': 'shadow',
```

Now updating the endpoint



Before updating two variant is there and in Endpoint config name is
Shadow-Ep…..



Data capture settings

Enable data capture	Current sampling percentage (%)	S3 location to store data collected	Data capture status
No	-		-

Endpoint runtime settings

Update weights | Update instance count | Configure auto scaling

	Variant name	Current weight	Desired weight	Elastic Inference	Instance type	Current instance count	Desired instance count	Inst
○	production	1	1	-	ml.m5.xlarge	2	2	n/a
○	shadow	0.5	0.5	-	ml.m5.xlarge	1	1	n/a

Endpoint configuration settings

Change | Clone

Endpoint configuration

Name	ARN	Encryption key	Creation time
Shadow-EpConfig-2023-12-13-02-52-20	arn:aws:sagemaker:us-east-1:015532774890:endpoint-config/shadow-epconfig-2023-12-13-02-52-20	-	12/12/2023, 9:52:21 PM

After updating and running describe we can see that endpoint config is updated

```
In [135]: update_endpoint_api_response = sm.update_endpoint(
              EndpointName=shadow_endpoint_name,
              EndpointConfigName=promote_ep_config_name,
          )

          wait_for_endpoint_in_service(shadow_endpoint_name)

          sm.describe_endpoint(EndpointName=shadow_endpoint_name)

          Waiting for endpoint in service
          .....
          Done!

Out[135]: {'EndpointName': 'xgb-prod-shadow-2023-12-13-02-53-43',
           'EndpointArn': 'arn:aws:sagemaker:us-east-1:015532774890:endpoint/xgb-prod-shadow-2023-12-13-02-53-43',
           'EndpointConfigName': 'PromoteShadow-EpConfig-2023-12-13-05-10-16',
           'ProductionVariants': [{'VariantName': 'shadow',
             'DeployedImages': [{'SpecifiedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:0.90-1-cpu-py3',
               'ResolvedImage': '683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost@sha256:4814427c3e0a6cf99e637704da3ada04219ac7cd5727ff62284153761d36d7d3',
               'ResolutionTime': datetime.datetime(2023, 12, 13, 5, 12, 38, 567000, tzinfo=tzlocal())}],
             'CurrentWeight': 1.0,
             'DesiredWeight': 1.0,
             'CurrentInstanceCount': 2,
             'DesiredInstanceCount': 2}],
           'EndpointStatus': 'InService',
           'CreationTime': datetime.datetime(2023, 12, 13, 2, 53, 44, 16000, tzinfo=tzlocal()),
           'LastModifiedTime': datetime.datetime(2023, 12, 13, 5, 14, 46, 661000, tzinfo=tzlocal()),
           'ResponseMetadata': {'RequestId': '647662c3-7b39-45f7-9cb3-d326cb35d611',
            'HTTPStatusCode': 200,
            'HTTPHeaders': {'x-amzn-requestid': '647662c3-7b39-45f7-9cb3-d326cb35d611',
             'content-type': 'application/x-amz-json-1.1',
             'content-length': '762',
             'date': 'Wed, 13 Dec 2023 05:15:08 GMT'},
            'RetryAttempts': 0}}

In [ ]:
```

Now there is nothing in shadow and production part its updated