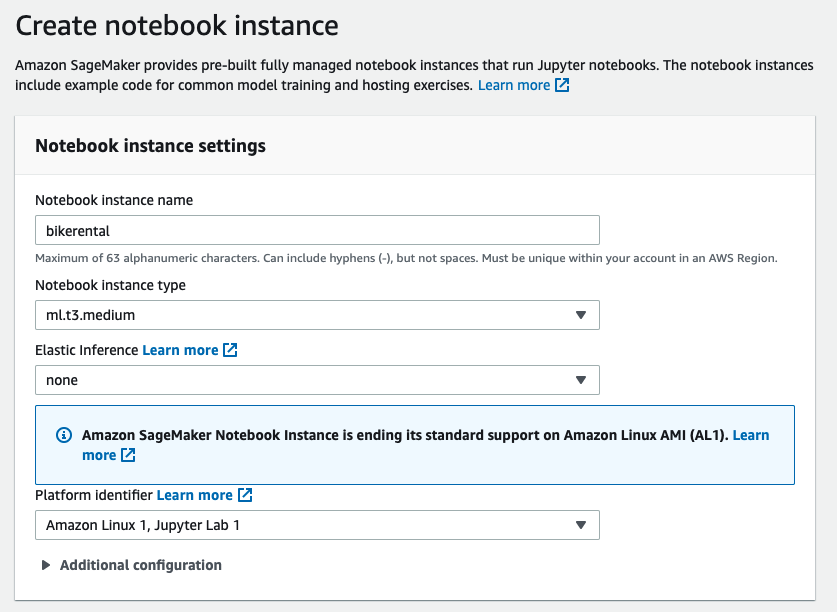
Sagemaker 🡪 notebook instances 🡪 create new instance

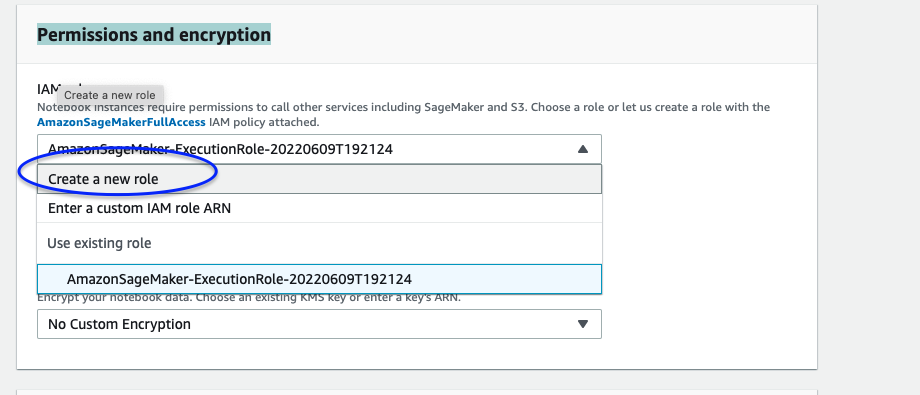
* Sadece notebook instance yeni adini verdik (**bikerental**)

ml.t3.medium veya t2.medium

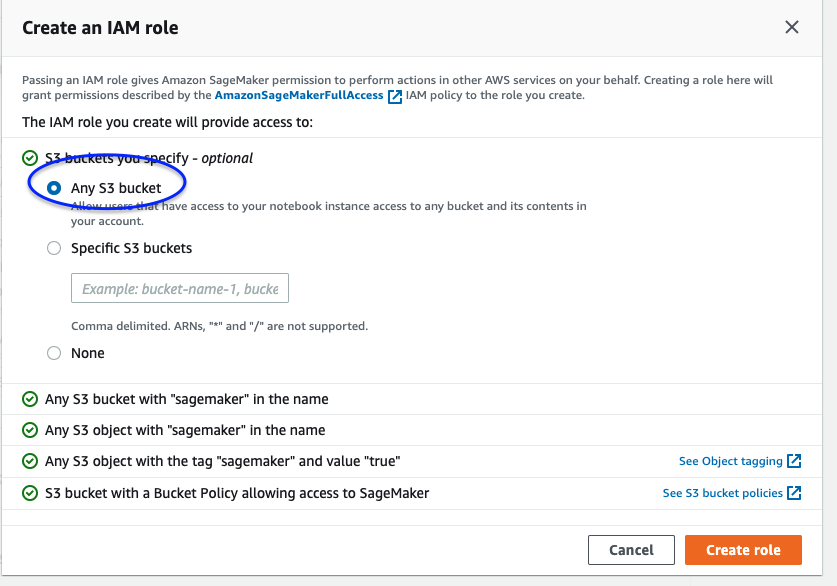


**Permissions and encryption**

**🡪 Create new role**



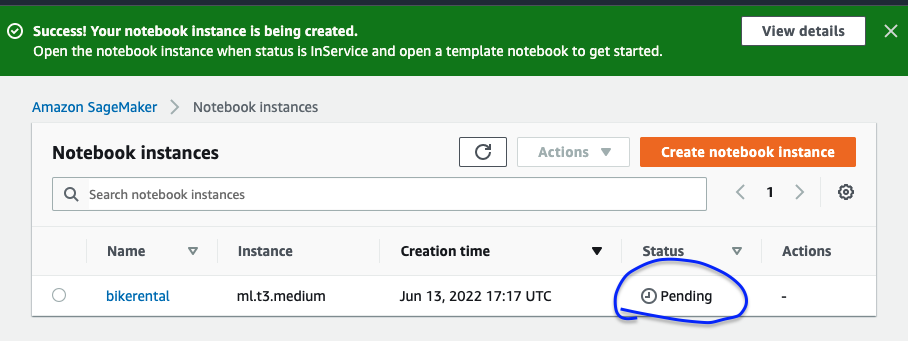
🡪 **Any S3 Bucket**



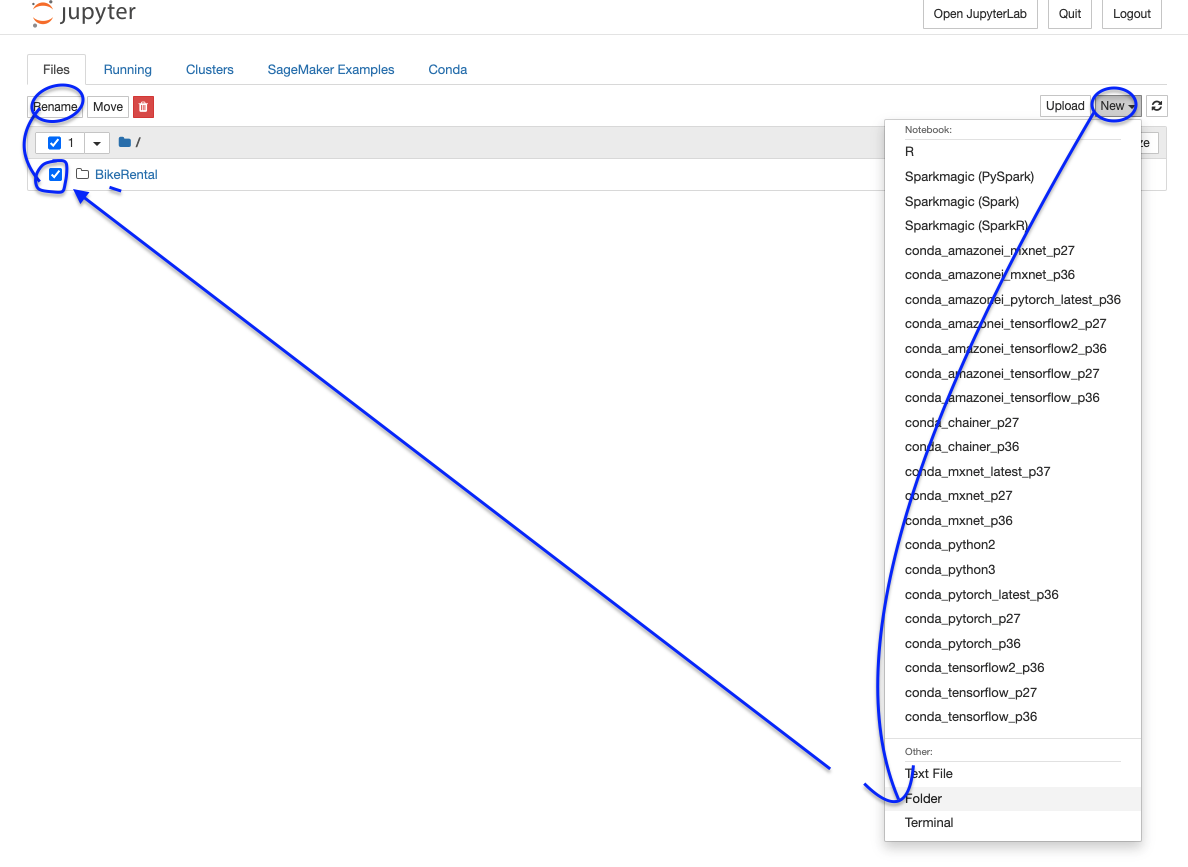
*None secersek, object isminde sagemaker var ise , tek olarak sagemaker kullanilmis ise bu dosyalara olasir*

**🡪 Create notebook instance**

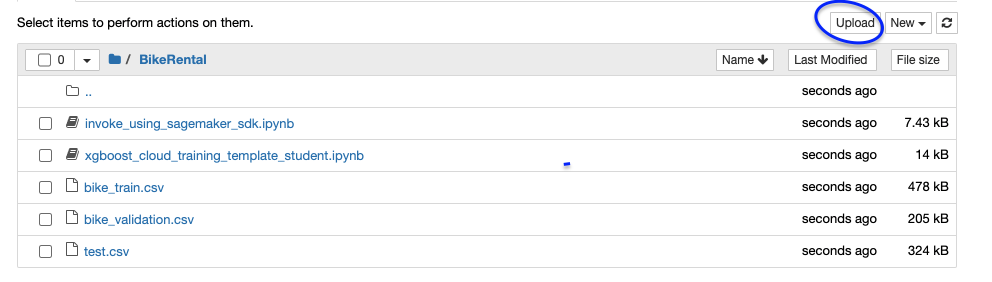
**Pending 🡪 in service olmasini bekliyoruz**

****

**🡪 Open jupyter 🡪 new folder 🡪 rename “BikeRental**

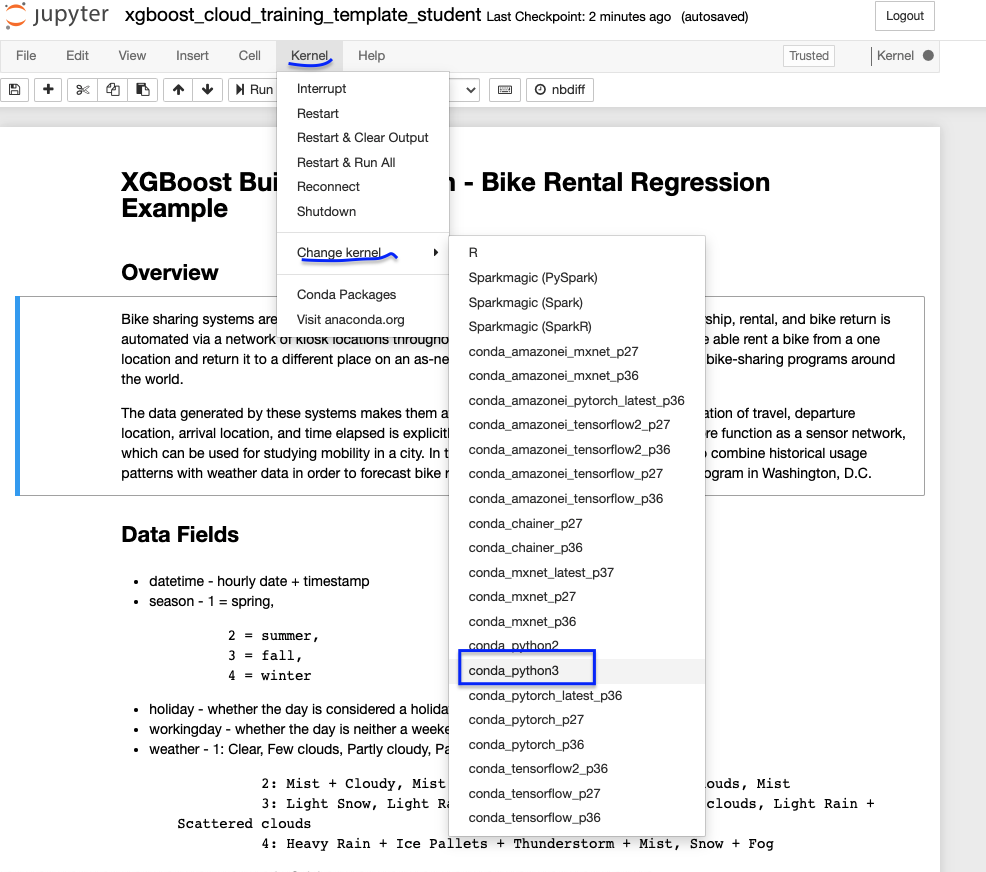
****

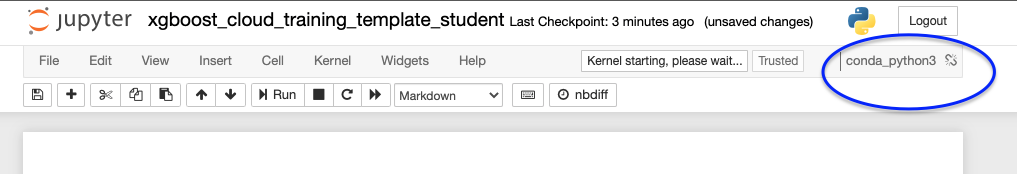
**🡪 Dosya icine derste gonderilen dosyalarin bazilarini upload ettik**

****

**🡪 xgboost\_cloud\_training…. Dosyasini actik**

**🡪 Kernel olarak conda\_python\_3 sectik**

****

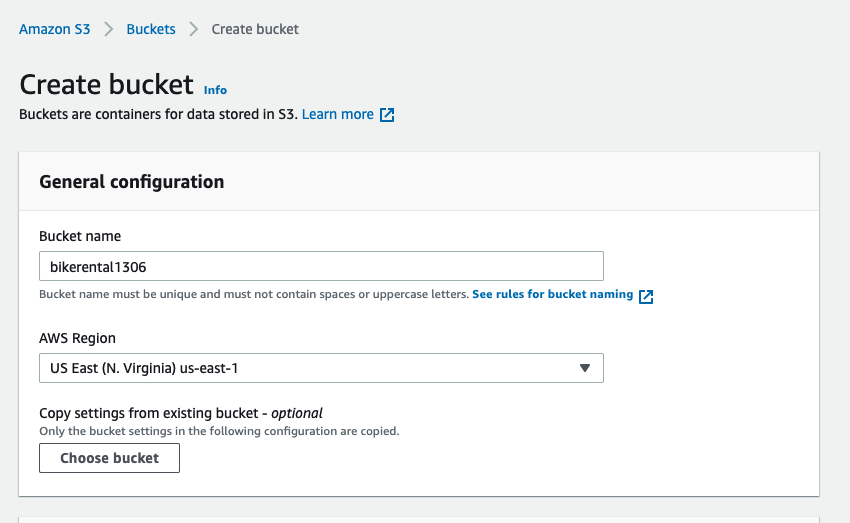
****

Projede kullanmak uzere yeni bucket aciyoruz ,

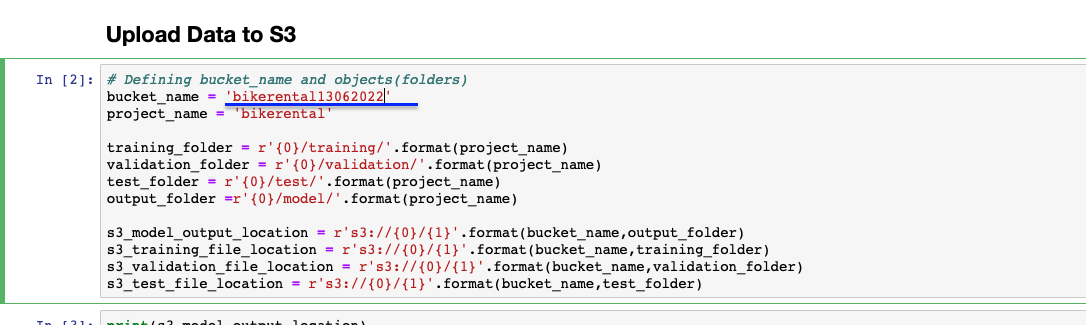
🡪 yeni sekmede S3 actik

🡪 Bucketname : [bikerental13062022](https://s3.console.aws.amazon.com/s3/buckets/bikerental13062022?region=us-east-1)

(bucket name worldwide icinde unique olmali)

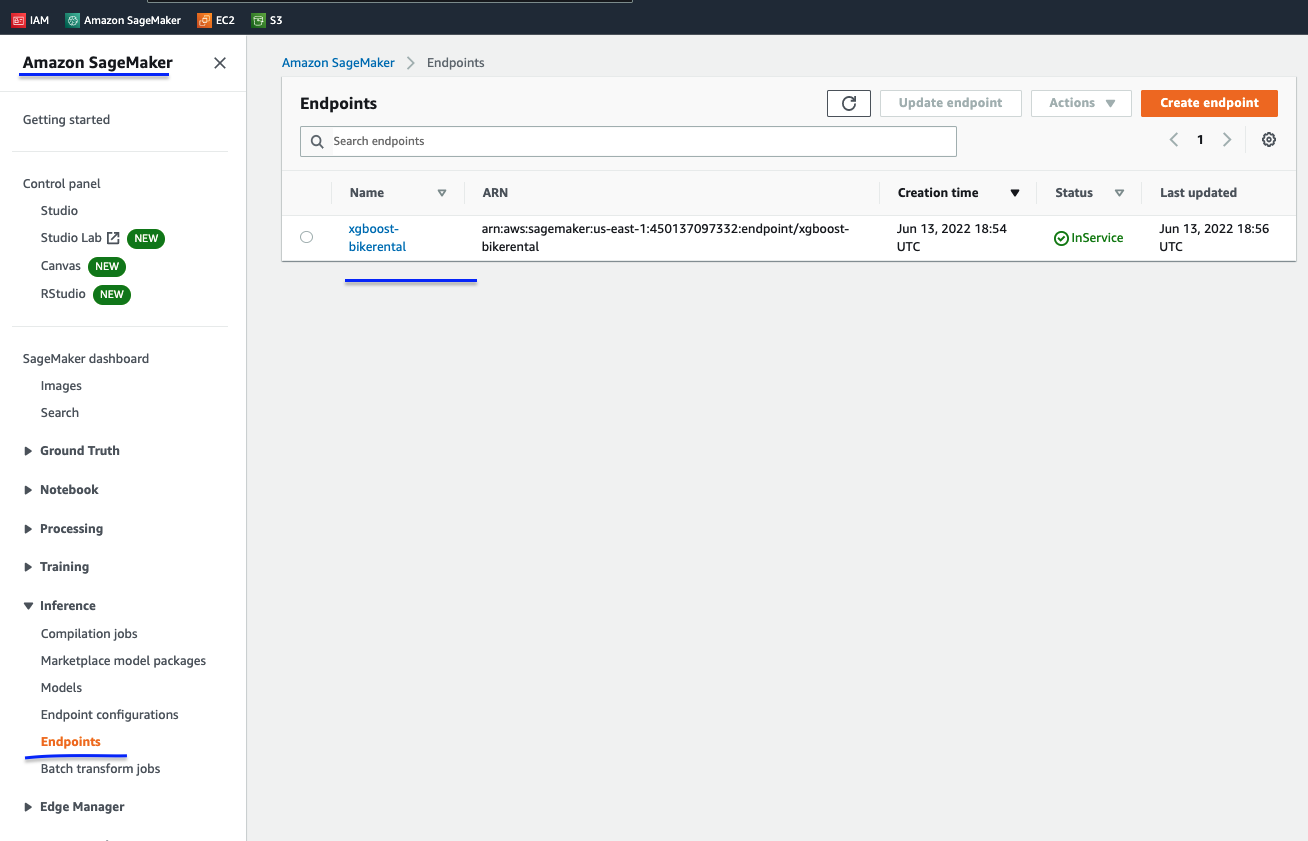


Jupyter notebook kodlari icinde bucket\_name adini guncelledik

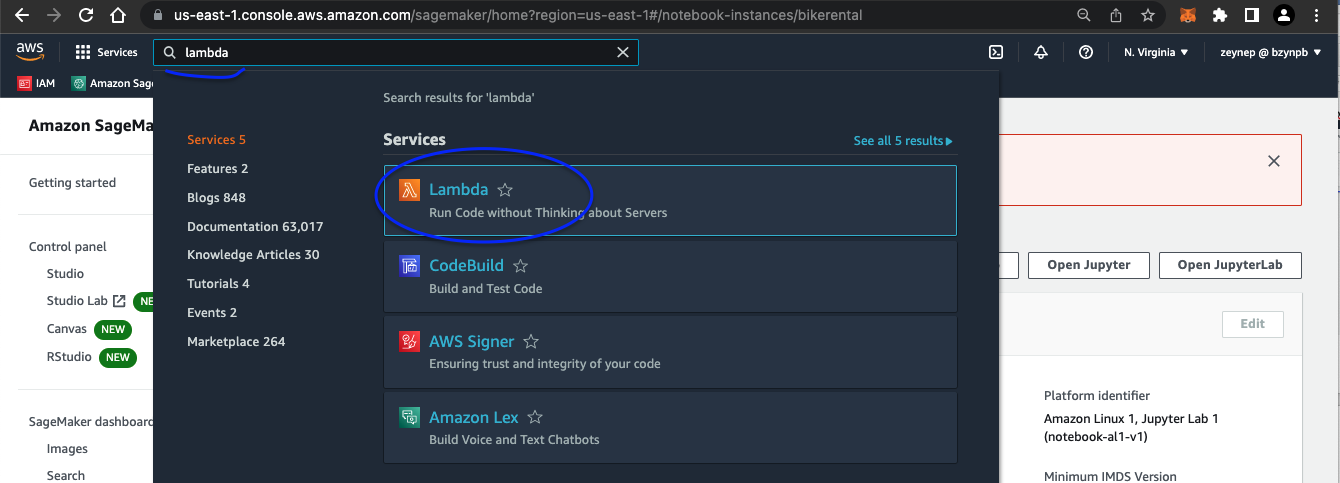


Kod satirlarini calistirmaya devam…

DEPLOY isleminden sonra endpoint icinde yeni dosya olusmali

  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LAMBDA : new browser**

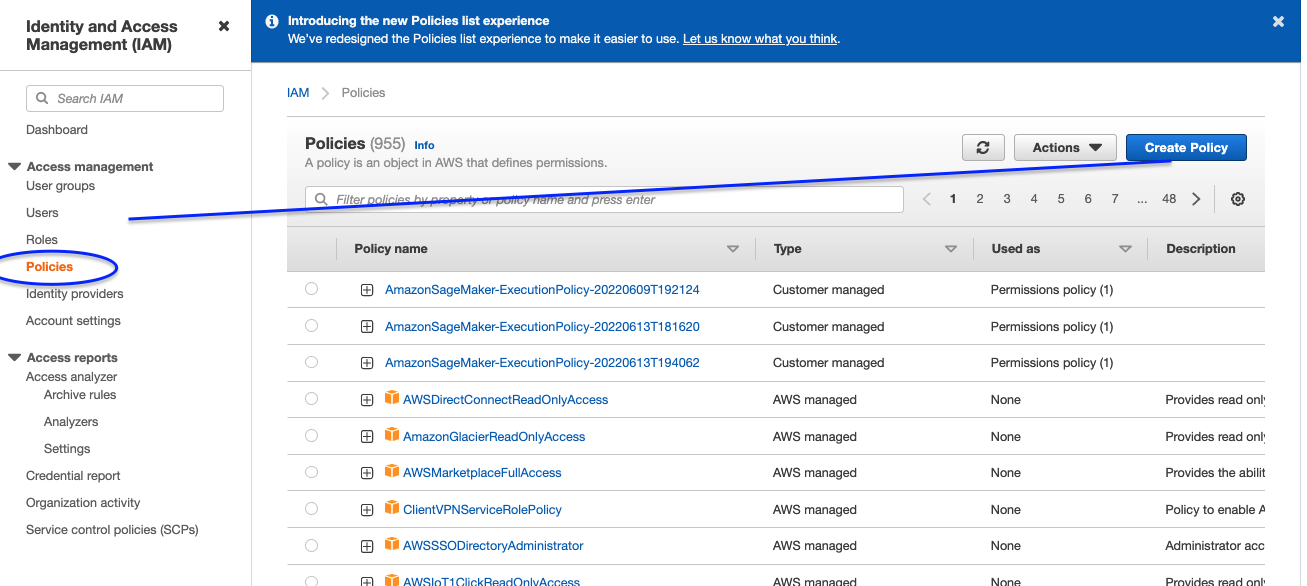


Lambda farkli islemleri, farkli aws source kullanarak islemlerini gerceklestiriyor. Sadece sagemaker kullanmiyor

🡪 create function

IAM sayfasina geldik (yeni browser)

🡪 policies 🡪 create policies



**JSON** 🡪 default veriyi silip asagidaki verileri yapistirdik

------------

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"sagemaker:DescribeTrainingJob",

"sagemaker:DescribeLabelingJob",

"sagemaker:DescribeDataQualityJobDefinition",

"sagemaker:DescribeModelPackage",

"sagemaker:Search",

"sagemaker:DescribeModelPackageGroup",

"sagemaker:DescribeApp",

"sagemaker:GetRecord",

"sagemaker:DescribeFlowDefinition",

"sagemaker:DescribeAlgorithm",

"sagemaker:GetLineageGroupPolicy",

"sagemaker:DescribeTransformJob",

"sagemaker:DescribeInferenceRecommendationsJob",

"sagemaker:DescribeHumanLoop",

"sagemaker:BatchDescribeModelPackage",

"sagemaker:DescribeAction",

"sagemaker:DescribeDeviceFleet",

"sagemaker:DescribeSubscribedWorkteam",

"sagemaker:DescribeHyperParameterTuningJob",

"sagemaker:DescribeAutoMLJob",

"sagemaker:DescribeWorkforce",

"sagemaker:DescribeProcessingJob",

"sagemaker:GetDeviceFleetReport",

"sagemaker:DescribeEndpointConfig",

"sagemaker:DescribeStudioLifecycleConfig",

"sagemaker:RenderUiTemplate",

"sagemaker:DescribeImageVersion",

"sagemaker:BatchGetRecord",

"sagemaker:DescribeHumanTaskUi",

"sagemaker:GetDeviceRegistration",

"sagemaker:DescribeProject",

"sagemaker:GetSagemakerServicecatalogPortfolioStatus",

"sagemaker:DescribeNotebookInstance",

"sagemaker:DescribeAppImageConfig",

"sagemaker:DescribeLineageGroup",

"sagemaker:DescribeNotebookInstanceLifecycleConfig",

"sagemaker:DescribeTrial",

"sagemaker:DescribeContext",

"sagemaker:DescribeModelExplainabilityJobDefinition",

"sagemaker:DescribeEndpoint",

"sagemaker:DescribeUserProfile",

"sagemaker:InvokeEndpoint",

"sagemaker:DescribeMonitoringSchedule",

"sagemaker:DescribeEdgePackagingJob",

"sagemaker:DescribeFeatureGroup",

"sagemaker:DescribeModelQualityJobDefinition",

"sagemaker:GetModelPackageGroupPolicy",

"sagemaker:DescribeModel",

"sagemaker:DescribePipeline",

"sagemaker:DescribeArtifact",

"sagemaker:DescribePipelineExecution",

"sagemaker:DescribeWorkteam",

"sagemaker:DescribeModelBiasJobDefinition",

"sagemaker:DescribeCompilationJob",

"sagemaker:BatchGetMetrics",

"sagemaker:GetSearchSuggestions",

"sagemaker:DescribeExperiment",

"sagemaker:DescribeImage",

"sagemaker:DescribeDomain",

"sagemaker:DescribeCodeRepository",

"sagemaker:InvokeEndpointAsync",

"sagemaker:DescribePipelineDefinitionForExecution",

"sagemaker:DescribeTrialComponent",

"sagemaker:DescribeDevice"

],

"Resource": "\*"

}

]

}

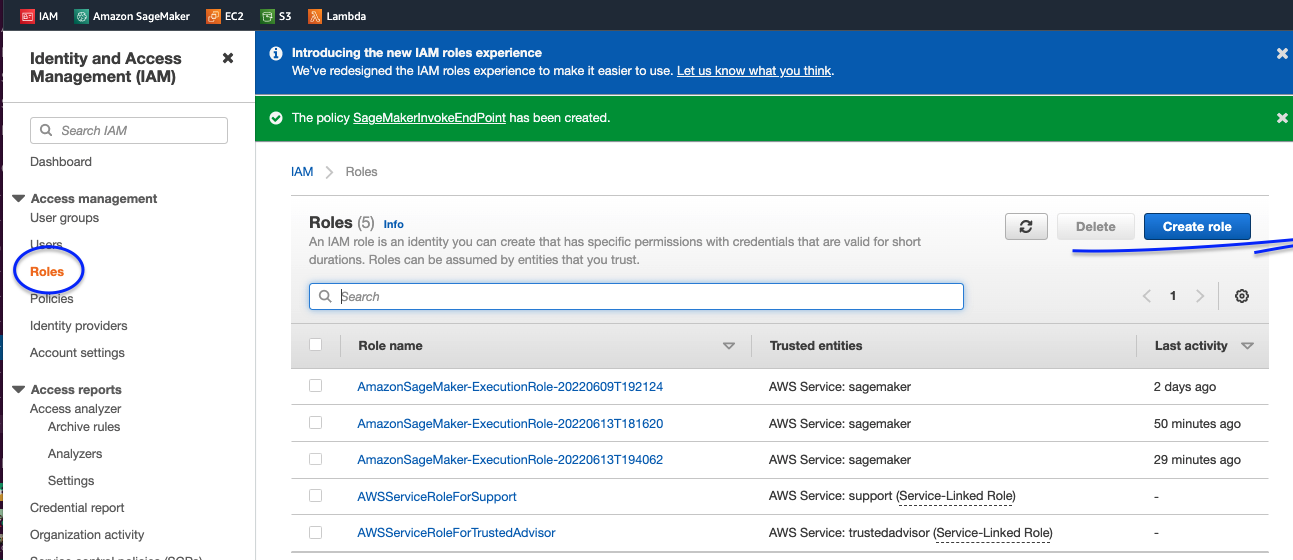
-------

🡪 next tags – tanimlama yapmadik

🡪 next review : isim verdik : **SageMakerInvokeEndPoint**

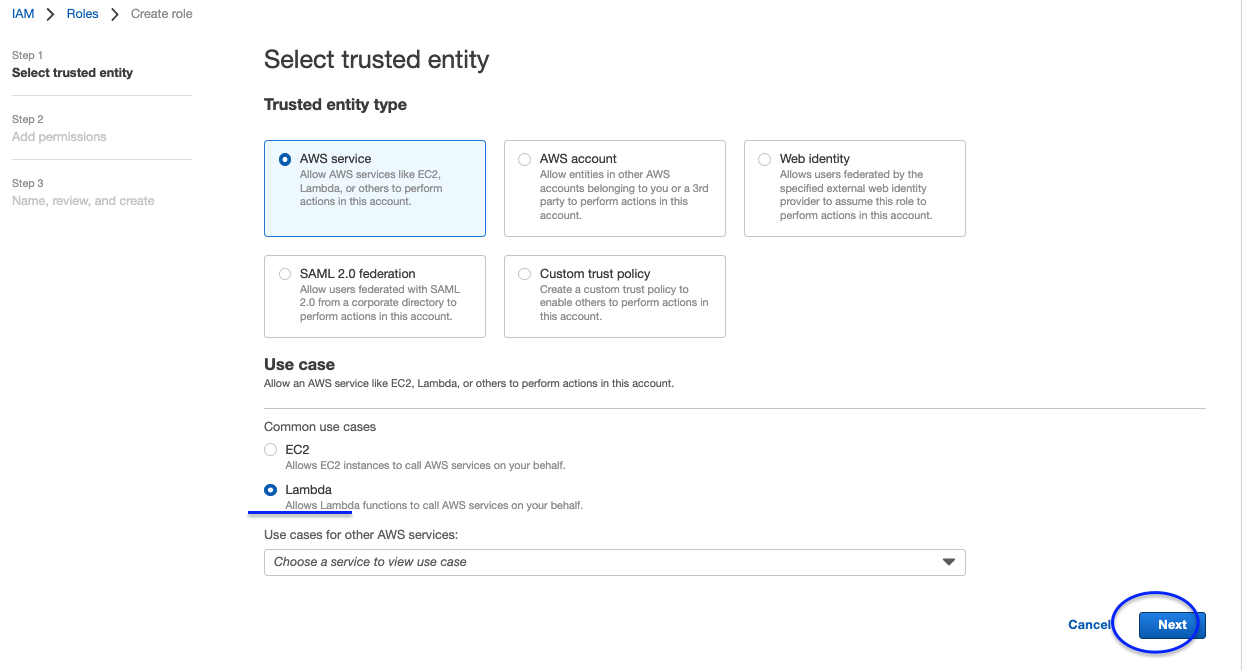
🡪 create policy

**ROLES**



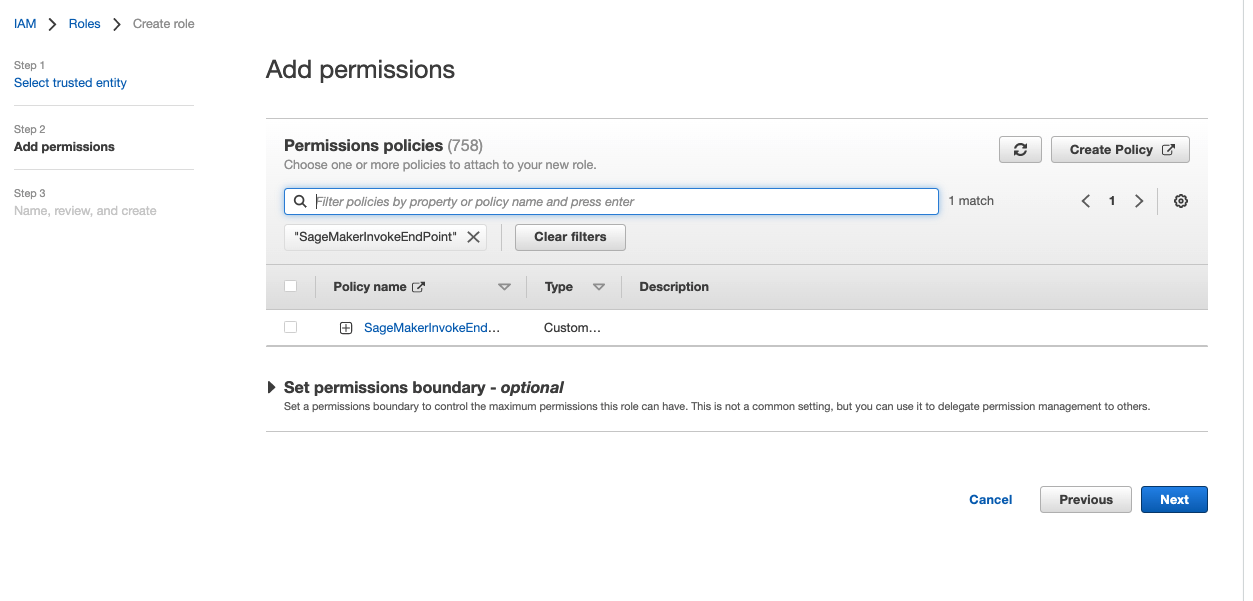
🡪 create role

🡪 use case



Cikan ekranda “SageMakerInvokeEndPoint” adini yazip enter’a bastik

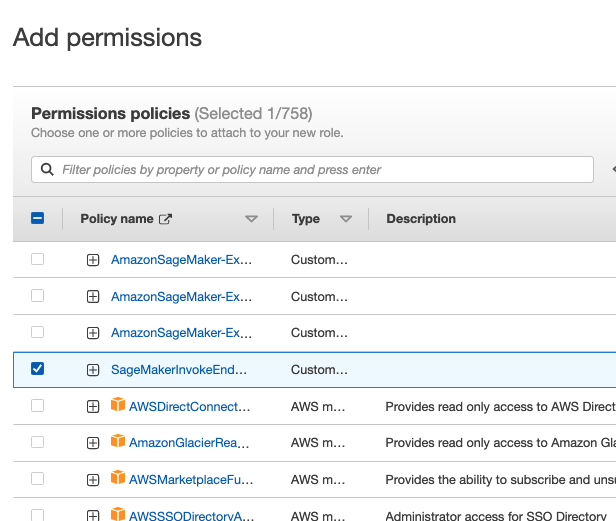
Asagidaki gibi bir goruntu oldu



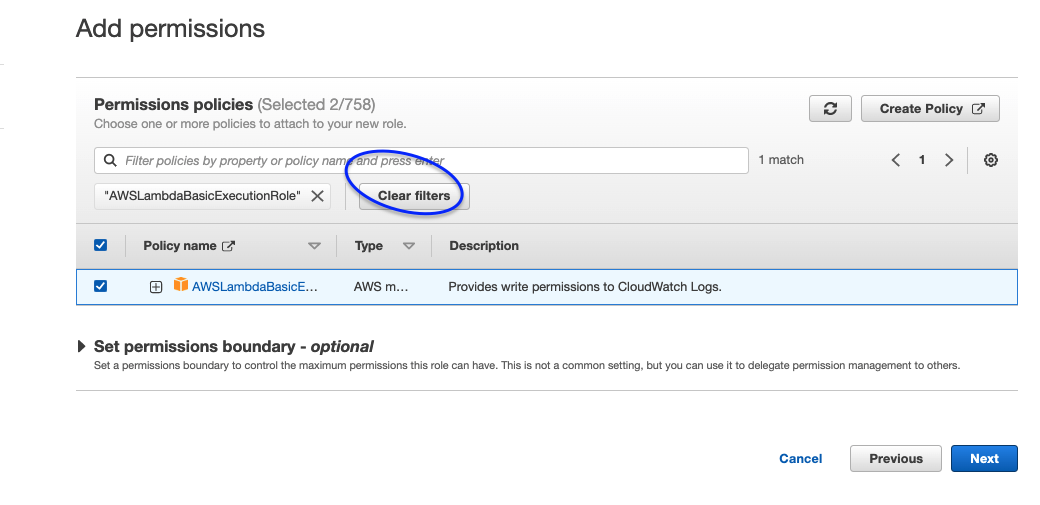
🡪 Yanina tik koyarak cikan satiri sectik

🡪 secili olarak kalsin, clear filtre

🡪 secimimiz tum listede gorunuyor



Yeniden arama ekranina “AWSLambdaBasicExecutionRole” bunu ekleyip enter’a bastik

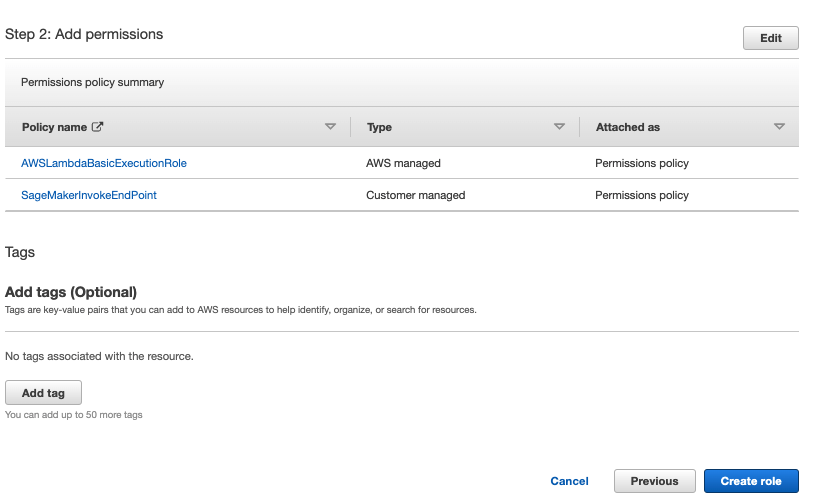


🡪 cikani sectik, clear filter dedik

Suan secili iki adet satir var

🡪 next

🡪 Role name : lambda-bzb



sayfanin altinda daha onceden sectigimiz 2 rol gorunmeli,

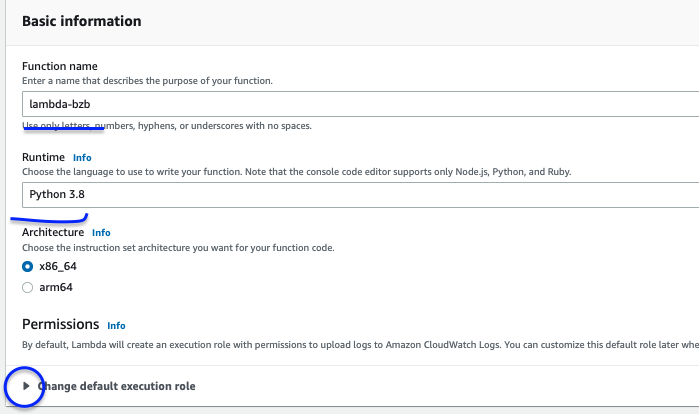
🡪 create role

LAMBDA SAYFASINI YENIDEN AC

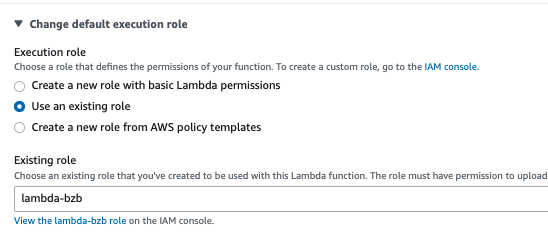
🡪 create function : lambda-bzb (farkli bir isim de verebiliriz)

(rolden bagimsiz olarak fonksiyona isim veriyoruz)

Runtime: pyhton



PERMISSION



🡪 create function

Lambda function olarak asagidaki kodu yapistirdik

# Reference:

# <https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/>

# <https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-output-format>

import boto3

import math

import dateutil

import json

import os

# grab environment variables

ENDPOINT\_NAME = os.environ['ENDPOINT\_NAME']

client = boto3.client(service\_name='sagemaker-runtime')

# Raw Data Structure:

# datetime,season,holiday,workingday,weather,temp,atemp,humidity,windspeed,casual,registered,count

# Model expects data in this format (it was trained with these features):

# season,holiday,workingday,weather,temp,atemp,humidity,windspeed,year,month,day,dayofweek,hour

def transform\_data(data):

try:

features = data.copy()

# Extract year, month, day, dayofweek, hour

dt = dateutil.parser.parse(features[0])

features.append(dt.year)

features.append(dt.month)

features.append(dt.day)

features.append(dt.weekday())

features.append(dt.hour)

# Return the transformed data. skip datetime field

return ','.join([str(feature) for feature in features[1:]])

except Exception as err:

print('Error when transforming: {0},{1}'.format(data,err))

raise Exception('Error when transforming: {0},{1}'.format(data,err))

def lambda\_handler(event, context):

try:

print("Received event: " + json.dumps(event, indent=2))

request = json.loads(json.dumps(event))

transformed\_data = [transform\_data(instance['features']) for instance in request["instances"]]

# XGBoost accepts data in CSV. It does not support JSON.

# So, we need to submit the request in CSV format

# Prediction for multiple observations in the same call

result = client.invoke\_endpoint(EndpointName=ENDPOINT\_NAME,

Body=('\n'.join(transformed\_data).encode('utf-8')),

ContentType='text/csv')

result = result['Body'].read().decode('utf-8')

# Apply inverse transformation to get the rental count

print(result)

import re # python regex module

s = result

pattern = r'[^0-9.]+'

result = re.split(pattern,s)

print(result)

#result = result.split(',')

predictions = [math.expm1(float(r)) for r in result if r != '']

return {

'statusCode': 200,

'isBase64Encoded':False,

'body': json.dumps(predictions)

}

except Exception as err:

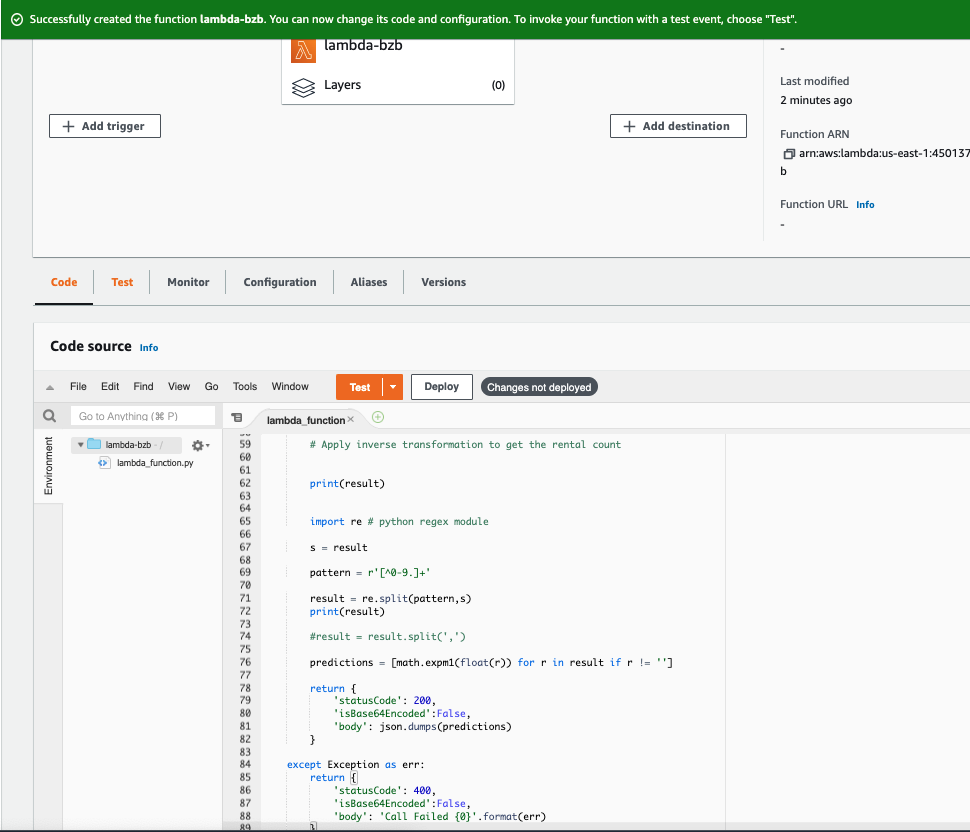
return {

'statusCode': 400,

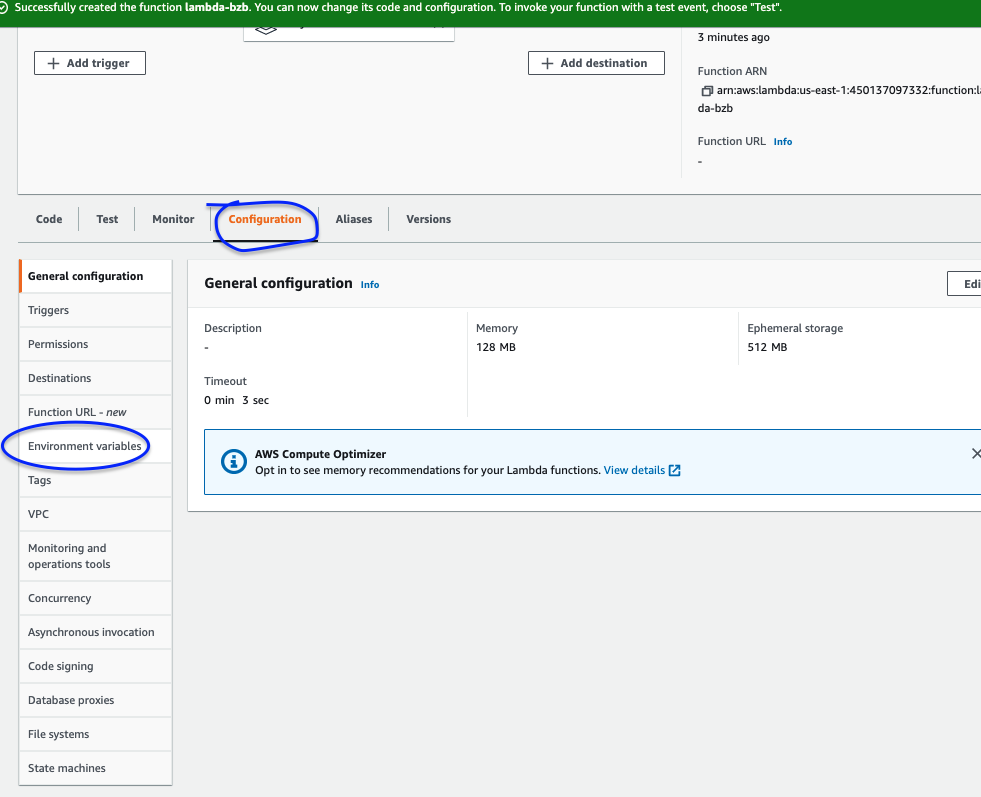
'isBase64Encoded':False,

'body': 'Call Failed {0}'.format(err)

}



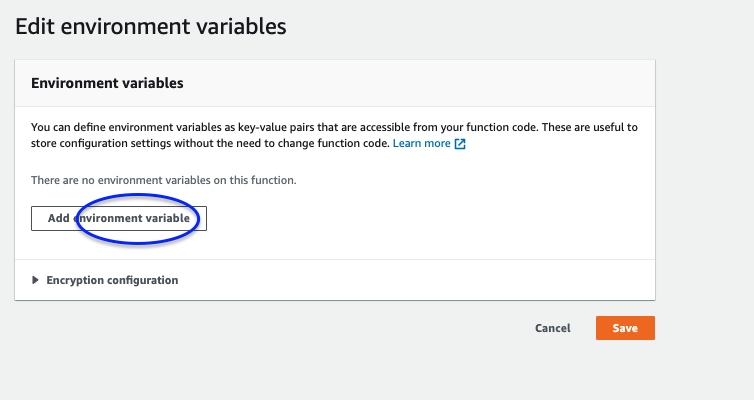
🡪 basit bir configuration yapicaz



🡪 environment variable tanimliyoruz

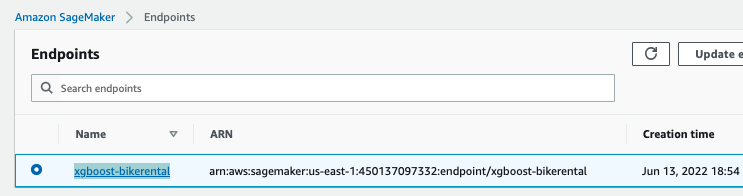
🡪 edit

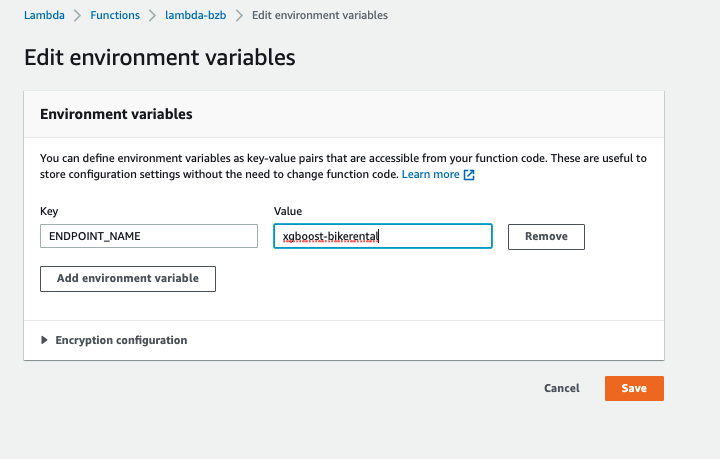
🡪 add environment



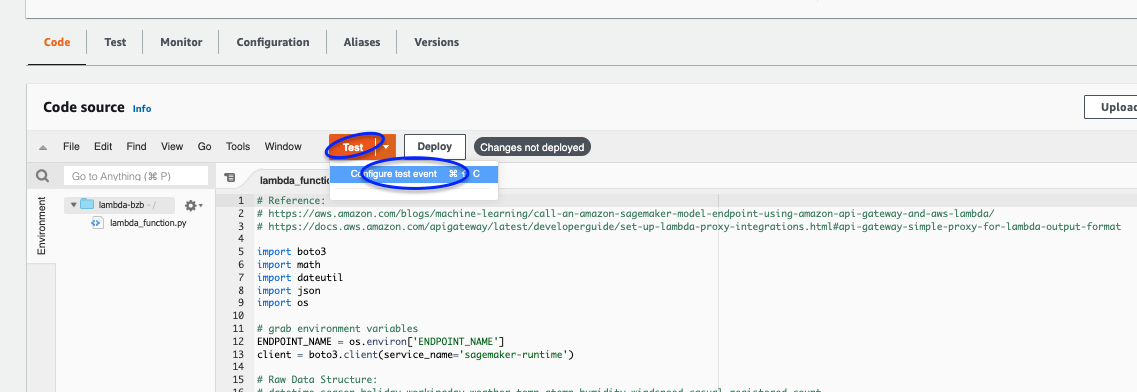
Value icin sagemaker 🡪 endpoints icindeki dosyayi aldik

“xgboost-bikerental”





--- Yaptiklarimizi test edelim



Acilan sayfada asagidaki kodu ekledik

{

"instances": [

{

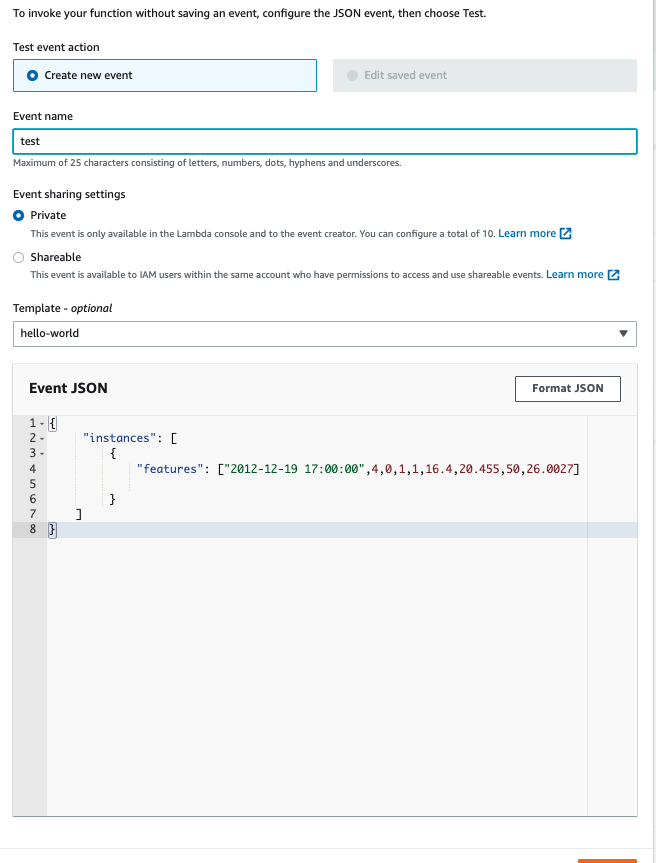
"features": ["2012-12-19 17:00:00",4,0,1,1,16.4,20.455,50,26.0027]

}

]

}

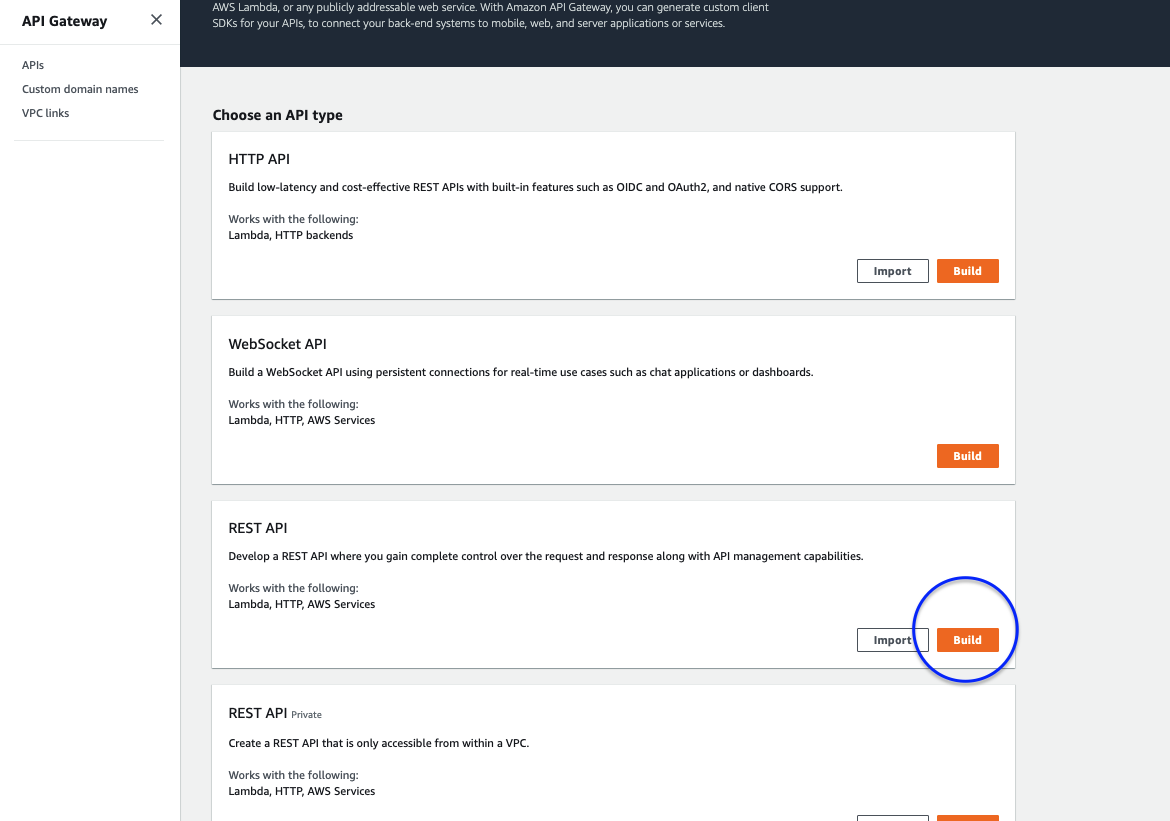
Asagidaki gibi gorunecek :

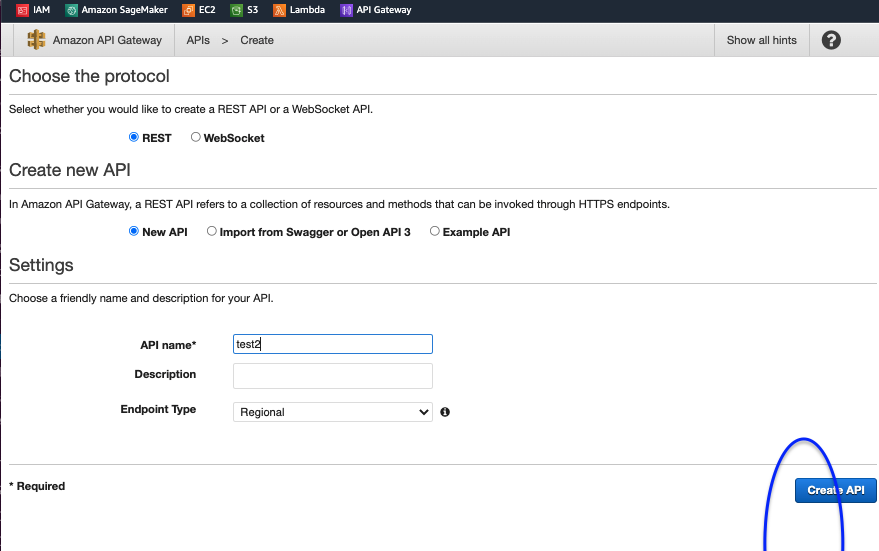


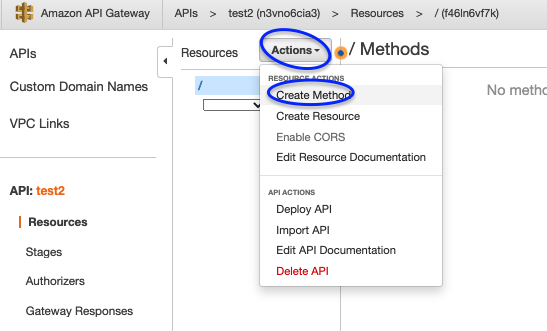
🡪 save dedik

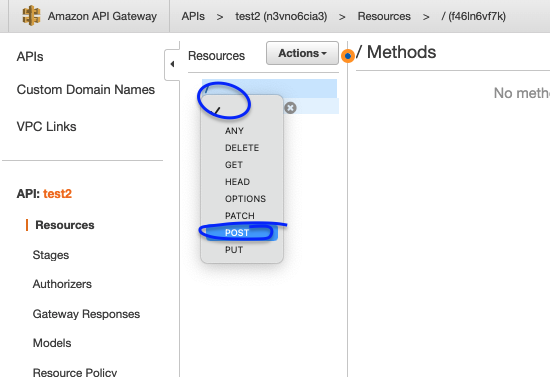
APIGATEWAY

3. olan REST API sec



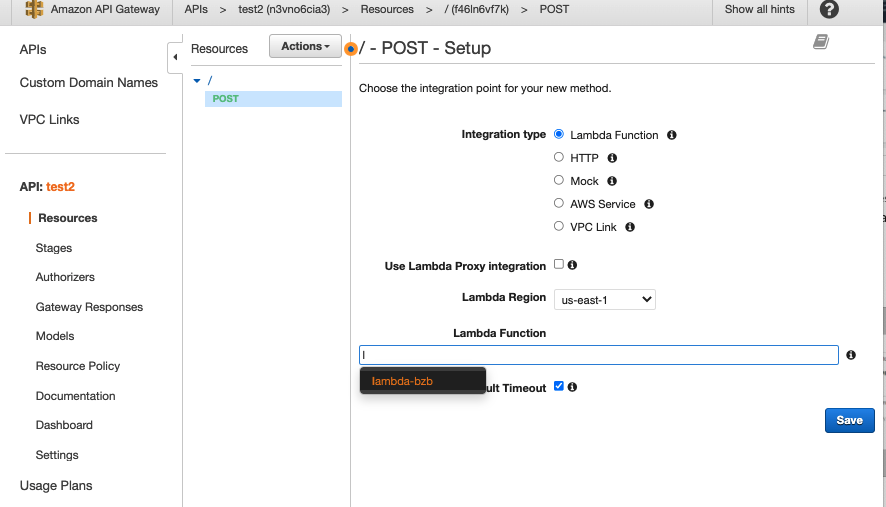


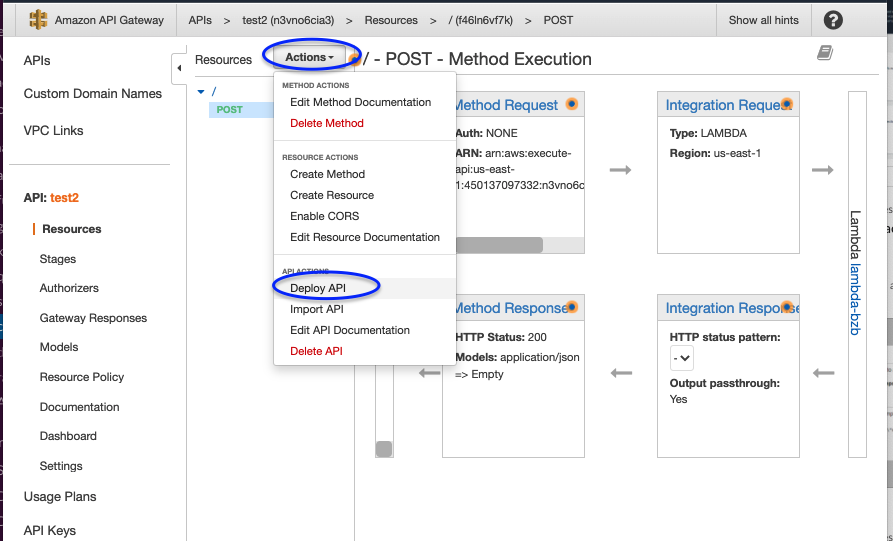


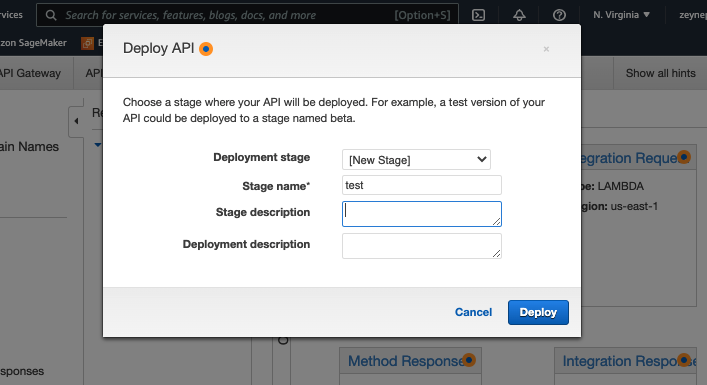


\

L yazinca otomatik cikti



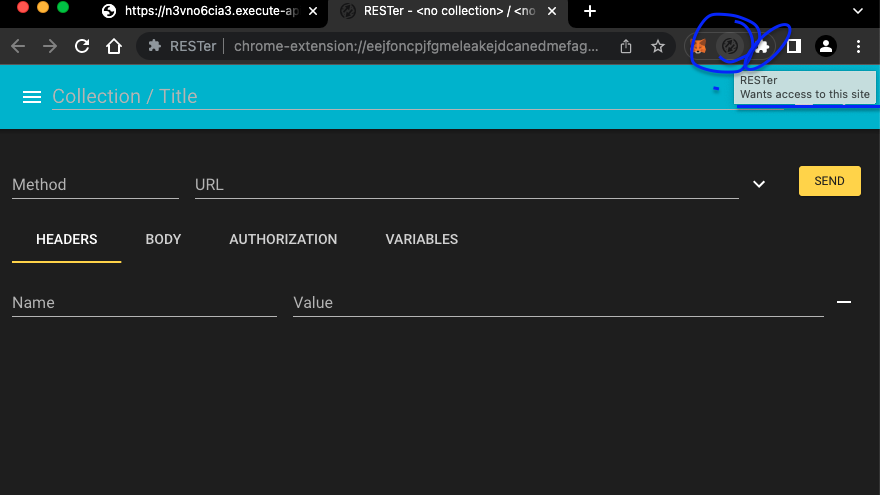


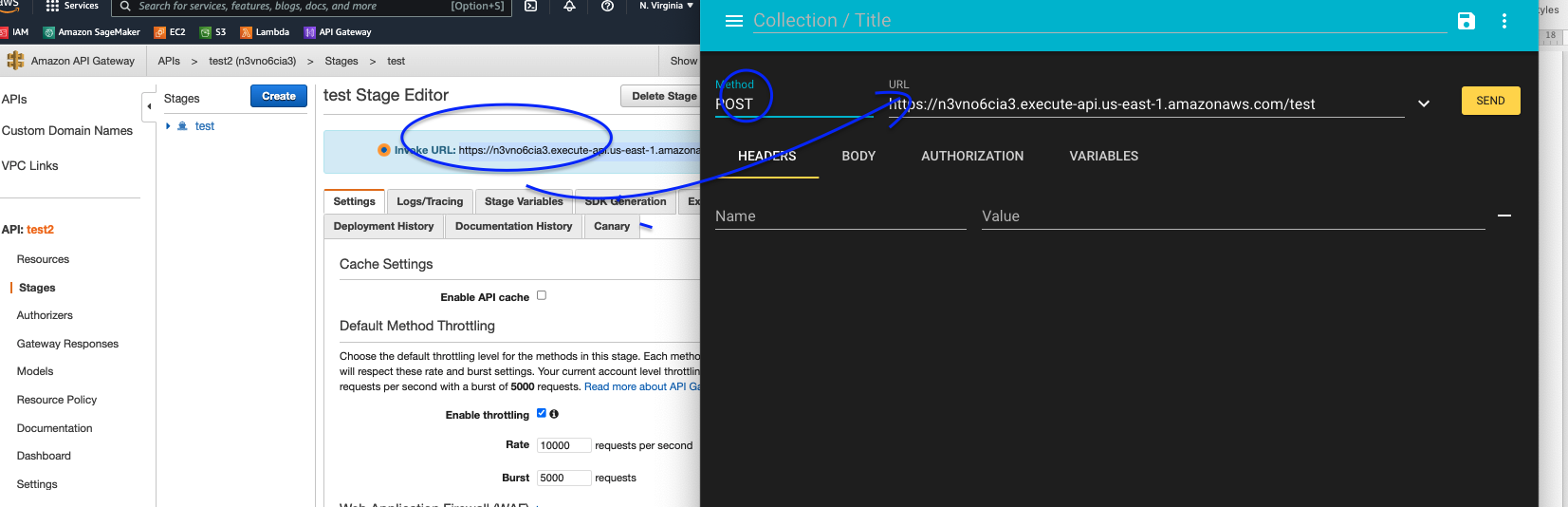


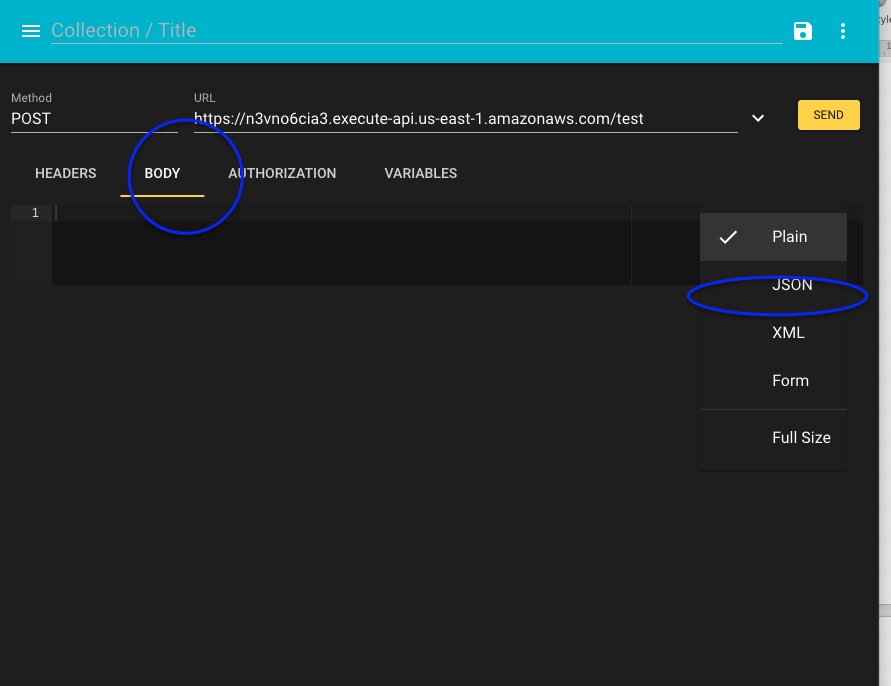
ENDPOINT acmis olduk,

Artik worldwide herkes ulasabilir

Chrom eklentisini ac,



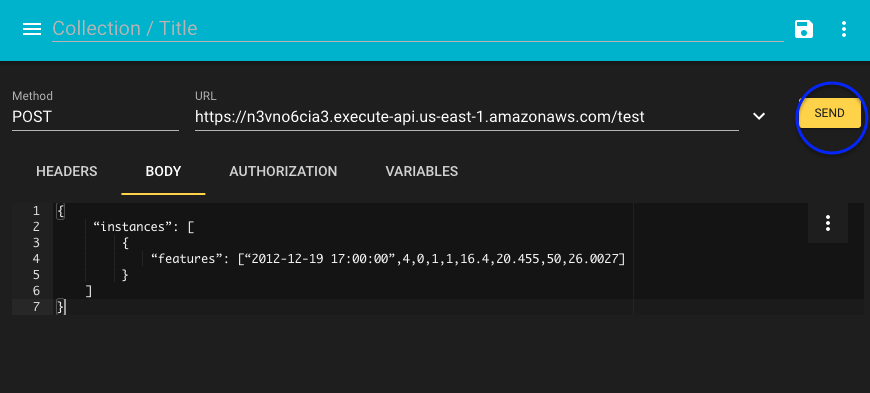




Asagidaki json file yapistir

Not: tirnak isaretlerini silip yeniden yazmak gerekebilir

{  
     “instances”: [  
         {  
             “features”: [“2012-12-19 17:00:00”,4,0,1,1,16.4,20.455,50,26.0027]         }  
    ]  
}



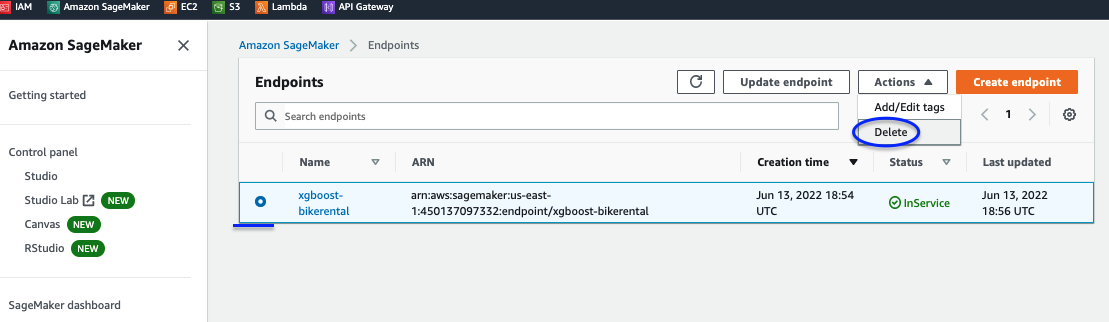
Yapistirdiktan sonra SEND

StatusCode: 200 almaliyiz



Vs code actik

ENDPOINT kapatmayi unutma



INSTANCE stop

