

AWS Model Deployment

Endpoint API Use Case

Key Steps


1. Use code file 'Iris_model_creation.ipynb' to build a model (say xgboost) locally using any dataset (say iris).
2. Start AWS Sagemaker → New Notebook
3. Upload model file in Notebook directory
4. Run 'my-ml-deploy-aws.ipynb' on notebook to create an endpoint
5. Verify that endpoint has been created, refer next slide

Endpoint created

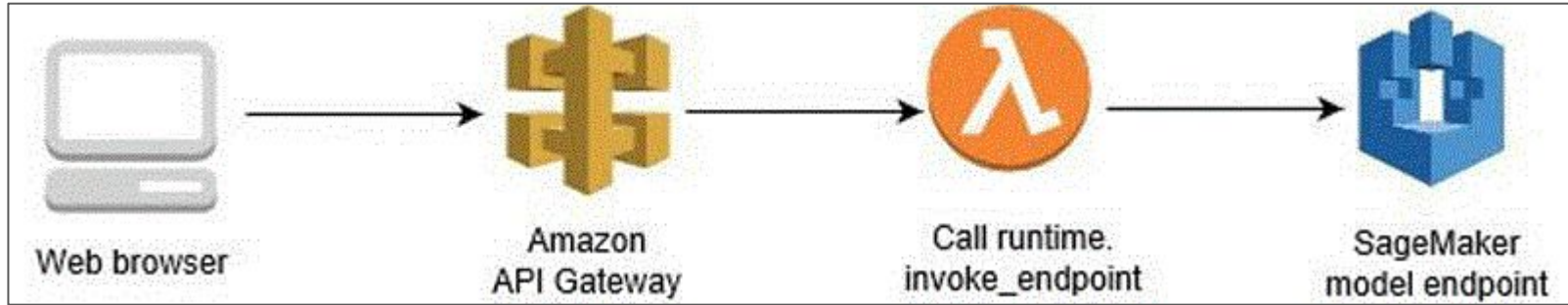
Amazon SageMaker > Endpoints > my-XGBoostEndpoint-2022-03-15-14-34-25

my-XGBoostEndpoint-2022-03-15-14-34-25

Endpoint settings

Name	Type
my-XGBoostEndpoint-2022-03-15-14-34-25	Real-time
ARN	Last updated
arn:aws:sagemaker:ap-south-1:288564358955:endpoint/my-xgboostendpoint-2022-03-15-14-34-25	Tue Mar 15 2022 20:07:06 GMT+0530 (India Standard Time)
Status	URL
 InService	https://runtime.sagemaker.ap-south-1.amazonaws.com/endpoints/my-XGBoostEndpoint-2022-03-15-14-34-25/invocations Learn more about the API

Next Step: Setting up API Gateway & Lambda Function



API gateway: Interface from where the HTTP request (POST method) will be received by AWS cloud

Lambda function: The code that receives input test data, queries the endpoint which has the ML model, obtains the result and gives back to the web browser

IAM role

Create IAM role that gives AWS Lambda permission to query the endpoint.

Go to IAM → create policy by putting following JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sagemaker:InvokeEndpoint",
      "Resource": "*"
    }
  ]
}
```

Then, go to IAM → create role using above policy

Lambda function

Go to AWS → Lambda → create a new function

Select runtime as 'python' and use the IAM role as execution role

Select configuration as assign a new environment variable

ENDPOINT_NAME = <name of your endpoint>

Add the code on next slide in it

```
import os
import boto3
import json

# grab environment variables

ENDPOINT_NAME = os.environ['ENDPOINT_NAME']

runtime= boto3.client('runtime.sagemaker')

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

    data = json.loads(json.dumps(event))
    payload = data['data']
    #print(payload)

    response = runtime.invoke_endpoint(EndpointName=ENDPOINT_NAME,
                                       ContentType='text/csv',
                                       Body=payload)

    #print(response)
    result = json.loads(response['Body'].read().decode())
    classes = ['Setosa', 'Versicolor', 'Virginica']
    res_list = [ float(i) for i in result]
    return classes[res_list.index(max(res_list))]
```

API Gateway

Go to AWS → API Gateway

Select REST API

Select 'actions', create a new method (POST)

Specify your lambda function created

Finally, select 'deploy'

Obtain the 'Invoke URL' (this will be the request URL)

Web Browser: Postman

