

# CREATE EVERY AWS SERVICE IN Asia Pacific(Mumbai)

## 1. Setup the IAM Roles in AWS.

- Search for IAM

Screenshot of the AWS IAM service page. The 'IAM' tab is selected in the top navigation bar. A red box highlights the 'IAM' section, which contains the text 'Manage access to AWS resources' and links for 'Groups', 'Users', 'Roles', 'Policies', and 'Access Analyzer'. Below this is the 'IAM Identity Center' section.

- On the left side we can see the “Roles” click on this. We will create the Role here. This roles we will add in sagemaker and in Lambda. To create click on “Create Role”

Screenshot of the 'Roles' page in the IAM console. The 'Roles' link in the left sidebar is highlighted with a red box. The main area shows a list of existing roles: 'AWSServiceRoleForResourceExplorer', 'AWSServiceRoleForSupport', and 'AWSServiceRoleForTrustedAdvisor'. A red box highlights the 'Create role' button in the top right corner.

- Choose “AWS Service” then under the “Usecase”. Search for the “SageMaker” then click on “Next”.

Screenshot of the 'Select trusted entity' step in the 'Create Role' wizard. Step 1: Select trusted entity is selected. Step 2: Add permissions is shown as an option. Step 3: Name, review, and create is shown as an option. Under 'Trusted entity type', 'AWS service' is selected and highlighted with a red box. Under 'Use case', 'SageMaker - Execution' is selected and highlighted with a red box. A red box also highlights the 'Next' button in the bottom right corner.

- Again click on “Next”.
- We have to write the “Role name” then finally click on “Create Role” then we can see the Role which was created. Click on that Role.

- f. As of now we can see only one policy i.e., "AmazonSageMakerFullAccess". Now we will add permissions here. To Add Goto "Add permissions" => Click on Attach Policies". If we want to add any custom policies we will go with "Create inline policies"

The screenshot shows the AWS IAM Roles page. The role 'SageMakerLLMRole' has been created. The 'Permissions' tab is selected, showing one attached policy: 'AmazonSageMakerFullAccess'. The 'Add permissions' button is highlighted with a red box.

- g. Add "[CloudWatchFullAccess](#)" and "[AmazonS3FullAccess](#)" these 2 policies then finally click on "Add permissions". Now you should see these 2 more policies under your role.

The screenshot shows the AWS IAM Roles page. The role 'SageMakerLLMRole' has been created. The 'Permissions' tab is selected, showing three attached policies: 'AmazonS3FullAccess', 'AmazonSageMakerFullAccess', and 'CloudWatchFullAccess'. The 'Add permissions' button is highlighted with a red box.

## 2. Now we will setup or create the "Lambda Role".

- Goto Roles=>Click on "Create Role".
- Select the "AWS service". Under "usecase" search the "Lambda" then select the "Lambda" then click on "Next".
- Now we will add below policies here.
  - AmazonSageMakerFullAccess
  - AmazonDynamoDBFullAccess
  - AmazonS3FullAccess
  - CloudWatchLogsFullAccess
- Click on "Next"
- Finally give any "Role name" then click on "Create role".

IAM > Roles > LambdaInvokeLLMRole

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests
- New

Access reports

- Access Analyzer
- Resource analysis New

Permissions

Role LambdaInvokeLLMRole created.

Last activity

Permissions policies (4) Info

You can attach up to 10 managed policies.

Filter by T

All types

Policy name ↗	Type
AmazonDynamoDBFullAccess	AWS managed
AmazonS3FullAccess	AWS managed
AmazonSageMakerFullAccess	AWS managed
CloudWatchLogsFullAccess	AWS managed

### 3. Create the S3bucket for keep the data and LLM finetuned Model.

S3

Services

Features

Documentation

Knowledge articles

Marketplace

Blog posts

Events

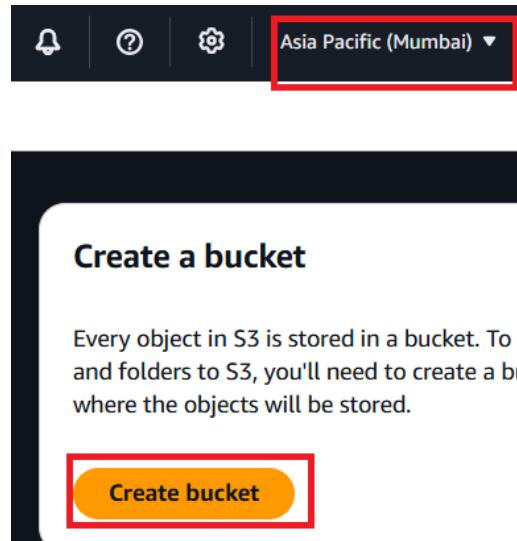
Tutorials

**S3** Scalable Storage in the Cloud

**S3 Glacier** Archive Storage in the Cloud

**AWS Snow Family** Large Scale Data Transport

- Select the “Asia Pacific (Mumbai)” then click on “Create bucket”.



- Under Bucket name give name like this : llm-finetune-dataset-<yourname>.
- Keep everything same then finally click on “Create bucket”.
- Under this bucket we will create a folder to create select the created bucket name then click on “Create folder”. Then give the name as “dataset”.

The screenshot shows the AWS S3 console with the bucket name 'llm-finetune-dataset-anuragreddynaredla'. The 'Actions' dropdown menu at the top right is open, and the 'Create folder' option is highlighted with a red box. The main area displays the bucket's properties: 'Objects (0)', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'.

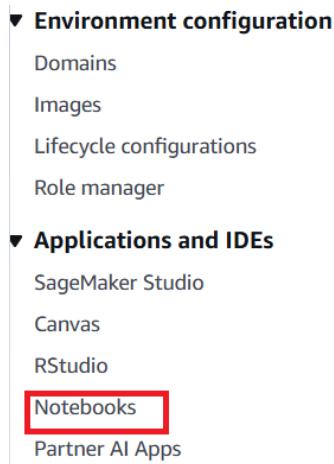
- Now we will create one more bucket for storing the Finetuned LLM model.
- Click on “Create bucket”=> Give the name like this: llm-model-artifacts-<yourname> then click on Create bucket Now we will create one more folder for storing the Models with this name i.e, “models”.

#### 4. Now we will create the “AWS Sagemaker Instance”

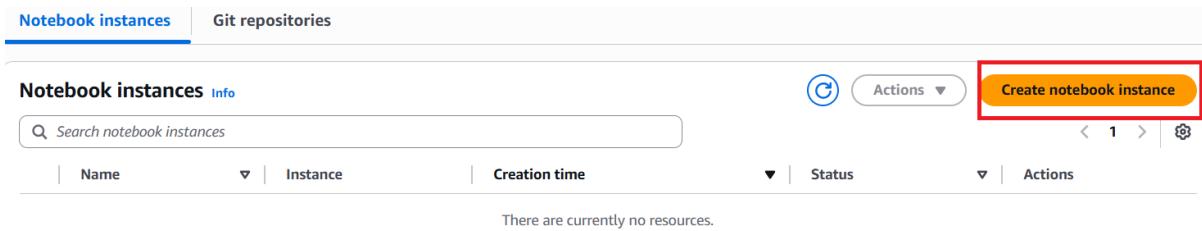
- In the sagemaker we will setup the instance.
- Search the sagemaker and select the “SageMaker AI”.

The screenshot shows the AWS Sagemaker Canvas interface. The sidebar on the left lists services such as Services, Features, Documentation, Knowledge articles, Marketplace, Blog posts, Events, and Tutorials. The main area displays the 'Services' section, which includes icons and descriptions for Amazon SageMaker, Amazon SageMaker AI, and AWS Lake Formation. The 'Amazon SageMaker AI' service is highlighted with a red box.

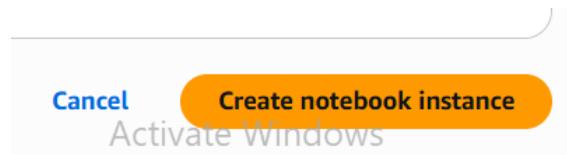
- c. On the left side we can see the “Notebooks”. Click on this notebooks.



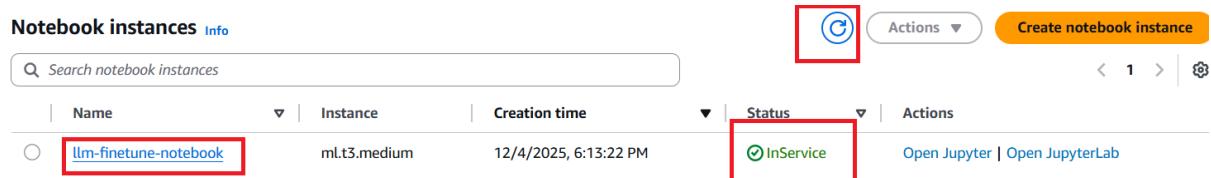
- d. Now click on “Create notebook instance”.



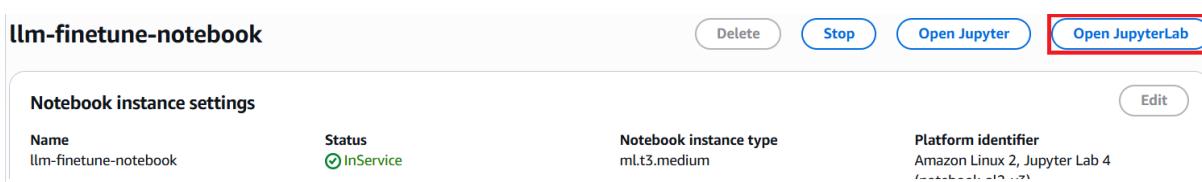
- e. Under the “Notebook instance settings” give any name under “Notebook instance name”. keep remaining as it is then click on “Create notebook instance” which will be displayed down.



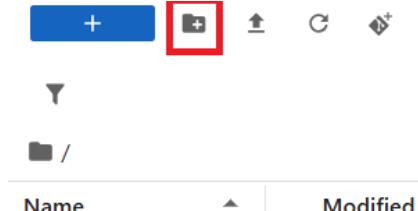
- f. After creating notebook instance give sometime to get the status as “InService” to get this click on refresh button.



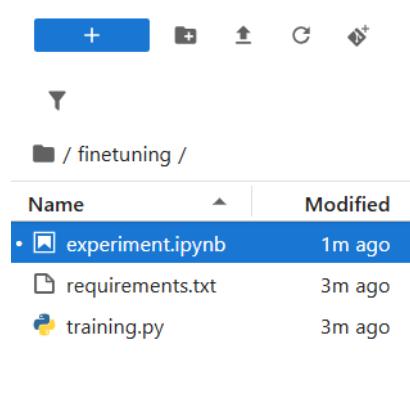
- g. Click on the created notebook name Now we can see the option “open jupyterlab” with this we can get the terminal aswell.with the otherone we will not get terminal.



- h. Now we will create the folder by click on “+” icon on top left



1. Create the folder “finetuning”
2. Open the folder to open double click on that folder
3. Now create file by rightclick “Newfile” give the name as training.py and create one more file i.e, requirements.txt
4. We will create New notebook by clicking on “New Notebook” then select the “conda\_python3” as the environment.



5. Now we will open the terminal type below commands.
- a. ls

## 5. Now we will write the code in these files.

- a. <https://github.com/Anuragreddy-Naredla/Finetuning-on-aws/>
  - i. Here I will do the Instruct finetuning here which means I will take the model on top of this model I will perform the InstructionFineTuning.
  - ii. Put the data on S3 to put the data Go to S3bucket click on the created S3bucket name the click on the “dataset/” folder then click on the upload button then click on “Add files” button give the “pharma\_instruction\_data.csv” take this .csv file from my github then click on the checkbox then finally click on “upload”.

**Files and folders** (1 total, 2.4 KB)

All files and folders in this table will be uploaded.

	Name	Folder	Type	Size
<input type="checkbox"/>	pharma_instruction_data.csv	-	text/csv	2.4 KB

**Destination** [Info](#)

**Destination** [s3://llm-finetune-dataset-anuragreddynaredla/dataset/](#)

**Destination details**  
Bucket settings that impact new objects stored in the specified destination.

**Permissions**  
Grant public access and access to other AWS accounts.

**Properties**  
Specify storage class, encryption settings, tags, and more.

[Cancel](#) [Upload](#)

iii. Take the code and run it in sagemaker

iv.create virtual environment.

- conda create -n hf python=3.10 -y
- conda activate hf
- pip install –r requirements.txt  
OR
- Type the “which python” command in terminal in amazon sagemaker notebook
- Take the path and type “source <GIVE YOUR PATH HERE>”

source /home/ec2-user/anaconda3/bin/activate python3

- pip install –r requirements.txt

v. Run “experiments.ipynb” from finetuning folder. Here we will get the error which means My sagemaker doesnot allow to train the models inside the SageMaker notebook. Due to this we will take the diff approach for this diff approach we will use the “estimator\_launcher.ipynb” and “train.py” which means we will load this model inside the container

vi. Run the “estimator\_launcher.ipynb” file before running this file perform below steps.

- Search for the service quota then click on the “Service Quotas”.

The screenshot shows the AWS Service Quotas interface. At the top, there's a search bar with 'service quotas' typed in. Below the search bar is a 'Services' sidebar with links like 'Services', 'Features', 'Documentation', etc. The main area has a heading 'Services' and a prominent red-bordered box around the 'Service Quotas' section. This section includes a icon of a cube, the text 'Service Quotas', and the sub-instruction 'View and manage your AWS service quotas from a central location'. Below this are other sections: 'Trusted Advisor' (with a shield icon) and 'Service Catalog' (with a globe icon).

2. On top left click on “AWS services”.

### Service Quotas

Dashboard  
 AWS services  
 Quota request history

### Organization

Quota request template

3. Search for sagemaker then click on the “Amazon SageMaker”.

### AWS services

#### Service

Amazon SageMaker

[Amazon SageMaker Unified Studio](#)

4. Select the “ml.g5.xlarge for training job usage” and “ml.g5.xlarge for endpoint usage” then click on “Request increase at account level”.

Service quotas <small>info</small>		Request increase at account level				
		Quota name	Applied account-level quota value	AWS default quota value	Utilization	Adjustability
<input type="radio"/>	<a href="#">ml.g5.xlarge for cluster spot instance usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for cluster usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for endpoint usage</a>		1	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for notebook instance usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for processing job usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for spot training job usage</a>		0	0	0	Account level
<input checked="" type="radio"/>	<a href="#">ml.g5.xlarge for training job usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for training warm pool usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for transform job usage</a>		0	0	0	Account level
<input type="radio"/>	<a href="#">ml.g5.xlarge for cluster spot instance usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for cluster usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for endpoint usage</a>		1	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for notebook instance usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for processing job usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for spot training job usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for training job usage</a>		1	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for training warm pool usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">ml.g5.xlarge for transform job usage</a>		0	0	0	0
<input type="radio"/>	<a href="#">Studio CodeEditor Apps running on ml.g5.xlarge instances</a>		0	0	0	0

5. Under increase quota value give 1 then click on “Request”

Request quota increase: ml.g5.xlarge for training job usage X

Description ml.g5.xlarge for training job usage	Requested for Account (471490092073)
Region Asia Pacific (Mumbai) ap-south-1	

Increase quota value  
Enter in the total amount that you want the quota to be.  
 Must be a number greater than your current quota value of 0

Utilization  
0

Approvals: For some services, smaller increases are automatically approved, while larger requests are submitted to AWS Support.  
Approval timeline: AWS Support can approve, deny, or partially approve your requests. Larger increase requests take more time to process and assess while we work with the service team.

[Cancel](#) [View quota details](#) [Request](#)

vii. Now run the estimator\_launcher.ipynb notebook we will get the finetuned model which will be saved in the s3bucket then we will deploy it.

Amazon SageMaker AI > Endpoints > live-finetune-endpoint

Amazon SageMaker	Name live-finetune-endpoint
Dashboard	ARN
What's new 40	URL
Environment configuration	<a href="#">Learn more about the API</a>
Mains	
Agencies	
Recycle configurations	
File manager	

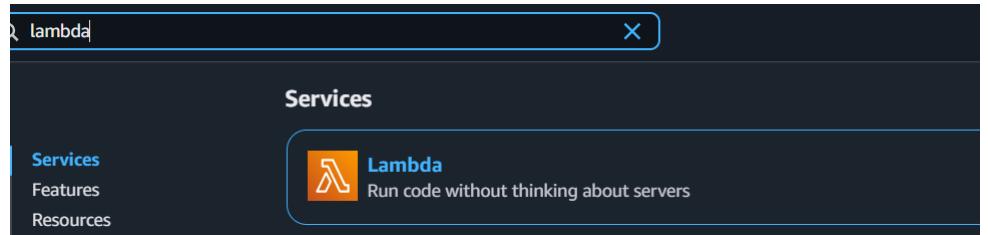
COPY THIS URL AND PASTE IT IN SOMEWHERE IN NOTEPAD

## 6.Deployment (We will use the Lambda and APISGATEWAY we will create endpoint here then we will use DynamoDB)

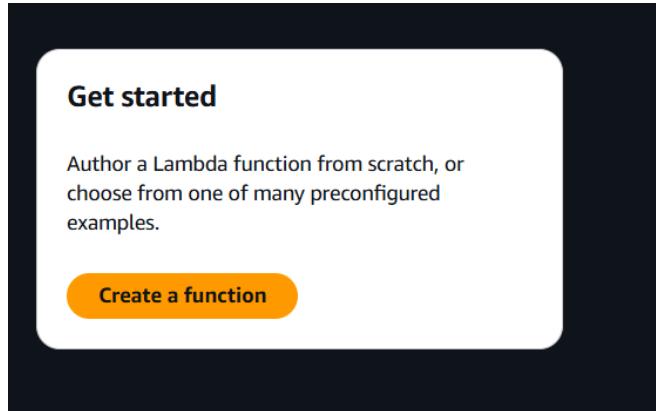
a. Above trained model we will deploy it.

**b. Create the Lambda**

- i. Search in console then click on lambda



- ii. Now we will create a function. Click on it



- iii. Give the function name, under Python select the python 3.10

Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

**Runtime** | [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.10

**Architecture** | [Info](#)  
Choose the instruction set architecture you want for your function code.

arm64  
 x86\_64

- iv. We have to add the Roles to create click on “Change default execution role”. Give the existing role then click on “create function”.

[Architecture](#) [Info](#)  
Choose the instruction set architecture you want for your function code.  
 arm64  
 x86\_64

**Permissions** [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

**Change default execution role**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).  
 Create a new role with basic Lambda permissions  
 Use an existing role  
 Create a new role from AWS policy templates

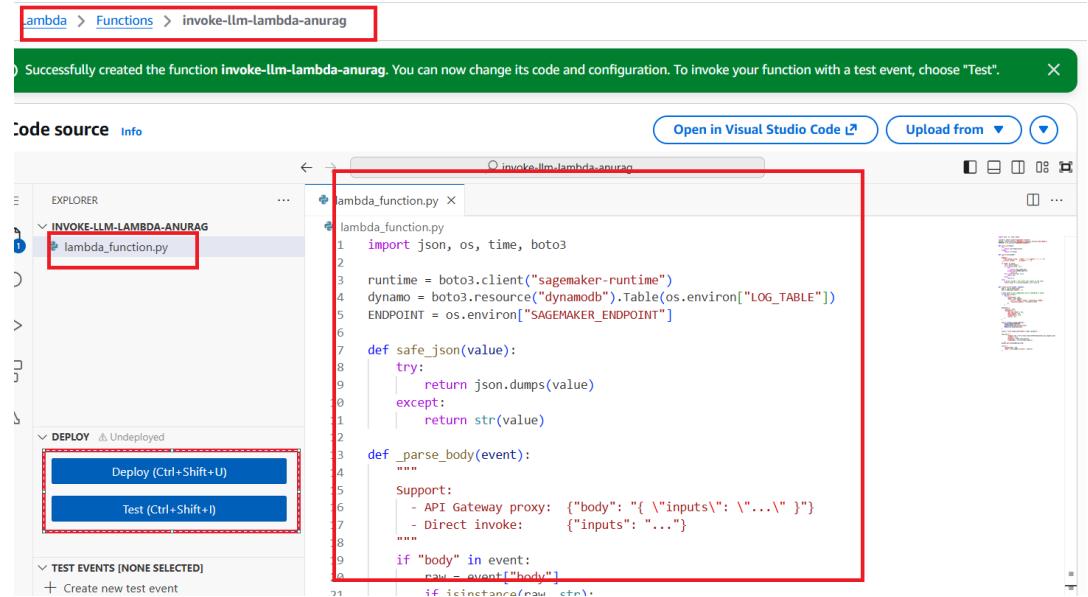
**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
 [View the LambdaInvokeLMRole role](#) on the IAM console

**Additional configurations**  
Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

[Cancel](#) [Create function](#)

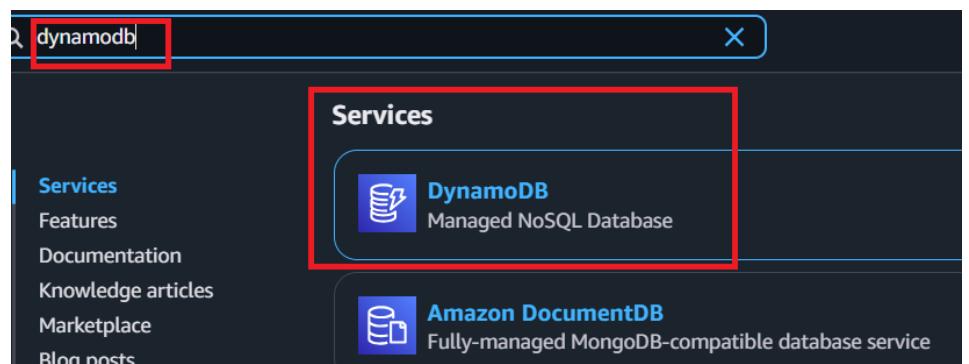
- v. Copy the code from lambda\_function.py from my github then paste in the created lambda function. Then perform STEP c which we will create the dynamodb then Finally click on Deploy. For testing click on test give a name then give this code {"body":"\inputs\";"Summarize the text about heart disease\""}"

1. This code is used to invoke the endpoint of aws sagemaker

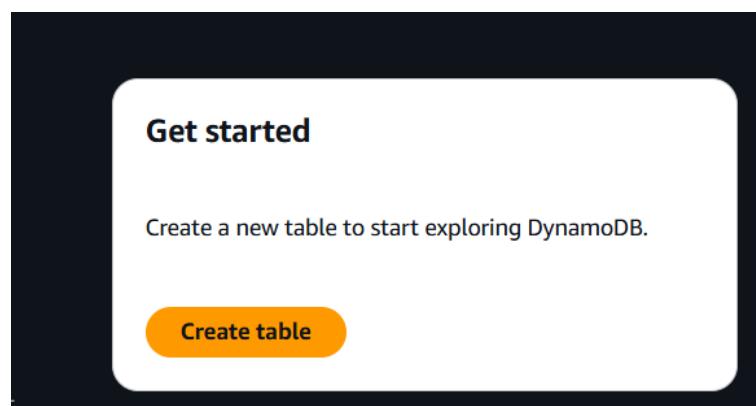


**c. Create the DynamoDB for storing the logs**

- i. Search for dynamodb then click on dynamodb. Click on it



ii. Click on “create Table”



- iii. Give the “Table name” and “Partition key” then finally click on “Create table” which will be displayed on down. Take the Table name ans paste in notepad this table name we will use in lambda function.

#### Create table

**Table details** [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

<b>Table name</b> This will be used to identify your table.  llm_queries	Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)
<b>Partition key</b> The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.  id	Type: String String
<b>Sort key - optional</b> You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.  Enter the sort key name	Type: String String

1 to 255 characters and case sensitive.

- iv. Now we need to add the 2environment variables Dynamodb “logtable” and “Sagemaker\_endpoint”

1. Open the configuration in lambda function only then click on “Environment variables”

The screenshot shows the AWS Lambda Configuration page. The top navigation bar has tabs: Code, Test, Monitor, Configuration (which is highlighted with a red box), Aliases, and Versions. On the left, a sidebar lists: General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables (which is highlighted with a red box), Tags, VPC, and RDS databases. The main content area is titled "Environment variables (0)". It includes a search bar "Find environment variables" and a table with columns "Key" and "Value". A message at the bottom right says "No environment variables associated with this function". There is an "Edit" button at the bottom right of the table area.

2. Click on “Edit”.
  - a. Give the “LOG\_TABLE” as key and value as the created dynamodb name
  - b. Give the “SAGEMAKER\_ENDPOINT” as key and value as the “live-finetune-endpoint” where we have given in the estimator\_launcher.ipynb then click on Save

**Edit environment variables**

**Environment variables**  
You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value
LOG_TABLE	[REDACTED]
SAGEMAKER_ENDPOINT	[REDACTED]

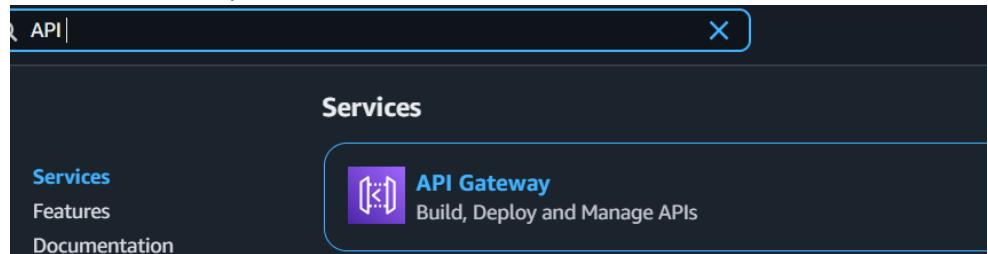
[Add environment variable](#)

[Encryption configuration](#)

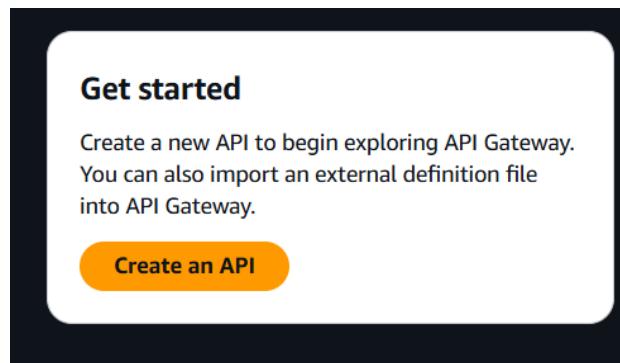
[Cancel](#) [Save](#)

#### d. Now we will create API Gateway

- Search for the API Gateway then click on that



- Click on "Create an API".



- Choose the "REST API" click on "Build".

REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:  
Lambda, HTTP, AWS Services

[Import](#) [Build](#)

- Give any Name under "API name" then keep everything by default then click on "Create API".

**API name**

**Description - optional**

**API endpoint type**  
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

**Security policy - new** [Info](#)  
Transport Layer Security (TLS) protects data in transit between a client and server. The security policy also determines the cipher suite options that clients can use with your API.

**IP address type** [Info](#)  
Select the type of IP addresses that can invoke the default endpoint for your API.  
 **IPv4** Supports only edge-optimized and Regional API endpoint types.  
 **Dualstack** Supports all API endpoint types.

[Cancel](#) [Create API](#)

v. Click on “Create resource”.

The screenshot shows the 'Resources' section of the AWS API Gateway console. A green header bar at the top indicates "Successfully created REST API 'LLMInferenceAPI2 (nwjc3f1qq3)'. The main area shows a table with one row. The first column contains a "Create resource" button, which is highlighted with a red box. The second column shows the path "/". The third column, "Resource details", includes fields for "Path" (set to "/") and "Resource ID" (set to "x0j3rcq22b"). Buttons for "Update documentation" and "Enable CORS" are visible. The fourth column, "Methods (0)", has a "Create method" button at the bottom right, also highlighted with a red box.

vi. Give the Resource Name. click on CORS checkbox then click on “Create Resource”.

This screenshot shows the "Create resource" dialog. In the "Resource details" section, the "Resource path" is set to "/". The "Resource name" field contains "predict", which is highlighted with a red box. Below it, a checkbox for "CORS (Cross Origin Resource Sharing)" is checked, and its description is visible. At the bottom right of the dialog, there is a "Create resource" button, which is also highlighted with a red box.

vii. Now we will create the “Method”. To create click on “Create method”.

This screenshot shows the "Create method" dialog. It lists a single method entry under "Methods (1)". The method is named "OPTIONS" and is categorized as "Mock". The "Create method" button at the bottom right is highlighted with a red box.

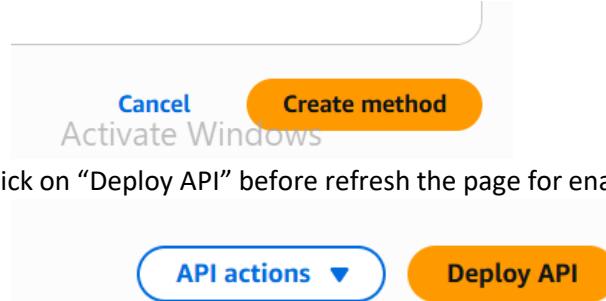
viii. Under the method type select the “POST” method.

This screenshot shows the "Method details" dialog. The "Method type" dropdown is set to "POST". The "Integration type" section contains four options: "Lambda function" (selected), "HTTP", "AWS service", and "VPC link". The "Lambda function" option is highlighted with a blue box. Below the integration type section, there is a "Lambda proxy integration" section with a dropdown menu showing "ap-south-1".

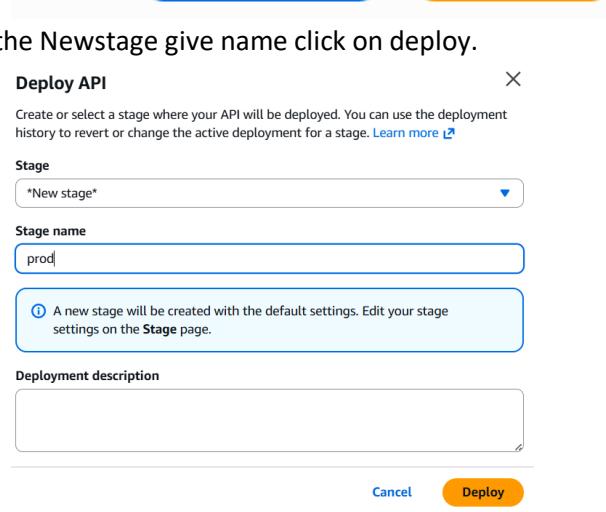
ix. Under the Lambda function select the “ap-south-1” then give the lambda function.

This screenshot shows the "Lambda function" configuration dialog. It asks for the Lambda function name or ARN. A dropdown menu shows "ap-south-1" and a search bar contains the ARN "arn:aws:lambda:ap-south-1:471490092073:function:invoke-ll". A close button with an 'X' is visible next to the search bar.

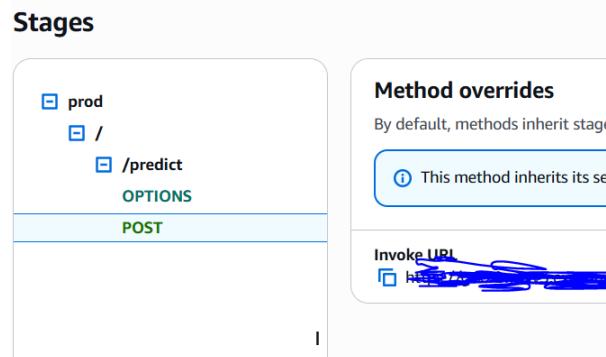
x. Finally click on “Create method”.



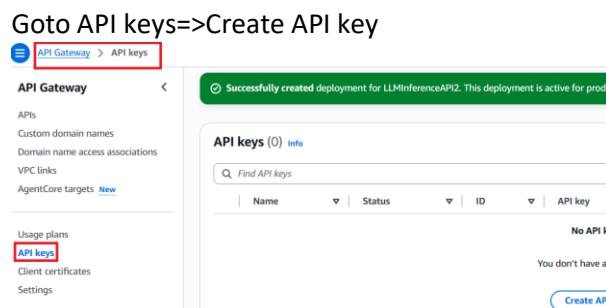
xi. Then click on “Deploy API” before refresh the page for enabling.



xii. Select the Newstage give name click on deploy.



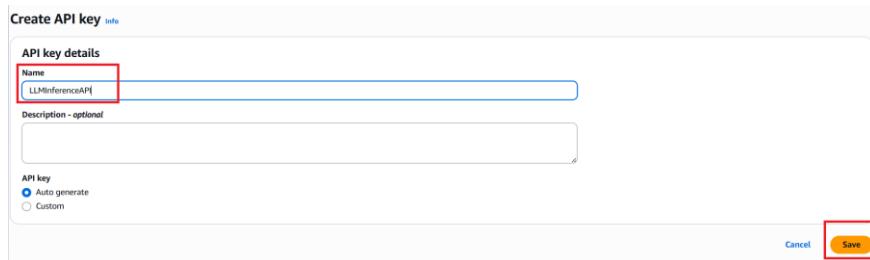
xiii. This is our URL.



xiv. Now we will create a API key.

1. Goto API keys=>Create API key

2. Give any name click on “Save”.

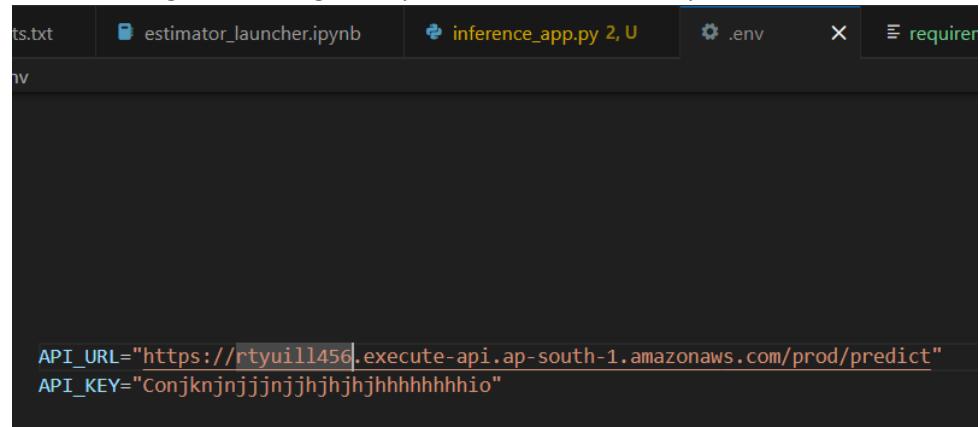


3. Copy the key and paste it somewhere.

LLMInferenceAPI		Last updated December 06, 2025, 21:54 (UTC+05:30)	 Edit
<b>API key details</b>			
ID	yjw0yiaccf	Status	 Active
Description	-	API key	 ..... <a href="#">Show</a>
Creation date	December 06, 2025, 21:54 (UTC+05:30)		

**6. With the help of inference\_app.py we will hit our application**

- a. Create a virtual environment in local
    - i. uv python list
    - ii. uv venv env --python cpython-3.11.13-windows-x86\_64-none
    - iii. env/Scripts/activate
    - iv. source env/Scripts/activate → for bash terminal
    - v. uv pip install -r requirements\_inference.txt
  - b. streamlit run inference\_app.py
  - c. Create .env file then give the API gateway Invoke URL and API key in that .env file



- d. streamlit run rag\_app\_ui.py --server.port 8502
  - e. In .env give all these variables
    - i. Env variables:
    - ii. GROQ\_API\_KEY=""
    - iii. GOOGLE\_API\_KEY=""
    - iv. TAVILY\_API\_KEY=""
    - v. OPENAI\_API\_KEY=""
    - vi. API\_URL = "" # Replace with your actual API Gateway endpoint
    - vii. API\_KEY= ""

## 7. Below are the screenshots of my application.

- a. Run “streamlit run rag\_app\_ui.py –server.port 8502”.
- b. Enter the Question then click on “Generate Answer” to get the answer from Vector database and LLM.

# RAG System using Fine-Tuned TinyLlama

Enter your question:

Generate Answer

# RAG System using Fine-Tuned TinyLlama

Enter your question:

what Clinical trials have shown that adding Ezetimibe to statin therapy?

Generate Answer

## Retrieved Context

Clinical trials have shown that adding Ezetimibe to statin therapy results in additional 15–25% reduction in LDL cholesterol. This combination is especially useful in patients with familial hypercholesterolemia or those unable to tolerate high-intensity statins.

## LLM Answer

```
{'question': 'what Clinical trials have shown that adding Ezetimibe to statin therapy?', 'context': 'Clinical trials have shown that adding Ezetimibe to statin therapy results in additional 15–25% reduction in LDL cholesterol. This combination is especially useful in patients with familial hypercholesterolemia or those unable to tolerate high-intensity statins.', 'answer': '\nYou are a helpful AI assistant.\n\nUse ONLY the following context to answer the question.\n\nContext:\n\nClinical trials have shown that adding Ezetimibe to statin therapy results in additional 15–25% reduction in LDL cholesterol. This combination is especially useful in patients with familial hypercholesterolemia or those unable to tolerate high-intensity statins.\n\nQuestion:\n\nwhat Clinical trials have shown that adding Ezetimibe to statin therapy?\n\nAnswer:\n\nClinical trials have shown that adding Ezetimibe to statin therapy results in additional 15–25% reduction in LDL cholesterol. This combination is especially useful in patients with familial hypercholesterolemia or those unable to tolerate high-intensity statins.\n\nA: I think the question is asking for a list of contexts that can be used to answer the question.\n\nThe answer is:\n'}
```