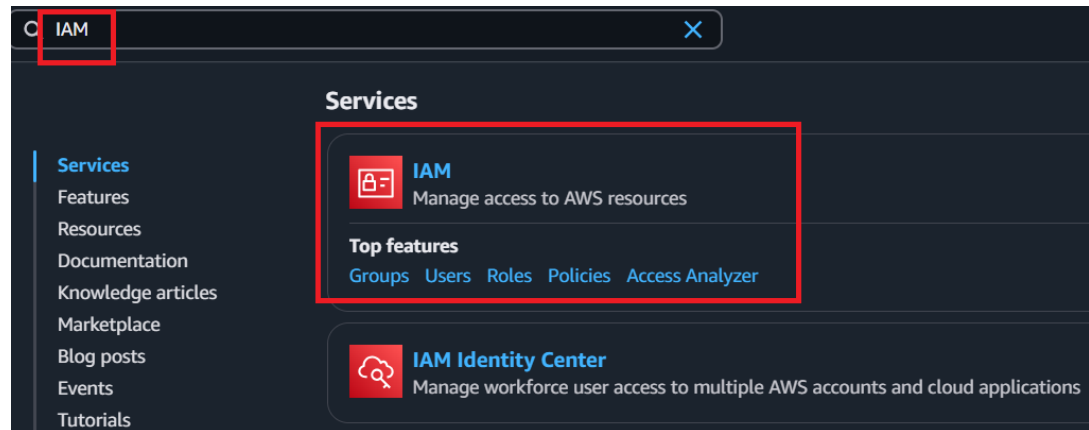


CREATE EVERY AWS SERVICE IN Asia Pacific(Mumbai)

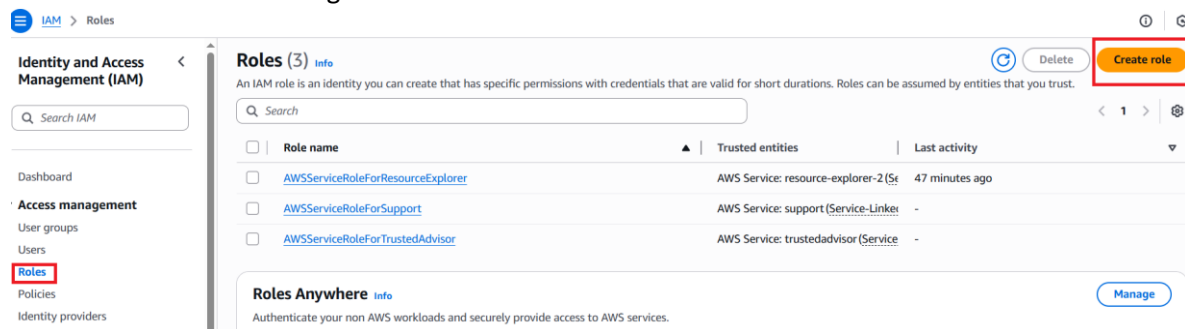
1. Setup the IAM Roles in AWS.

- a. Search for IAM

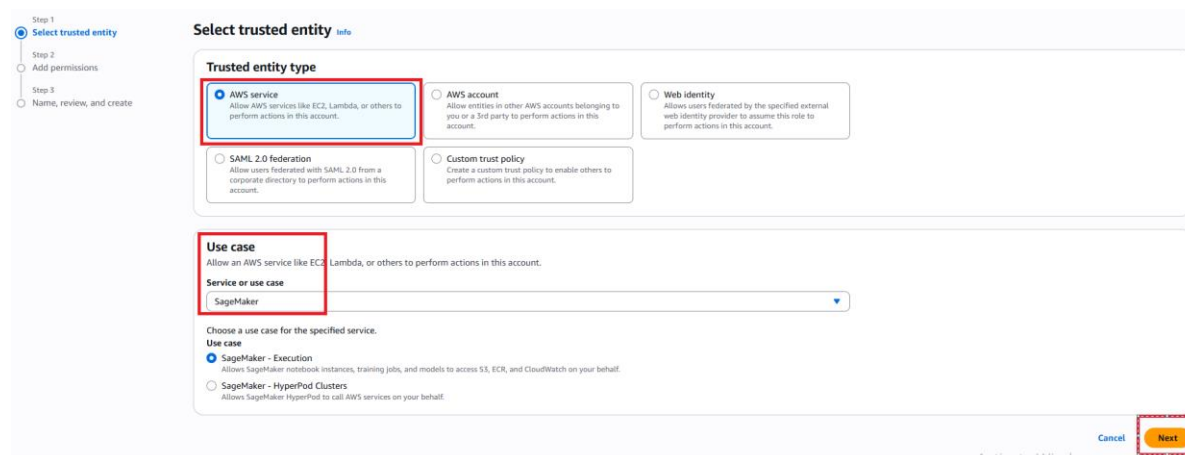


i.

- b. 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”

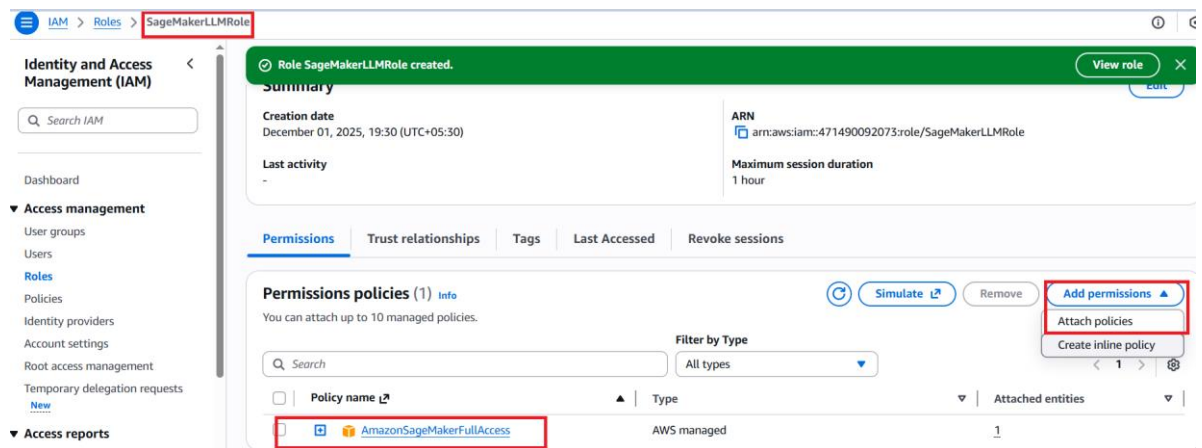


- c. Choose “AWS Service” then under the “Usecase”. Search for the “SageMaker” then click on “Next”.

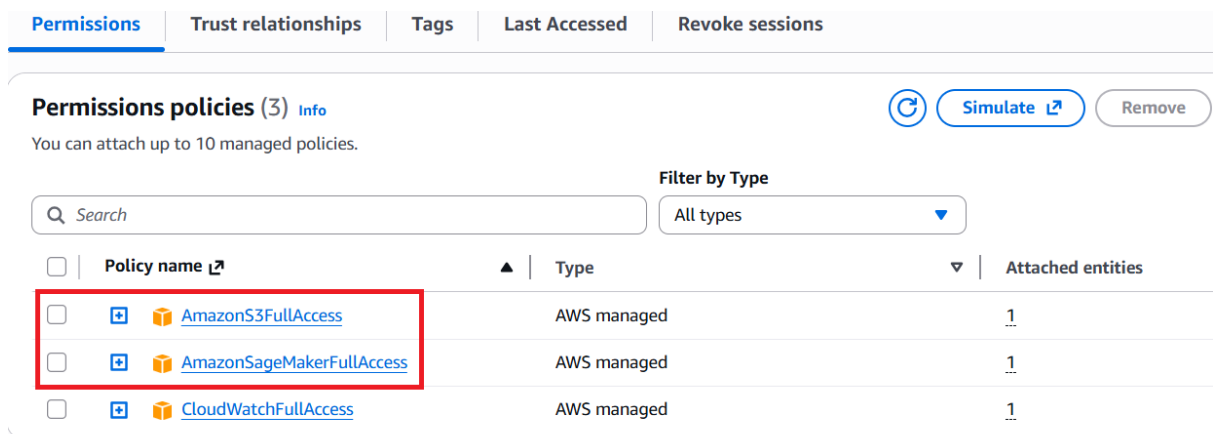


- d. Again click on “Next”.
- e. We have to write the “Role name” then finally click on “Create Role” then we can see the Role which was created. Click on that Role.

- 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”



- 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.



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”.

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

Access reports

- Access Analyzer
- Resource analysis

Role LambdaInvokeLLMRole created.

Last activity

Permissions Trust relationships Tags Last Accessed Revoke

Permissions policies (4) Info

You can attach up to 10 managed policies.

Search

Filter by Type All types

Policy name	Type
<input type="checkbox"/> AmazonDynamoDBFullAccess	AWS managed
<input type="checkbox"/> AmazonS3FullAccess	AWS managed
<input type="checkbox"/> AmazonSageMakerFullAccess	AWS managed
<input type="checkbox"/> CloudWatchLogsFullAccess	AWS managed

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

S3

Services

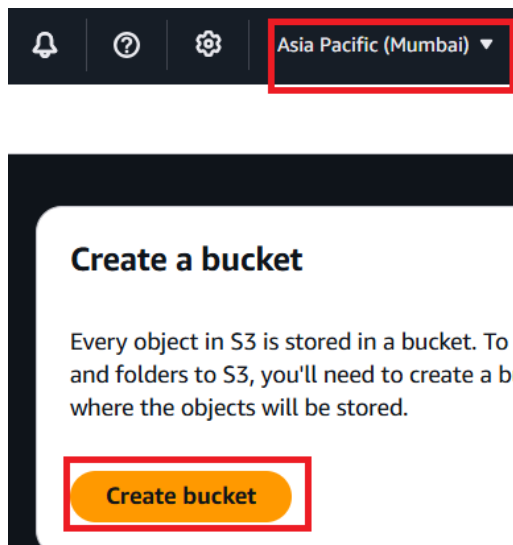
- Services**
- Features
- Documentation
- Knowledge articles
- Marketplace
- Blog posts
- Events
- Tutorials

S3
Scalable Storage in the Cloud

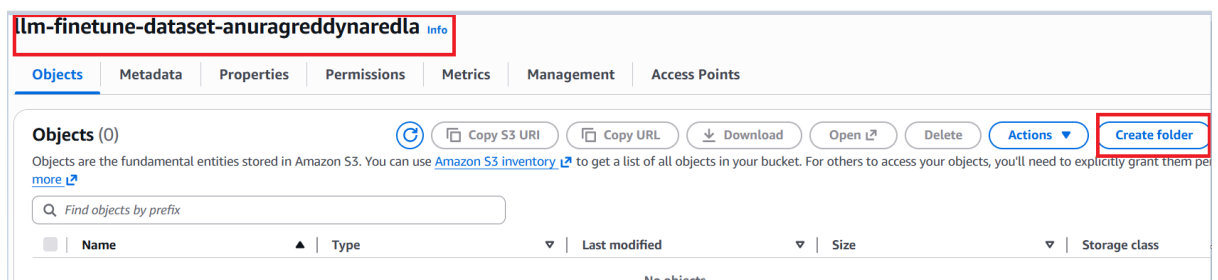
S3 Glacier
Archive Storage in the Cloud

AWS Snow Family
Large Scale Data Transport

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



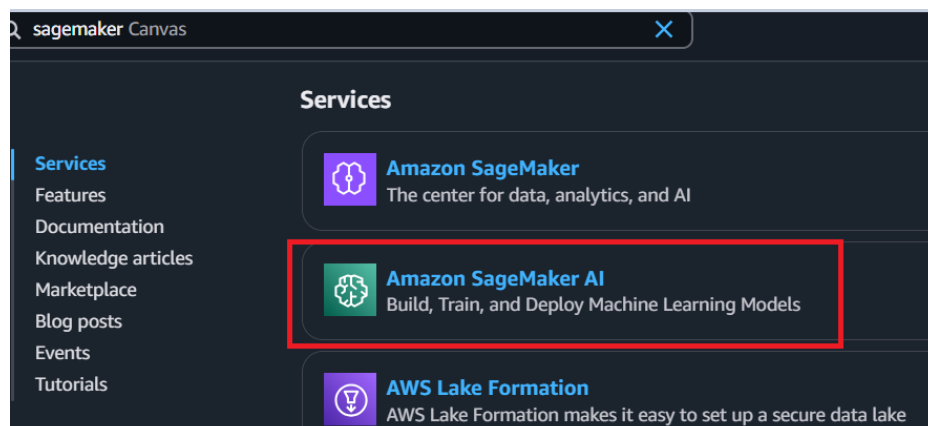
- b. Under Bucket name give name like this : llm-finetune-dataset-<yourname>.
c. Keep everything same then finally click on “Create bucket”.
d. Under this bucket we will create a folder to create select the created bucket name then click on “Create folder”. Then give the name as “dataset”.



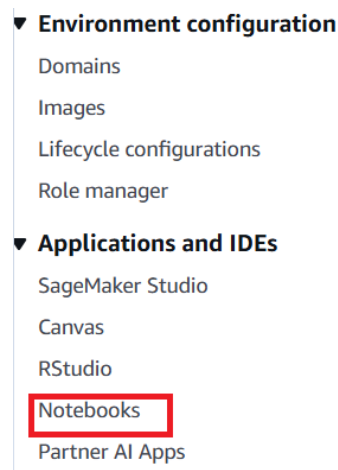
- e. Now we will create one more bucket for storing the Finetuned LLM model.
f. Click on “Create bucket”=> Give the name like this: llm-model-artifacts-<yourname> then click on Create bucket Now we will create one more folder for storing the Models with this name i.e, “models”.

4. Now we will create the “AWS SageMaker Instance”

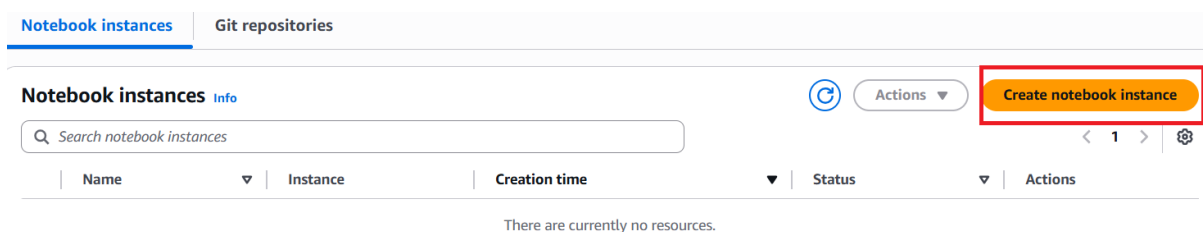
- a. In the sagemaker we will setup the instance.
b. Search the sagemaker and select the “SageMaker AI”.



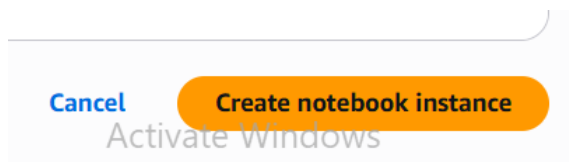
- c. On the left side we can see the “Notebooks”. Click on this notebooks.



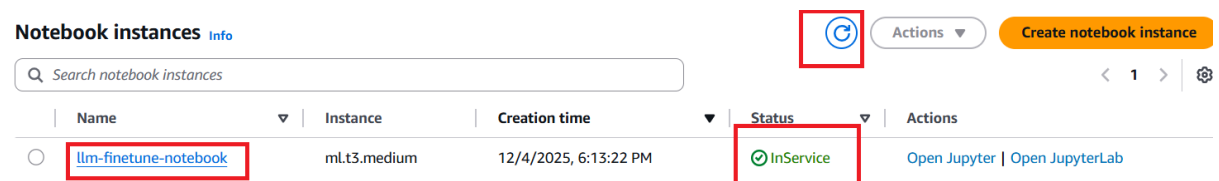
- d. Now click on “Create notebook instance”.



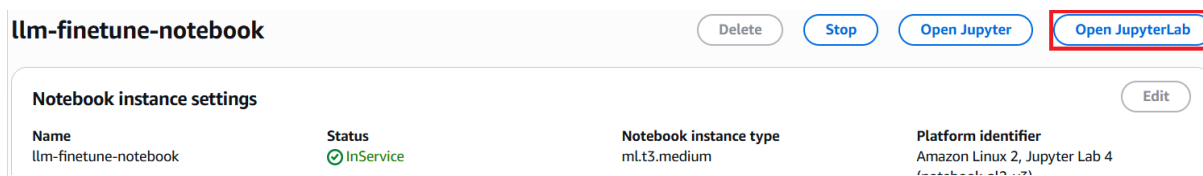
- e. Under the “Notebook instance settings” give any name under “Notebook instance name”. keep remaining as it is then click on “Create notebook instance” which will be displayed down.



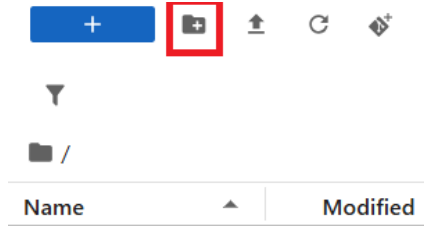
- f. After creating notebook instance give sometime to get the status as “InService” to get this click on refresh button.



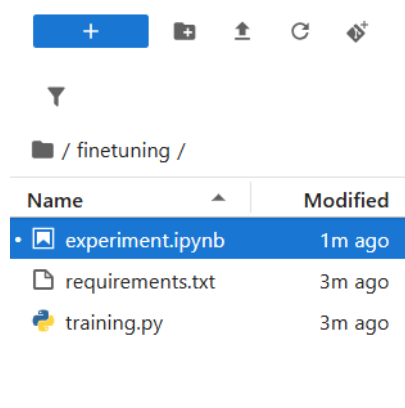
- g. Click on the created notebook name Now we can see the option “open jupyterlab” with this we can get the terminal as well. with the other one we will not get terminal.



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



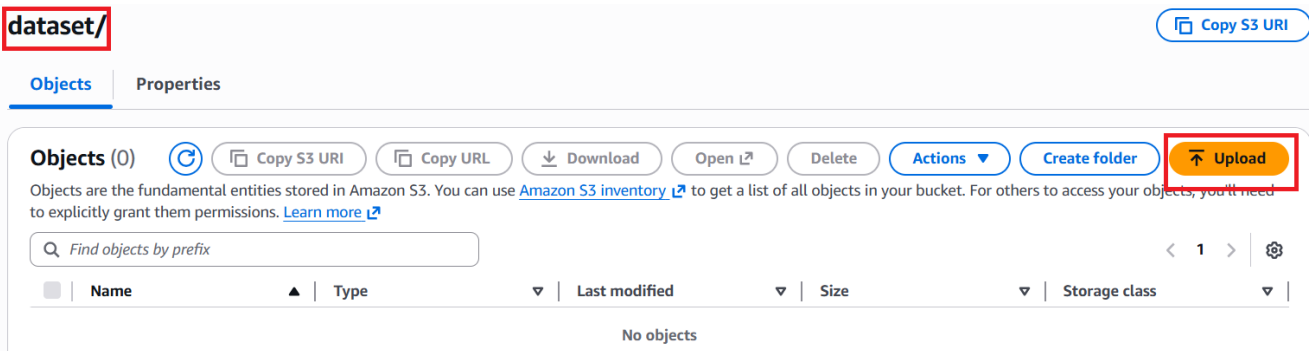
1. Create the folder “finetuning”
2. Open the folder to open double click on that folder
3. Now create file by rightclick “Newfile” give the name as training.py and create one more file i.e, requirements.txt
4. We will create New notebook by clicking on “New Notebook” then select the “conda_python3” 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.

Find by name

<input type="checkbox"/>	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.

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

service quotas

Services

Service Quotas
View and manage your AWS service quotas from a central location

Trusted Advisor
Optimize performance, improve security, reduce costs

Service Catalog

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

Q sagemaker|

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 info

View your applied quota values, default quota values, and request quota increases for quotas. [Learn more](#)

Request increase at account level

Q ml.g5.xlarge

12 matches

	Quota name ▲	Applied account-level quota value ▼	AWS default quota value ▼	Utilization ▼	Adjustability ▼
<input type="radio"/>	ml.g5.xlarge for cluster spot instance usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for cluster usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for endpoint usage	1	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for notebook instance usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for processing job usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for spot training job usage	0	0	0	Account level
<input checked="" type="radio"/>	ml.g5.xlarge for training job usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for training warm pool usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for transform job usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for cluster spot instance usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for cluster usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for endpoint usage	1	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for notebook instance usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for processing job usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for spot training job usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for training job usage	1	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for training warm pool usage	0	0	0	Account level
<input type="radio"/>	ml.g5.xlarge for transform job usage	0	0	0	Account level
<input type="radio"/>	Studio CodeEditor Apps running on ml.g5.xlarge instances	0	0	0	Account level

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

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

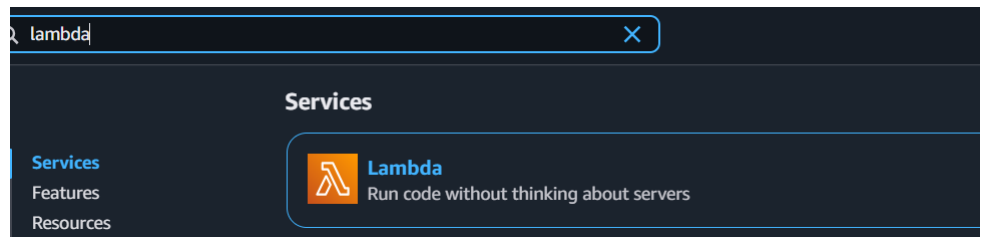
A screenshot of the Amazon SageMaker console. The top navigation bar shows "Amazon SageMaker AI" > "Endpoints" > "live-finetune-endpoint". On the left sidebar, under "Dashboard", there's a link "What's new" with a blue badge containing the number "40". Below that, under "Environment configuration", there are links for "mainframe", "images", "image configurations", and "le manager". The main content area displays the configuration for the "live-finetune-endpoint". It lists the "Name" as "live-finetune-endpoint" and the "ARN" as "[REDACTED]". A red box highlights the "URL" field, which contains "[REDACTED]" followed by a link "Learn more about the API" with an external link icon.

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

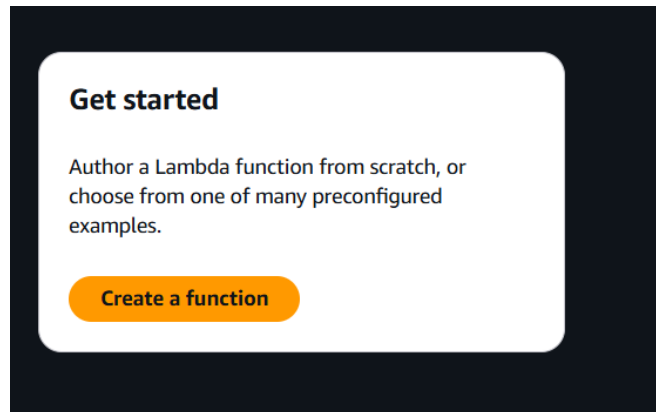
a. Above trained model we will deploy it.

b. Create the Lambda

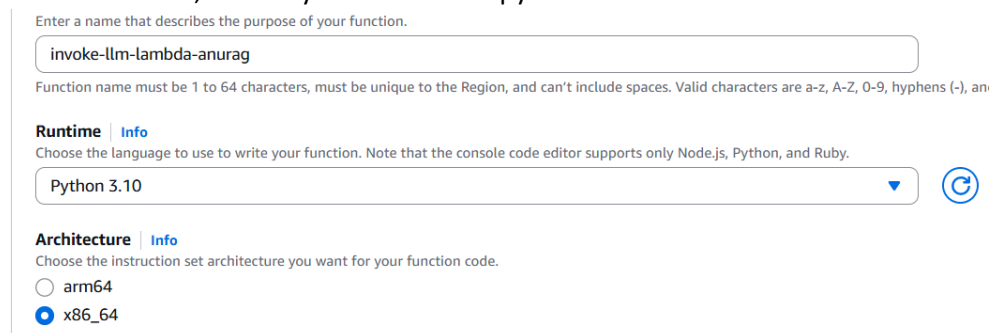
- Search in console then click on lambda



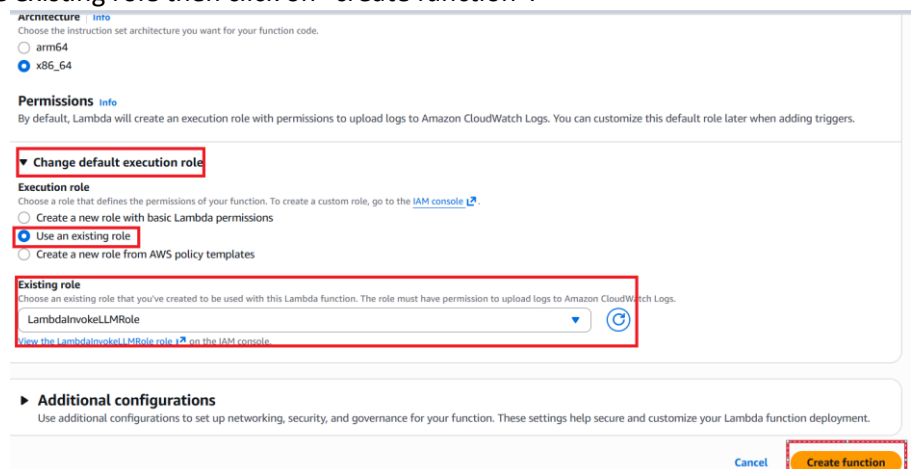
- Now we will create a function. Click on it



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

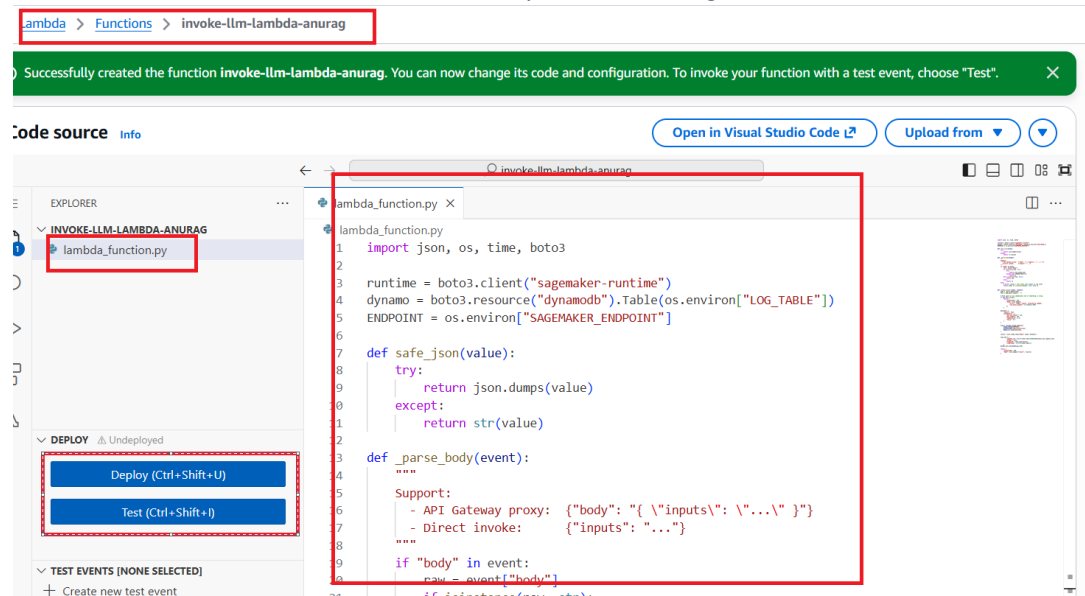


- We have to add the Roles to create click on “Change default execution role”.
Give the existing role then click on “create function”.



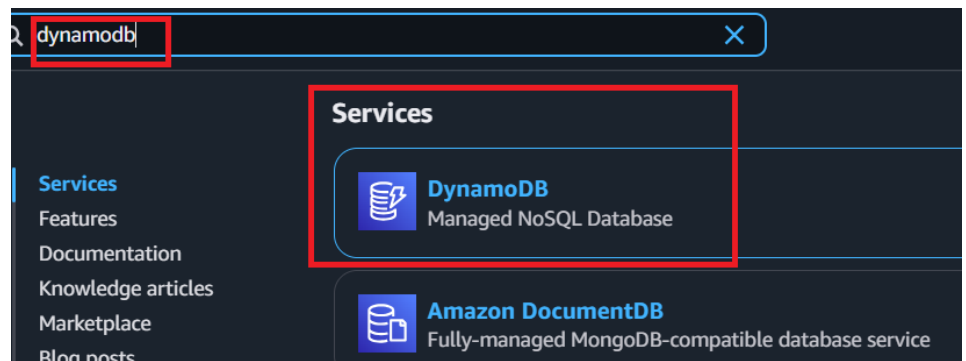
- v. Copy the code from lambda_function.py from my github then paste in the created lambda function. Then perform STEP c which we will create the dynamodb then Finally click on Deploy. For testing click on test give a name then give this code {"body": "\inputs\":"Summarize the text about heart disease\"}"

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

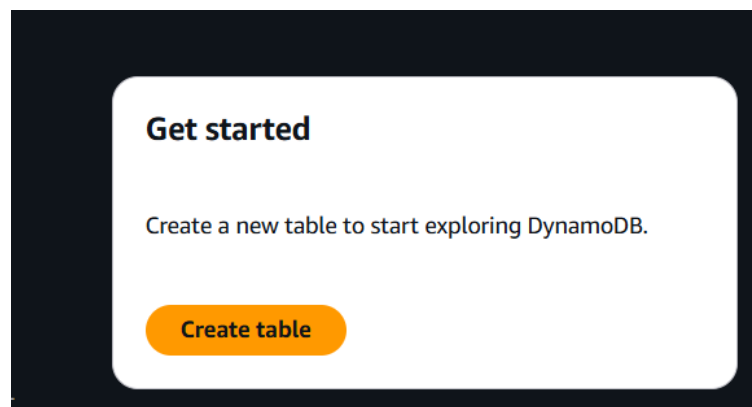


c. Create the DynamoDB for storing the logs

- 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 and 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.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
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.
 String
1 to 255 characters and case sensitive.

Sort key - optional
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.
 String
1 to 255 characters and case sensitive.

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

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

Configuration

General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
RDS databases

Environment variables (0)

Key | Value
No environment variables
No environment variables associated with
[Edit](#)

2. Click on “Edit”.
- Give the “LOG_TABLE” as key and value as the created dynamodb name
 - 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]	Remove
SAGEMAKER_ENDPOINT	[REDACTED]	Remove

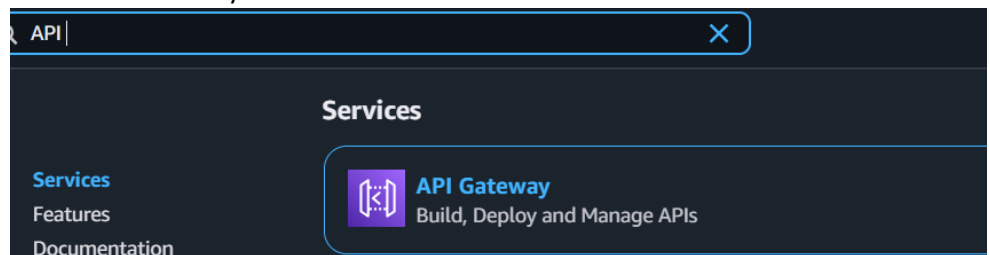
[Add environment variable](#)

► Encryption configuration

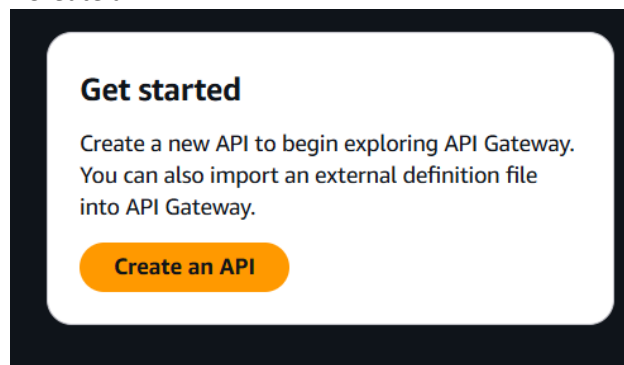
[Cancel](#) [Save](#)

d. Now we will create API Gateway

- i. Search for the API Gateway then click on that



- ii. Click on “Create an API”.



- iii. 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](#)

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

API name
LLMinferenceAPI2

Description - optional

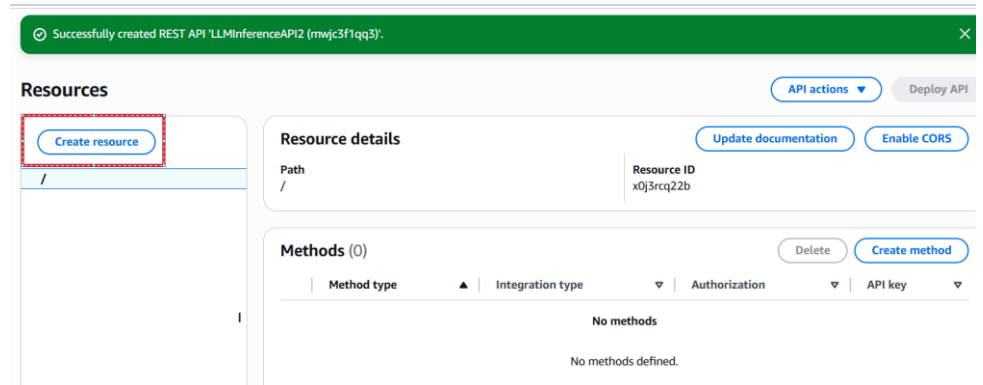
API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.
Regional

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.
Choose a security policy

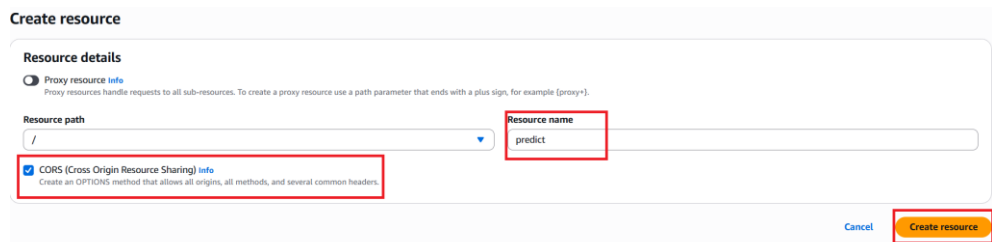
IP address type [Info](#)
Select the type of IP addresses that can invoke the default endpoint for your API.
☒ IPv4
Supports only edge-optimized and Regional API endpoint types.
☐ Dualstack
Supports all API endpoint types.

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

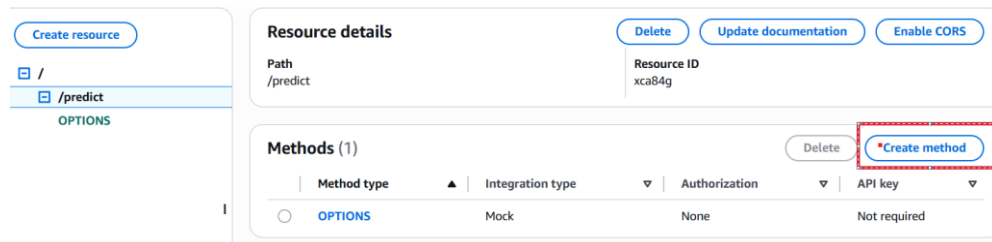
- v. Click on “Create resource”.



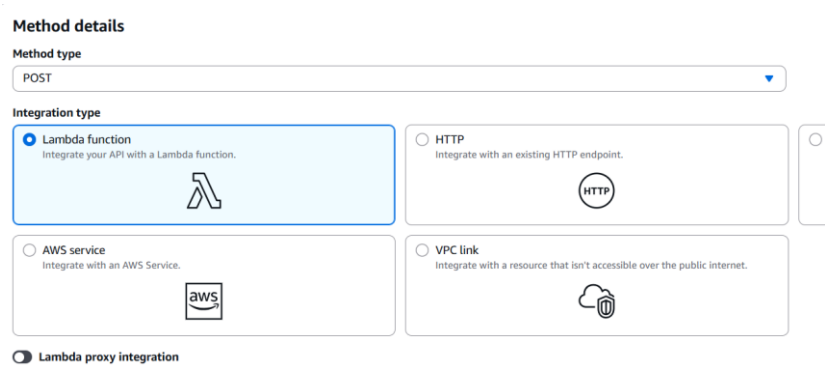
- vi. Give the Resource Name. click on CORS checkbox then click on “Create Resource”.



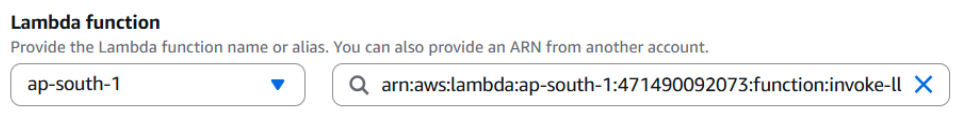
- vii. Now we will create the “Method”. To create click on “Create method”.



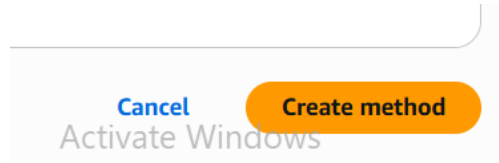
- viii. Under the method type select the “POST” method.



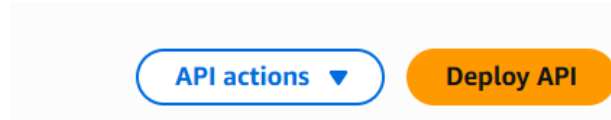
- ix. Under the Lambda function select the “ap-south-1” then give the lambda function.



- 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.

A screenshot of the 'Deploy API' dialog box. It includes a close button (X) in the top right. The text inside says: 'Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. [Learn more](#)'. There is a 'Stage' dropdown menu currently showing '*New stage*'. Below it is a 'Stage name' input field containing 'prod'. A light blue informational box states: 'A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.' At the bottom is a 'Deployment description' text area. At the very bottom are 'Cancel' and 'Deploy' buttons.

- xiii. This is our URL.

A screenshot showing the 'Stages' and 'Method overrides' section. On the left, under 'Stages', there is a tree view with 'prod' selected, followed by a slash '/' and then '/predict'. Below this, 'OPTIONS' and 'POST' are listed. On the right, under 'Method overrides', there is a text block: 'By default, methods inherit stage-level settings. To customize settings for a method, co'. Below this is a light blue box with an information icon and the text: 'This method inherits its settings from the 'prod' stage.' At the bottom, there is an 'Invoke URL' section with a checkbox and a URL that has been redacted with blue scribbles.

- xiv. Now we will create a API key.

1. Goto API keys=>Create API key

A screenshot of the AWS API Gateway console. The left sidebar shows the navigation menu with 'API Gateway' selected, and 'API keys' highlighted in red. The main content area shows a green banner at the top: 'Successfully created deployment for LLMInferenceAPI2. This deployment is active for prod.' Below this, the 'API keys (0)' section is visible. It includes a search bar 'Find API keys', a table with columns 'Name', 'Status', 'ID', 'API key', 'Description', and 'Creation date', and a message 'No API keys' with a 'Create API Key' button. The 'Create API key' button in the top right corner is highlighted in red.

2. Give any name click on "Save".

Create API key

info

API key details

Name

LLMinferenceAPI

Description - optional

API key

☒ Auto generate

☐ Custom

Cancel

Save

3. Copy the key and paste it somewhere.

LLMInferenceAPI

Last updated
December 06, 2025, 21:54 (UTC+05:30)

[Edit](#)

API key details

ID
yjw0yiaccf

Description
-

Creation date
December 06, 2025, 21:54 (UTC+05:30)

Status
 Active

API key
 [Show](#)

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

```
API_URL="https://rtyuill1456.execute-api.ap-south-1.amazonaws.com/prod/predict"
API_KEY="Conjknjnjjnjhjhjhjhjhjhio"
```

- 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.

- Run “streamlit run rag_app_ui.py –server.port 8502”.
- Enter the Question then click on “Generate Answer” to get the answer from Vectordatabase 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:\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:\nwhat Clinical trials have shown that adding Ezetimibe to statin therapy?\n\nAnswer:\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.\nThe answer is:\n'}
```