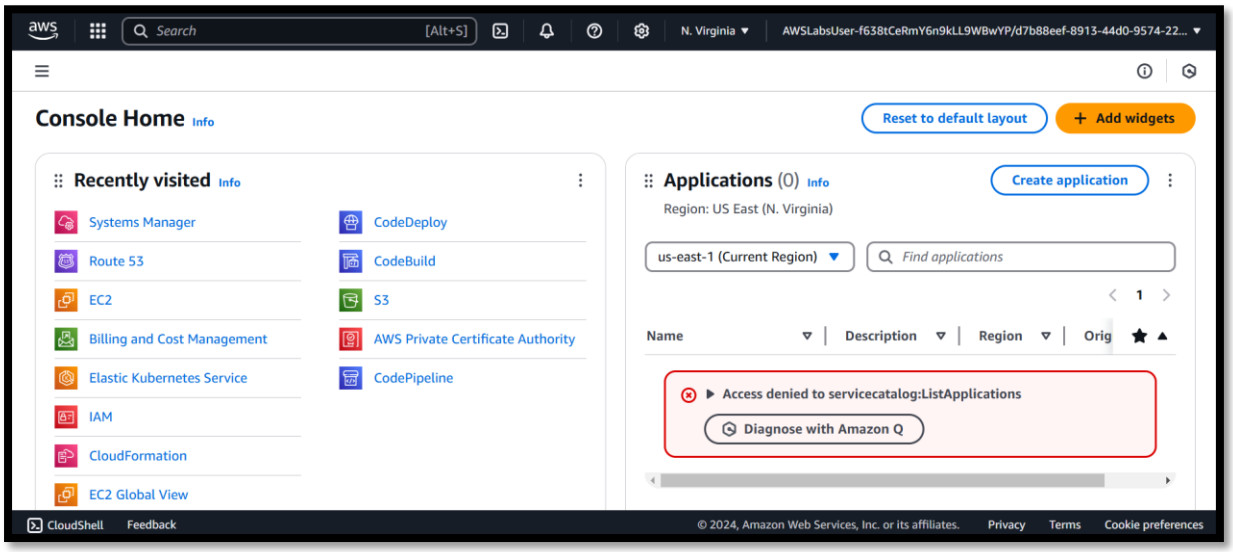
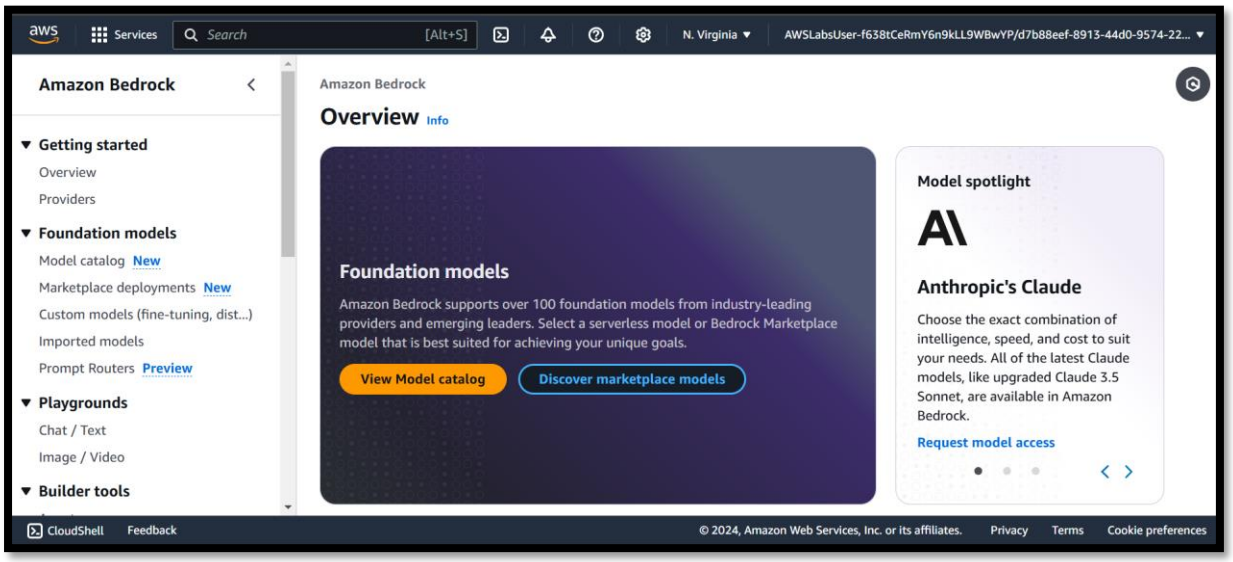
Objective: To create text summarizations by running two notebook files: Task2a.ipynb, which summarizes text with small files using Titan Text Premier, and Task2b.ipynb, which uses chunking to summarize long texts with Amazon Titan.

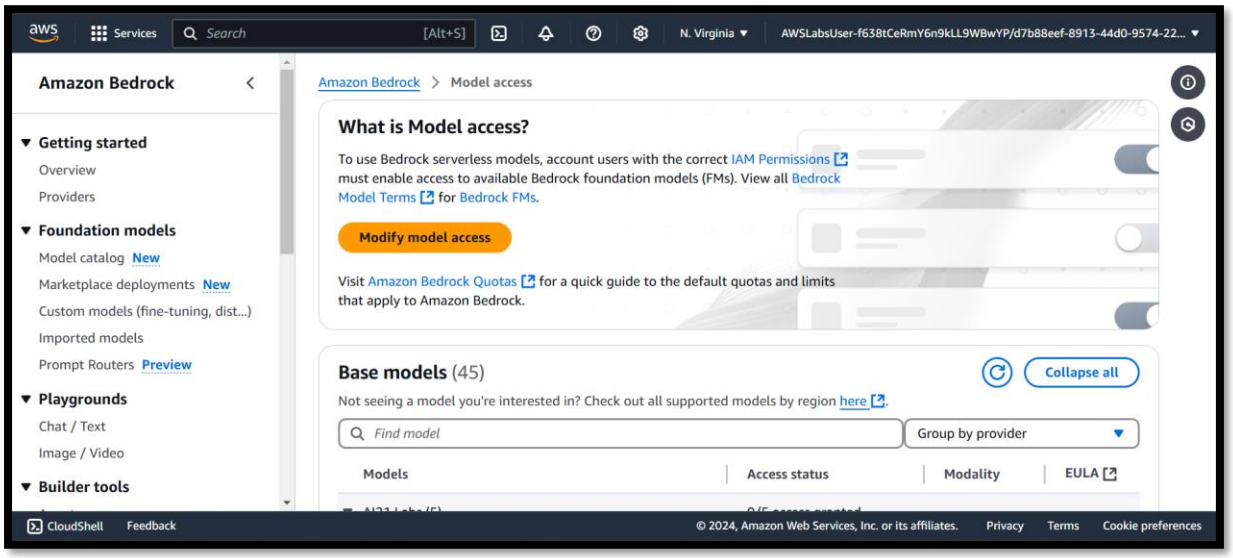
Task 0: Set up the environment

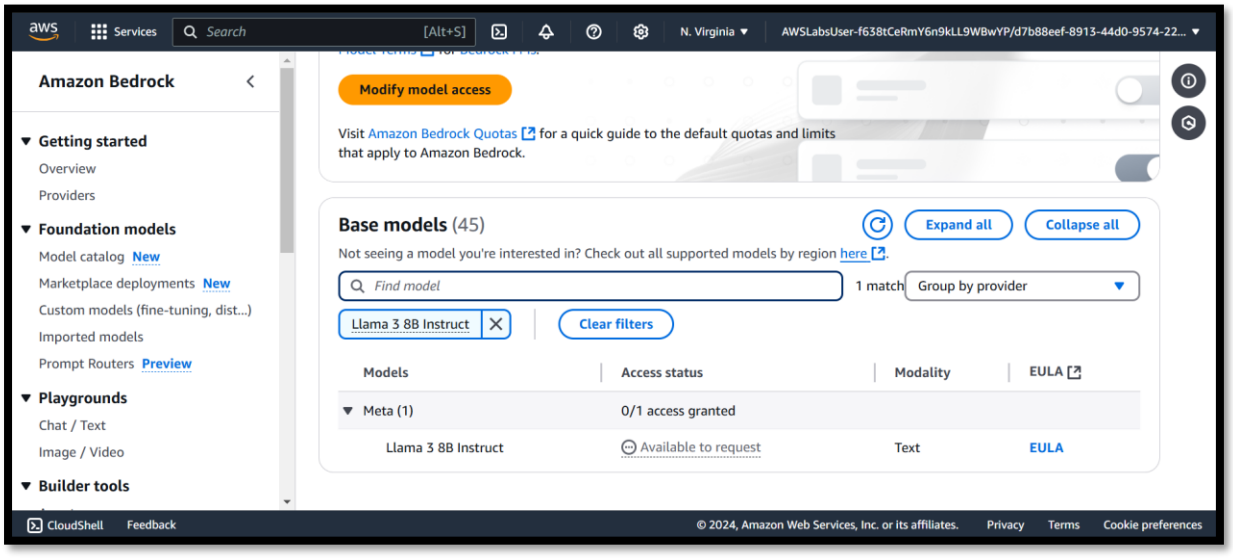
In this task, I registered the base models in the Amazon Bedrock console and launched an Amazon SageMaker Studio application to access my lab resources.



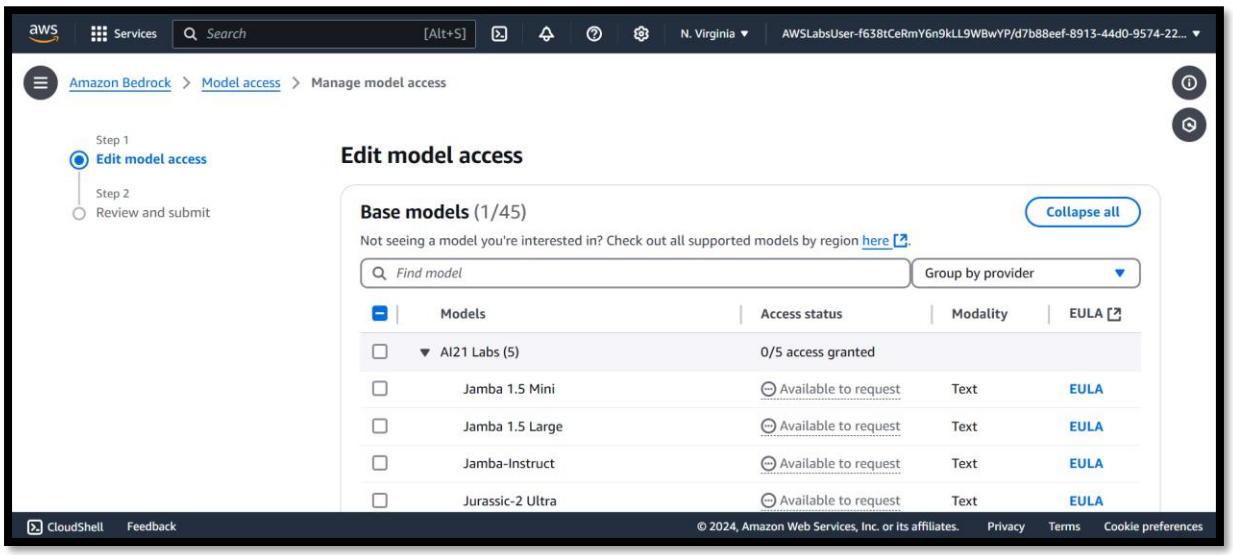


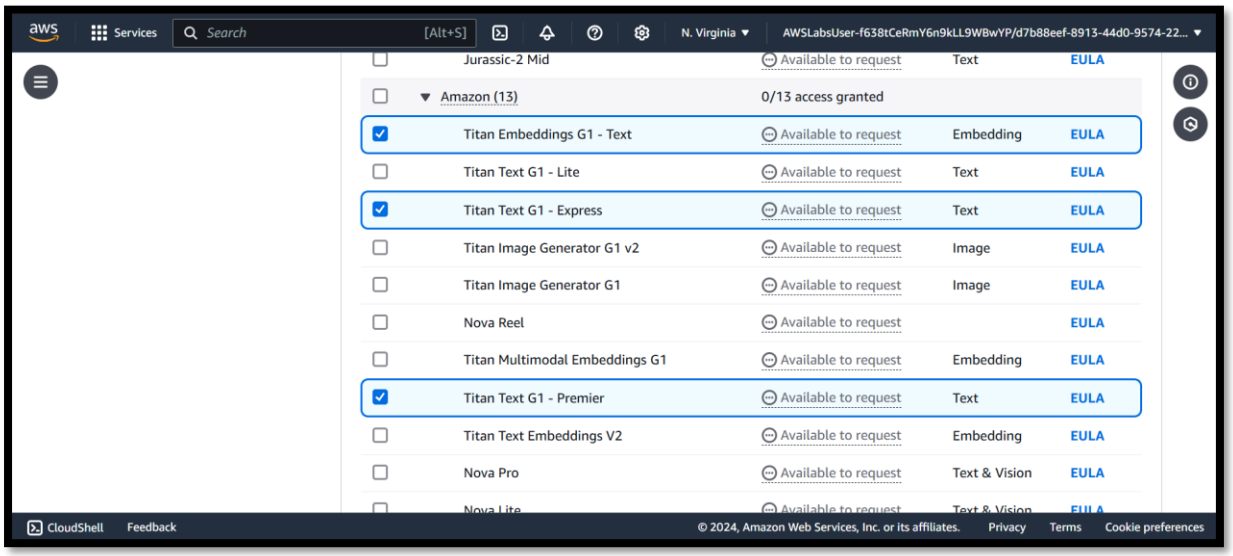
I reviewed the Access Status for each of the models. If the Access Status for one or more of the models was set to Available to request, I expanded this menu and followed the steps to enable access for them.

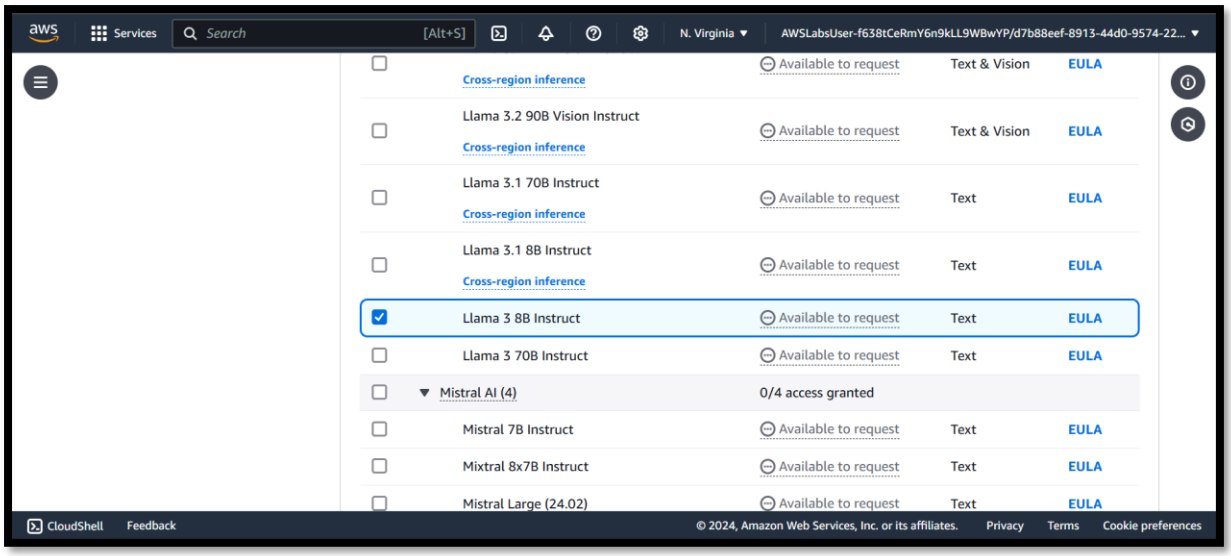


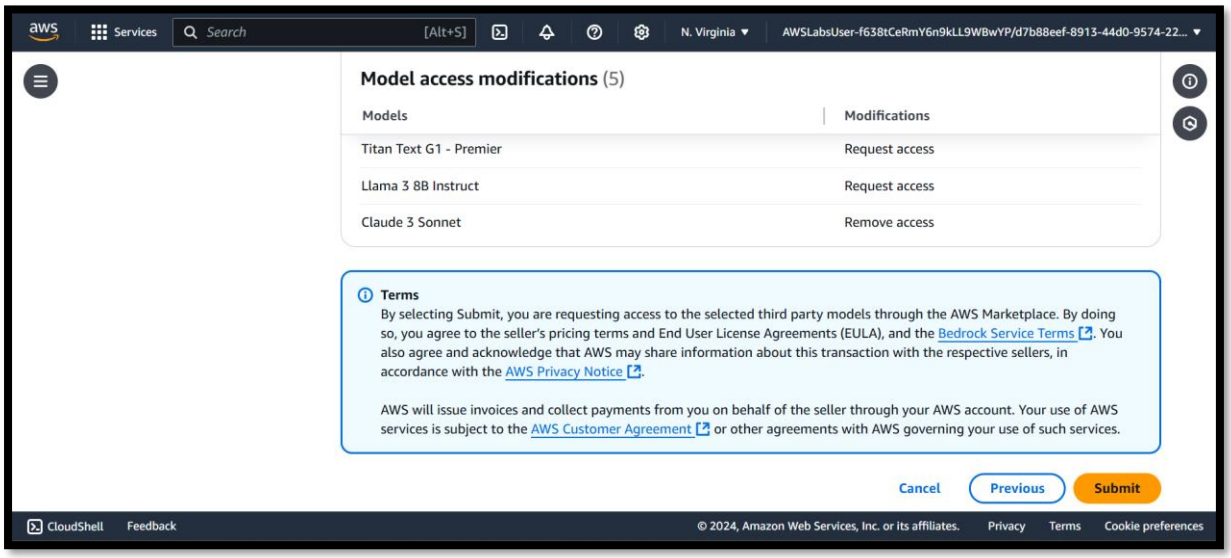


I chose Modify model access at the top of the screen.

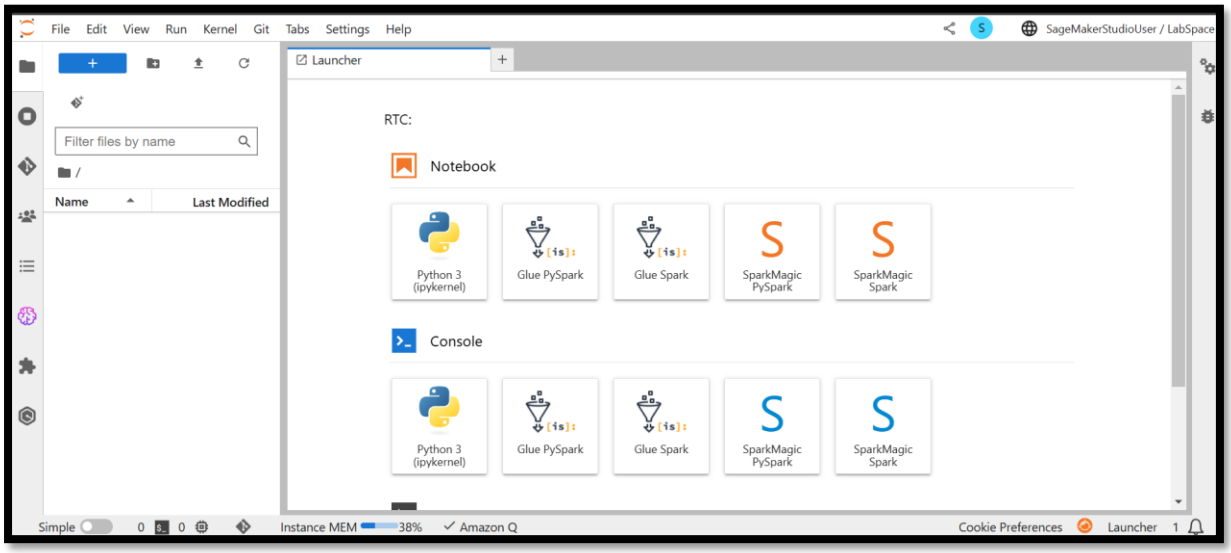


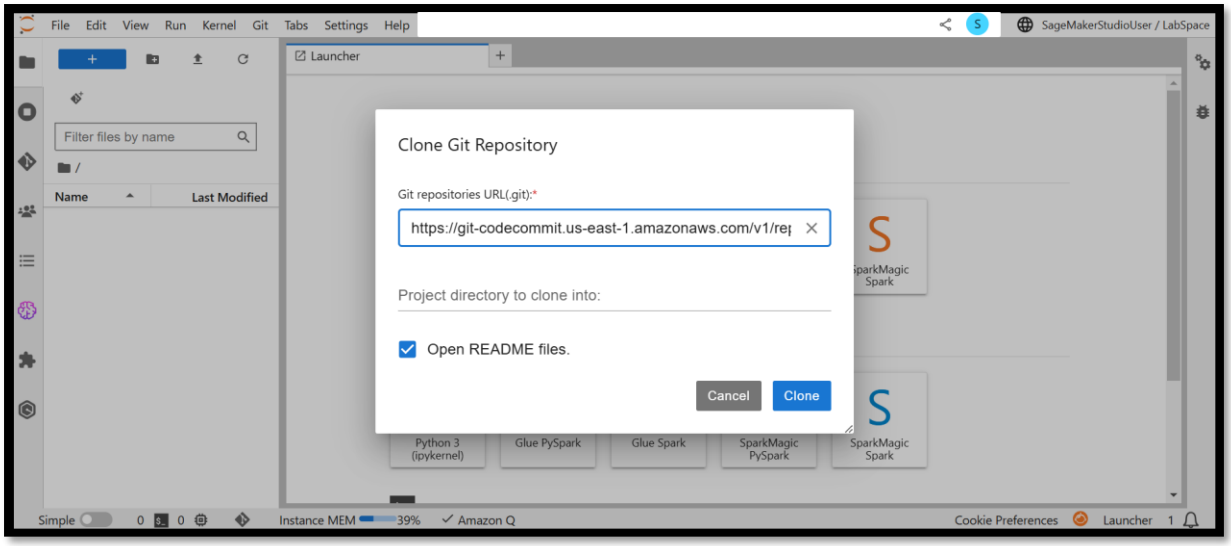



aws
Services
Search
[Alt+S]
N. Virginia
AWSLabsUser-f638tCeRmY6n9kLL9WBwYP/d7b88eef-8913-44d0-9574-22...
Cross-region inference
Available to request
Text & Vision
EULA
Llama 3.2 90B Vision Instruct
Cross-region inference
Available to request
Text & Vision
EULA
Llama 3.1 70B Instruct
Cross-region inference
Available to request
Text
EULA
Llama 3.1 8B Instruct
Cross-region inference
Available to request
Text
EULA
Llama 3 8B Instruct
Available to request
Text
EULA
Llama 3 70B Instruct
Available to request
Text
EULA
Mistral AI (4)
0/4 access granted
Mistral 7B Instruct
Available to request
Text
EULA
Mixtral 8x7B Instruct
Available to request
Text
EULA
Mistral Large (24.02)
Available to request
Text
EULA
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
N. Virginia
AWSLabsUser-f638tCeRmY6n9kLL9WBwYP/d7b88eef-8913-44d0-9574-22...
Model access modifications (5)
Models
Modifications
Titan Text G1 - Premier
Request access
Llama 3 8B Instruct
Request access
Claude 3 Sonnet
Remove access
Terms
By selecting Submit, you are requesting access to the selected third party models through the AWS Marketplace. By doing so, you agree to the seller's pricing terms and End User License Agreements (EULA), and the Bedrock Service Terms. You also agree and acknowledge that AWS may share information about this transaction with the respective sellers, in accordance with the AWS Privacy Notice.
AWS will issue invoices and collect payments from you on behalf of the seller through your AWS account. Your use of AWS services is subject to the AWS Customer Agreement or other agreements with AWS governing your use of such services.
Cancel
Previous
Submit
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

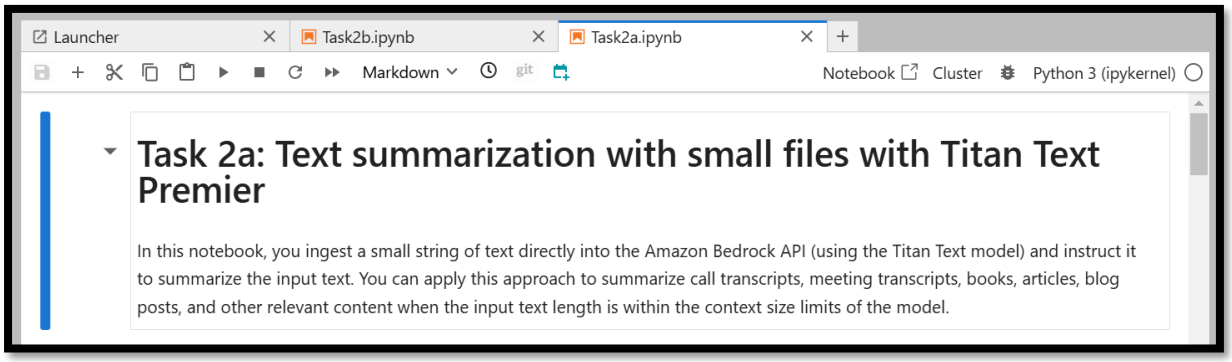
Launch an Amazon SageMaker Studio application

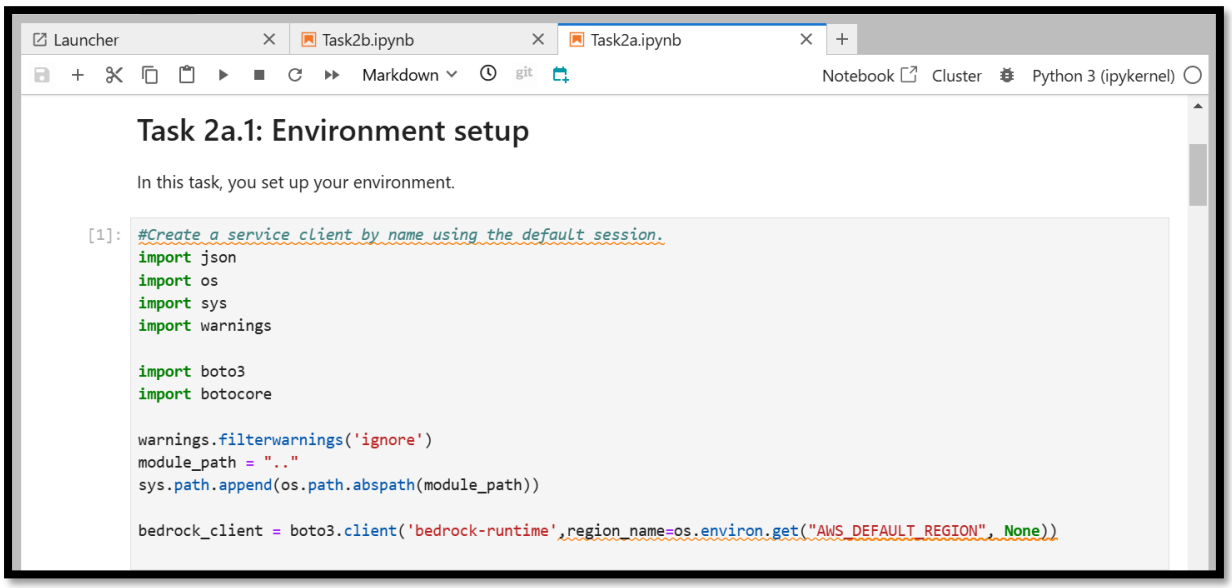


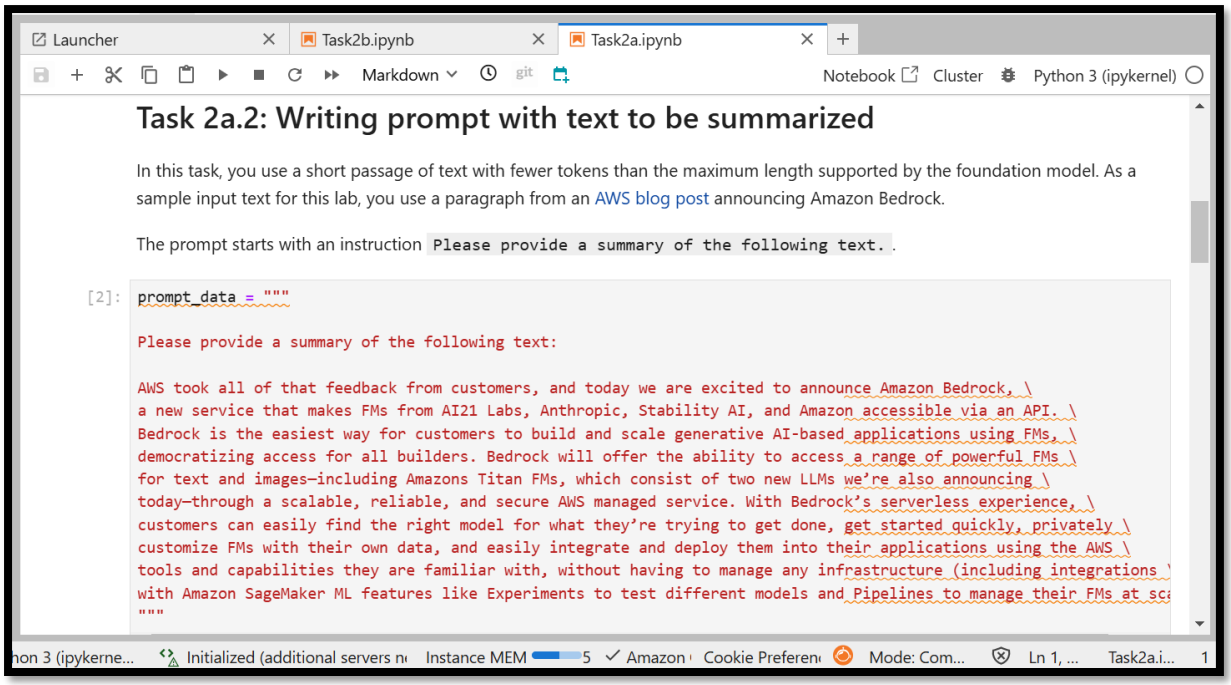


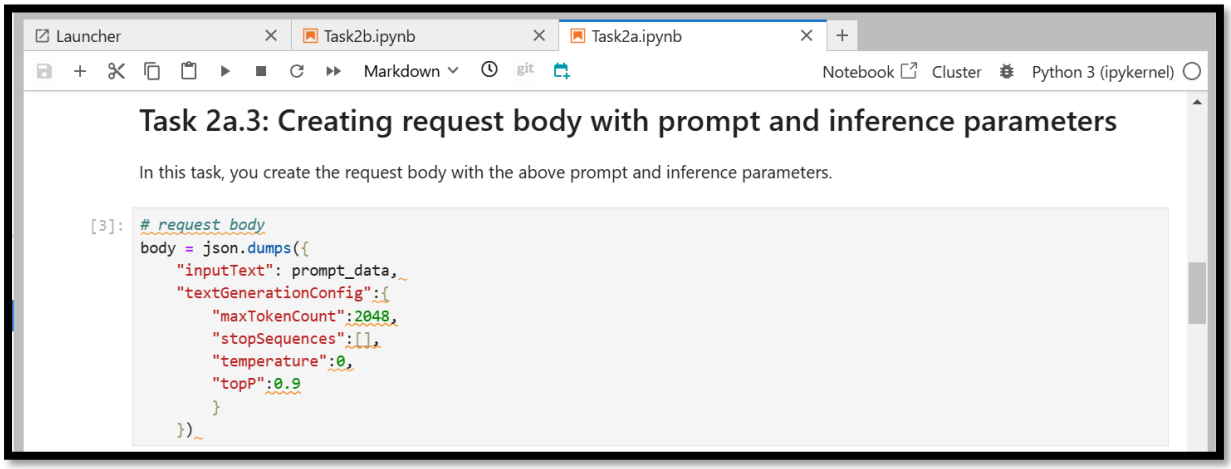
Task 1: Create Text Summarization

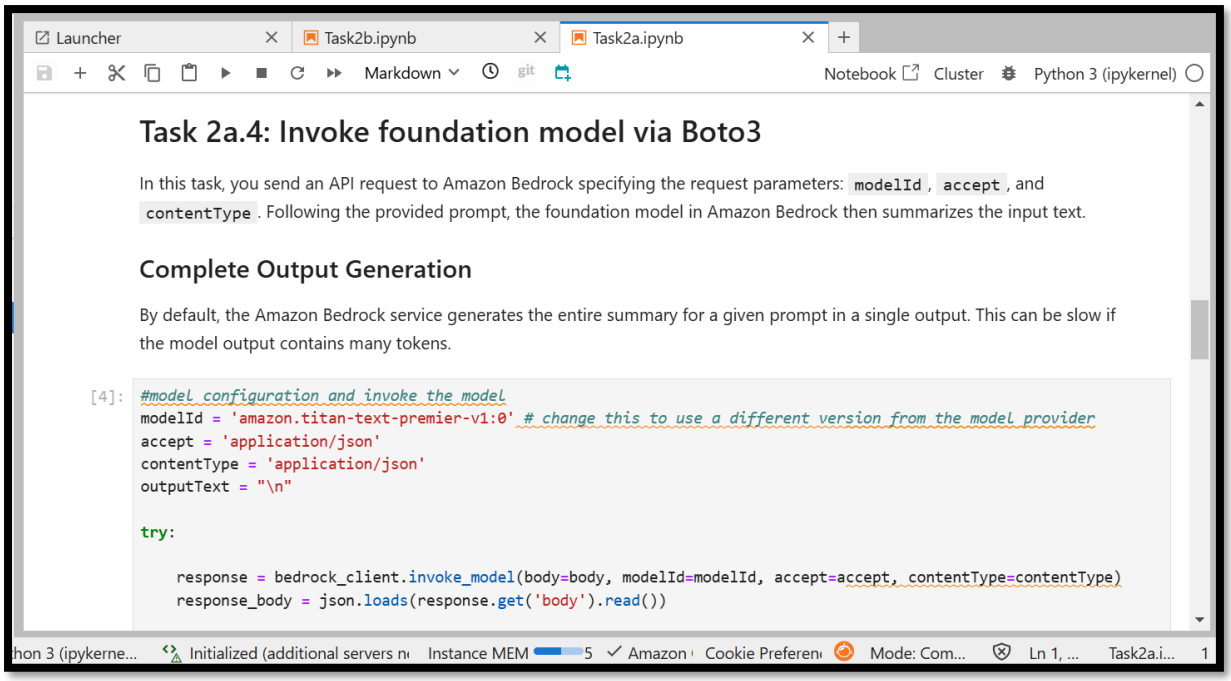
Task2a.ipynb, which summarized the text with small files using Titan Text Premier

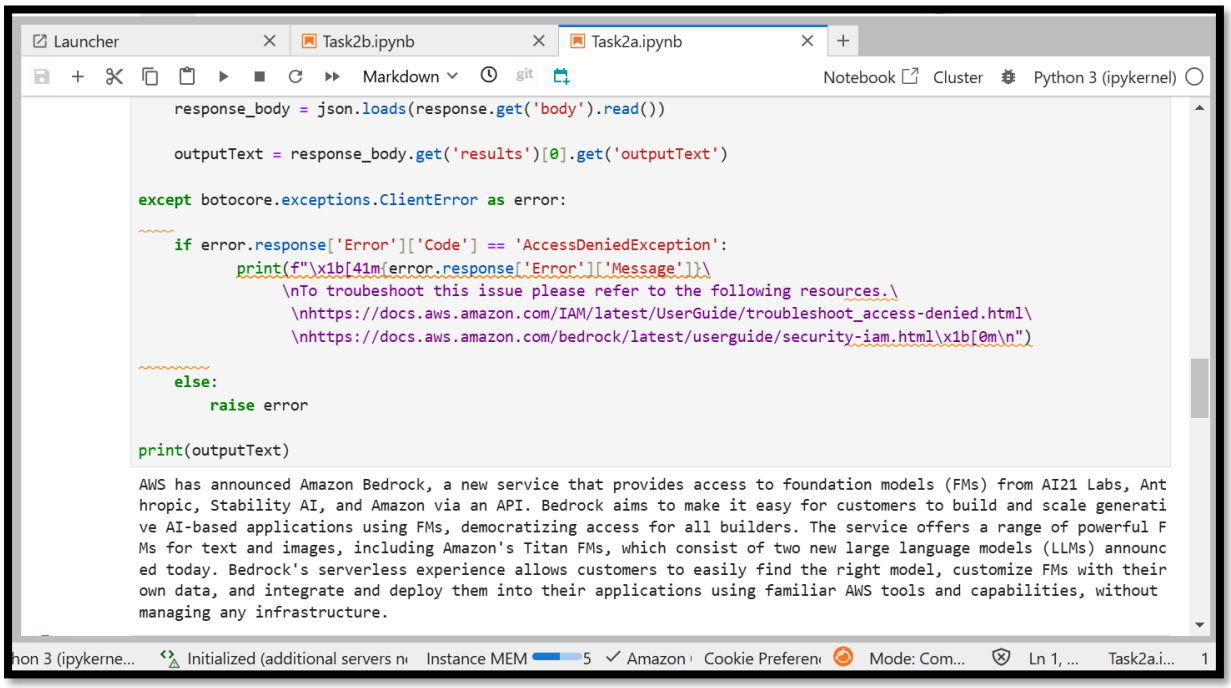


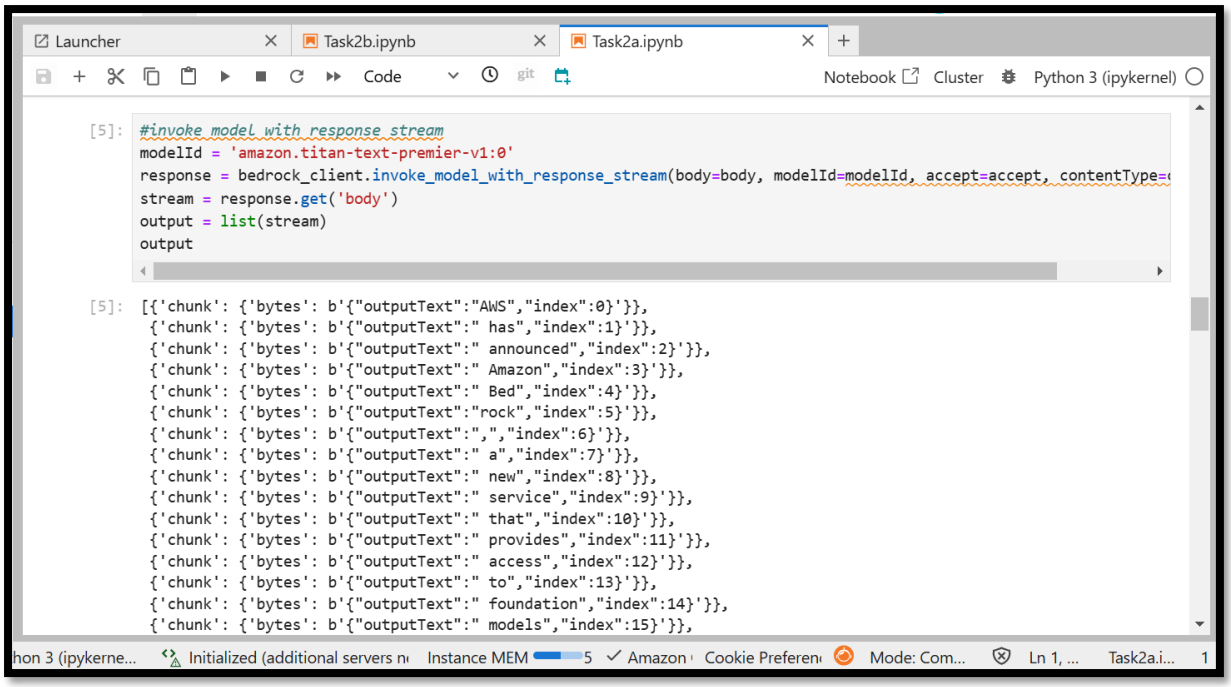


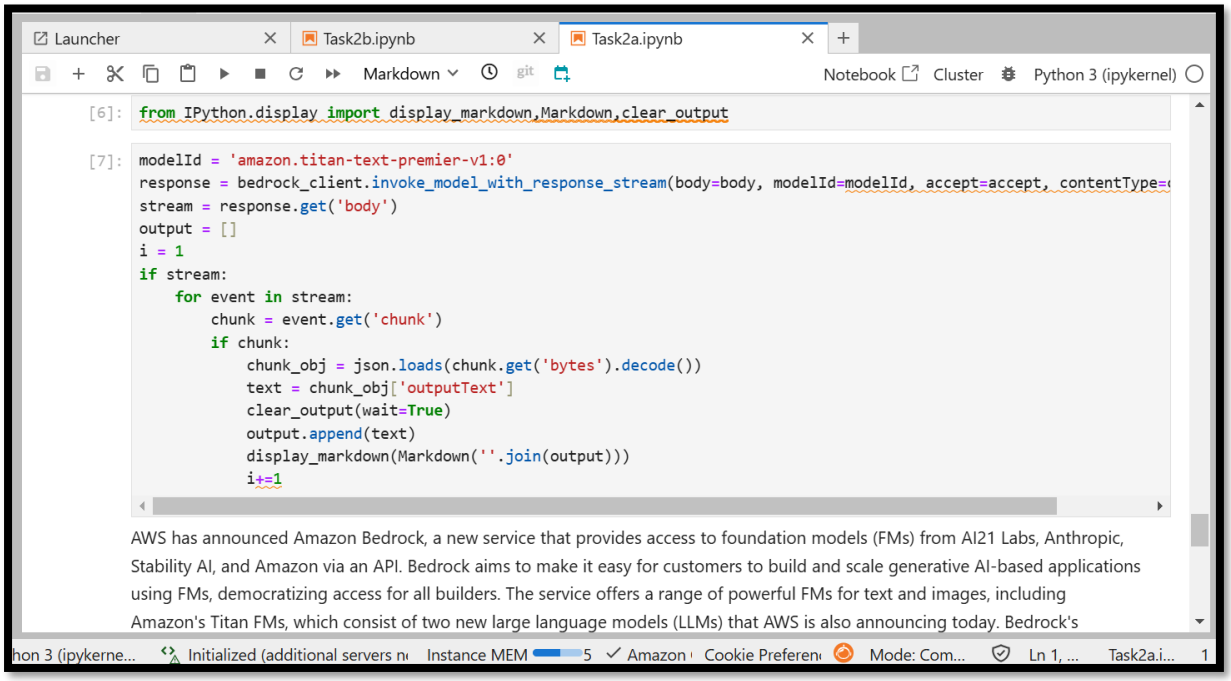

Launcher
Task2b.ipynb
Task2a.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
Task 2a.2: Writing prompt with text to be summarized
In this task, you use a short passage of text with fewer tokens than the maximum length supported by the foundation model. As a sample input text for this lab, you use a paragraph from an AWS blog post announcing Amazon Bedrock.
The prompt starts with an instruction Please provide a summary of the following text. .
[2]:
prompt_data = """
Please provide a summary of the following text:
AWS took all of that feedback from customers, and today we are excited to announce Amazon Bedrock, \
a new service that makes FMs from AI21 Labs, Anthropic, Stability AI, and Amazon accessible via an API. \
Bedrock is the easiest way for customers to build and scale generative AI-based applications using FMs, \
democratizing access for all builders. Bedrock will offer the ability to access a range of powerful FMs \
for text and images—including Amazons Titan FMs, which consist of two new LLMs we're also announcing \
today—through a scalable, reliable, and secure AWS managed service. With Bedrock's serverless experience, \
customers can easily find the right model for what they're trying to get done, get started quickly, privately \
customize FMs with their own data, and easily integrate and deploy them into their applications using the AWS \
tools and capabilities they are familiar with, without having to manage any infrastructure (including integrations \
with Amazon SageMaker ML features like Experiments to test different models and Pipelines to manage their FMs at sca
"""
Initialized (additional servers n
Instance MEM
Amazon
Cookie Preferen
Mode: Com...
Ln 1, ...
Task2a.i...


Launcher
Task2b.ipynb
Task2a.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
Task 2a.3: Creating request body with prompt and inference parameters
In this task, you create the request body with the above prompt and inference parameters.
[3]:
# request body
body = json.dumps({
"inputText": prompt_data,
"textGenerationConfig":{
"maxTokenCount":2048,
"stopSequences":[],
"temperature":0,
"topP":0.9
}
})

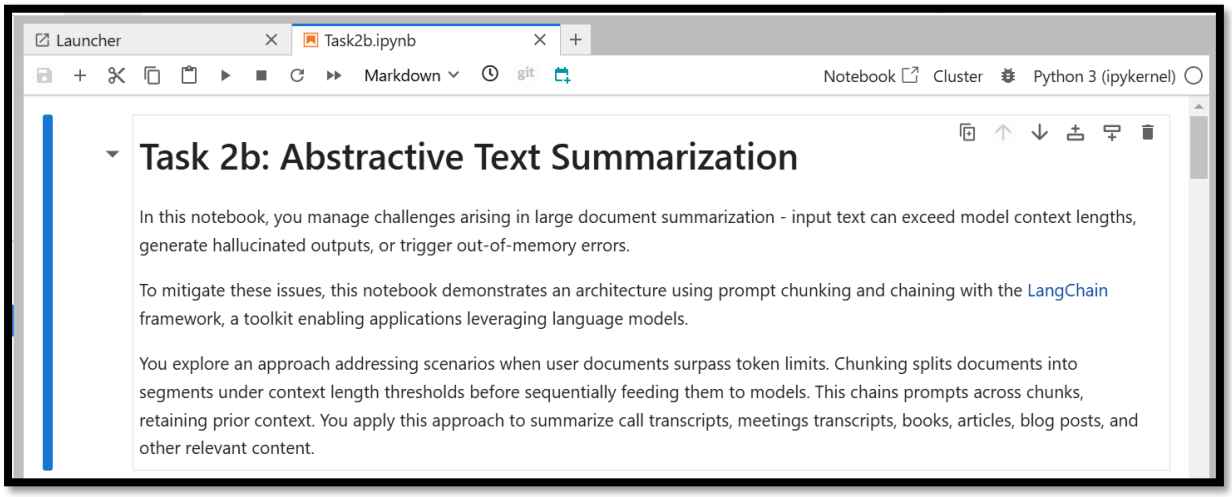

Launcher
Task2b.ipynb
Task2a.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
Task 2a.4: Invoke foundation model via Boto3
In this task, you send an API request to Amazon Bedrock specifying the request parameters: modelId , accept , and contentType . Following the provided prompt, the foundation model in Amazon Bedrock then summarizes the input text.
Complete Output Generation
By default, the Amazon Bedrock service generates the entire summary for a given prompt in a single output. This can be slow if the model output contains many tokens.
[4]:
#model configuration and invoke the model
modelId = 'amazon.titan-text-premier-v1:0' # change this to use a different version from the model provider
accept = 'application/json'
contentType = 'application/json'
outputText = "\n"
try:
response = bedrock_client.invoke_model(body=body, modelId=modelId, accept=accept, contentType=contentType)
response_body = json.loads(response.get('body').read())
Initialized (additional servers n
Instance MEM
Amazon
Cookie Preferen
Mode: Com...
Ln 1, ...
Task2a.i...

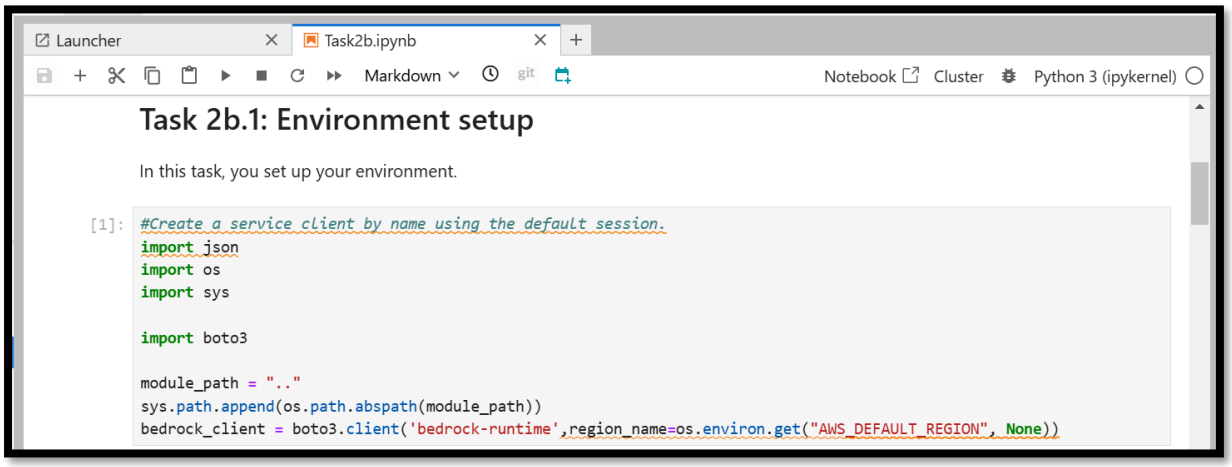

Launcher
Task2b.ipynb
Task2a.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
response_body = json.loads(response.get('body').read())
outputText = response_body.get('results')[0].get('outputText')
except botocore.exceptions.ClientError as error:
if error.response['Error']['Code'] == 'AccessDeniedException':
print(f"\x1b[41m{error.response['Error']['Message']}\
\nTo troubleshoot this issue please refer to the following resources.\
\nhttps://docs.aws.amazon.com/IAM/latest/UserGuide/troubleshoot_access-denied.html\
\nhttps://docs.aws.amazon.com/bedrock/latest/userguide/security-iam.html\x1b[0m\n")
else:
raise error
print(outputText)
AWS has announced Amazon Bedrock, a new service that provides access to foundation models (FMs) from AI21 Labs, Anthropic, Stability AI, and Amazon via an API. Bedrock aims to make it easy for customers to build and scale generative AI-based applications using FMs, democratizing access for all builders. The service offers a range of powerful FMs for text and images, including Amazon's Titan FMs, which consist of two new large language models (LLMs) announced today. Bedrock's serverless experience allows customers to easily find the right model, customize FMs with their own data, and integrate and deploy them into their applications using familiar AWS tools and capabilities, without managing any infrastructure.
Initialized (additional servers n
Instance MEM
Amazon
Cookie Preferen
Mode: Com...
Ln 1, ...
Task2a.i...

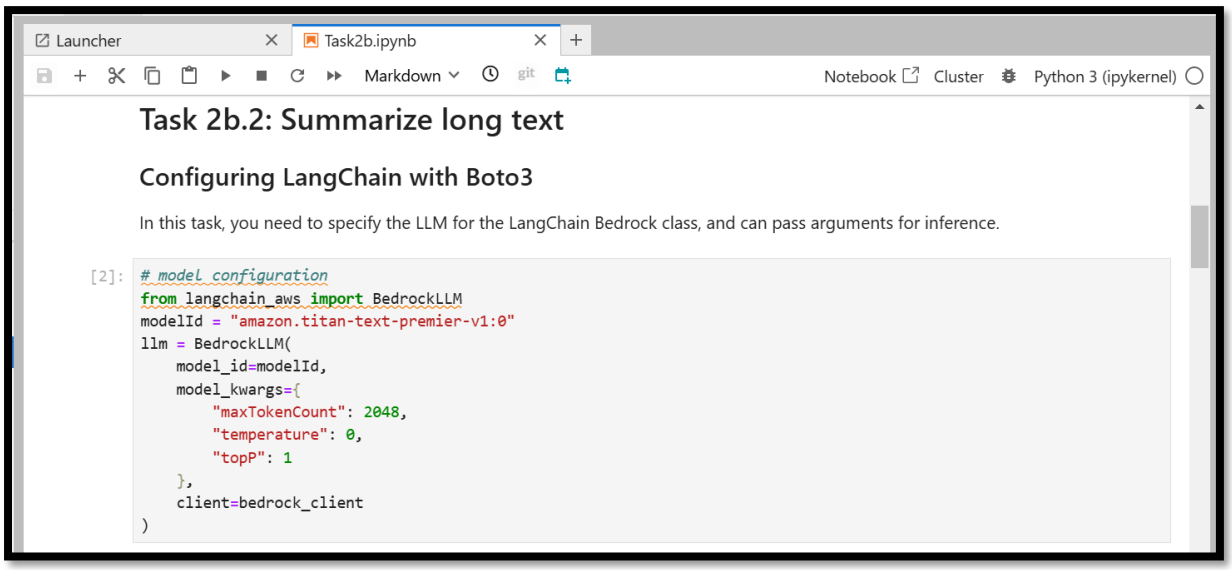

Launcher
Task2b.ipynb
Task2a.ipynb
Code
Notebook
Cluster
Python 3 (ipykernel)
[5]:
#invoke model with response stream
modelId = 'amazon.titan-text-premier-v1:0'
response = bedrock_client.invoke_model_with_response_stream(body=body, modelId=modelId, accept=accept, contentType=
stream = response.get('body')
output = list(stream)
output
[5]:
[{'chunk': {'bytes': b'{"outputText":"AWS","index":0}'}},
{'chunk': {'bytes': b'{"outputText":" has","index":1}'}},
{'chunk': {'bytes': b'{"outputText":" announced","index":2}'}},
{'chunk': {'bytes': b'{"outputText":" Amazon","index":3}'}},
{'chunk': {'bytes': b'{"outputText":" Bed","index":4}'}},
{'chunk': {'bytes': b'{"outputText":"rock","index":5}'}},
{'chunk': {'bytes': b'{"outputText":",","index":6}'}},
{'chunk': {'bytes': b'{"outputText":" a","index":7}'}},
{'chunk': {'bytes': b'{"outputText":" new","index":8}'}},
{'chunk': {'bytes': b'{"outputText":" service","index":9}'}},
{'chunk': {'bytes': b'{"outputText":" that","index":10}'}},
{'chunk': {'bytes': b'{"outputText":" provides","index":11}'}},
{'chunk': {'bytes': b'{"outputText":" access","index":12}'}},
{'chunk': {'bytes': b'{"outputText":" to","index":13}'}},
{'chunk': {'bytes': b'{"outputText":" foundation","index":14}'}},
{'chunk': {'bytes': b'{"outputText":" models","index":15}'}},
hon 3 (ipykerne...
Initialized (additional servers n
Instance MEM
5
Amazon
Cookie Preferen
Mode: Com...
Ln 1, ...
Task2a.i...
1

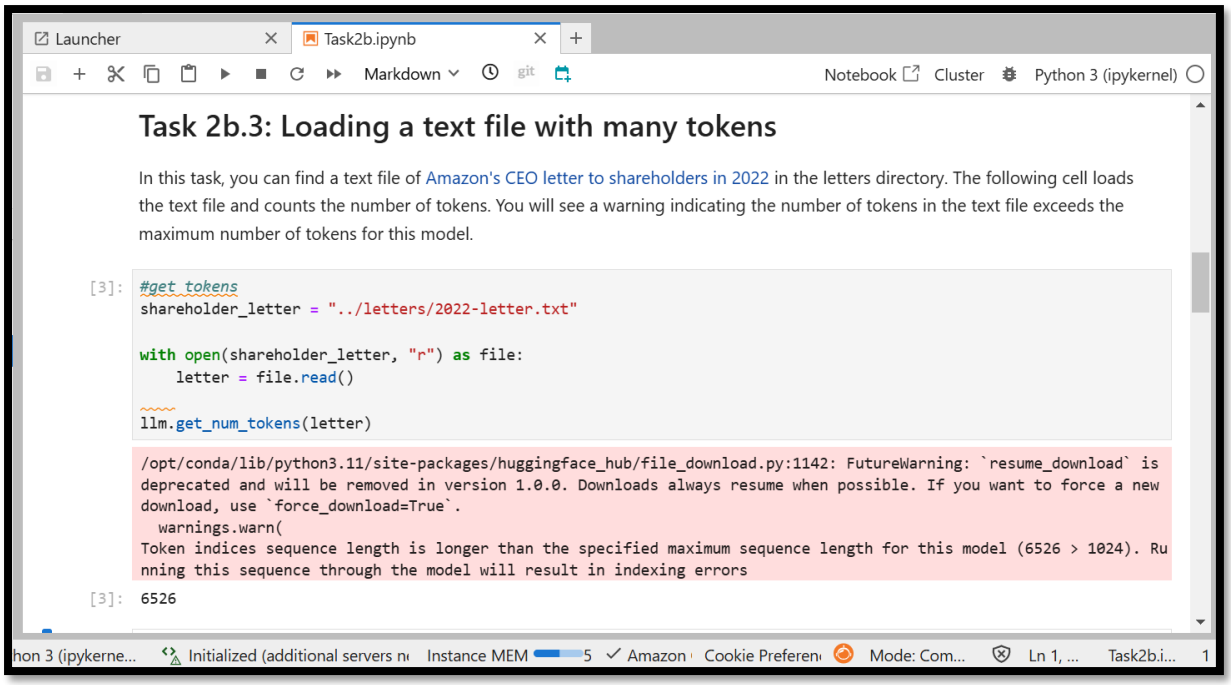

Launcher
Task2b.ipynb
Task2a.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
[6]:
from IPython.display import display_markdown,Markdown,clear_output
[7]:
modelId = 'amazon.titan-text-premier-v1:0'
response = bedrock_client.invoke_model_with_response_stream(body=body, modelId=modelId, accept=accept, contentType=
stream = response.get('body')
output = []
i = 1
if stream:
for event in stream:
chunk = event.get('chunk')
if chunk:
chunk_obj = json.loads(chunk.get('bytes').decode())
text = chunk_obj['outputText']
clear_output(wait=True)
output.append(text)
display_markdown(Markdown(''.join(output)))
i+=1
AWS has announced Amazon Bedrock, a new service that provides access to foundation models (FMs) from AI21 Labs, Anthropic, Stability AI, and Amazon via an API. Bedrock aims to make it easy for customers to build and scale generative AI-based applications using FMs, democratizing access for all builders. The service offers a range of powerful FMs for text and images, including Amazon's Titan FMs, which consist of two new large language models (LLMs) that AWS is also announcing today. Bedrock's
hon 3 (ipykerne...
Initialized (additional servers n
Instance MEM
5
Amazon
Cookie Preferen
Mode: Com...
Ln 1, ...
Task2a.i...
1

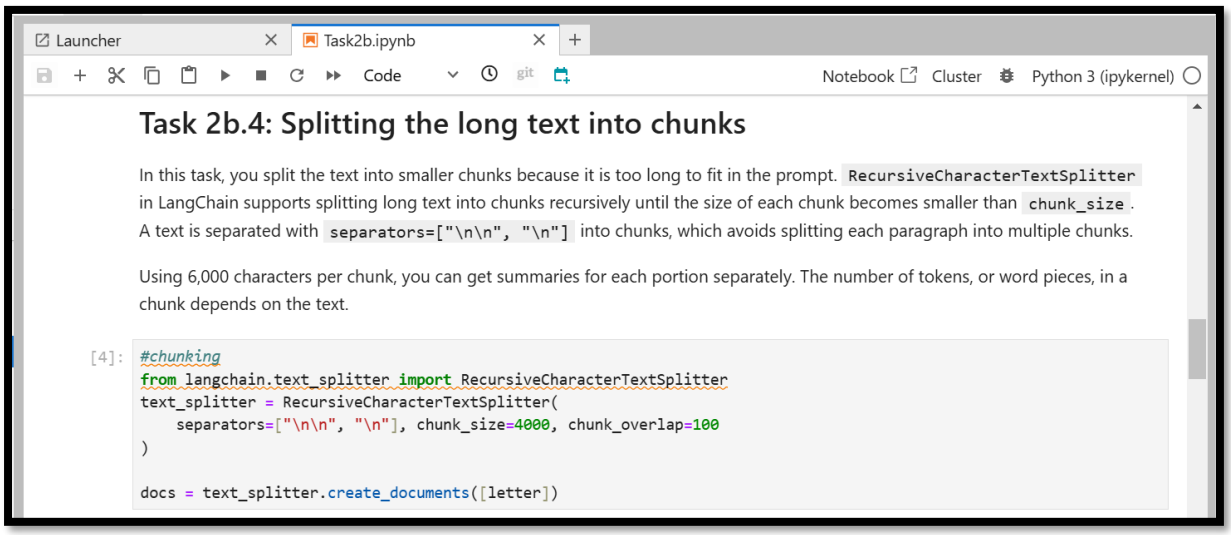
Task2b.ipynb, which used chunking to summarize long texts with Amazon Titan.

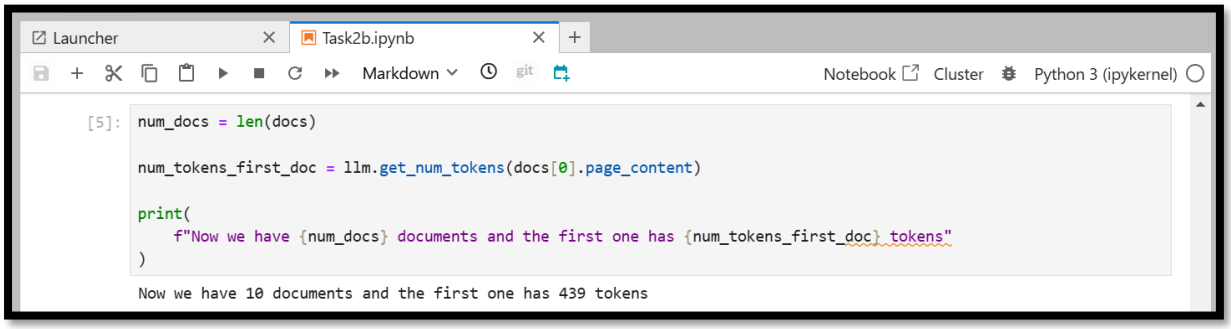


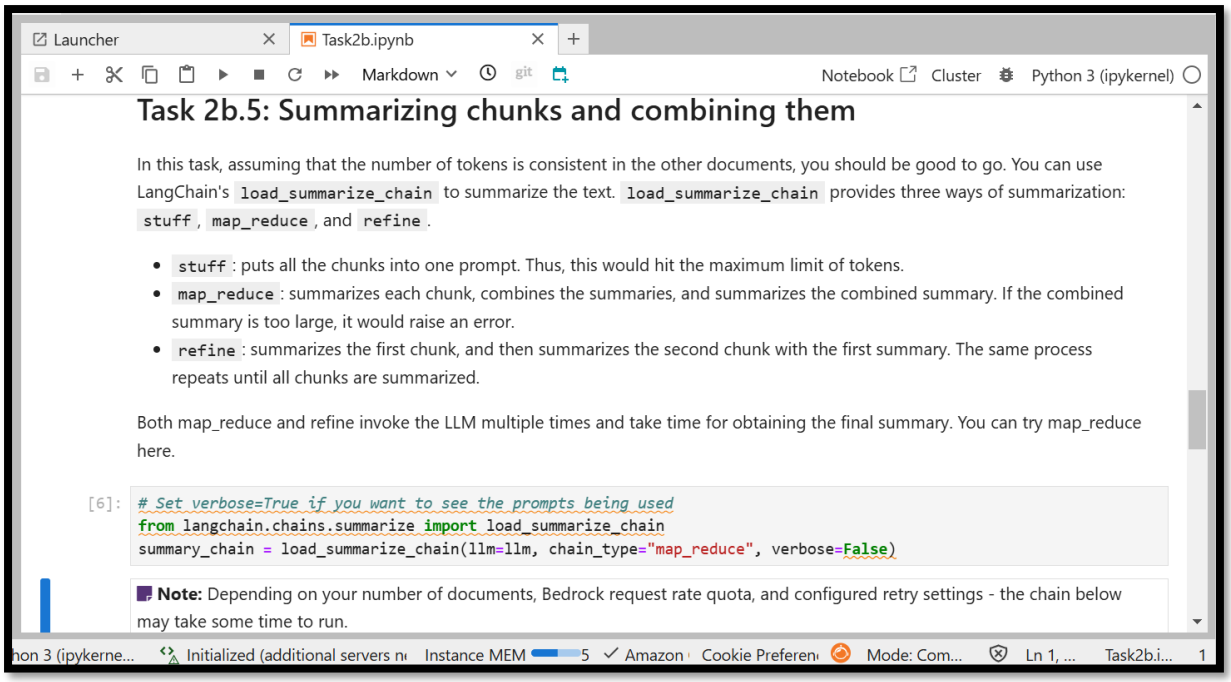


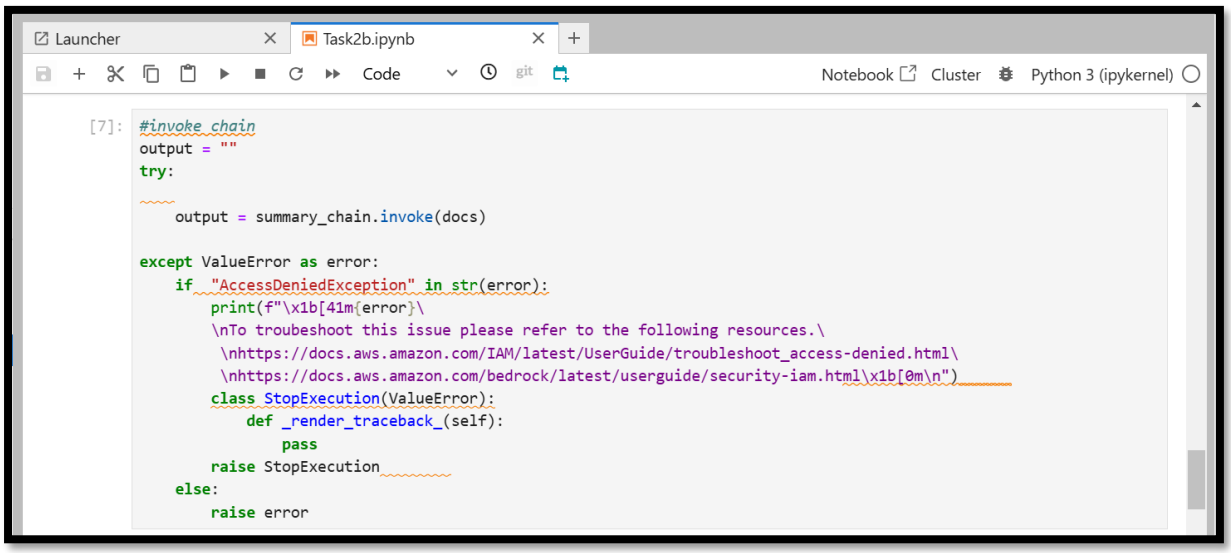


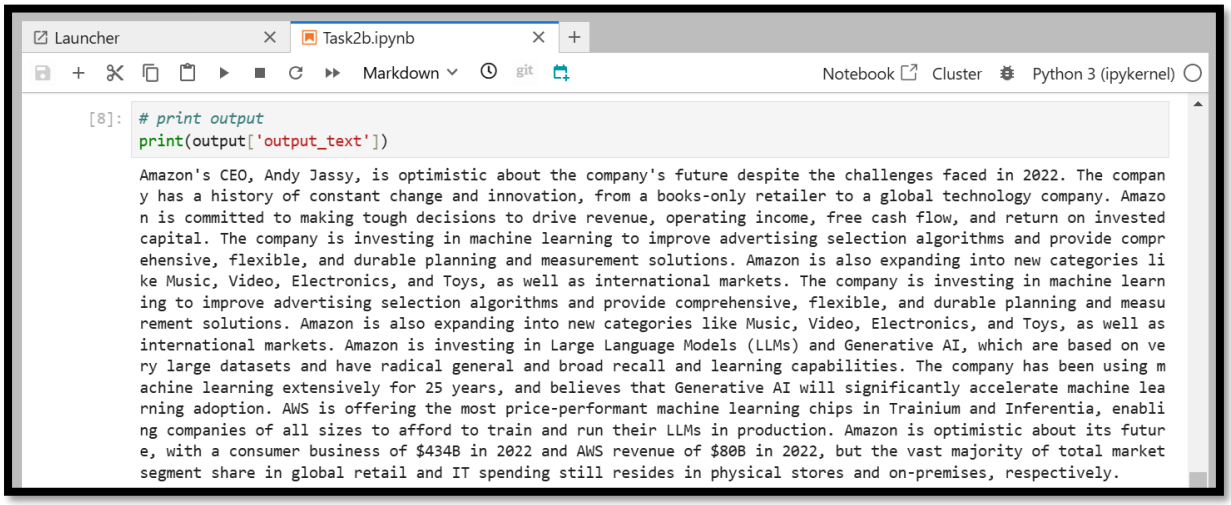

Launcher
Task2b.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
Task 2b.3: Loading a text file with many tokens
In this task, you can find a text file of Amazon's CEO letter to shareholders in 2022 in the letters directory. The following cell loads the text file and counts the number of tokens. You will see a warning indicating the number of tokens in the text file exceeds the maximum number of tokens for this model.
[3]:
#get_tokens
shareholder_letter = "../letters/2022-letter.txt"
with open(shareholder_letter, "r") as file:
letter = file.read()
llm.get_num_tokens(letter)
/opt/conda/lib/python3.11/site-packages/huggingface_hub/file_download.py:1142: FutureWarning: `resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to force a new download, use `force_download=True`.
warnings.warn(
Token indices sequence length is longer than the specified maximum sequence length for this model (6526 > 1024). Running this sequence through the model will result in indexing errors
[3]:
6526


Launcher
Task2b.ipynb
Code
Notebook
Cluster
Python 3 (ipykernel)
Task 2b.4: Splitting the long text into chunks
In this task, you split the text into smaller chunks because it is too long to fit in the prompt. RecursiveCharacterTextSplitter in LangChain supports splitting long text into chunks recursively until the size of each chunk becomes smaller than chunk_size. A text is separated with separators=["\n\n", "\n"] into chunks, which avoids splitting each paragraph into multiple chunks.
Using 6,000 characters per chunk, you can get summaries for each portion separately. The number of tokens, or word pieces, in a chunk depends on the text.
[4]:
#chunking
from langchain.text_splitter import RecursiveCharacterTextSplitter
text_splitter = RecursiveCharacterTextSplitter(
separators=["\n\n", "\n"], chunk_size=4000, chunk_overlap=100
)
docs = text_splitter.create_documents([letter])


Launcher
Task2b.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
[5]:
num_docs = len(docs)
num_tokens_first_doc = llm.get_num_tokens(docs[0].page_content)
print(
f"Now we have {num_docs} documents and the first one has {num_tokens_first_doc} tokens"
)
Now we have 10 documents and the first one has 439 tokens


Launcher
Task2b.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
Task 2b.5: Summarizing chunks and combining them
In this task, assuming that the number of tokens is consistent in the other documents, you should be good to go. You can use LangChain's load_summarize_chain to summarize the text. load_summarize_chain provides three ways of summarization: stuff, map_reduce, and refine.
stuff : puts all the chunks into one prompt. Thus, this would hit the maximum limit of tokens.
map_reduce : summarizes each chunk, combines the summaries, and summarizes the combined summary. If the combined summary is too large, it would raise an error.
refine : summarizes the first chunk, and then summarizes the second chunk with the first summary. The same process repeats until all chunks are summarized.
Both map_reduce and refine invoke the LLM multiple times and take time for obtaining the final summary. You can try map_reduce here.
[6]:
# Set verbose=True if you want to see the prompts being used
from langchain.chains.summarize import load_summarize_chain
summary_chain = load_summarize_chain(llm=llm, chain_type="map_reduce", verbose=False)
Note: Depending on your number of documents, Bedrock request rate quota, and configured retry settings - the chain below may take some time to run.
Initialized (additional servers n
Instance MEM
Amazon
Cookie Preferen
Mode: Com...
Ln 1, ...
Task2b.i...


Launcher
Task2b.ipynb
Code
Notebook
Cluster
Python 3 (ipykernel)
[7]:
#invoke_chain
output = ""
try:
output = summary_chain.invoke(docs)
except ValueError as error:
if "AccessDeniedException" in str(error):
print(f"\x1b[41m{error}\
\nTo troubeshoot this issue please refer to the following resources.\
\nhttps://docs.aws.amazon.com/IAM/latest/UserGuide/troubleshoot_access-denied.html\
\nhttps://docs.aws.amazon.com/bedrock/latest/userguide/security-iam.html\x1b[0m\n")
class StopExecution(ValueError):
def _render_traceback_(self):
pass
raise StopExecution
else:
raise error


Launcher
Task2b.ipynb
Markdown
Notebook
Cluster
Python 3 (ipykernel)
[8]:
# print output
print(output['output_text'])
Amazon's CEO, Andy Jassy, is optimistic about the company's future despite the challenges faced in 2022. The compan
y has a history of constant change and innovation, from a books-only retailer to a global technology company. Amazo
n is committed to making tough decisions to drive revenue, operating income, free cash flow, and return on invested
capital. The company is investing in machine learning to improve advertising selection algorithms and provide compr
ehensive, flexible, and durable planning and measurement solutions. Amazon is also expanding into new categories li
ke Music, Video, Electronics, and Toys, as well as international markets. The company is investing in machine learn
ing to improve advertising selection algorithms and provide comprehensive, flexible, and durable planning and measu
rement solutions. Amazon is also expanding into new categories like Music, Video, Electronics, and Toys, as well as
international markets. Amazon is investing in Large Language Models (LLMs) and Generative AI, which are based on ve
ry large datasets and have radical general and broad recall and learning capabilities. The company has been using m
achine learning extensively for 25 years, and believes that Generative AI will significantly accelerate machine lea
rning adoption. AWS is offering the most price-performant machine learning chips in Trainium and Inferentia, enabli
ng companies of all sizes to afford to train and run their LLMs in production. Amazon is optimistic about its futur
e, with a consumer business of $434B in 2022 and AWS revenue of $80B in 2022, but the vast majority of total market
segment share in global retail and IT spending still resides in physical stores and on-premises, respectively.