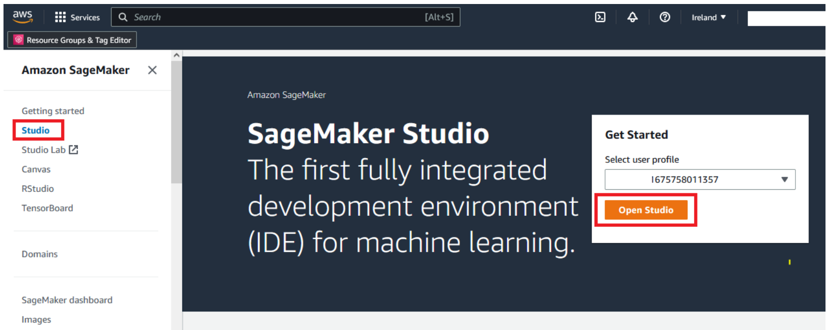
# Lab – Develop POC App

**Prerequisite**: This lab assumes that you have already configured SageMaker Studio Domain in the Ireland region.

This lab has 4 sections:

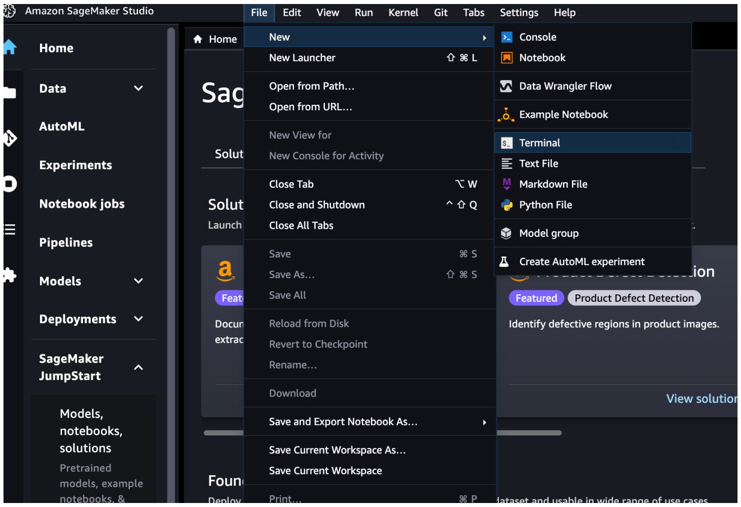
## Section 1- Create Streamlit HelloWorld App

1. Login to your AWS Account, select **Ireland (eu-west-1)** as the region and navigate to Amazon SageMaker Management Console. Click on **Studio** link in the left and then click on the **Open Studio** link.

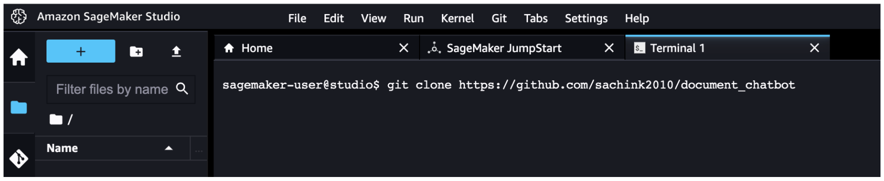


The first fully integrated development environment (IDE) for machine learning.

2. It will launch Amazon SageMaker Studio in a new browser window or tab. In the studio, click on **File** in the top menu. Next Open Terminal

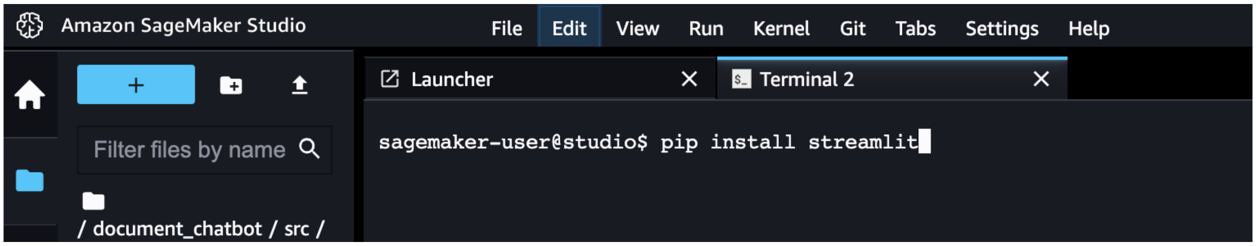


3. It will open **Terminal** tab. You can git clone here to download repository: https://github.com/sachink2010/document_chatbot



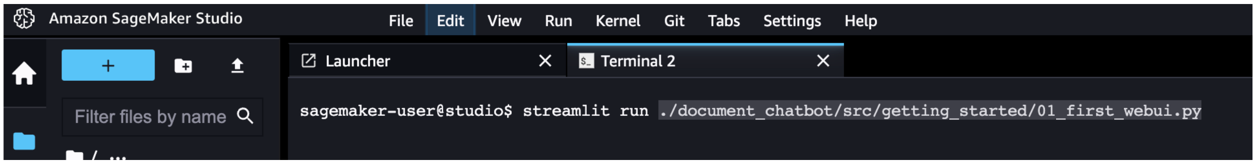
```
sagemaker-user@studio$ git clone https://github.com/sachink2010/document_chatbot
```

4. In the terminal, run command: pip install streamlit



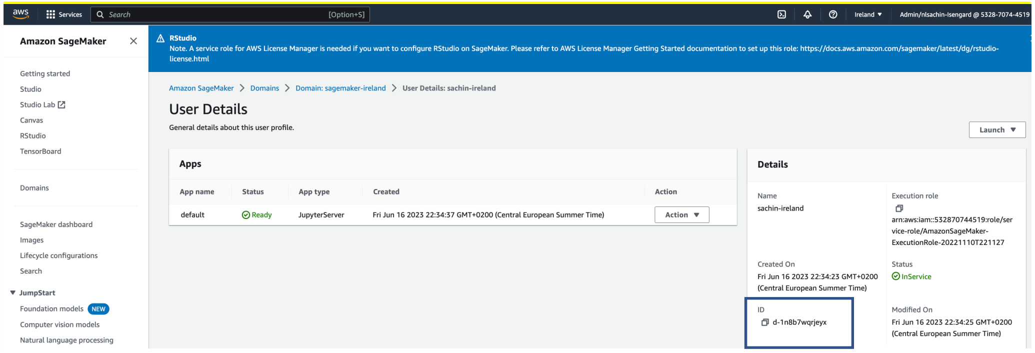
```
sagemaker-user@studio$ pip install streamlit
```

5. In the terminal select file path of streamlit and run:
streamlit run ./document_chatbot/src/getting_started/01_first_webui.py --server.port 6006



```
sagemaker-user@studio$ streamlit run ./document_chatbot/src/getting_started/01_first_webui.py
```

6. Copy Sagemaker domain id:



RStudio
Note. A service role for AWS License Manager is needed if you want to configure RStudio on SageMaker. Please refer to AWS License Manager Getting Started documentation to set up this role: https://docs.aws.amazon.com/sagemaker/latest/dg/rstudio-license.html

Amazon SageMaker > Domains > Domain: sagemaker-ireland > User Details: sachin-ireland

User Details

General details about this user profile.

Apps

| App name | Status | App type | Created | Action |
|---|---|---|---|---|
| default | Ready | JupyterServer | Fri Jun 16 2023 22:34:37 GMT+0200 (Central European Summer Time) | Action |

Details

Name
sachin-ireland

Execution role
arn:aws:iam::532870744519:role/service-role/AmazonSageMaker-ExecutionRole-20221110T221127

Created On
Fri Jun 16 2023 22:34:23 GMT+0200 (Central European Summer Time)

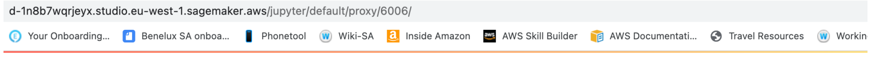
Status
InService

ID
d-1n8b7wqrjeyx

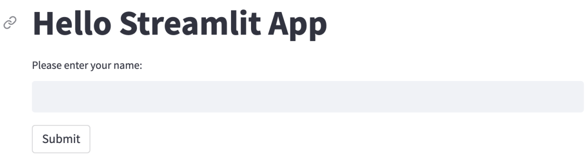
Modified On
Fri Jun 16 2023 22:34:25 GMT+0200 (Central European Summer Time)

7. When you run the Streamlit run command, app will be deployed and you will get the port number where it is deployed. In a new same browser window in a new tab, open link: https://studio-id.studio.region.sagemaker.aws/jupyter/default/proxy/port/

https://d-1n8b7wqrjeyx.studio.eu-west-1.sagemaker.aws/jupyter/default/proxy/6006/



d-1n8b7wqrjeyx.studio.eu-west-1.sagemaker.aws/jupyter/default/proxy/6006/

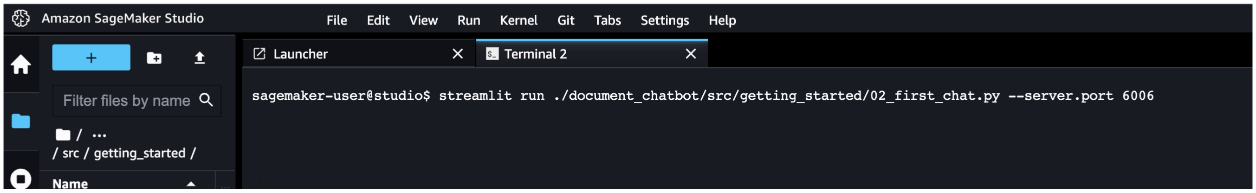


Hello Streamlit App

Please enter your name:

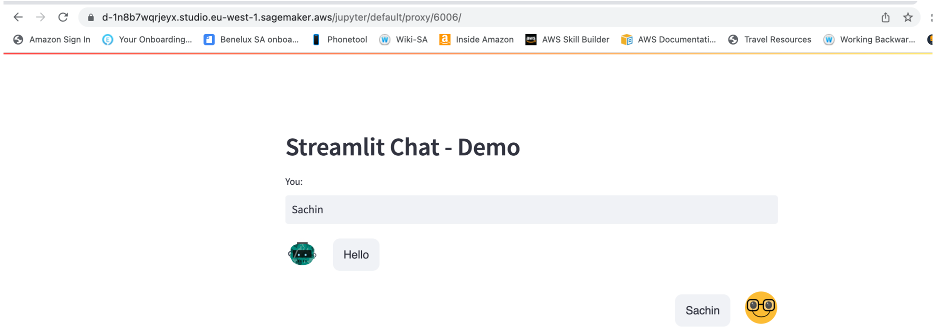
## Section 2- Create Streamlit Chat App

1. In Terminal window pip install streamlit_chat
2. In the same terminal window enter command:
   streamlit run ./document_chatbot/src/getting_started/02_first_chat.py --server.port 6006



```
sagemaker-user@studio$ streamlit run ./document_chatbot/src/getting_started/02_first_chat.py --server.port 6006
```

3. In a new same browser window in a new tab, open link:
   https://studio-id.studio.region.sagemaker.aws/jupyter/default/proxy/port/
   https://d-1n8b7wqrjeyx.studio.eu-west-1.sagemaker.aws/jupyter/default/proxy/6006/


d-1n8b7wqrjeyx.studio.eu-west-1.sagemaker.aws/jupyter/default/proxy/6006/

Streamlit Chat - Demo

You:

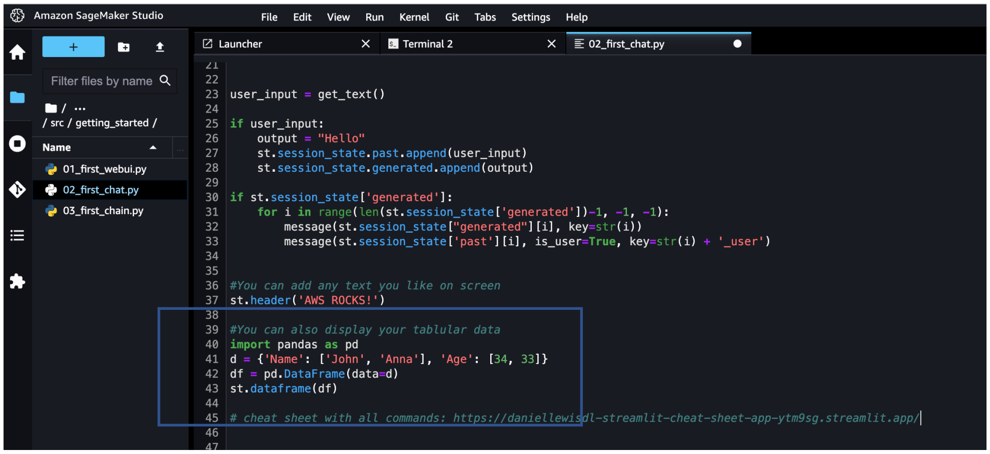
Sachin

Hello

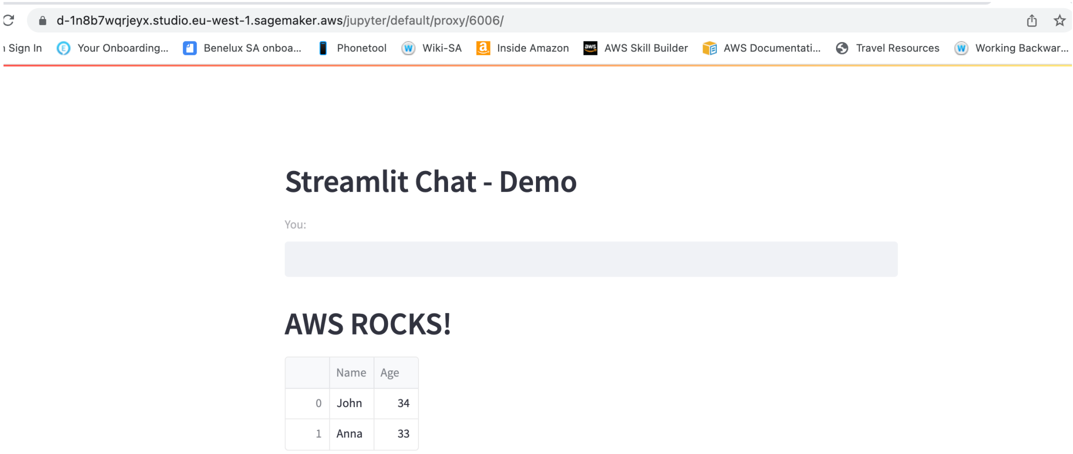
Sachin

4. You can add more Streamlit commands to the 02_first_chat.py file to modify the front-end. See some sample commands here: https://daniellewisdl-streamlit-cheat-sheet-app-ytm9sg.streamlit.app/ For e.g. you can add a data frame, or add text: Add this code and refresh app:

```
#You can also display your tabular data

import pandas as pd

d = {'Name': ['John', 'Anna'], 'Age': [34, 33]}

df = pd.DataFrame(data=d)

st.dataframe(df)
```



```python


user_input = get_text()

if user_input:
    output = "Hello"
    st.session_state.past.append(user_input)
    st.session_state.generated.append(output)

if st.session_state['generated']:
    for i in range(len(st.session_state['generated'])-1, -1, -1):
        message(st.session_state["generated"][i], key=str(i))
        message(st.session_state['past'][i], is_user=True, key=str(i) + '_user')


#You can add any text you like on screen
st.header('AWS ROCKS!')

#You can also display your tablular data
import pandas as pd
d = {'Name': ['John', 'Anna'], 'Age': [34, 33]}
df = pd.DataFrame(data=d)
st.dataframe(df)

# cheat sheet with all commands: https://daniellewisdl-streamlit-cheat-sheet-app-ytm9sg.streamlit.app/

```



# Streamlit Chat - Demo

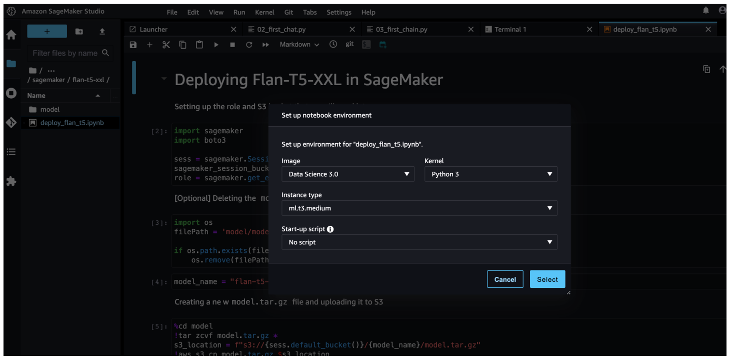
You:

## AWS ROCKS!

| | Name | Age |
|---|---|---|
| 0 | John | 34 |
| 1 | Anna | 33 |

## Section 3- Deploy a LLM and work with Langchain

1. Go to document_chatbot/src/sagemaker/flan-t5-xxl/deploy_flan_t5.ipynb, Select Image as Data Science 3.0 and click on Select



Deploying Flan-T5-XXL in SageMaker

Set up notebook environment

Set up environment for "deploy_flan_t5.ipynb".

Image: Data Science 3.0

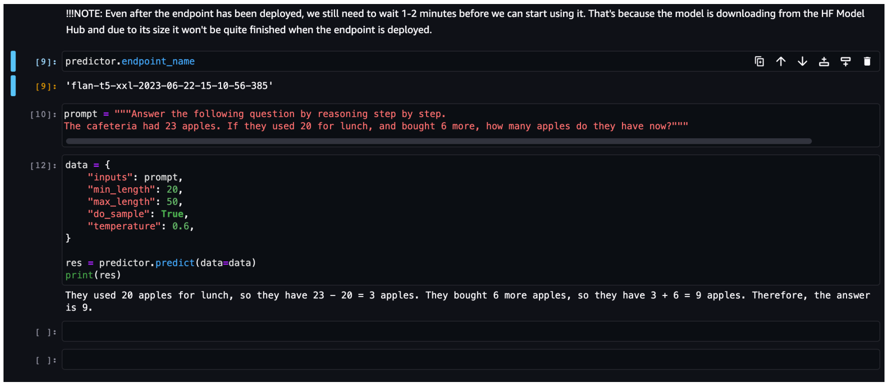
Kernel: Python 3

Instance type: ml.t3.medium

Start-up script: No script

Cancel | Select

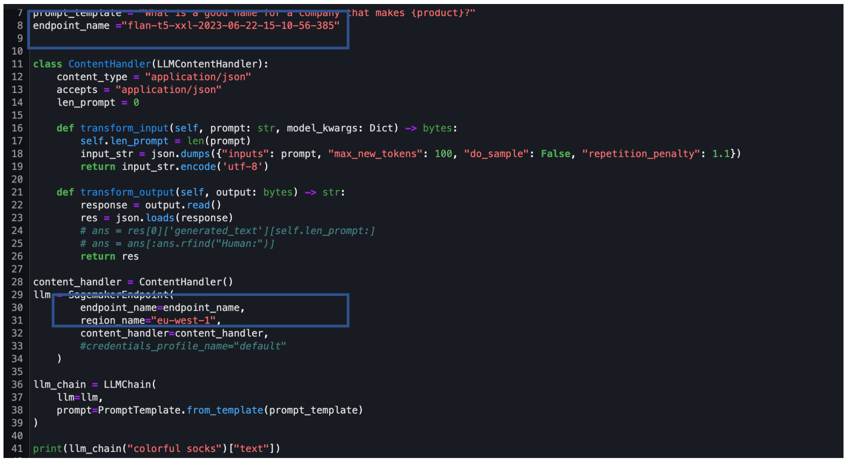
2. Run the cells in the notebook (Shift+enter) to deploy the end-point and check that predictor is working (See if you get response for the prompt entered).



!!!NOTE: Even after the endpoint has been deployed, we still need to wait 1-2 minutes before we can start using it. That's because the model is downloading from the HF Model Hub and due to its size it won't be quite finished when the endpoint is deployed.

```
[9]: predictor.endpoint_name

[9]: 'flan-t5-xxl-2023-06-22-15-10-56-385'

[10]: prompt = """Answer the following question by reasoning step by step.
The cafeteria had 23 apples. If they used 20 for lunch, and bought 6 more, how many apples do they have now?"""

[12]: data = {
          "inputs": prompt,
          "min_length": 20,
          "max_length": 50,
          "do_sample": True,
          "temperature": 0.6,
      }

      res = predictor.predict(data=data)
      print(res)

      They used 20 apples for lunch, so they have 23 - 20 = 3 apples. They bought 6 more apples, so they have 3 + 6 = 9 apples. Therefore, the answer
      is 9.

[ ]:

[ ]:
```

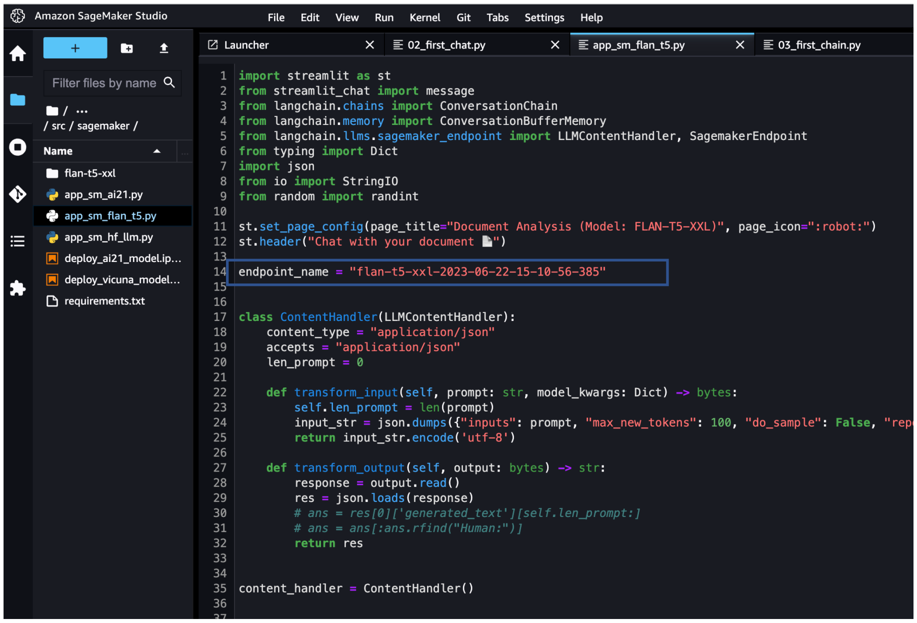
3. In Terminal window, enter pip install -r ./document_chatbot/src/sagemaker/requirements.txt
4. Note down the predictor.endpoint_name, and go to document_chatbot/src/getting_started/03_first_chain.py
   - Enter endpoint name from previous step
   - Check region name is eu-west-1



```python
prompt_template = "What is a good name for a company that makes {product}?"
endpoint_name ="flan-t5-xxl-2023-06-22-15-10-56-385"


class ContentHandler(LLMContentHandler):
    content_type = "application/json"
    accepts = "application/json"
    len_prompt = 0

    def transform_input(self, prompt: str, model_kwargs: Dict) -> bytes:
        self.len_prompt = len(prompt)
        input_str = json.dumps({"inputs": prompt, "max_new_tokens": 100, "do_sample": False, "repetition_penalty": 1.1})
        return input_str.encode('utf-8')

    def transform_output(self, output: bytes) -> str:
        response = output.read()
        res = json.loads(response)
        # ans = res[0]['generated_text'][self.len_prompt:]
        # ans = ans[:ans.rfind("Human:")]
        return res

content_handler = ContentHandler()
llm = SagemakerEndpoint(
        endpoint_name=endpoint_name,
        region_name="eu-west-1",
        content_handler=content_handler,
        #credentials_profile_name="default"
    )

llm_chain = LLMChain(
    llm=llm,
    prompt=PromptTemplate.from_template(prompt_template)
)

print(llm_chain("colorful socks")["text"])
```

## Section 4- Create Document App

1. Open file: document_chatbot/src/sagemaker/app_sm_flan_t5.py
2. Change End-point name to match your end point



```python
import streamlit as st
from streamlit_chat import message
from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory
from langchain.llms.sagemaker_endpoint import LLMContentHandler, SagemakerEndpoint
from typing import Dict
import json
from io import StringIO
from random import randint

st.set_page_config(page_title="Document Analysis (Model: FLAN-T5-XXL)", page_icon=":robot:")
st.header("Chat with your document 📄")

endpoint_name = "flan-t5-xxl-2023-06-22-15-10-56-385"


class ContentHandler(LLMContentHandler):
    content_type = "application/json"
    accepts = "application/json"
    len_prompt = 0

    def transform_input(self, prompt: str, model_kwargs: Dict) -> bytes:
        self.len_prompt = len(prompt)
        input_str = json.dumps({"inputs": prompt, "max_new_tokens": 100, "do_sample": False, "rep
        return input_str.encode('utf-8')

    def transform_output(self, output: bytes) -> str:
        response = output.read()
        res = json.loads(response)
        # ans = res[0]['generated_text'][self.len_prompt:]
        # ans = ans[:ans.rfind("Human:")]
        return res


content_handler = ContentHandler()

```

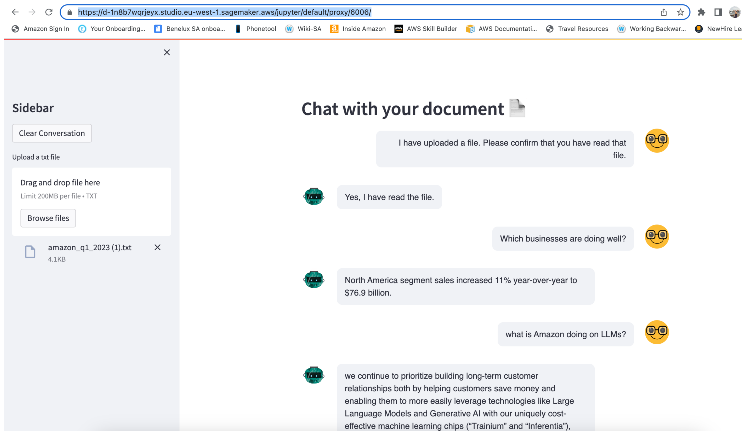
3. In terminal window enter command:
   streamlit run ./document_chatbot/src/sagemaker/app_sm_flan_t5.py --server.port 6006

   Open in the browser streamlit app:

   https://d-1n8b7wqrjeyx.studio.eu-west-1.sagemaker.aws/jupyter/default/proxy/6006/

4. Download documents from assets folder to your local machine:
   document_chatbot/assets/amazon_q1_2023_short.txt

5. Upload the document to the browser and start sending questions to the chatbot:



Chat with your document 📄

I have uploaded a file. Please confirm that you have read that file.

Yes, I have read the file.

Which businesses are doing well?

North America segment sales increased 11% year-over-year to $76.9 billion.

what is Amazon doing on LLMs?

we continue to prioritize building long-term customer relationships both by helping customers save money and enabling them to more easily leverage technologies like Large Language Models and Generative AI with our uniquely cost-effective machine learning chips (“Trainium” and “Inferentia”),