[1]:
```
import sagemaker, boto3, json
from sagemaker.session import Session

sagemaker_session = Session()
aws_role = sagemaker_session.get_caller_identity_arn()
aws_region = boto3.Session().region_name
sess = sagemaker.Session()
print(aws_role)
print(aws_region)
print(sess)
```

```
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
arn:aws:iam::512229249619:role/service-role/SageMaker-udacitySagemakerRole
us-west-2
<sagemaker.session.Session object at 0x7ff012ff7190>
```

2. Select Text Generation Model Meta Llama 2 7B

Run the next cell to set variables that contain the values of the name of the model we want to load and the version of the model .

[2]:
```
(model_id, model_version,) = ("meta-textgeneration-llama-2-7b","2.*",)
```

Running the next cell deploys the model This Python code is used to deploy a machine learning model using Amazon SageMaker's JumpStart library.

[3]:
```
from sagemaker.jumpstart.model import JumpStartModel

model = JumpStartModel(model_id=model_id, model_version=model_version, instance_type="ml.g5.2xlarge")
predictor = model.deploy()
```

```
For forward compatibility, pin to model_version='2.*' in your JumpStartModel or JumpStartEstimator definitions. N
ote that major version upgrades may have different EULA acceptance terms and input/output signatures.
Using model 'meta-textgeneration-llama-2-7b' with wildcard version identifier '2.*'. You can pin to version '2.1.
8' for more stable results. Note that models may have different input/output signatures after a major version upg
rade.
----------------!
```

[4]:
```
def print_response(payload, response):
    print(payload["inputs"])
    print(f"> {response[0]['generation']}")
    print("\n==================================\n")
```

[5]:
```
payload = {
    "inputs": "Traditional approaches to data management such as",
    "parameters": {
        "max_new_tokens": 64,
        "top_p": 0.9,
        "temperature": 0.6,
        "return_full_text": False,
    },
}
try:
    response = predictor.predict(payload, custom_attributes="accept_eula=true")
    print_response(payload, response)
except Exception as e:
    print(e)
```

```
Traditional approaches to data management such as
>  the relational database have been around for decades. But they are no longer enough to support the modern data
management needs of enterprises. The emergence of new data sources and the explosion in the volume and variety of
data have led to the development of new approaches to data management. These new approaches include No

==================================
```

After you've filled out the report, run the cells below to delete the model deployment

`IF YOU FAIL TO RUN THE CELLS BELOW YOU WILL RUN OUT OF BUDGET TO COMPLETE THE PROJECT`

[6]:

```
# Delete the SageMaker endpoint and the attached resources
predictor.delete_model()
predictor.delete_endpoint()
```

Verify your model endpoint was deleted by visiting the Sagemaker dashboard and choosing `endpoints` under 'Inference' in the left navigation menu. If you see your endpoint still there, choose the endpoint, and then under "Actions" select **Delete**