

# Lab

- Model Fine-tuning of image generation with Bedrock/Titan Image Generator

<https://github.com/aws-samples/amazon-bedrock-samples/tree/main/bedrock-fine-tuning/amazon-titan-image-generator>



+    Filter files by name    Q

Filter files by name    Q

...

Name    Last Modified

- data    a day ago
- prompts    15 hours ago
- 1-TIGFT-customiz...    an hour ago
- 2-TIGFT-provisio...    an hour ago
- README.md    a day ago

00\_setup.ipynb X 01\_fine-tuning-titan-lit... X 02\_fine-tuning\_llama2... X 03\_continued\_pretraini... X 1-TIGFT-customization- X 2-TIGFT-provisioned-thi... X

Cluster Data Science 3.0 | Python 3 | 4 vCPU + 8 GiB Share ⚡

# Fine-Tuning Amazon Titan Image Generator G1

This notebook has been tested with the **SageMaker Data Science 3.0** kernel in Amazon SageMaker Studio.

In this notebook, we will show how to fine tune [Amazon Titan Image Generator G1](#) on Amazon Bedrock model.

We will teach our model to recognize two new classes:

Ron the dog



and Smila the cat

Amazon SageMaker Studio Classic

File Edit View Run Kernel Git Tabs Settings Help

default-20240221t112169 / Personal Studio

00\_setup.ipynb X 01\_fine-tunin X 02\_fine-tunin X 03\_continue X 1-TIGFT-cust X ron\_01.jpg X ron\_ft\_1.png X ron\_ft\_2.png X 2-TIGFT-prov X

Filter files by name

Name Last Modified

- ron\_16.jpg a day ago
- ron\_17.jpg a day ago
- ron\_18.jpg a day ago
- ron\_19.jpg a day ago
- ron\_20.jpg a day ago
- ron\_21.jpg a day ago
- ron\_22.jpg a day ago
- ron\_23.jpg a day ago
- ron\_24.jpg a day ago
- ron\_25.jpg a day ago
- ron\_26.jpg a day ago
- ron\_27.jpg a day ago
- ron\_28.jpg a day ago
- ron\_29.jpg a day ago
- ron\_30.jpg a day ago
- ron\_ft\_1.png a day ago
- ron\_ft\_2.png a day ago
- smila\_01.jpg a day ago
- smila\_02.jpg a day ago
- smila\_03.jpg a day ago
- smila\_04.jpg a day ago
- smila\_05.jpg a day ago
- smila\_06.jpg a day ago



Simple 2 \$ 6 main

Cookie Preferences ron\_ft\_1.png 0

Type here to search

12:50 PM

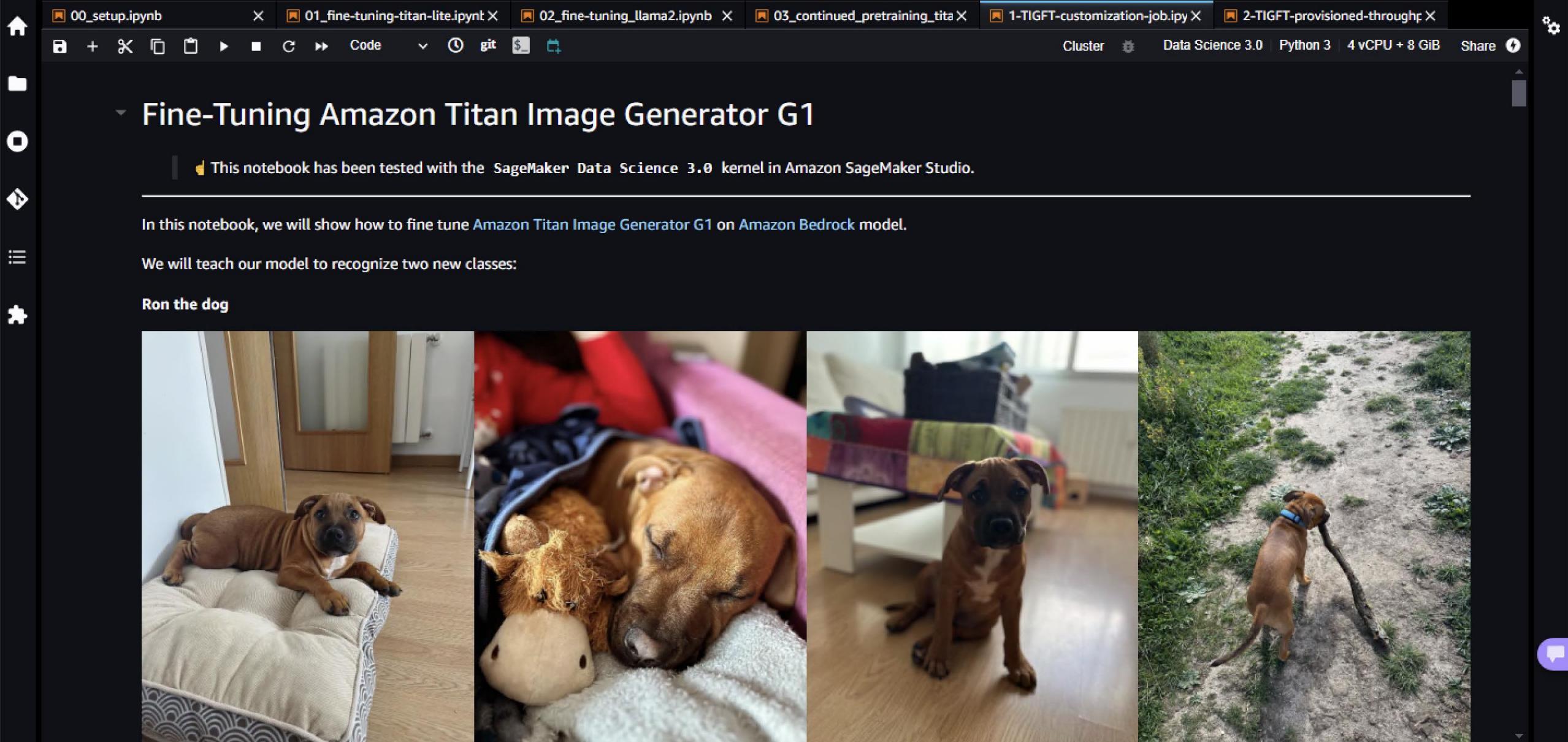
+    -    ↑    C    ⌂

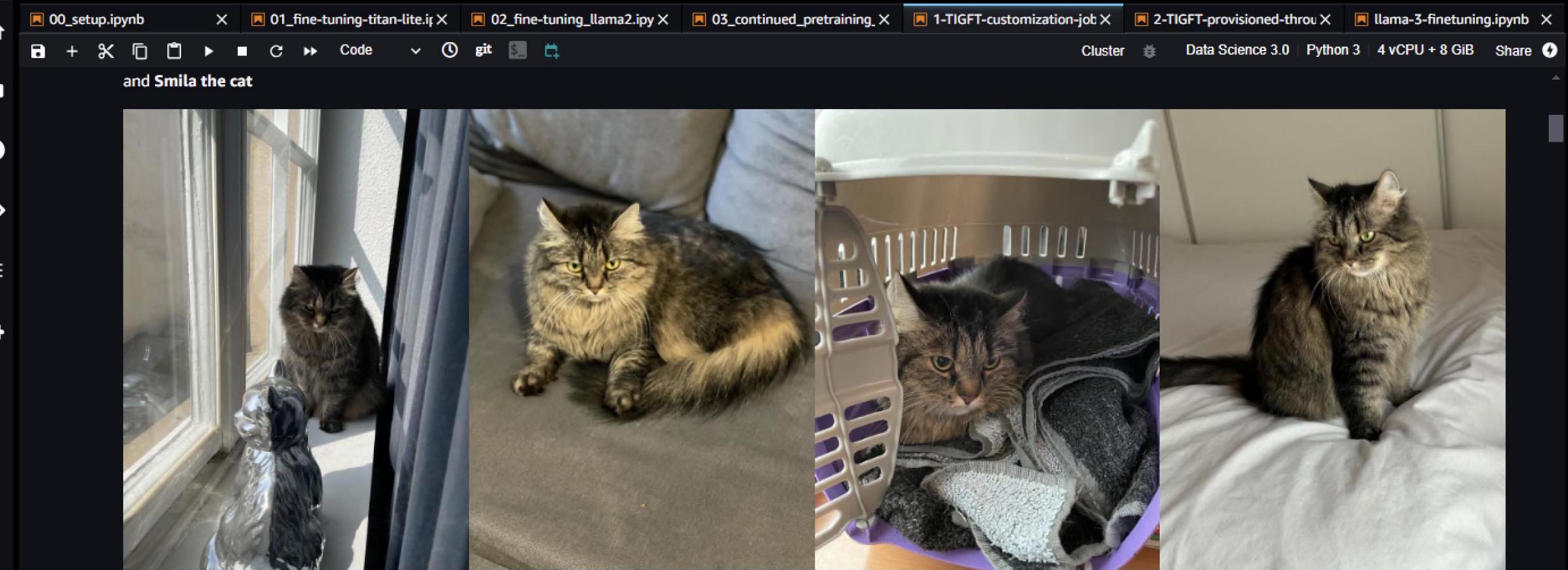
Filter files by name

amazon-titan-image-generator / data /

Name	Last Modified
smila_10.jpg	a day ago
smila_11.jpg	a day ago
smila_12.jpg	a day ago
smila_13.jpg	a day ago
smila_14.jpg	a day ago
smila_15.jpg	a day ago
smila_16.jpg	a day ago
smila_17.jpg	a day ago
smila_18.jpg	a day ago
smila_19.jpg	a day ago
smila_20.jpg	a day ago
smila_21.jpg	a day ago
smila_22.jpg	a day ago
smila_23.jpg	a day ago
smila_24.jpg	a day ago
smila_25.jpg	a day ago
smila_26.jpg	a day ago
smila_27.jpg	a day ago
smila_28.jpg	a day ago
smila_29.jpg	a day ago
smila_30.jpg	a day ago
smila_ft_1.png	a day ago
smila_ft_2.png	a day ago







```
[2]: !pip install --upgrade --force-reinstall --no-cache boto3  
!pip install --upgrade --force-reinstall --no-cache botocore  
!pip install --upgrade --force-reinstall --no-cache awscli
```

```
Collecting boto3  
  Downloading boto3-1.34.143-py3-none-any.whl.metadata (6.6 kB)  
Collecting botocore<1.35.0,>=1.34.143 (from boto3)  
  Downloading botocore-1.34.143-py3-none-any.whl.metadata (5.7 kB)  
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
```

Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ipynb X 2-TIGFT-provisioned-throughp X

Cluster Data Science 3.0 Python 3 4 vCPU + 8 GiB Share

## Data preparation

- To fine-tune a text-to-image or image-to-embedding model, prepare a training dataset by create a JSONL file with multiple JSON lines.
- Validation datasets are not supported.
- Each JSON line is a sample containing an image-ref, the Amazon S3 URI for an image, and a caption that could be a prompt for the image.

The images must be in JPEG or PNG format.

```
{"image-ref": "s3://bucket/path/to/image001.png", "caption": "<prompt text>"}  
{"image-ref": "s3://bucket/path/to/image002.png", "caption": "<prompt text>"}  
{"image-ref": "s3://bucket/path/to/image003.png", "caption": "<prompt text>"}
```

The following is an example item:

```
{"image-ref": "s3://my-bucket/my-pets/cat.png", "caption": "an orange cat with white spots"}
```

### Locate your sample json file

We are going to use a json file which contains the image captions in the following format:

```
{  
    "imagefile": "caption",  
    "imagefile": "caption",  
    "imagefile": "caption"  
}
```

[5]: raw\_data\_file = "prompts/captions.json"

```
with open(raw_data_file, 'r') as file:  
    raw_data = json.load(file)
```

```
[5]: raw_data_file = "prompts/captions.json"

with open(raw_data_file, 'r') as file:
    raw_data = json.load(file)

print(raw_data)

{'ron_01.jpg': 'Ron the dog laying on a white dog bed', 'ron_02.jpg': 'Ron the dog sitting on a tile floor, possibly in a kitchen or living room', 'ron_03.jpg': 'Ron the dog laying on a car seat', 'ron_04.jpg': 'Ron the dog looking directly at the camera. He is laying down on a wooden floor.', 'ron_05.jpg': 'Ron the dog sitting on a couch, looking at the camera with a smile on his face.', 'ron_06.jpg': 'Ron the dog lying on a couch, covered in a blanket.', 'ron_07.jpg': 'Ron the dog sleeping or resting, with his head on the stuffed animal.', 'ron_08.jpg': 'Ron the dog sitting on a box with a red leash.', 'ron_09.jpg': 'Ron the dog sitting in the snow, wearing a red collar.', 'ron_10.jpg': 'Ron the dog lying on a couch, chewing on a tennis shoe.', 'ron_11.jpg': 'Ron the dog sitting on a sandy beach, wearing a blue collar.', 'ron_12.jpg': 'Ron the dog wearing a yellow raincoat and is sitting on the floor.', 'ron_13.jpg': 'Ron the dog looking at the camera in front of a table.', 'ron_14.jpg': 'Ron the dog lying on a white couch, playing with a pink toy.', 'ron_15.jpg': 'Ron the dog sleeping on a white couch.', 'ron_16.jpg': 'Ron the dog laying on a couch.', 'ron_17.jpg': 'Ron the dog laying on a couch.', 'ron_18.jpg': 'Ron the dog playing with a ball.', 'ron_19.jpg': 'Ron the dog sitting on a couch.', 'ron_20.jpg': 'Ron the dog holding a large stick in his mouth.', 'ron_21.jpg': 'Ron the dog sitting on a brick floor.', 'ron_22.jpg': 'Ron the dog laying on a blanket.', 'ron_23.jpg': 'Ron the dog sitting on a couch.', 'ron_24.jpg': 'Ron the dog standing on a tile floor, wearing a leash.', 'ron_25.jpg': 'Ron the dog wearing a blue and white jacket and is sitting on the grass.', 'ron_26.jpg': 'Ron the dog standing in front of a large brown teddy bear.', 'ron_27.jpg': 'Ron the dog is standing in a kitchen, looking at the camera.', 'ron_28.jpg': 'Ron the dog sitting on a couch.', 'ron_29.jpg': 'Ron the dog wearing a festive sweater with snowflakes on it.', 'ron_30.jpg': 'Ron the dog wearing a red sweater with a white snowflake pattern.', 'smila_01.jpg': 'Smila the cat lying on a couch.', 'smila_02.jpg': 'Smila the cat sitting next to the window next to a statue cat.', 'smila_03.jpg': 'Smila the cat sitting inside a colorful cat tunnel.', 'smila_05.jpg': 'Smila the cat lying on a sofa.', 'smila_06.jpg': 'Smila the cat sitting on a couch.', 'smila_07.jpg': 'Smila the cat sitting on a black cupboard.', 'smila_08.jpg': 'Smila the cat is sitting inside a cat carrier.', 'smila_09.jpg': 'Smila the cat is sitting on the floor next to a door.', 'smila_10.jpg': 'Smila the cat is sitting in a pet carrier.', 'smila_12.jpg': 'Smila the cat lying on the sofa.', 'smila_13.jpg': 'Smila the cat lying on a grey couch.', 'smila_14.jpg': 'Smila the cat sitting inside a colorful tunnel.', 'smila_15.jpg': 'Smila the cat sitting in a cat transporter.', 'smila_16.jpg': 'Smila the cat is sitting next to a window.', 'smila_17.jpg': 'Smila the cat laying on a white table.', 'smila_18.jpg': 'Smila the cat lying on a cat scratcher.', 'smila_19.jpg': 'Smila the cat sitting on a wicker basket in the bathroom.', 'smila_20.jpg': 'Smila the cat sitting on a couch, eating yogurt.', 'smila_21.jpg': 'Smila the cat sitting on a wicker basket in a bathroom.', 'smila_22.jpg': 'Smila the cat laying on a bed.', 'smila_23.jpg': 'Smila the cat sitting on a cat bed.', 'smila_24.jpg': 'Smila the cat sitting on a white bed.', 'smila_25.jpg': 'Smila the cat laying on a cat scratcher.', 'smila_26.jpg': 'Smila the cat is laying on a gray sofa, surrounded by pillows.', 'smila_27.jpg': 'Smila the cat sitting on a couch.', 'smila_28.jpg': 'Smila the cat laying on a gray sofa, looking at the camera.', 'smila_29.jpg': 'Smila the cat sitting on a bench, staring at the camera.', 'smila_30.jpg': 'Smila the cat lying on a white bed.'}
```

## Create the dataset file and upload the images to Amazon S3

Create the `jsonl` file with the images prompt based on the image's s3 path.

Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

[11]: ts = datetime.datetime.now().strftime("%Y-%m-%d-%H-%M-%S")

# Select the foundation model you want to customize (you can find this from the "modelId" from listed foundation model above)  
base\_model\_id = "amazon.titan-image-generator-v1:0"

# Select the customization type from "FINE\_TUNING" or "CONTINUED\_PRE\_TRAINING".  
customization\_type = "FINE\_TUNING"

# Specify the roleArn for your customization job  
customization\_role = role\_arn

# Create a customization job name  
customization\_job\_name = f"image-gen-ft-{ts}"

# Create a customized model name for your fine-tuned Llama2 model  
custom\_model\_name = f"image-gen-ft-{ts}"

# Define the hyperparameters for fine-tuning Llama2 model  
hyper\_parameters = {  
 "stepCount": "8000",  
 "batchSize": "8",  
 "learningRate": "0.00001",  
}

# Specify your data path for training, validation(optional) and output  
s3\_train\_uri = s3\_bucket\_path + "/" + output\_file  
training\_data\_config = {"s3Uri": s3\_train\_uri}

output\_data\_config = {"s3Uri": f's3://{bucket\_name}/outputs/output-{custom\_model\_name}'}

# Create the customization job  
bedrock\_client.create\_model\_customization\_job(  
 customizationType=customization\_type,  
 jobName=customization\_job\_name,



Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ipynb X 2-TIGFT-provisioned-throughp X

Cluster Data Science 3.0 Python 3 4 vCPU + 8 GiB Share

```
# Create the customization job
bedrock_client.create_model_customization_job(
    customizationType=customization_type,
    jobName=customization_job_name,
    customModelName=custom_model_name,
    roleArn=customization_role,
    baseModelIdentifier=base_model_id,
    hyperParameters=hyper_parameters,
    trainingDataConfig=training_data_config,
    outputDataConfig=output_data_config
)
```

[11]: { 'ResponseMetadata': { 'RequestId': 'c1c09ec1-3658-4593-b986-e13cb80ae3e9',
 'HTTPStatusCode': 201,
 'HTTPHeaders': { 'date': 'Fri, 12 Jul 2024 13:43:03 GMT',
 'content-type': 'application/json',
 'content-length': '122',
 'connection': 'keep-alive',
 'x-amzn-requestid': 'c1c09ec1-3658-4593-b986-e13cb80ae3e9'},
 'RetryAttempts': 0},
 'jobArn': 'arn:aws:bedrock:us-east-1:094784590684:model-customization-job/amazon.titan-image-generator-v1:0/hw5vebkzf8c1'}

## Waiting until customization job is completed

Once the customization job is finished, you can check your existing custom model(s) and retrieve the `modelArn` of your fine-tuned model.

**Warning:** The model customization job can take hours to run. With 5000 steps, 0.000001 learning rate, 64 of batch size and 60 images, it takes around 4 hours to complete

```
[ ]: # check model customization status
status = bedrock_client.list_model_customization_jobs(
    nameContains=customization_job_name
)[["modelCustomizationJobSummaries"]][0]["status"]
```

A Search fo X | a Search fo X | a DevRel - X | Isengard X | Amazon X | Amazon X | 2-TIGFT- X | a Domain- X | Domain- X | The Ama X | +

← → C ⌂ d-c9qeuo6x91qp.studio.us-east-1.sagemaker.aws/jupyter/default/lab/tree/amazon-bedrock-samples/bedrock-fine-tuning/amazon-titan-image-generator/2-TIGFT-pr... ☆

Amazon SageMaker Studio Classic File Edit View Run Kernel Git Tabs Settings Help default-20240221t112169 / Personal Studio

00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ip X 2-TIGFT-provisioned-through X

Cluster Data Science 3.0 Python 3 4 vCPU + 8 GiB Share

# Image Generation with Fine-tuned Amazon Titan Image Generator G1 model

This notebook has been tested with the `SageMaker Data Science 3.0` kernel in Amazon SageMaker Studio.

---

In this notebook, we will show how to invoke a fine-tuned Amazon Titan Image Generator G1 on Amazon Bedrock trained with Ron the dog and Smila the cat



## Pre-requisites

Import needed libraries, instantiate the boto3 clients and setup any style tag that was added during model fine-tuning

Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

[9]:

```
def decode_base64_image(img_b64):
    return Image.open(io.BytesIO(base64.b64decode(img_b64)))

def invoke_model_tgi(prompt, seed, model_arn):
    img_gen_conf = {
        "cfgScale": 8,
        "seed": seed,
        "quality": "standard",
        "width": 1024,
        "height": 1024,
        "numberOfImages": 2
    }

    body = json.dumps({
        "textToImageParams": {
            "text": prompt
        },
        "taskType": "TEXT_IMAGE",
        "imageGenerationConfig": img_gen_conf
    })

    response = bedrock_runtime.invoke_model(
        body=body,
        modelId=model_arn,
        accept="application/json",
        contentType="application/json"
    )

    response_body = json.loads(response.get("body").read())
    return [decode_base64_image(img) for img in response_body["images"]]

def compare_models_outputs(prompts, seeds, model_arn_base, model_arn_custom):
    for prompt in prompts:
        for seed in seeds:
            # Invoke the base model
            base_image = invoke_model_tgi(prompt=prompt, seed=seed, model_arn=model_arn_base)
```

Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ipynb X 2-TIGFT-provisioned-throughp X

Cluster Data Science 3.0 Python 3 4 vCPU + 8 GiB Share

```
response = bedrock_runtime.invoke_model(
    body=body,
    modelId=model_arn,
    accept="application/json",
    contentType="application/json"
)

response_body = json.loads(response.get("body").read())
return [decode_base64_image(img) for img in response_body["images"]]

def compare_models_outputs(prompts, seeds, model_arn_base, model_arn_custom):
    for prompt in prompts:
        for seed in seeds:
            # Invoke the base model
            base_images = invoke_model_tgi(prompt, seed, model_arn_base)

            # Invoke the customized model
            custom_images = invoke_model_tgi(prompt, seed, model_arn_custom)

            # Display the images for comparison
            plot_images(base_images, custom_images, prompt, seed)

def plot_images(base_images, custom_images, prompt, seed):
    fig, axes = plt.subplots(1, 2, figsize=(10, 5))

    axes[0].imshow(np.array(base_images[0]))
    axes[0].set_title('Base Model\nSeed: {}'.format(seed))
    axes[0].axis('off')

    axes[1].imshow(np.array(custom_images[0]))
    axes[1].set_title('Custom Model\nSeed: {}'.format(seed))
    axes[1].axis('off')
    print("Prompt:{}\n".format(prompt))
    plt.show()
```

Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

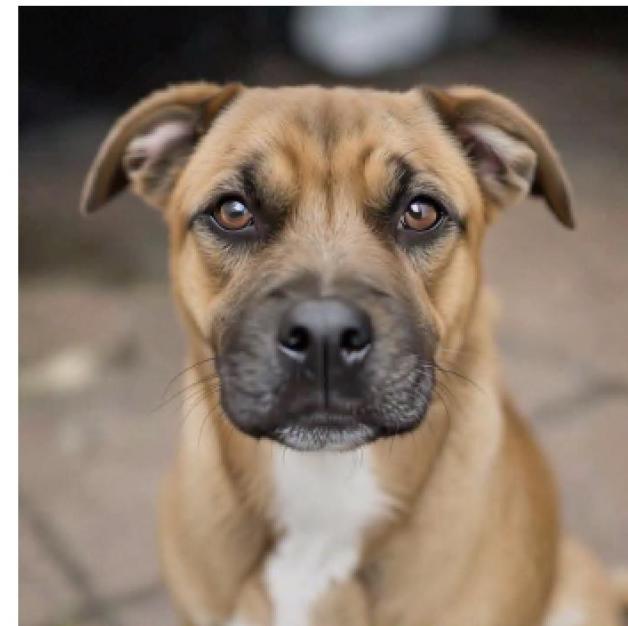
00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ipynb X 2-TIGFT-provisioned-throughp X

Prompt:'Ron the dog' is staring at the camera.

Base Model  
Seed: 3000



Custom Model  
Seed: 3000



Prompt:'Smila the cat' is staring at the camera.

Base Model  
Seed: 0



Custom Model  
Seed: 0



Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

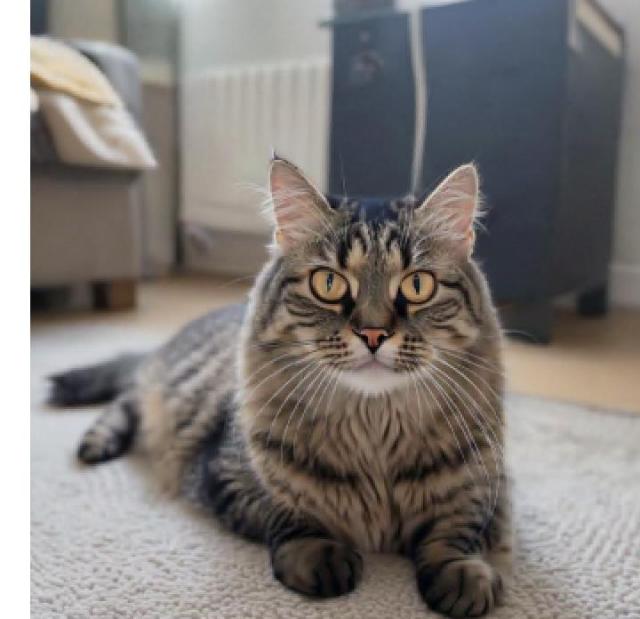
00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ipynb X 2-TIGFT-provisioned-throughp X

Prompt:'Smila the cat' is staring at the camera.

Base Model  
Seed: 0



Custom Model  
Seed: 0



Prompt:'Smila the cat' is staring at the camera.

Base Model  
Seed: 3000



Custom Model  
Seed: 3000



Amazon SageMaker Studio Classic      File Edit View Run Kernel Git Tabs Settings Help      default-20240221t112169 / Personal Studio

00\_setup.ipynb X 01\_fine-tuning-titan-lite.ipynb X 02\_fine-tuning\_llama2.ipynb X 03\_continued\_pretraining\_tita X 1-TIGFT-customization-job.ipynb X 2-TIGFT-provisioned-throughp X

Prompt: 'Ron the dog' is running on the beach.

Base Model  
Seed: 3000



Custom Model  
Seed: 3000



Prompt: 'Smila the cat' is running on the grass.

Base Model  
Seed: 0



Custom Model  
Seed: 0



Prompt:'Smila the cat' is running on the grass.

Base Model  
Seed: 3000



Custom Model  
Seed: 3000



[ ]:

## Clean Up

To avoid unnecessary costs, let's now delete the provisioned throughput model