

# Building Owly a personalised Comic video Generator with Stable Diffusion 2.1 on Amazon SageMaker

Agustinus Nalwan  
GM AI, Data Platform & Data Science  
Carsales  
AWS ML Hero



# About me



# Owly – a Personalised Comic Video Generator



# Why?



# Why?



# Why?



# What does Owly do?



Story title



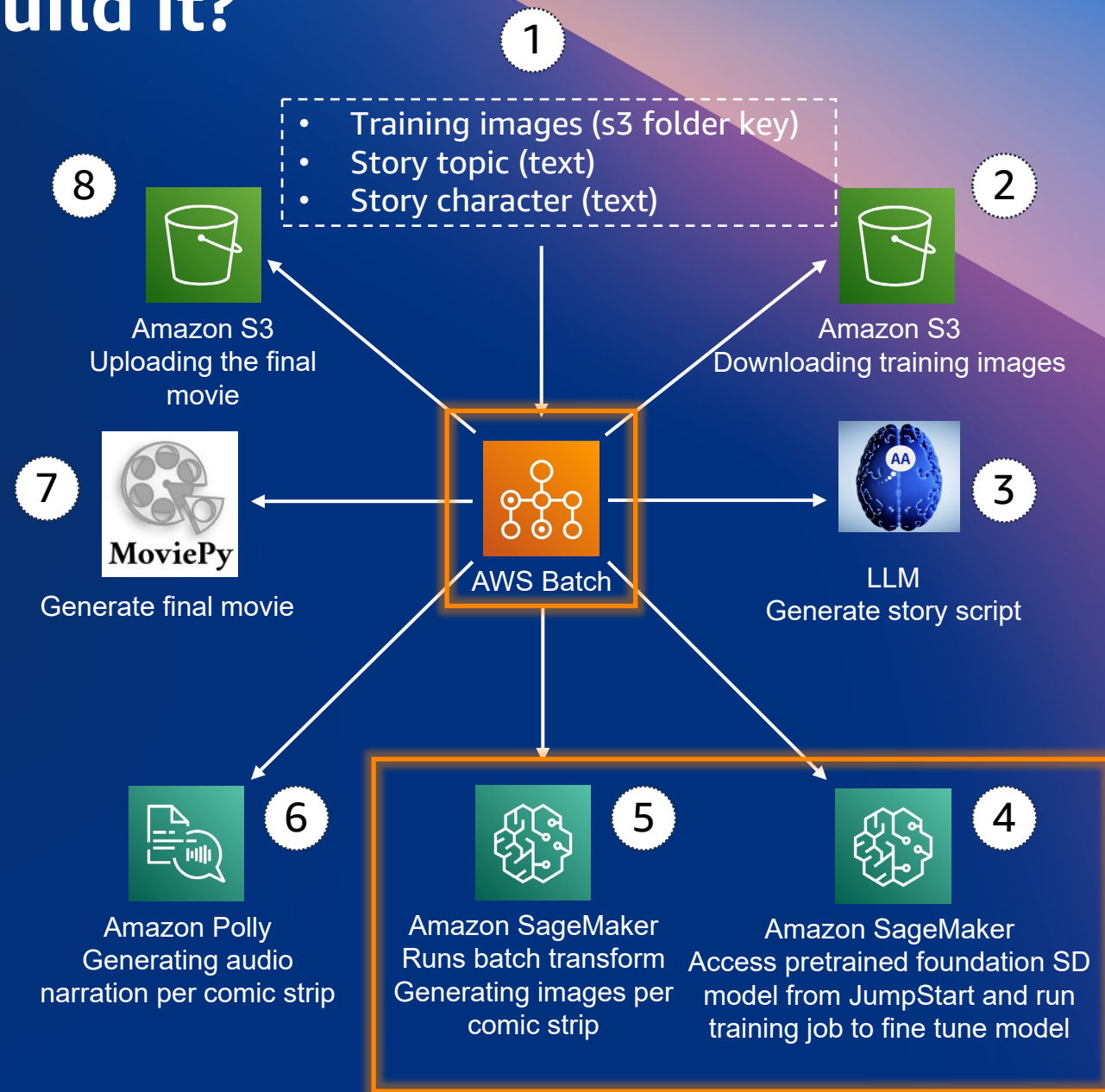
# What does Owly do?







# How did I build it?



# Stable Diffusion Model



Stable Diffusion

Captions

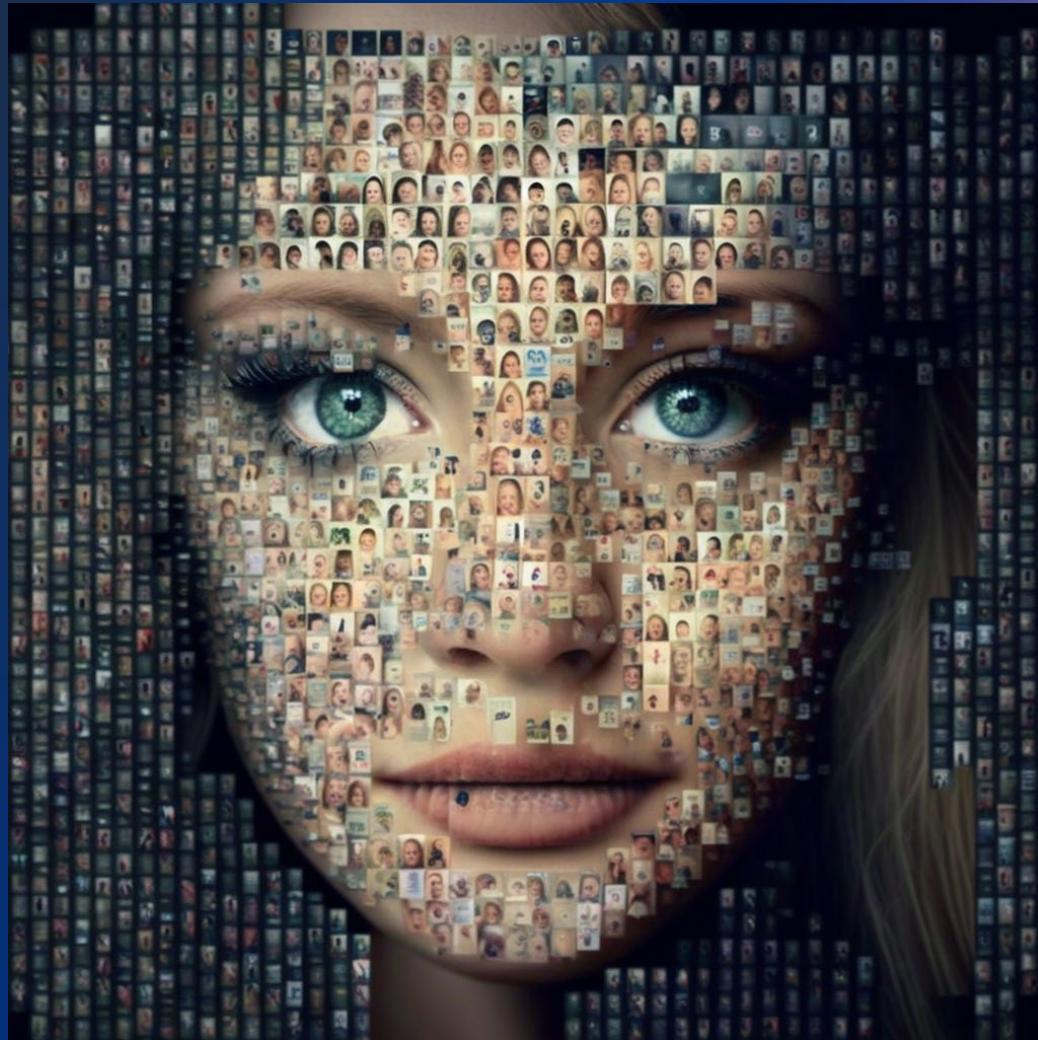
# Stable Diffusion Model

Turtle is swimming in the sea

Stable Diffusion



# Training Stable Diffusion Model



# Fine Tuning Stable Diffusion Model

Stable Diffusion 2.1  
Pretrained model

Stable Diffusion 2.1  
Fine-tuned model

Fine-tuned



# Building a fine-tuned Stable Diffusion Model

- Access a Stable Diffusion 2.1 Pre-trained Model
- Generate images
- Fine-tune the model
- Generate a lots of images with Batch Transform

Amazon SageMaker JumpStart and Amazon SageMaker Ecosystem

# Access a Stable Diffusion 2.1 Model – Amazon SageMaker JumpStart

The screenshot displays the SageMaker JumpStart interface. At the top, there's a navigation bar with 'Solutions', 'ML tasks', 'Data types', 'Notebooks', 'Frameworks', and 'Resources'. A search bar is on the right. The main content is divided into 'Foundation Models' and 'Vision Models' sections. The 'Stable Diffusion 2.1 base' model card is highlighted with a red border. It features a 'Featured' badge and a 'Text To Image' category. Other models shown include 'FLAN-T5 XXL' (Text2text Generation), 'Alexa TM 20B' (Text Generation), 'Stable Diffusion x4 upscaler...' (Text), 'Image Classification' (Swin Base 4 7 224), and 'Object Detection' (Ssd Resnet50 V1 Fpn 640x640 Coco 17).

## SageMaker JumpStart

[Show introduction](#) [Browse Shared Models](#)

[Solutions](#) [ML tasks](#) [Data types](#) [Notebooks](#) [Frameworks](#) [Resources](#)

### Foundation Models

Deploy foundation models trained on broad dataset and usable in wide range of use cases. [Explore All Foundation Models \(13\)](#)

**Stable Diffusion 2.1 base**

**Featured** **Text To Image**

Fine-tunable: Yes  
Source: **Stability AI**  
Pre-training Dataset: **LAION-5B**

[View model >](#)

**FLAN-T5 XXL**

**Text2text Generation**

Pre-training Dataset: **English Text**  
Fine-tunable: No  
Source: **Hugging Face**

[View model >](#)

**Alexa TM 20B**

**Featured** **Text Generation**

Pre-training Dataset: **Common Crawl (mC4) and...**  
Fine-tunable: No  
Source: **Alexa**

[View model >](#)

### Vision Models

Fine-tune and deploy pretrained vision models with one click. [Explore All Vision Models \(293\)](#)

**Stable Diffusion x4 upscaler...**

**Featured** **Text**

Fine-tunable: No  
Source: **Stability AI**

**Image Classification**

**Featured** **Swin Base 4 7 224**

Dataset: **ImageNet**  
Fine-tunable: Yes  
Source: **TensorFlow Hub**

**Object Detection**

**Ssd Resnet50 V1 Fpn 640x640 Coco 17**

Pre-training Dataset: **COCO '17**  
Fine-tunable: Yes  
Source: **Tensorflow**




# Access a Stable Diffusion 2.1 Model – Amazon SageMaker JumpStart

MODEL

## Stable Diffusion 2.1 base

text · text to image · foundation models · featured

Open notebook

 Browse JumpStart

Deploy Train Notebook Model details

### Deploy Model

Deploy a pretrained model to an endpoint for inference. Deploying on SageMaker hosts the model on the specified compute instance and creates an internal API endpoint. JumpStart will provide you an example notebook to access the model after it is deployed. [Learn more.](#)

> Deployment Configuration

> Security Settings

Deploy

# Load a Stable Diffusion 2

JumpStart

The screenshot shows the Amazon SageMaker Studio interface. On the left, a file browser displays a folder named 'StableDiffusion' containing an 'old' folder and a file named 'Amazon\_JumpStart\_Text\_T...'. The main area shows a code editor with Python code. A red box highlights a section of the code that defines a model dropdown menu. Below the code, a 'Select a model' dropdown menu is visible, with a red box around the selected option 'model-txt2img-stabilityai-stable-diffusion-v2-1-base'.

```
[3]: from ipywidget
from sagemake

# Retrieves all T
filter_value = "t
txt2img_model

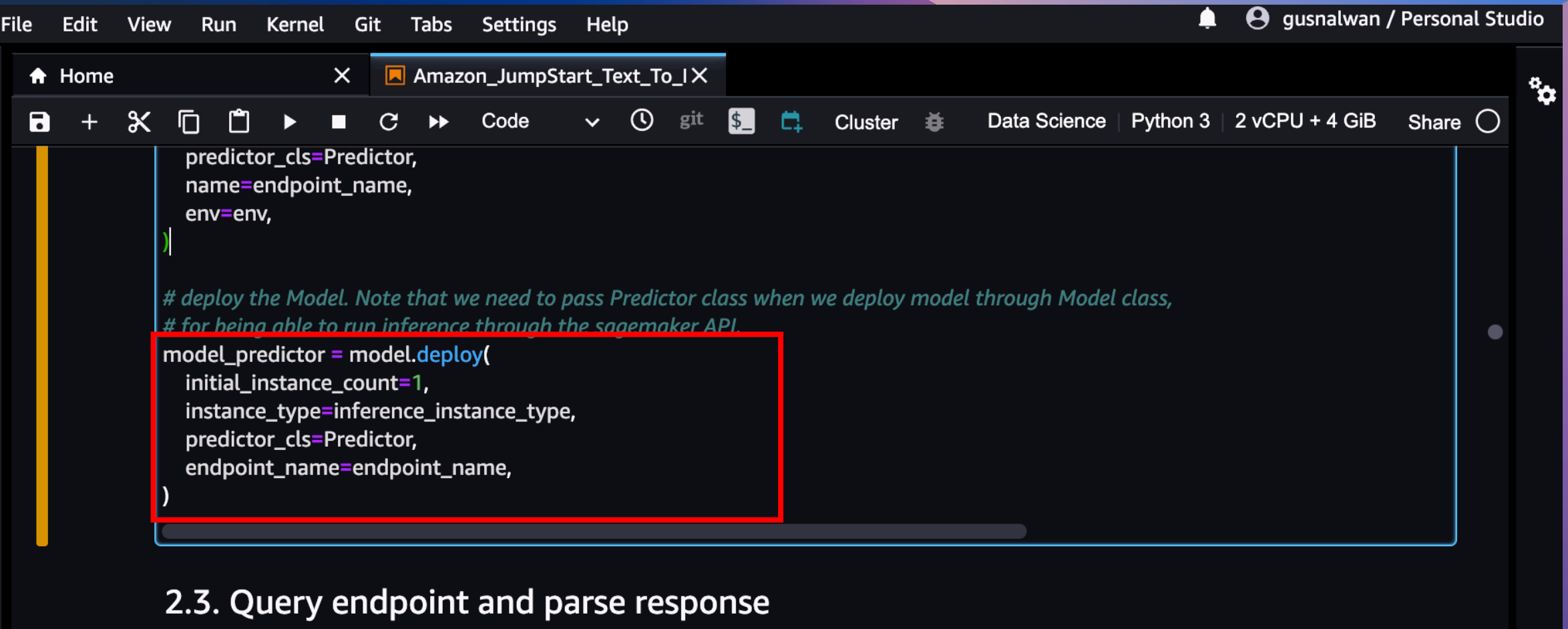
# display the mo
model_dropdov
options=txt2
value="mode
description="
style={"descr
layout={"wid
)
display(model_

Select a model

[5]: # model_versior
model_id, mod
print(model_id)
print(model_ve
```

- huggingface-txt2img-nitrososocke-redshift-diffusion
- huggingface-txt2img-nitrososocke-spider-verse-diffusion
- huggingface-txt2img-nousr-robo-diffusion
- huggingface-txt2img-ogkalu-comic-diffusion
- huggingface-txt2img-openjourney-openjourney
- huggingface-txt2img-piesposito-openpotionbottle-v2
- huggingface-txt2img-plasmo-voxel-ish
- huggingface-txt2img-plasmo-woolitize
- huggingface-txt2img-progamergov-min-illust-background-diffusion
- huggingface-txt2img-prompthero-linkedln-diffusion
- huggingface-txt2img-prompthero-openjourney
- huggingface-txt2img-qilex-magic-diffusion
- huggingface-txt2img-rabidgremlin-sd-db-epic-space-machine
- huggingface-txt2img-rayhell-popupbook-diffusion
- huggingface-txt2img-runwayml-stable-diffusion-v1-5
- huggingface-txt2img-s3nh-beksinski-style-stable-diffusion
- huggingface-txt2img-sd-dreambooth-library-original-character-cyclps
- huggingface-txt2img-sd-dreambooth-library-persona-5-shigenori-style
- huggingface-txt2img-sd-dreambooth-library-seraphm
- huggingface-txt2img-shirayu-sd-tohoku-v1
- huggingface-txt2img-thelastben-hrrzg-style-768px
- huggingface-txt2img-timothepearce-gina-the-cat
- huggingface-txt2img-trystar-clonediffusion
- huggingface-txt2img-tuwonga-dbluth
- huggingface-txt2img-tuwonga-rotoscopee
- huggingface-txt2img-volrath50-fantasy-card-diffusion
- huggingface-txt2img-yayab-sd-onepiece-diffusers4
- model-txt2img-stabilityai-stable-diffusion-v1-4
- model-txt2img-stabilityai-stable-diffusion-v1-4-fp16
- model-txt2img-stabilitvai-stable-diffusion-v2
- ✓ model-txt2img-stabilityai-stable-diffusion-v2-1-base
- model-txt2img-stabilityai-stable-diffusion-v2-fp16

# Access a Stable Diffusion 2.1 Model – Amazon SageMaker JumpStart

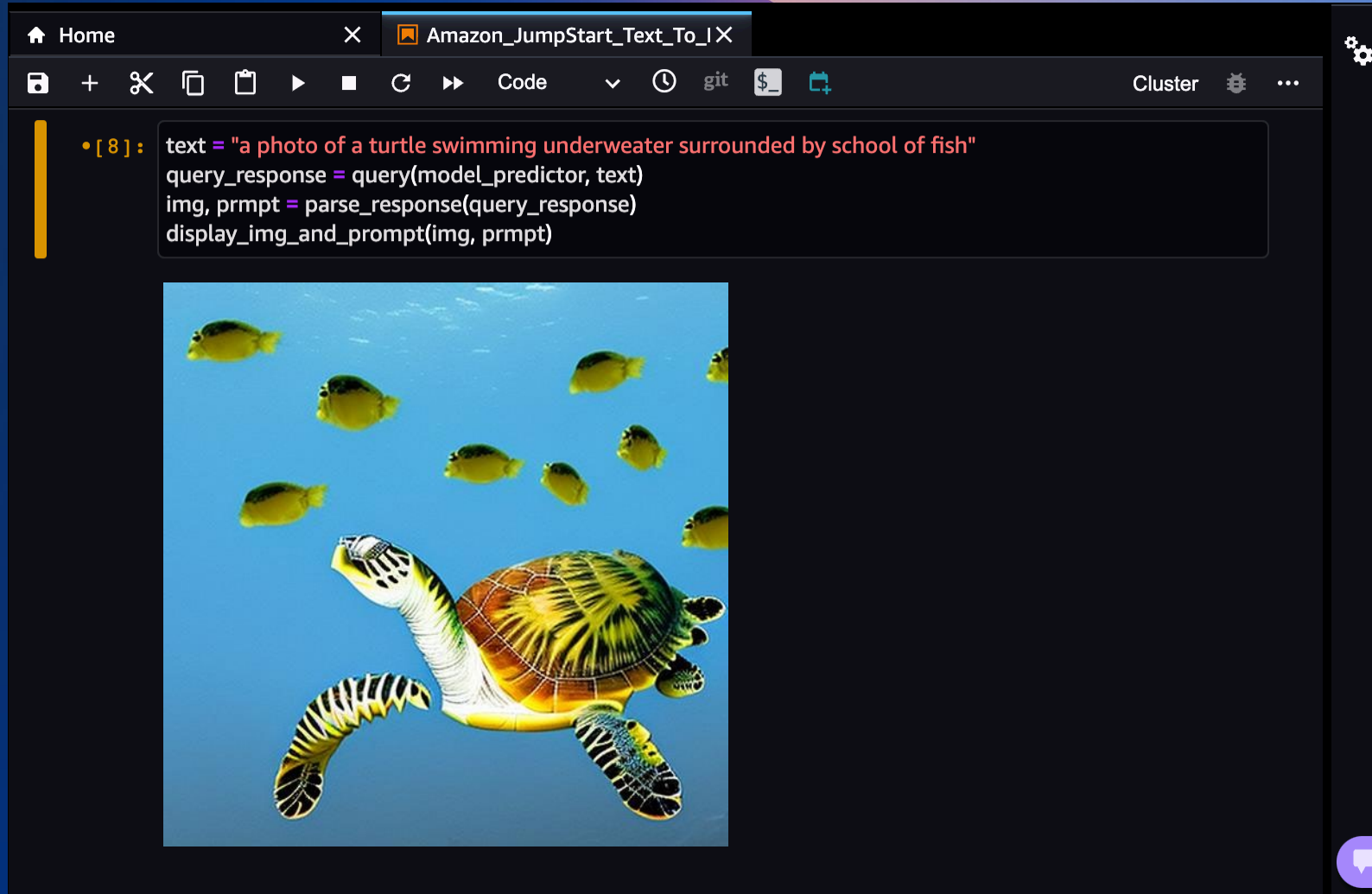


```
File Edit View Run Kernel Git Tabs Settings Help gusnalwan / Personal Studio
Home Amazon_JumpStart_Text_To_I X
+ ✂ 📄 📌 ▶ ■ ↺ ▶ Code git $ _ Cluster Data Science Python 3 2 vCPU + 4 GiB Share
predictor_cls=Predictor,
name=endpoint_name,
env=env,
)

# deploy the Model. Note that we need to pass Predictor class when we deploy model through Model class,
# for being able to run inference through the sagemaker API
model_predictor = model.deploy(
    initial_instance_count=1,
    instance_type=inference_instance_type,
    predictor_cls=Predictor,
    endpoint_name=endpoint_name,
)
```

### 2.3. Query endpoint and parse response

# Generating Images

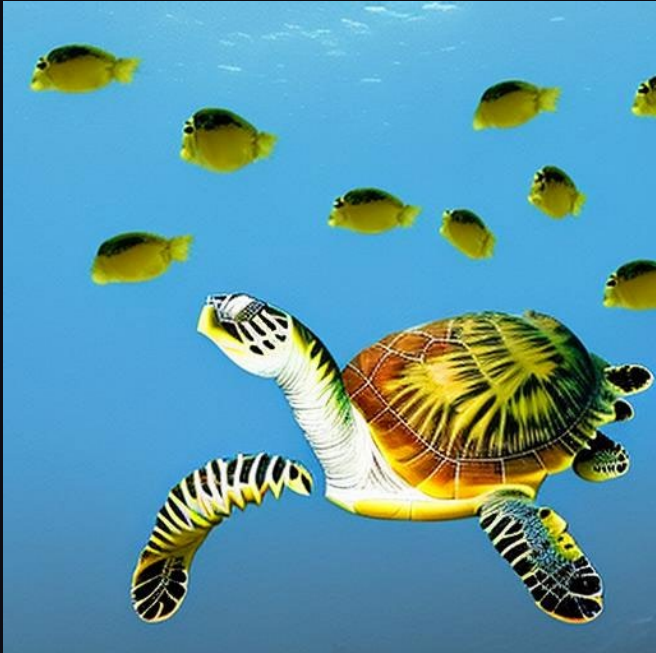


Home Amazon\_JumpStart\_Text\_To\_I X

Code git \$

Cluster

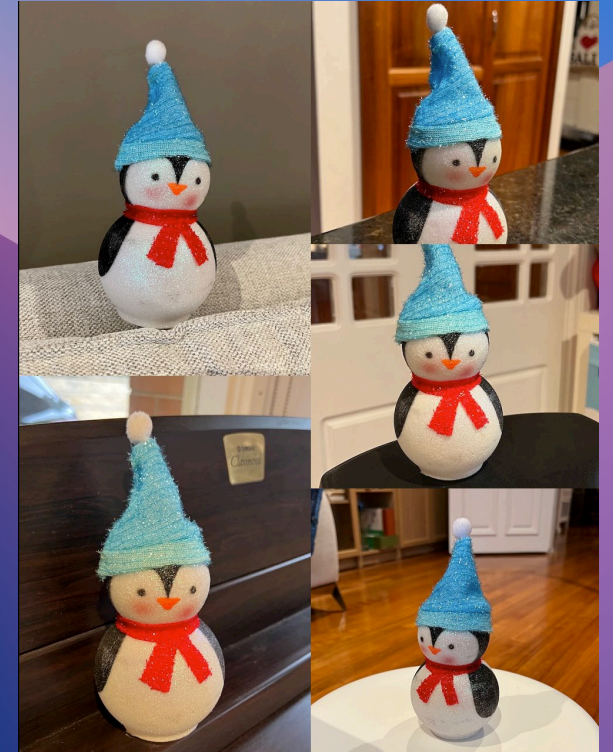
```
• [8]: text = "a photo of a turtle swimming underwater surrounded by school of fish"  
query_response = query(model_predictor, text)  
img, prmp = parse_response(query_response)  
display_img_and_prompt(img, prmp)
```



# Fine-tune Stable Diffusion – Run training on Amazon SageMaker

```
input_directory
|---instance_image_1.png
|---instance_image_2.png
|---instance_image_3.png
|---instance_image_4.png
|---instance_image_5.png
|---dataset_info.json
|---class_data_dir
|   |---class_image_1.png
|   |---class_image_2.png
|   |---class_image_3.png
|   |---class_image_4.png
```

```
{
  "instance_prompt": "a photo of xyz penguin",
  "class_prompt": "a photo of penguin"
}
```



S3 bucket

# Fine-tune Stable Diffusion – Run training on Amazon SageMaker

```
Cluster [16]: from sagemaker.estimator import Estimator
           from sagemaker.utils import name_from_base
           from sagemaker.tuner import HyperparameterTuner

           training_job_name = name_from_base(f"stable-diffusion-{train_model_id}-transfer-learning")

           # Create SageMaker Estimator instance
           sd_estimator = Estimator(
               role=aws_role,
               image_uri=train_image_uri,
               source_dir=train_source_uri,
               model_uri=train_model_uri,
               entry_point="transfer_learning.py", # Entry-point file in source_dir and present in train_source_uri.
               instance_count=1,
               instance_type=training_instance_type,
               max_run=360000,
               hyperparameters=hyperparameters,
               output_path=s3_output_location,
               base_job_name=training_job_name,
           )

           # Launch a SageMaker Training job by passing s3 path of the training data
           sd_estimator.fit({"training": training_dataset_s3_path}, logs=True)

           ...

           [17]: training_job_name = sd_estimator.latest_training_job.name
                 training_job_name

           [17]: 'stable-diffusion-model-txt2img-stabilit-2023-03-13-01-22-14-342'
```

# Fine-tune Stable Diffusion – Run training on Amazon SageMaker

```
inference_instance_type = "ml.g4dn.2xlarge"

# Retrieve the inference docker container uri
deploy_image_uri = image_uris.retrieve(
    region=None,
    framework=None, # automatically inferred from model_id
    image_scope="inference",
    model_id=train_model_id,
    model_version=train_model_version,
    instance_type=inference_instance_type,
)

# Retrieve the inference script uri. This includes scripts for model loading, inference handling etc.
deploy_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version, script_scope="inference"
)

endpoint_name = name_from_base(f"generator-FT-{train_model_id}")

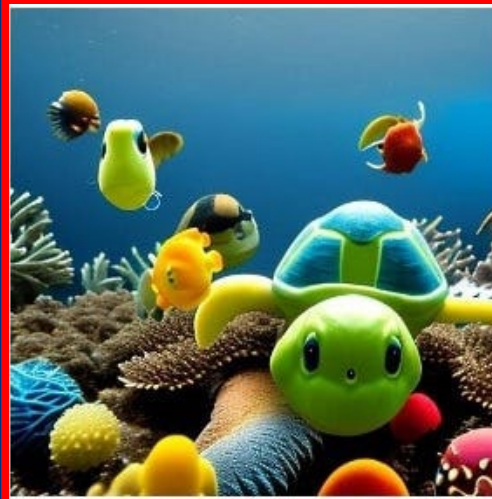
# Use the estimator from the previous step to deploy to a SageMaker endpoint
finetuned_predictor = sd_estimator.deploy(
    initial_instance_count=1,
    instance_type=inference_instance_type,
    entry_point="inference.py", # entry point file in source_dir and present in deploy_source_uri
    image_uri=deploy_image_uri,
    source_dir=deploy_source_uri,
    endpoint_name=endpoint_name,
)
```

# Fine-tune Stable Diffusion – Run training on Amazon SageMaker



Nick the kangaroo is sitting next to the lake

Rex the turtle is swimming around school of fish





# Generating images – SageMaker Batch Transform



S3 bucket

```
{ "prompt": "a photo of Bob the penguin in Antarctica" }
```

```
{ "prompt": "a photo of Bob the penguin in the jungle" }
```

```
{ "prompt": "a photo of an elephant and a tiger" }
```

# Generating images – SageMaker Batch Transform

```

Edit View Run Kernel Git Tabs Settings Help gusnalwan / Personal Studio
Amazon_JumpStart_Text_To_I
Code $ git Cluster Data Science Python 3 2 vCPU + 4 GiB ...

[ 21 ]: import sagemaker
        from sagemaker.estimator import Estimator
        from sagemaker import image_uris, model_uris, script_uris, hyperparameters
        from sagemaker import model
        from sagemaker.predictor import Predictor
        from sagemaker.utils import name_from_base

        # Get model artifact location by estimator.model_data, or give an S3 key directly
        model_artifact_s3_location = f"s3://comgen/output-model/test-101/{training_job_name}/output/model.tar.gz"

        env = {
            "MMS_MAX_RESPONSE_SIZE": "20000000",
        }

        # Create model from saved model artifact
        model = model.Model(
            model_data=model_artifact_s3_location,
            role=aws_role,
            entry_point="inference.py", # entry point file in source_dir and present in deploy_source_uri
            image_uri=deploy_image_uri,
            source_dir=deploy_source_uri,
            env=env
        )

[ 22 ]: transformer = model.transformer(instance_count=1, instance_type="ml.p3.2xlarge",
                                       output_path="s3://comgen/processing/test-101/output-images",
                                       accept='application/json')
        transformer.transform(data="s3://comgen/processing/test-101/batch_transform_input/", content_type='application/json')












```

# Generating images – SageMaker Batch Transform

## Objects (210)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

< 1 >

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	 <a href="#">story-0-0-0.txt.out</a>	out	April 11, 2023, 21:39:29 (UTC+10:00)	3.9 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-1.txt.out</a>	out	April 11, 2023, 21:39:36 (UTC+10:00)	3.8 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-2.txt.out</a>	out	April 11, 2023, 21:39:43 (UTC+10:00)	3.9 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-3.txt.out</a>	out	April 11, 2023, 21:39:51 (UTC+10:00)	4.0 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-4.txt.out</a>	out	April 11, 2023, 21:39:58 (UTC+10:00)	3.9 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-5.txt.out</a>	out	April 11, 2023, 21:40:05 (UTC+10:00)	4.0 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-6.txt.out</a>	out	April 11, 2023, 21:40:13 (UTC+10:00)	3.8 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-7.txt.out</a>	out	April 11, 2023, 21:40:20 (UTC+10:00)	3.9 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-8.txt.out</a>	out	April 11, 2023, 21:40:28 (UTC+10:00)	3.7 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-0-9.txt.out</a>	out	April 11, 2023, 21:40:35 (UTC+10:00)	4.0 MB	Standard
<input type="checkbox"/>	 <a href="#">story-0-1-0.txt.out</a>	out	April 11, 2023, 21:40:42 (UTC+10:00)	3.9 MB	Standard

# Generating images – SageMaker Batch Transform

```

[21]: import sagemaker
      from sagemaker.estimator import Estimator
      from sagemaker import image_uris, model_uris, script_uris, hyperparameters
      from sagemaker import model
      from sagemaker.predictor import Predictor
      from sagemaker.utils import name_from_base

      # Get model artifact location by estimator.model_data, or give an S3 key directly
      model_artifact_s3_location = f"s3://comgen/output-model/test-101/{training_job_name}/output/model.tar.gz"

      env = {
          "MMS_MAX_RESPONSE_SIZE": "20000000",
      }

      # Create model from saved model artifact
      model = model.Model(
          model_data=model_artifact_s3_location,
          role=aws_role,
          entry_point="inference.py", # entry point file in source_dir and present in deploy_source_uri
          image_uri=deploy_image_uri,
          source_dir=deploy_source_uri,
          env=env
      )

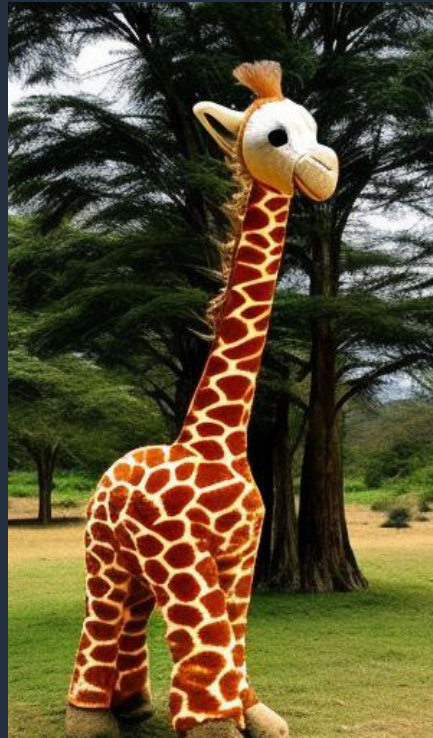
[22]: transformer = model.transformer(instance_count=1, instance_type="ml.p3.2xlarge",
      output_path="s3://comgen/processing/test-101/output-images",
      accept="application/json")
      transformer.transform(data="s3://comgen/processing/test-101/batch_transform_input/", content_type="application/json")

```



AWS Batch

# Generated Stories



<https://shorturl.at/stACV>



# Thank you!

Agustinus Nalwan

<https://www.linkedin.com/in/agustinus-nalwan>

<https://medium.com/@agustinus-nalwan>

<https://github.com/msubzero2000>