Introduction to Cloud Computing Project

Creative Imagery: Text-To-Image Generation

Prof: Raja Loganantharaj

By-

Group 6

Shailendra Nutakki - 2211269

Meher Gajula - 2217304

Alexa De La Garza – 1584551

Deekshitha Teegala - 2247690

Pradeep Reddy Mallepally- 2200020

Contents :

ABSTRACT:

This project delves into the impressive capabilities of the Stable Diffusion XL (SDXL) model, integrated within Amazon Bedrock, with a primary emphasis on text-to-image generation. The innovative utilization of text prompts, in combination with style presets and negative prompts, allows for the creation of captivating and diverse images. This technology opens the door to a wide array of creative applications, spanning from the generation of artistic compositions to the production of visually engaging product images.

Through the utilization of AWS infrastructure and the Bedrock API, this project highlights the seamless fusion of cutting-edge AI models into image-related tasks. It serves as a testament to the harmonious blend of technology and artistic expression, offering boundless possibilities for the generation and manipulation of images.

INTRODUCTION:

Crafting effective prompts for image generation is a nuanced art, often requiring a blend of precision and creativity. Predicting the outcome of a given prompt with a specific model can be challenging, but by breaking prompts into three key components, we can enhance our control over the creative process.

The first element, the type of image, sets the foundation. It can range from a "photograph" to a "sketch" or a "painting," allowing us to define the visual medium. The second component, the description of content, specifies what the image should portray—be it a subject, object, environment, or scene. The inclusion of adjectives and rich details adds depth to the creative process. The third facet, the style of the image, enables us to guide the aesthetics. Whether "realistic," "artistic," or inspired by a specific artist, this component provides a crucial layer of artistic direction.

By skillfully manipulating these elements, we can generate an array of images, each with its unique flair. It's worth noting that even with the same prompt, the model produces diverse results, offering the flexibility to select the most suitable image for any application. This project explores the

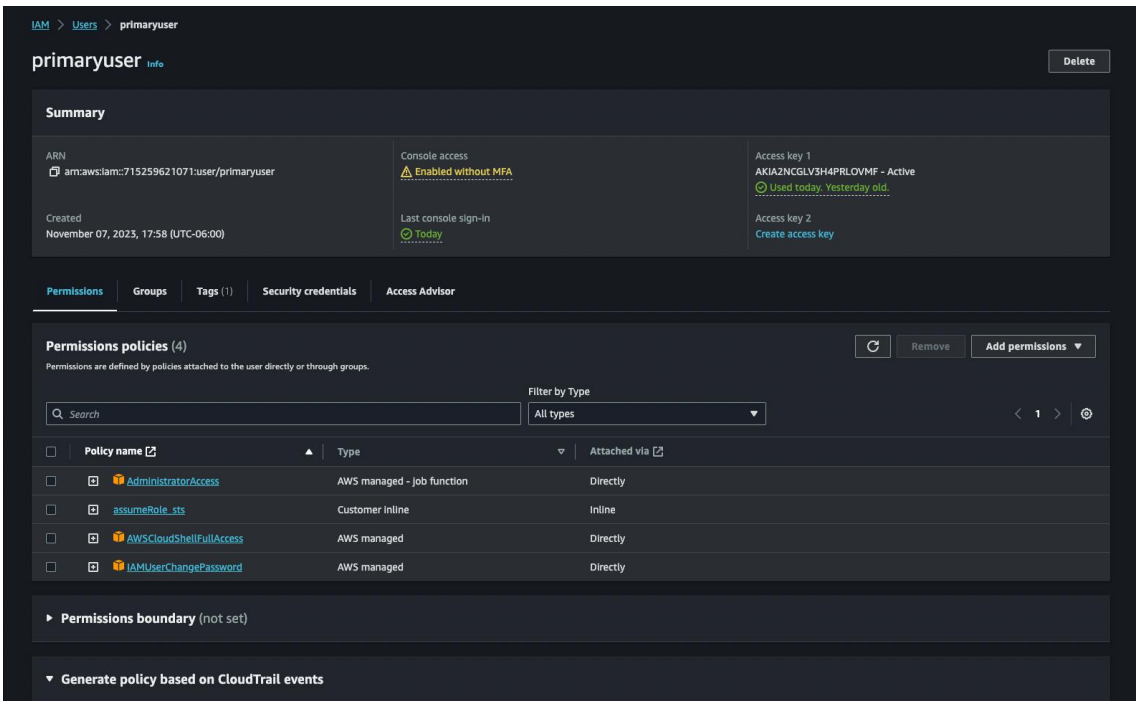
intricacies of prompt engineering within the context of image generation, unlocking the potential for a wide range of creative possibilities.

SETUP:

AWS Configuration:

IAM:

To successfully utilize the Amazon Bedrock service within your notebook environment on AWS, it is imperative that the AWS identity assumed from your notebook environment has the requisite IAM (Identity and Access Management) permissions to invoke the Bedrock service. Whether it's the Studio/notebook Execution Role from SageMaker or a role/IAM User for self-managed notebooks, proper permissions are crucial.



It's essential to note that, particularly with Amazon SageMaker, the notebook execution role may be distinct from the user or role used to log in to the AWS Console. If you intend to explore the AWS Console for Amazon Bedrock, ensure permissions are granted to your Console user/role as well.

METHODOLGY:

Models and Components:

The CLIP text encoder: Encodes text prompts.

The VAE decoder: Decodes encoded text prompts into images.

The UNet: A neural network architecture for image generation.

The VAE_post_quant_conv: Part of the image generation process.

Text-to-Image Generation:

Images are generated by providing a text prompt that describes what the image should depict.SDXL allows the specification of style presets to influence image generation. Clip guidance presets and samplers are used to enhance the quality and desirability of generated images.

Resolution and Aspect Ratio:

The SDXL model can generate images at various resolutions and non-squared aspect ratios, not limited to the default 512px x 512px square resolution.

Negative Prompts:

To influence image generation by avoiding specific content or style choices, negative prompts with corresponding weights are used. These negative prompts help steer the model away from undesired results.

Invoke Model with Amazon Bedrock:

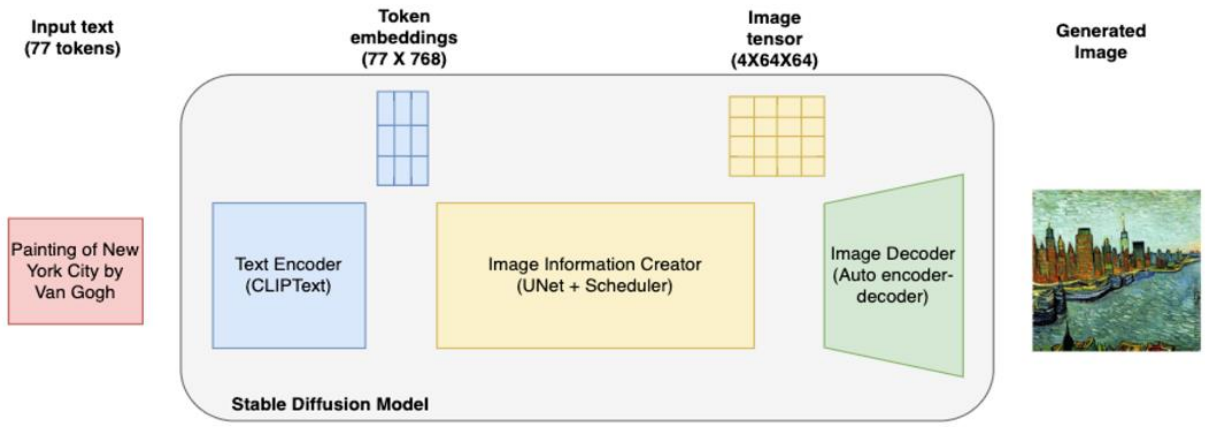
The Amazon Bedrock Invoke Model is employed to access the SDXL model.

It takes various input parameters, including text prompts, negative prompts, scale, seed, steps, style presets, clip guidance presets, sampler, and width.

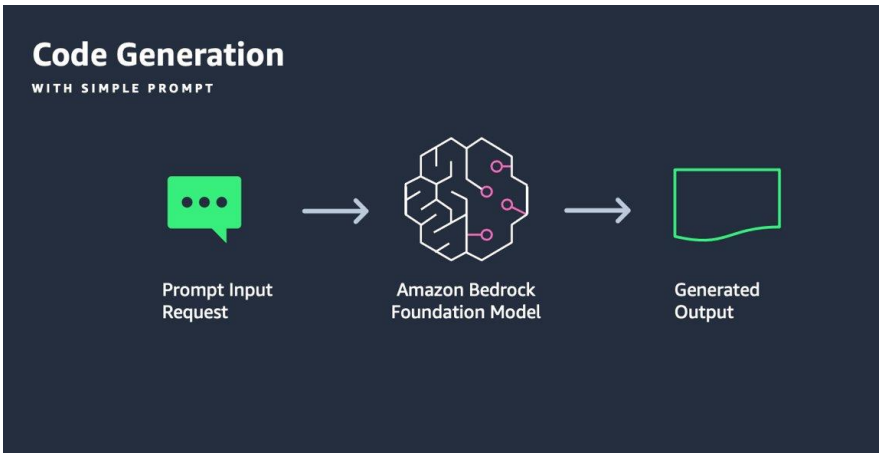
Output Processing:

The code decodes the Base64 string returned by the SDXL model to binary.

It utilizes an image processing library like Pillow to load and display the generated image within the notebook.



MODEL BUILDING:



The architecture diagram presented in the PowerPoint highlights the straightforward yet effective implementation of our creative imagery project. This implementation leverages the formidable capabilities of the AWS Bedrock API through the utilization of Boto 3, the AWS SDK version for Python, seamlessly integrated within our architectural framework. Through this integration, we can communicate with the AWS Bedrock API by formulating a text prompt that defines our desired output image.

Upon initiating the request, we set in motion the bedrock stable diffusion model. This model processes the provided textual input and translates it into a one-of-a-kind generated image. This symbiotic interaction between AWS Bedrock and the stable diffusion model empowers us to yield innovative and visually captivating results from text-based prompts. With this live demonstration of IM user implementation and the setup of the Boto 3 Client connection. This final step will illustrate the seamless execution of the project's capabilities in real-time.

RESULTS:

In this live demonstration, we have showcased the configuration and operation of our model. we emphasized the importance of setting up a user to establish connections with Amazon client services and the Bedrock API. This user was endowed with both admin access and a custom role, ensuring comprehensive access to AWS Services

The process of connecting to the Stability XL AI Diffusion Model through the Amazon Boto 3 client. We highlighted the critical role of storing credentials in a secure JSON file for authentication. Once loaded, we successfully initiated the Bedrock client, paving the way for further interactions.

Next, we conducted a comprehensive review of the available models, focusing specifically on the Text-to-Image capability of the Stability AI Diffusion Model. Through a practical example, we demonstrated the input and output process, showcasing the seamless flow of API requests and responses.

The culmination of our efforts resulted in the successful generation of an image based on the provided prompt, which depicted a landscape with trees and mountains. The image was securely encoded for transmission, ensuring a safe exchange of data.

Stability Stable Diffusion XL

In [6]:
```
prompt_data = "a landscape with trees and mountains"
body = json.dumps({
    "text_prompts": [{"text": prompt_data}],
    "cfg_scale": 10,
    "seed": 20,
    "steps": 50
})
modelId = "stability.stable-diffusion-xl"
accept = "application/json"
contentType = "application/json"

try:

    response = bedrock_runtime.invoke_model(
        body=body, modelId=modelId, accept=accept, contentType=contentType
    )
    response_body = json.loads(response.get("body").read())

    print(response_body["result"])
    print(f'{response_body.get("artifacts")[0].get("base64")[0:80]}...')

except botocore.exceptions.ClientError as error:

    if error.response['Error']['Code'] == 'AccessDeniedException':
           print(f"\x1b[41m{error.response['Error']['Message']}\
                \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")

    else:
        raise error
```

```
success
iVBORw0KGgoAAAANSUhEUgAAAgAAAAIACAIAAAB7GkOtAAABemVYSWZNTQAqAAAACAAGAQAABAAAAAEA...
```

```
image = Image.open(io.BytesIO(base64.decodebytes(bytes(base_64_img_str, "utf-8"))))
```

```
In [7]: import base64
        import io
        from PIL import Image

        base_64_img_str = response_body.get("artifacts")[0].get("base64")
        image = Image.open(io.BytesIO(base64.decodebytes(bytes(base_64_img_str, "utf-8"))))
        image
```

Out[7]:

In summary, this live demo exemplifies the effective integration of IBM configurations, AWS services, and the Bedrock API, culminating in the generation of visually compelling images through the Stability AI Diffusion Model. This achievement underscores the innovative potential of our project.

CONCLUSION:

In conclusion, our project centered on exploring the capabilities of the SDXL model within Amazon Bedrock, with a specific focus on text-to-image generation. What sets our endeavor apart is its remarkable proficiency in transforming text prompts into visually captivating images. Through the incorporation of style presets and the strategic use of negative prompts, we go beyond mere image generation; we actively mold outcomes, unlocking a realm of creative potential.

This application exhibits versatility, whether it involves translating words into artistic expressions or producing compelling product imagery. Nevertheless, what truly distinguishes our project is the seamless integration of AWS infrastructure and the Bedrock API. This collaboration underscores the powerful constructive collaboration between innovative AI models and image-related tasks.

Our presentation guides you through a journey into the realm of image generation and manipulation, demonstrating how this fusion expands the horizons of creativity. This venture not only highlights the potential of modern technology but also underscores the transformative impact it can have on various creative fields.

REFERENCES : Amazon Bedrock, ChatGPT, Github