

re:Invent2024体験記

Amazonの最新モデルを使ってみよう

(Amazon Nova Canvas / Reelと時々Proのデモ)

Yu Otsubo

Yu Otsubo (tubone24)



↑ 名刺



特徴

KDDIアジャイル開発センター

ゆるふわエンジニア

(@meitante1conan)

犬と遊んでいる



Rerank

API マーケットプレイス

蒸留

プロンプトキャッチャー

Amazon

Nova

マルチエージェント

プロンプトルール・テイキング

カスタムコネクタ

GenAI

Index

kendra

GraphRAG

BOA

↳ インテリジェント最適化

Rerank

APIマーケットプレイス

蒸留

コンテキストキャッチ

たくさん

a マルチエージェント

ありますね

プロ

カスタムコ

kendra GenAI

Index

BOA

最適化

GraphRAG

やっぱり気になるますよね？

やっぱり気になるますよね？

 Nova 

Amazon NovaはAWSが2024年12月3日に発表した新しい基盤モデル

精度などはこれから見ていくとして...

新しいモデルが出るとワクワクしますよね。



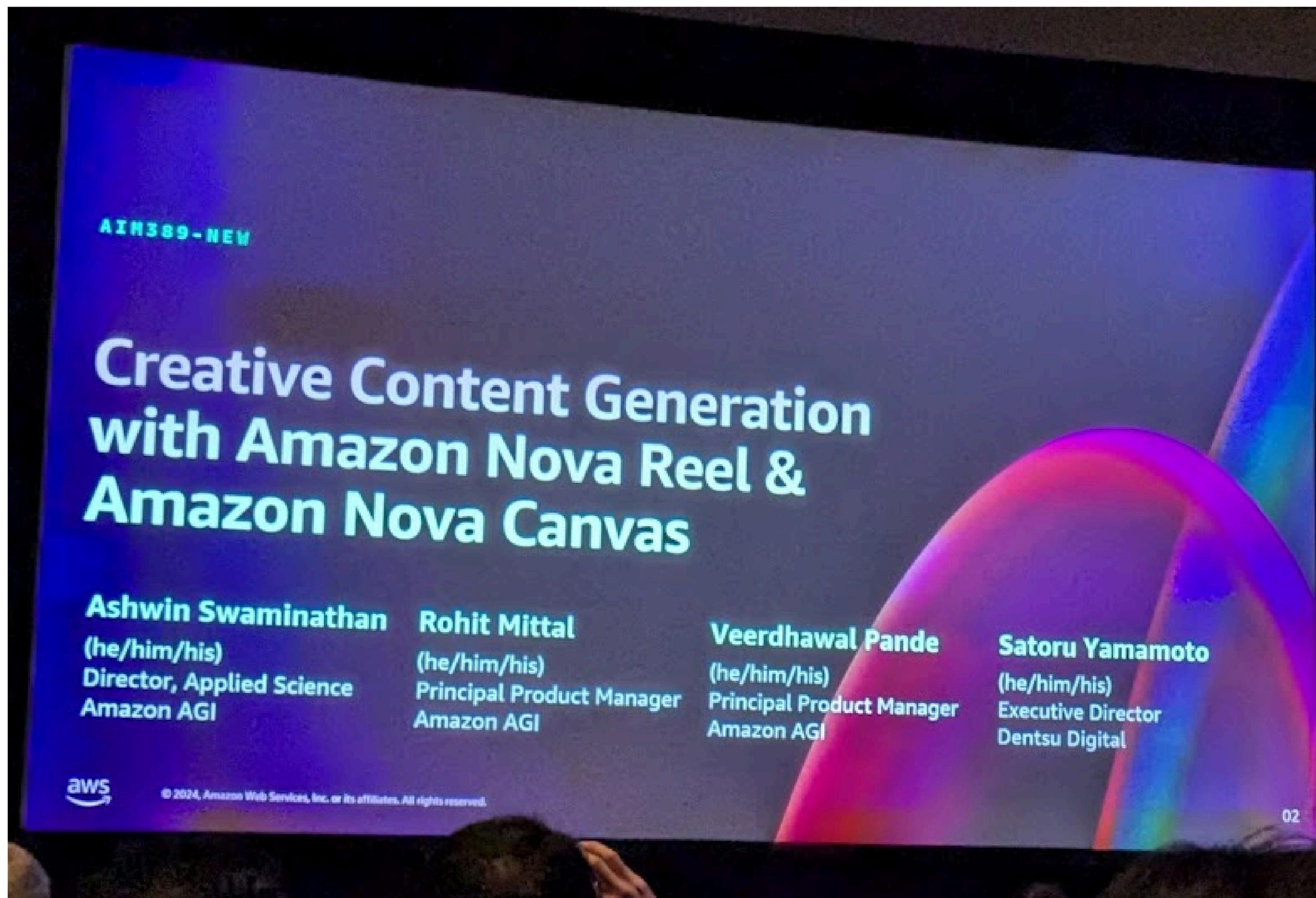
Amazon Nova CanvasとAmazon Nova Reelについて

もちろんUnderstanding Modelsも気になりますが...

やっぱり気になる2つ
(Creative Content Generation Models)

でもどんな特徴や使い所があるんだろう...

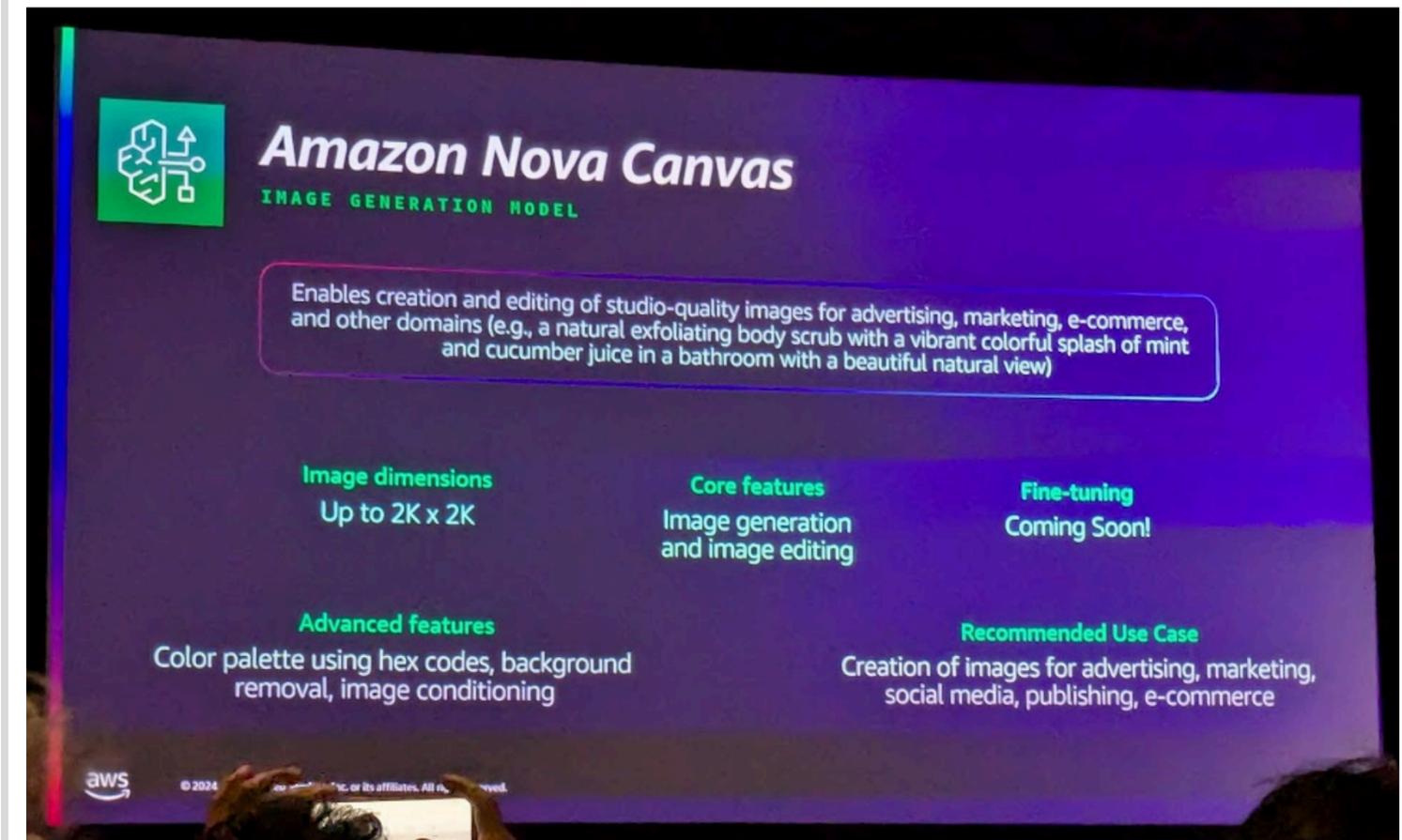




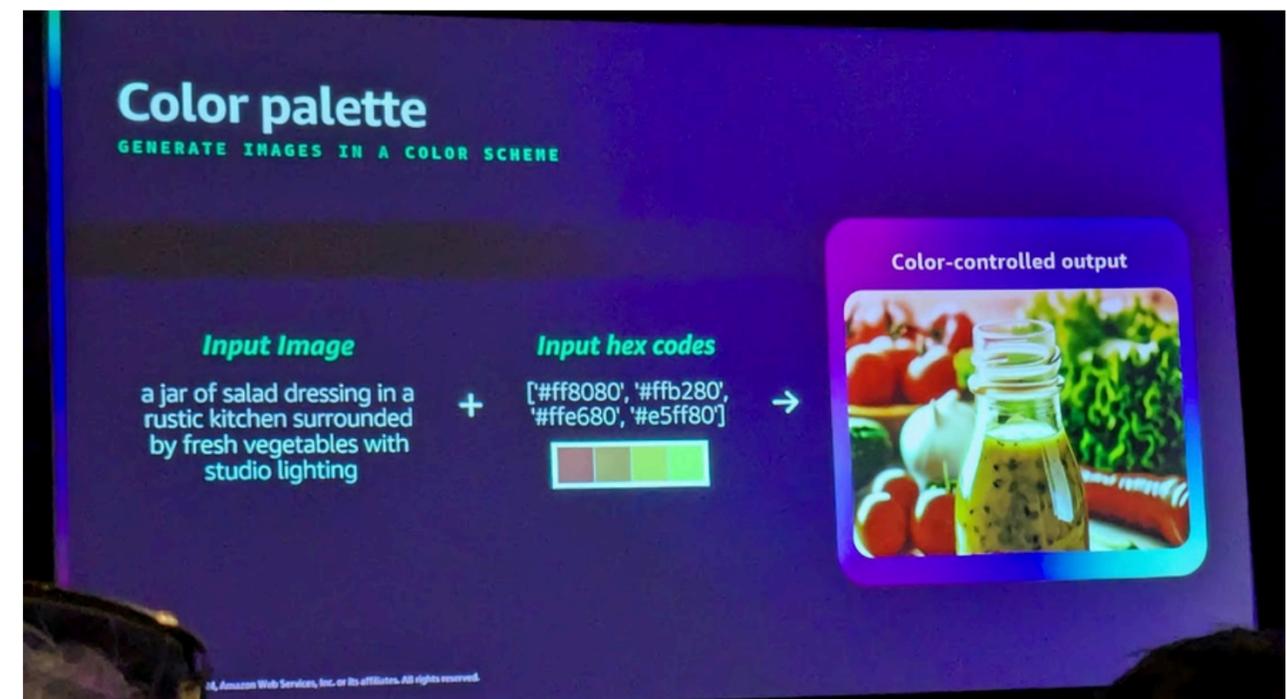
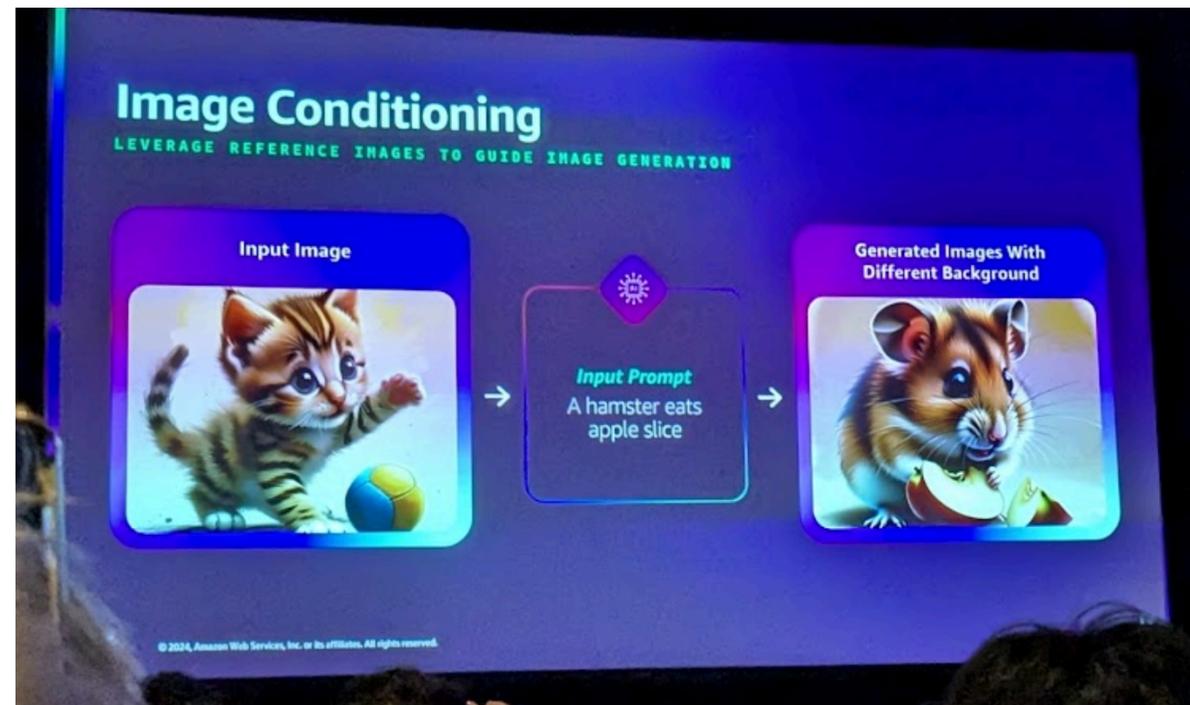
いわゆるTextToImageだけでなく、
高度な画像編集機能も具備

広告、マーケティング、eコマース、ソーシャル
メディア、出版など、商用目的の画像制作に！

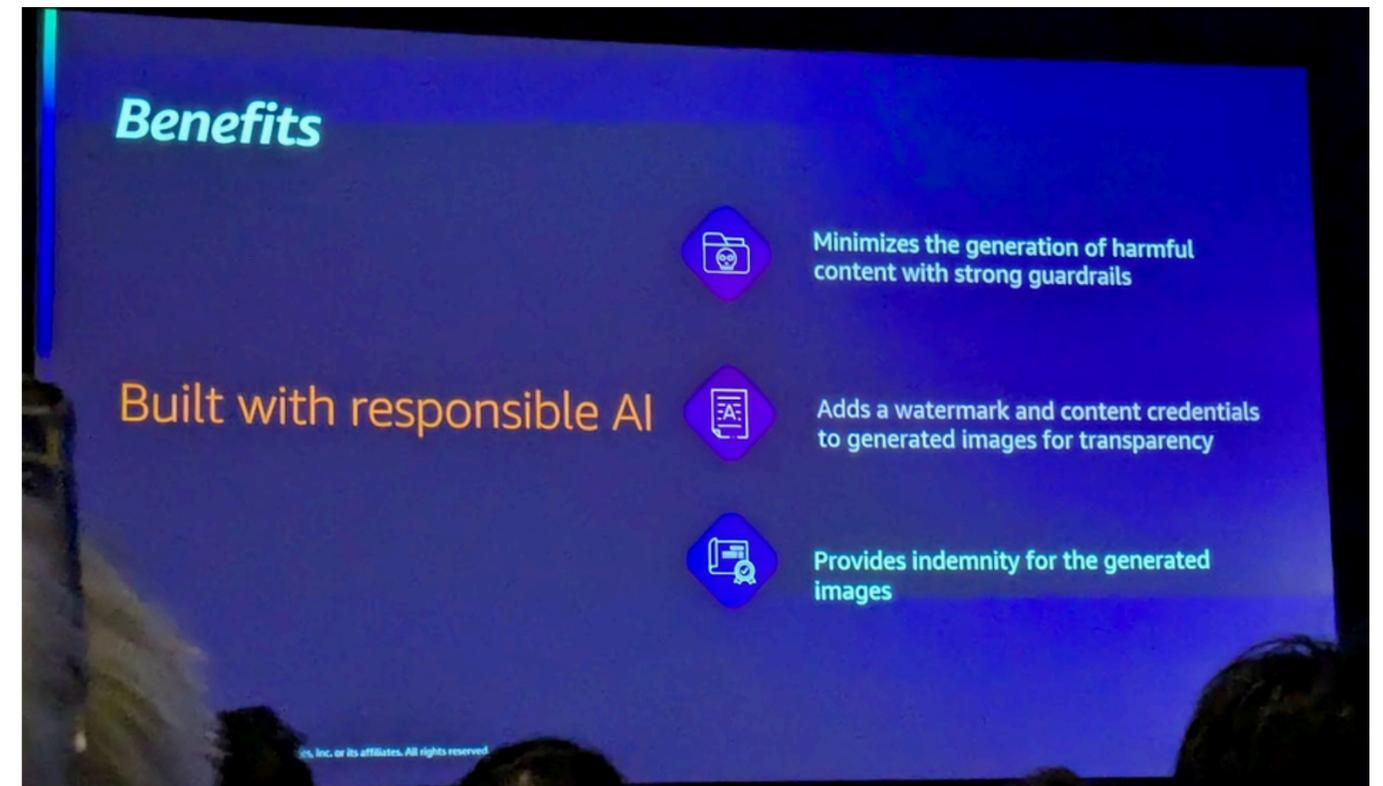
今のところTextToImageは英語のみ？



Amazon Nova Canvas



他のBedrockモデルと同じように
「責任あるAI」を謳っている。





きわどい文言での生成は
ValidationExceptionになる

元の画像 ⇄

画像生成プロンプト

poop: [redacted]

画像を生成

エラーが発生しました: An error occurred (ValidationException) when calling the InvokeModel operation: This request has been blocked by our content filters. Please adjust your text prompt to submit a new request.

現状はテキストor画像から6秒の動画が作成可能。

生成には5分ほどかかります。

こちらにも広告、マーケティング、eコマース、ソーシャルメディア、出版など、商用目的の動画素材制作に！





Pasta City, created with Amazon Nova Reel by Amazon Ads



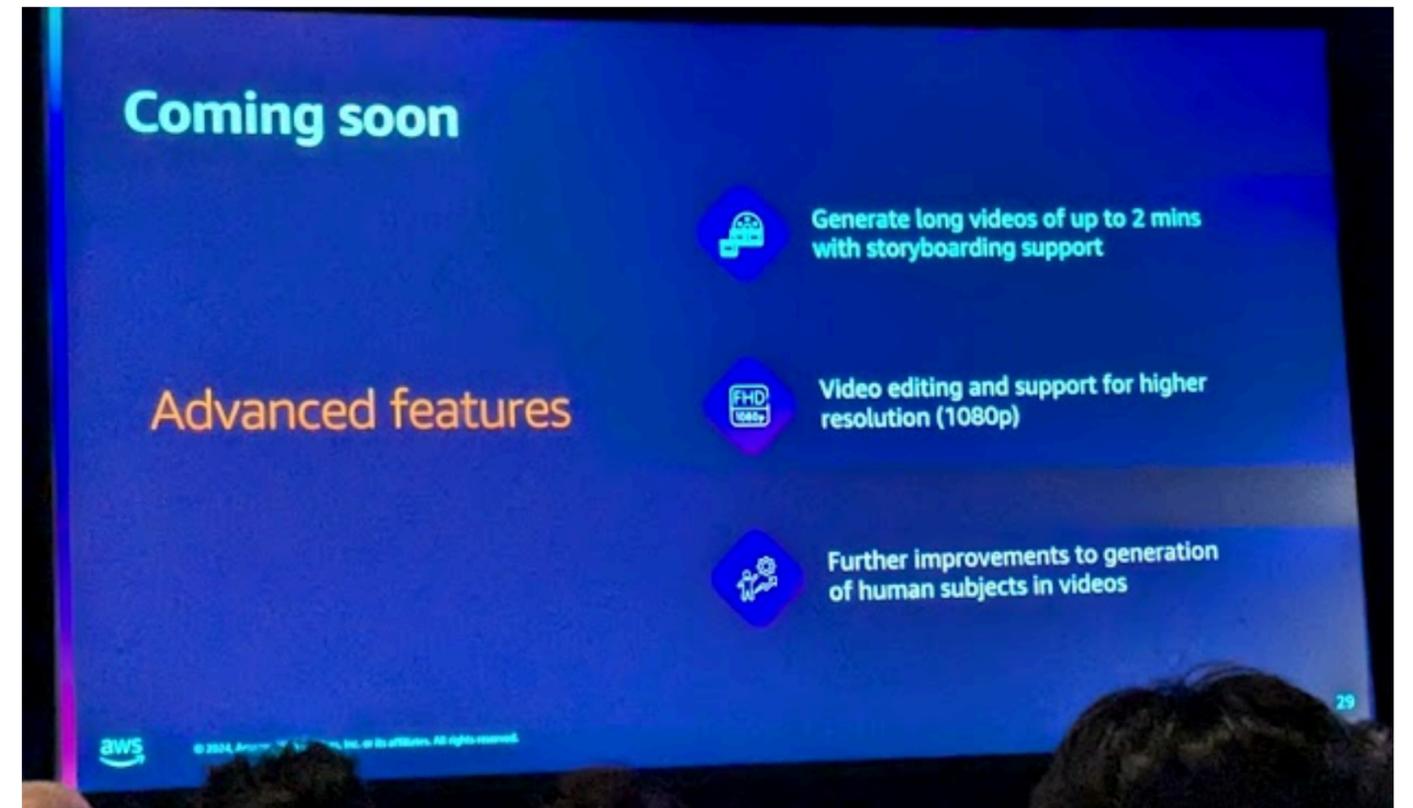
Share



Watch on  YouTube

<https://www.youtube.com/watch?v=g87NaWcszq0>

今後は2分までに拡大するほか、
解像度も上がるらしい。



ちよっと
飽きませんでしたよね

実際に使っ てみましよう！

Novaモデルを駆使して一連の動きをやってみましょう！



tubone24/**bedrock-reel-canvas-test**



1 Contributor 0 Issues 0 Stars 0 Forks



Star me!





```
def translate_and_enhance_prompt(prompt):
    client = boto3.client(
        "bedrock-runtime",
        region_name=REGION
    )

    body = [
        {
            "role": "user",
            "content": [{"text": f"以下の文言は生成AIの画像生成のText-Imageのプロンプトです。"}
            f"画像が生成しやすいように英語に翻訳の上、具体性がないものは具体的に描写できるように文言を足してください。"
            f"また、出力は英語文そのもののみ出力してください。枕詞は不要です。"
            "\n\n{prompt}"}]

    try:
        response = client.converse(
            modelId=PRO_MODEL_ID,
            messages=body,
        )
        translated_prompt = response['output']['message']['content'][0]['text']
        return translated_prompt
    except Exception as e:
        st.error(f"翻訳エラー: {str(e)}")
        return prompt
```



英訳と
詳細化



```
image_prompt = st.text_input("画像生成プロンプト")
if st.button("画像を生成"):
    image_prompt = translate_and_enhance_prompt(image_prompt)
    st.write(f"生成AIの画像生成のText-Imageのプロンプト")
    st.write(image_prompt)

canvas_payload = {
    "taskType": "TEXT_IMAGE",
    "textToImageParams": {
        "text": image_prompt,
        "negativeText": "bad quality, low res",
    },
    "imageGenerationConfig": {
        "numberOfImages": 1,
        "quality": "standard",
        "height": 720,
        "width": 1280,
        "seed": random.randint(0, 2147483648)
    }
}

try:
    canvas_response = bedrock_runtime.invoke_model(
        modelId='amazon.nova-canvas-v1:0',
        body=json.dumps(canvas_payload)
    )
    response_body = json.loads(canvas_response.get('body').read())
```

taskType=TEXT_IMAGE

Base64で
画像が返ってきます。





```
def process_image_inpainting_main_object(base_image, additional_prompt, base_prompt):
    canvas_payload = {
        "taskType": "INPAINTING",
        "inPaintingParams": {
            "text": additional_prompt,
            "negativeText": "bad quality, low res",
            "image": base_image,
            "maskPrompt": base_prompt,
        },
        "imageGenerationConfig": {
            "numberOfImages": 1,
            "height": 720,
            "width": 1280,
        }
    }
    try:
        canvas_response = bedrock_runtime.invoke_model(
            modelId=CANVAS_MODEL_ID,
            body=json.dumps(canvas_payload)
        )
        response_body = json.loads(canvas_response.get('body').read())
        if 'images' in response_body:
            st.session_state.image_data = response_body['images'][0]
            st.session_state.image = base64.b64decode(st.session_state.image_data)
            st.session_state.display_image = True
            st.session_state.image_data2 = base64.b64encode(st.session_state.image).decode("utf-8")
            st.rerun()
    except Exception as e:
        st.error(f"エラーが発生しました: {str(e)}")
```

taskType=INPAINTING

Base64で元画像を指定し、
柴犬に置き換えてみます。

置き換え対象を
マスク画像かプロンプトで
指定します。





```
def generate_video_with_nova_reel(prompt, input_image_base64, duration=6, fps=24, dimension="1280x720"):
    bedrock_runtime = boto3.client("bedrock-runtime", region_name=REGION)
    model_input = {
        "taskType": "TEXT_VIDEO",
        "textToVideoParams": {
            "text": prompt,
            "images": [{
                "format": "png",
                "source": {
                    "bytes": input_image_base64
                }
            }]
        },
        "videoGenerationConfig": {
            "durationSeconds": duration,
            "fps": fps,
            "dimension": dimension,
            "seed": random.randint(0, 2147483648)
        }
    }
    invocation = bedrock_runtime.start_async_invoke(
        modelId=REEL_MODEL_ID,
        modelInput=model_input,
        outputDataConfig={"s3OutputDataConfig": {"s3Uri": f"s3://{S3_DESTINATION_BUCKET}"}}
    )
    return invocation["invocationArn"]
```

taskType=TEXT_VIDEO

元画像をbase64で
指定します。

Invocation ARNが
返却されるので、
statusを確認しましょう。

S3にoutput.mp4が
出力されます

```
def generate_video_with_nova_reel(prompt, input_image_base64, duration=6, fps=24, dimension="1280x720"):
    bedrock_runtime = boto3.client("bedrock-runtime", region_name=REGION)
    model_input = {
        "taskType": "TEXT_VIDEO",
        "textToVideoParams": {
            "text": prompt,
            "images": [{
                "format": "png",
                "source": {
                    "bytes": input_image_base64
                }
            }]
        },
        "videoGenerationConfig": {
            "durationSeconds": duration,
            "fps": fps,
            "dimension": dimension,
            "seed": random.randint(0, 2147483648)
        }
    }
    invocation = bedrock_runtime.start_async_invoke(
        modelId=REEL_MODEL_ID,
        modelInput=model_input,
        outputDataConfig={"s3OutputDataConfig": {"s3Uri": f"s3://{S3_DESTINATION_BUCKET}"}}
    )
    return invocation["invocationArn"]
```

taskType=TEXT_VIDEO

元画像をbase64で
指定します。

Invocation ARNが
返却されるので、
statusを確認しましょう。

```
def wait_for_video_generation(invocation_arn):
    bedrock_runtime = boto3.client("bedrock-runtime", region_name=REGION)
    while True:
        response = bedrock_runtime.get_async_invoke(
            invocationArn=invocation_arn
        )
        status = response["status"]
        st.write(f"ステータス: {status}")

        if status == "Completed":
            s3_prefix = invocation_arn.split('/')[-1]
            # 動画をダウンロードしてセッション状態に保存
            s3_client = boto3.client('s3', region_name=REGION)
            try:
                local_file = f"temp_video_{int(time.time())}.mp4"
                s3_client.download_file(S3_DESTINATION_BUCKET, f"{s3_prefix}/output.mp4", local_file)
                with open(local_file, 'rb') as video_file:
                    st.session_state.video_data = video_file.read()
                    st.session_state.video_generated = True
                    st.rerun()
            except Exception as e:
                st.error(f"動画の取得に失敗しました: {str(e)}")

        if status != "InProgress":
            break
        time.sleep(SLEEP_TIME)
```

Invocation ARNを指定して
statusを取得できます。

S3にoutput.mp4が
出力されます

すみません。
時間がかかるので
3分クッキングします。

Amazonの最新モデルNova CanvasとNova Reelを試してみましょう！

時差ボケで作ったアプリなので例外処理がいろいろ加減です。エラーが起きたらやり直すこと

元の画像



画像生成プロンプト

猫

画像を生成

加工後の画像



柴犬にする

背景を消す

色を変える

アニメ風にする

メインオブジェクトを消す

動画を作成

プロンプト
「猫」

Nova Proで
英訳+詳細化

画像作成
(Canvas)

画像編集
(Canvas)

動画作成
(Reel)

楽しかった!


tubone24/bedrock-reel- canvas-test



 1

Contributor

 0

Issues

 0

Stars

 0

Forks





THANK YOU FOR LISTENING