

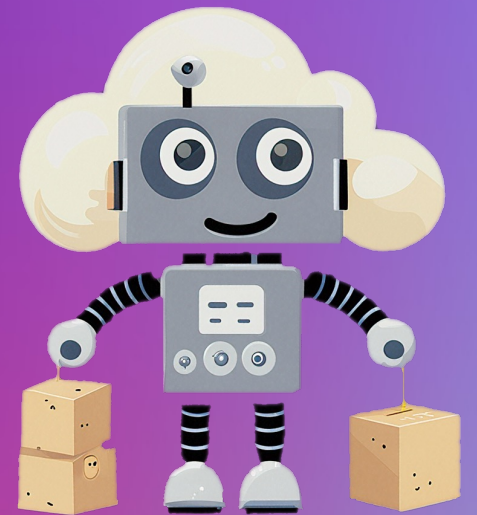


LIG104

Why I built my own chat app (twice), and why you should, too

Constantin Gonzalez (he/him)

Principal Solutions Architect
Amazon Web Services



The competent person

“A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.”

— Robert A. Heinlein, *Time Enough for Love*

The competent person

“A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly.
Specialization is for insects.”

— Robert A. Heinlein, *Time Enough for Love*

How can you become more competent?



Pet projects!

- They are a learning opportunity: that's the **goal**
- They are motivating: this helps you **finish** them
- They should be from the real-world: so you learn something **relevant**
- They should be your own: so you're in **control**



Pet project #1: AI for my family

Making Stable Diffusion family-ready

1. Deploy Stable Diffusion model
2. Add a UI
3. Done!

Simple?

Making Stable Diffusion family-ready

- Deploy Stable Diffusion model
 - Production-ready
 - CI/CD
 - Pre/post-processing
- Add a UI
 - Front-end development
 - User-management
 - Hosting



Amazon SageMaker Jumpstart

AWS Machine Learning Blog

Generate images from text with the stable diffusion model on Amazon SageMaker JumpStart

by Vivek Madan, Leonardo Bachege, and Santoshkumar Kulkarni | on 07 NOV 2022 | in Amazon SageMaker, Amazon SageMaker JumpStart, Artificial Intelligence, Foundational (100) | Permalink | Comments | Share

March 2023: This post was reviewed and updated with support for Stable Diffusion inpainting model.

Today, we announce that [Stable Diffusion 1](#) and [Stable Diffusion 2](#) are available in [Amazon SageMaker JumpStart](#). JumpStart is the machine learning (ML) hub of SageMaker that provides hundreds of built-in algorithms, pre-trained models, and end-to-end solution templates to help you quickly get started with ML. Stable Diffusion is an image generation model that can generate realistic images given a raw input text. You can use Stable Diffusion to design products and build catalogs for e-commerce business needs or to generate realistic art pieces or stock images.

In this post, we provide an overview of how to deploy and run inference with Stable Diffusion in two ways: via JumpStart's user interface (UI) in [Amazon SageMaker Studio](#), and programmatically through [JumpStart APIs](#) available in the [SageMaker Python SDK](#).

Stable Diffusion

Generative AI technology is improving rapidly, and it's now possible to generate text and images simply based on text input. Stable Diffusion is a text-to-image model that empowers you to create photorealistic applications.

A diffusion model trains by learning to remove noise that was added to a real image. This de-noising process generates a realistic image. These models can also generate images from text alone by conditioning the generation process on the text. For instance, Stable Diffusion is a latent diffusion where the model learns to recognize shapes in a pure noise image and gradually brings these shapes into focus if the shapes match the words in the input text. The text must first be embedded into a latent space using a language model. Then, a series of noise addition and noise removal operations are performed in the latent space with a U-Net architecture. Finally, the de-noised output is decoded into the pixel space.

The following are some examples of input texts and the corresponding output images generated by Stable Diffusion.

The following images are in response to the inputs "a photo of an astronaut riding a horse on mars," "a painting of new york city in impressionist style," and "dog in a suit."



Resources

[Getting Started](#)
[What's New](#)

Blog Topics

[Amazon Comprehend](#)
[Amazon Kendra](#)
[Amazon Lex](#)
[Amazon Polly](#)
[Amazon Rekognition](#)
[Amazon SageMaker](#)
[Amazon Textract](#)

Follow

[Twitter](#)
[Facebook](#)
[LinkedIn](#)
[Twitch](#)
[Email Updates](#)

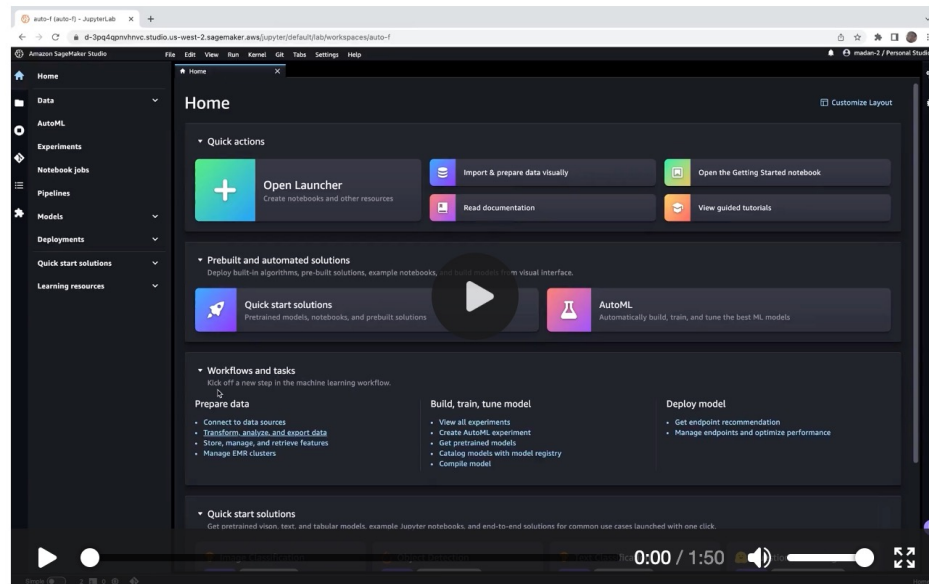


AWS Events
Discover the latest AWS events in your region
[Learn more >](#)

Access JumpStart through the Studio UI

In this section, we demonstrate how to train and deploy JumpStart models through the Studio UI. The following video shows you how to find the pre-trained Stable Diffusion model on JumpStart and deploy it. The model page contains valuable information about the model and how to use it. You can deploy any of the pre-trained models available in JumpStart. For inference, we pick the ml.p3.2xlarge instance type because it provides the GPU acceleration needed for low inference latency at a low price point. After you configure the SageMaker hosting instance, choose **Deploy**. It may take 5–10 minutes until your persistent endpoint is up and running.

After a few minutes, your endpoint is operational and ready to respond to inference requests!



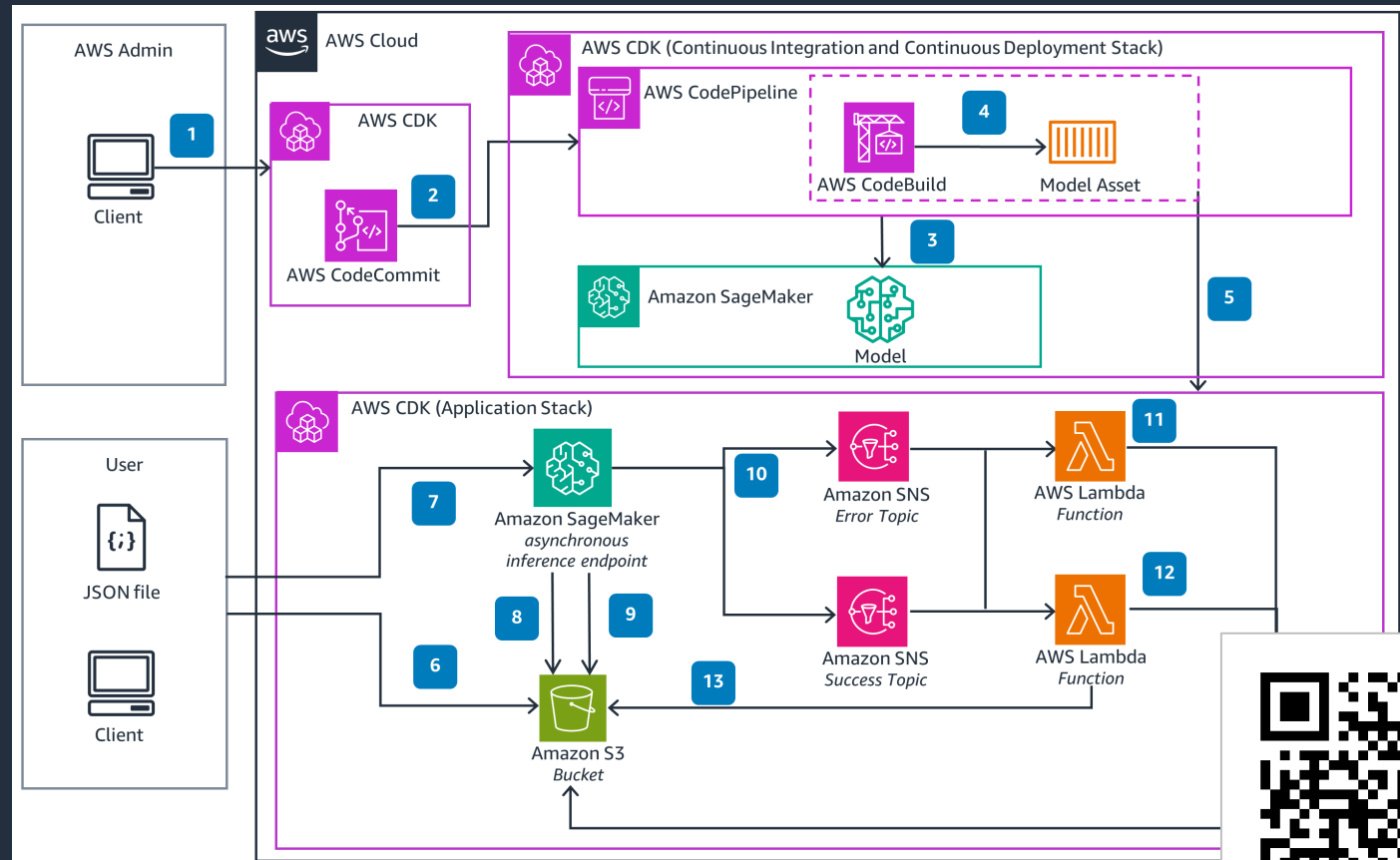
But:

- DevOps?
- Changes?
- CI/CD?
- Pre/Post-processing?
- Infrastructure-as-code?



Deploy JumpStart models easily with AWS CDK

- Choose a JumpStart model
- Write one line of AWS CDK (Python)
- Deploy automatically
- CI/CD pipeline included
- go.aws/3KrQuTw



Self-mutating CDK Pipeline

Source Succeeded
Pipeline execution ID: a31a78f7-2671-4afe-8063-9d4b96e26c15

StableDiffusionService
AWS CodeCommit
Succeeded - Just now
14d82b93

14d82b93 StableDiffusionService: Merge pull request #8 from aws-solutions-library-samples/dependabot/pip/simple_g

Disable transition

Build In progress
Pipeline execution ID: a31a78f7-2671-4afe-8063-9d4b96e26c15

Synth
AWS CodeBuild
In progress - Just now
Details

View logs

14d82b93 StableDiffusionService: Merge pull request #8 from aws-solutions-library-samples/dependabot/pip/simple_g

Disable transition



Source Succeeded
Pipeline execution ID: a31a78f7-2671-4afe-8063-9d4b96e26c15

Build In progress
Pipeline execution ID: a31a78f7-2671-4afe-8063-9d4b96e26c15

UpdatePipeline Completed
Pipeline execution ID: a31a78f7-2671-4afe-8063-9d4b96e26c15

SelfMutate
AWS CodeBuild
Didn't Run
No executions yet

Disable transition

Assemble Didn't Run

Frontend1
AWS CodeBuild
Didn't Run
No executions yet

Frontend2
AWS CodeBuild
Didn't Run
No executions yet

Disable transition

BackendDeploy Didn't Run

StableDiffusionServiceTest Didn't Run

Produce
AWS CodeBuild
Didn't Run
No executions yet

Deploy
AWS CodeBuild
Didn't Run
No executions yet

...and it works!

Executing

You can use [util/generate_image.py](#) to test the image generation. The file [util/test_request.json](#) in the same folder works with the StableDiffusion model configured in the standard config.toml.

Example

```
./util/generate_image.py --request-input-file util/test_request.json
```

[guidance-for-generative-ai-deployments-using-amazon-sagemaker-jumpstart](#) / util / [test_request.json](#)

Code Blame 8 lines (8 loc) · 168 Bytes

```
1  {
2      "prompt": "astronaut on a horse",
3      "width": 400,
4      "height": 400,
5      "num_images_per_prompt": 2,
6      "num_inference_steps": 50,
7      "guidance_scale": 7.5
8  }
```



Making Stable Diffusion family-ready

- Deploy Stable Diffusion model ✓
 - Production-ready ✓
 - Pre/post-processing ✓
 - CI/CD ✓
- Add a UI
 - Front-end development
 - User-management
 - Hosting



AWS Amplify

CREATE AND CONNECT TO BACKEND FOR YOUR IOS, ANDROID, FLUTTER, WEB, OR REACT NATIVE APP

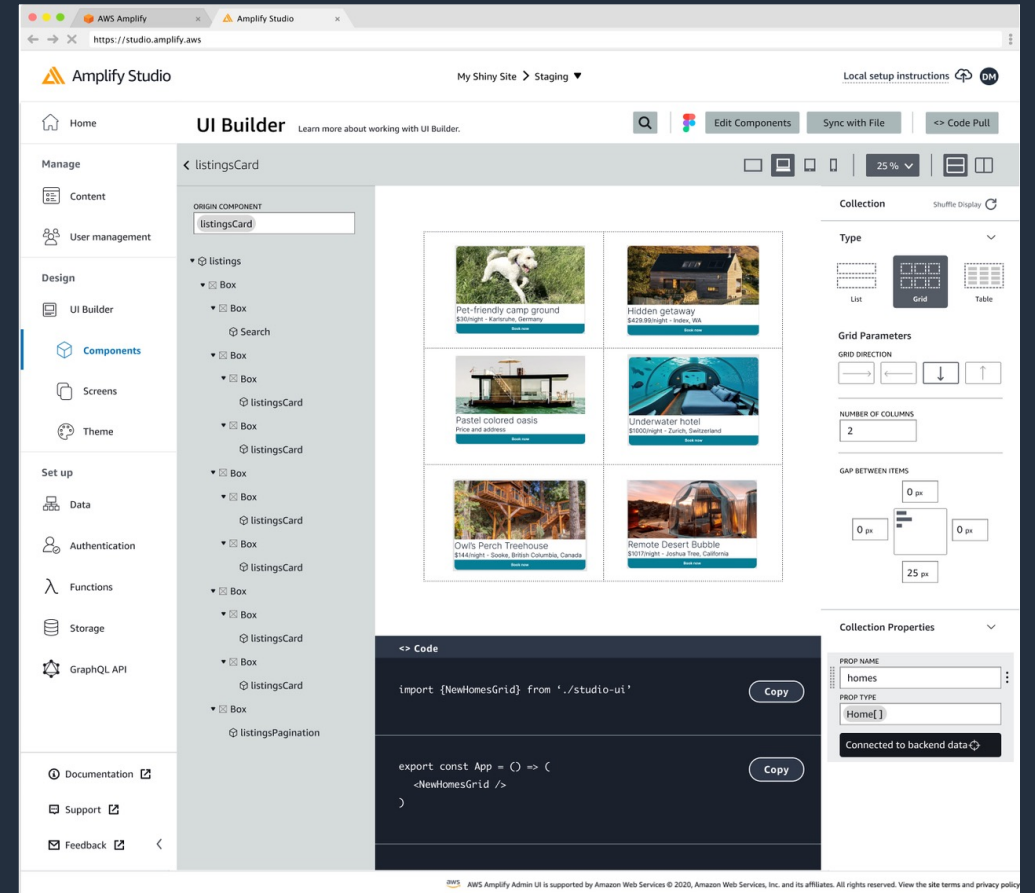
**Your app needs a backend?
It's just one command away**

amplify add

AWS Amplify components

BUILD A FRONTEND UI WITH PREBUILT COMPONENTS

- Prebuilt component library
- Themeable
- Accessible
- Bind components to data in a few steps



Amplify hosting

EASILY DEPLOY TO THE FAST, SECURE, AND RELIABLE AWS CONTENT DELIVERY NETWORK

How it works

1

Connect your repository



2

Configure build settings



3

Deploy your app

```
02:33:00 Preparing Repository
02:33:05 Reticulating Splines
02:34:11 Launch Prep Initiated
02:34:57 Launch Prep Complete
02:35:03 We Have Lift-Off
```



With a little help...



You

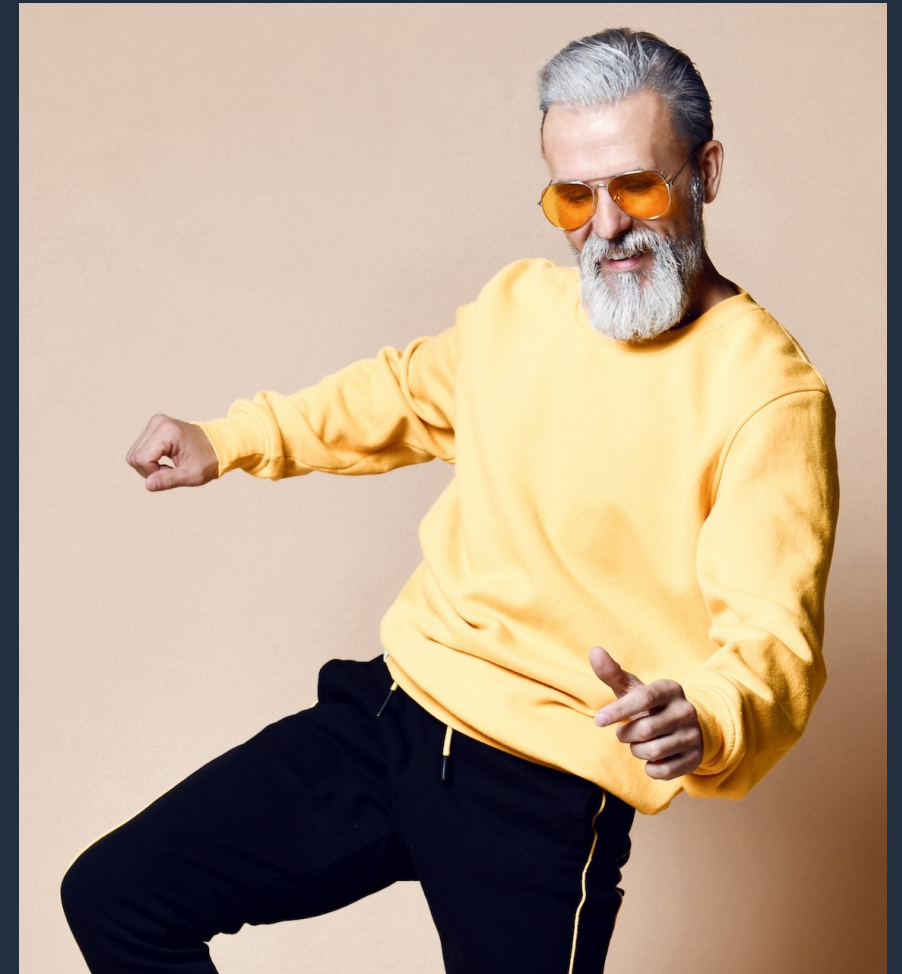
Hi! You are a senior React developer with 20 years of JavaScript experience and 10 years of React experience. I'd like you to act as a helpful React coach who always suggests the most helpful programming patterns and guides me towards implementing things according to good programming practices. Do you understand?




ChatGPT

Absolutely, I understand. I can certainly provide guidance and advice regarding JavaScript and React programming best practices, design patterns, performance improvements, code organization, testing, and more. Keep in mind, however, that while I'm a sophisticated AI model capable of discussing complex programming topics, I don't have personal experience or emotions. I base my advice on the large amount of data I've been trained on, including many books, articles, and other materials about software development.

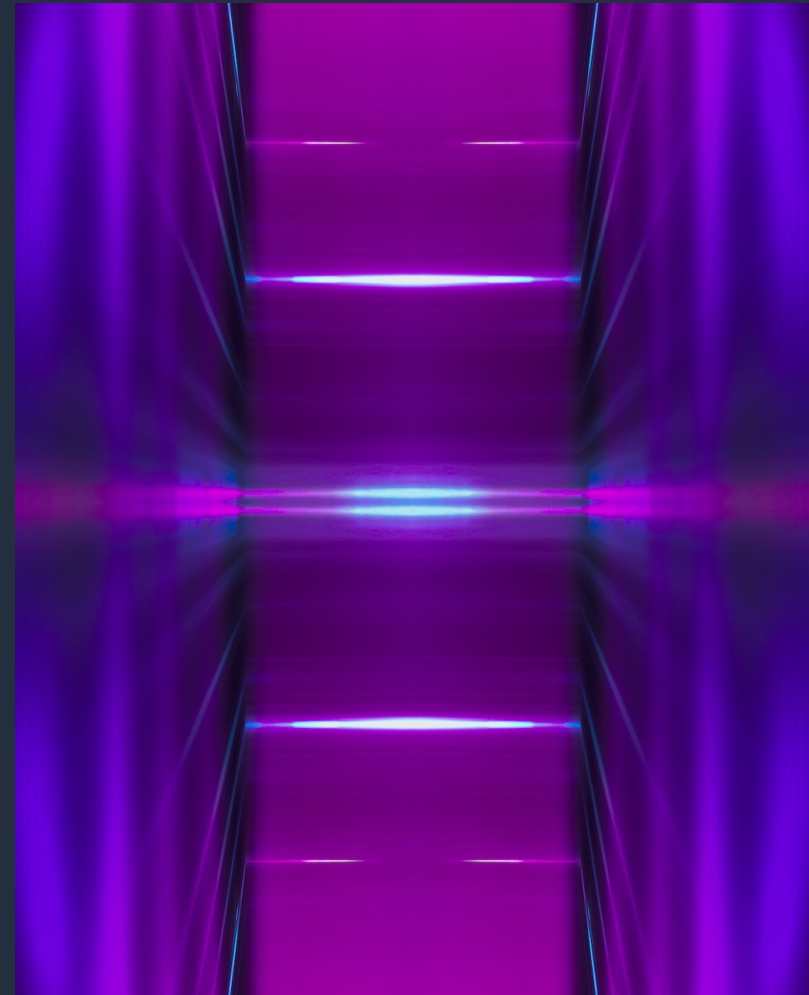
Now, what would you like to discuss or ask about?



...and some assistance



```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```





Welcome to Eddie text2image!

Prompt

Describe the image you'd like to generate as clearly as possible.

A parrot in an astronaut suit, riding a horse

Negative prompt

Describe things you'd like to avoid in the image.

Generate

Looking for ideas? Check out this [Stable Diffusion prompt guide](#), the article ["How to write an awesome Stable Diffusion prompt"](#), the [OpenArt Prompt book](#), the [Public Prompts collection](#), or this [illustrated guide to Stable Diffusion](#)

Width

The width of the image in pixels.

512





Logged in as Constantin.

[Log out](#)



Chat with Eddie!

Personality

The AI model parameters to use for chatting.

 Eddie, der Wellensittich ▼

Stop generating

New chat

Eddie ist ein hübscher, grüner Wellensittich, der schlau und hilfreich ist.



Hi Eddie, wie geht es Dir? Kannst Du mir einen Witz über das Wetter in Deutschland erzählen?

Hallo! Mir geht es wunderbar, danke, dass du fragst. Ich hoffe, dir geht es genauso gut. Natürlich kann ich dir einen Witz über das Wetter in Deutschland erzählen. Also, hier kommt er:



Warum ist das Wetter in Deutschland manchmal so launisch?

Weil es sich immer noch nicht entscheiden kann, ob es regnen oder scheinen soll. Es ist wohl ein echter "Wetterzwitscher"! Tschilp!

Ich hoffe, der Witz hat dir ein kleines Lächeln ins Gesicht gezaubert. Wenn du noch weitere Fragen hast, stehe ich dir gerne zur Seite.

Making Stable Diffusion family-ready

- Deploy Stable Diffusion model ✓
 - Production-ready ✓
 - Pre/post-processing ✓
 - CI/CD ✓
- Add a UI ✓
 - Front-end development ✓
 - User-management ✓
 - Hosting ✓



Some stuff I learned

- Amazon SageMaker Jumpstart
- Amazon SageMaker SDK source
- Amazon SageMaker Asynchronous inference
- Hugging Face Deep Learning Containers for Amazon SageMaker
- AWS CDK Pipelines
- Proper Python static code checking
 - pre-commit.com
- AWS Amplify
- AWS Amplify Hosting
- React (from scratch)
- JavaScript (almost from scratch)
- Backend-for-frontend concepts
- How to tunnel HTTP chunked responses over S3 by polling
- Prompt engineering

Pet project #2: A terminal-based Bedrock client

Adding a Terminal interface to Bedrock

1. Use the Bedrock API
2. Add a bit of Python on top
3. Done!

How hard can it be?

“React for the terminal”

The screenshot shows the Textual framework documentation website. The header is dark blue with the 'Textual' logo on the left, a search bar in the center, and GitHub repository information on the right (v0.48.1, 22.7k stars, 706 forks). The main content area is white and features a 'Welcome' section with a 'Get started' button and a link to the 'Tutorial'. A 'Tip' box on the right suggests checking navigation links in the header or side-bar. Below this is a 'What is Textual?' section with a description of the framework and its benefits. Three feature cards are displayed: 'Rapid development' (uses Python skills), 'Low requirements' (runs on a single board computer), and 'Cross platform' (runs everywhere).

Textual

Search

GitHub v0.48.1 ☆ 22.7k 🗨 706

Welcome

Welcome to the [Textual](#) framework documentation.

[Get started](#) or go straight to the [Tutorial](#)

Tip

See the navigation links in the header or side-bar.

Click ☰ (top left) on mobile.

What is Textual? 🗨

Textual is a *Rapid Application Development* framework for Python, built by [Textualize.io](#).

Build sophisticated user interfaces with a simple Python API. Run your apps in the terminal or a [web browser](#)!

⌚ 🔄 Rapid development

Uses your existing Python skills to build beautiful user interfaces.

🖨 Low requirements

Run Textual on a single board computer if you want to.

🪟 🍏 🐧 Cross platform

Textual runs just about everywhere.

textual.textualize.io

○ Chatrock – A chat interface to Amazon Bedrock
Suggest a developer joke for my presentation
us-east-1 - Anthropic Claude v2.1 - 179->227 (\$0.00688)

😊 Hi Claude, I'm working on a presentation for an upcoming developer event. Can you suggest a joke I can use in the beginning of my talk about presenting to a developer audience or a similar theme?

🗣️ Here are a few lighthearted developer-themed jokes I could suggest for the beginning of your presentation:

"Why was the developer unhappy at the baseball game? Because there were too many bugs and not enough hits!"

"Did you hear about the developer who got stuck in the shower? They were debugging for half an hour before realizing it was a soap issue."

"Why do developers prefer dark mode? Because light attracts bugs."

"How do you tell an extroverted developer from an introverted one? The extroverted one looks at YOUR shoes when talking."

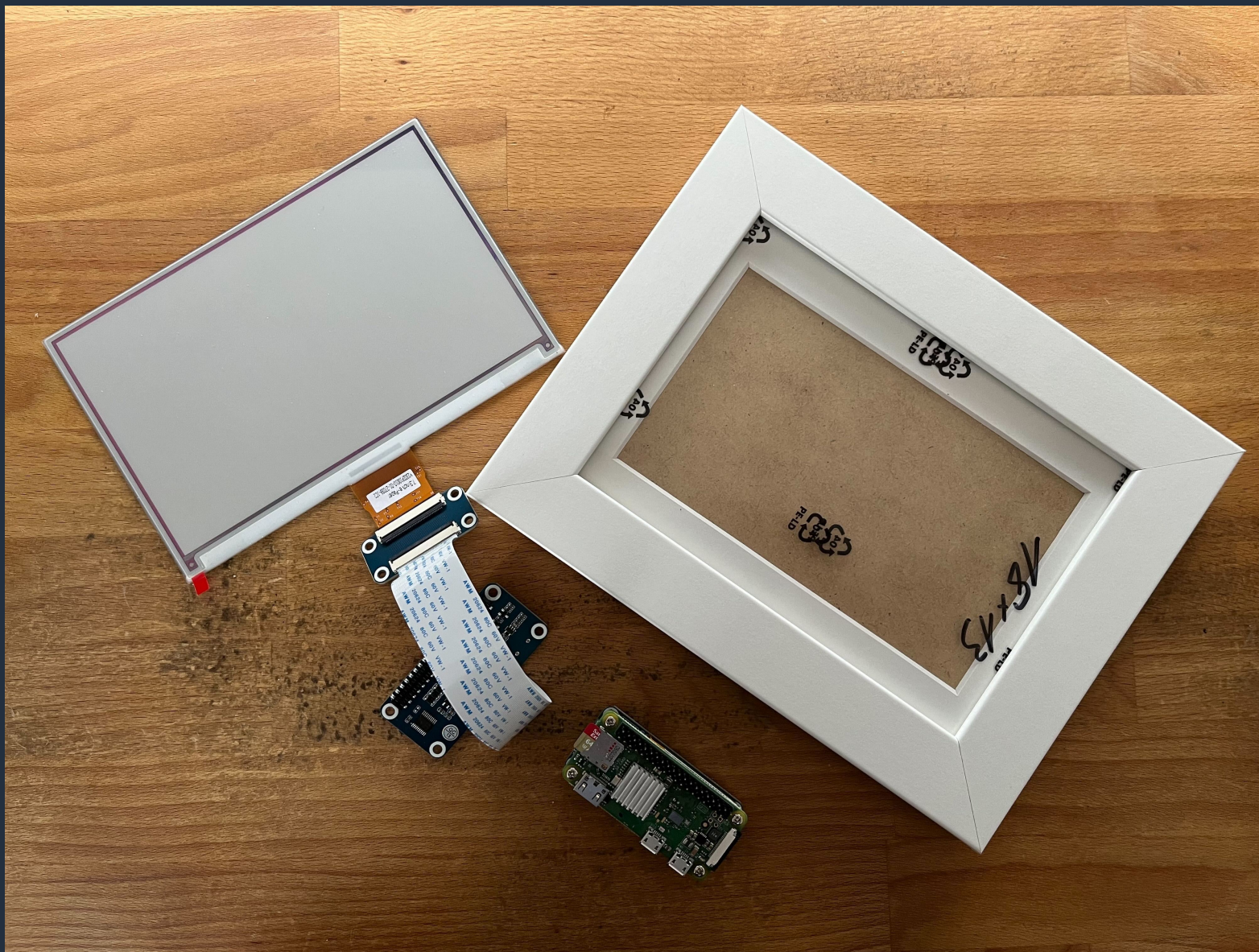
Input your message here.

CTRL+L Show help screen CTRL+Q Quit CTRL+S (Or: double-enter) Submit

Stuff I learned

- Figuring out the proper user experience
- The devil is in the details
- Terminals can't handle Shift+Return
- Proper modularization can be challenging
- Word-wrapping can be *hard*
- How to or not to abstract away Bedrock models
- Token counting/estimation
- Iteration time matters
- Async programming in Python
- Anthropic Claude is my new best friend

Pet project #3: An e-ink display for Amazon Titan Image Generator



Stuff I learned already...

- AWS IoT Greengrass
- Raspberry Pi power management
- Raspberry Pi real-time clock
- SPI-interfaces and low-level HW signaling
- ...to be continued

Summary

- Learning one thing is not enough — **Become a competent person**
- Pick **good pet projects** and run with them
- Check out some non-AI services to **complete** your projects:
 - AWS Amplify: great for **building apps** with generative AI
 - AWS CDK and CDK Pipelines for **Infrastructure-as-code** and **CI/CD**
 - AWS IoT Greengrass for building **robots** and **physical things**
- The devil is in the details — and they are **important**
- Stick to your projects **until the end** — and have **fun**



Thank you!

Constantin Gonzalez

 constantingonzalez



Please complete the session survey.



pulse.aws/survey/VI2LGBRD