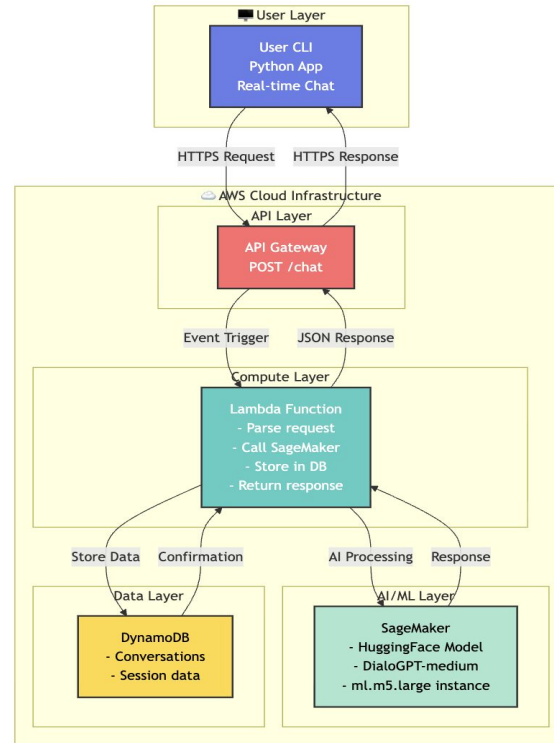


NLP Cloudy Bot

Present by Qiuhaio Gu

System Architecture Overview

User types in CLI → API Gateway →
Lambda processes → SageMaker AI
responds → DynamoDB stores history



System Architecture Overview

1. **Lambda** handles requests (pay per use)
2. **SageMaker** powers the AI with HuggingFace models
3. **DynamoDB** stores conversations
4. **Terraform** defines everything as code



Hugging Face is way more fun with friends and colleagues! 😊 [Join an organization](#)

microsoft/**DialoGPT-medium** 📄 like 396 Follow Microsoft 14.1k

Text Generation Transformers PyTorch TensorFlow JAX Rust gpt2

Model card Files and versions xet Community 21



> 1+1 equal to
🤖 Thinking...
🌐 Bot: 1,000, 000

> smart bot
🤖 Thinking...
🌐 Bot: I'm here to help! Could you please rephrase your question?



Demo/example

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: bash

[illegible]

Tradeoff

- **Tradeoff:** SageMaker Serverless was chosen for pay-per-millisecond pricing with zero idle costs and access to the full Hugging Face model ecosystem, despite cold start delays of 3-10 seconds and additional setup complexity for IAM and container configurations.
- **Alternative:** Amazon Lex + Bedrock would provide simpler setup with minimal cold start issues at \$0.00075 per turn, but lacks access to specialized Hugging Face models and custom fine-tuning capabilities needed for domain-specific chatbot behavior.

Challenges

1. HuggingFace Model Images
2. IAM Role Timing Issues
3. Terraform Complexity

Next Steps

- Web-based UI with React frontend
- Multi-model support (GPT-4, Claude, etc.)
- Vector database for context retrieval
- Real-time collaboration features

Thanks