Agents on Edge

Deploying LangChain Agents on Cloudflare

Karim Lalani

What is Edge?

Proximity to End Users: The edge refers to servers and resources located geographically closer to end users, reducing the distance data travels and improving performance.

Edge in Cloud Computing: In cloud computing, edge locations are distributed data centers that process data and run applications closer to the user, improving speed and reducing latency.

Serverless at the Edge: Serverless platforms like Cloudflare Workers or AWS Lambda@Edge allow developers to run code closer to the user, without needing to manage or provision servers.

Edge and Content Delivery Networks (CDNs): CDNs cache content at edge nodes, ensuring faster delivery of static assets like images, scripts, and videos from servers closer to the end-user's location.

Real-Time AI & Edge: Running AI agents at the edge allows real-time decision-making and interaction with users, particularly for latency-sensitive tasks like chatbots or recommendation engines.

Cloudflare’s Global Network

https://www.cloudflare.com/network/

Why Edge?

General Cloud Deployment Benefits:

- **Scalability:** Seamlessly handle growing workloads by dynamically scaling resources based on demand.
- **Cost Efficiency:** Pay-as-you-go pricing reduces upfront infrastructure costs and enables flexible resource allocation.

Specific to AI Agents:

- **Global Accessibility & Low Latency:** Deploy AI agents closer to users through edge computing, reducing latency and improving response times for real-time interactions.
- **Integration with Cloud Services:** Easily connect AI agents with cloud-based AI/ML tools, data storage, and analytics platforms for enhanced capabilities.
- **Access to LLM APIs and Inference Services:** Most large language models (LLMs) are accessed via APIs from providers like OpenAI, Anthropic, and Google, or through cloud inference services such as Azure, Bedrock, Groq, and NIMs.

LLM / Inference Providers

LangChain already supports:

- Anthropic
- AWS
- Google
- Hugging Face
- Microsoft
- OpenAI
- And many more

https://python.langchain.com/v0.2/docs/integrations/providers/

Why Cloudflare?

- **Edge Network for Low Latency:** Cloudflare's global edge network ensures that your AI agents are deployed close to users, minimizing latency and delivering faster responses.
- **Integrated Security Solutions:** Built-in Web Application Firewall (WAF), DDoS protection, and SSL/TLS encryption safeguard your AI agents without added complexity.
- **Scalability Without Complexity:** Cloudflare Workers and Durable Objects allow your agents to scale seamlessly without managing traditional infrastructure.
- **Developer-Friendly Ecosystem:** Cloudflare Workers support JavaScript, TypeScript, and modern frameworks, making it easy to deploy and manage AI agents.
- **Built-in AI Platform Integration:** Cloudflare Workers AI allows you to run AI models directly on the edge, supporting deployments without external API dependencies.

Getting started with Cloudflare Workers

Run the command

```
npm create cloudflare@latest
```

Provide a name for the application: cf-worker

Select starter code: Hello World example

Select a template: Hello World Worker

Select language: Typescript

Git integration: No

Deploy: No

Running Worker locally

```
cd cf-worker
npm run start
```

http://localhost:8787

Deploy to Cloudflare

```
npm run deploy
```

Cloudflare Worker Hello World Code

src/index.ts

```
1 export default {
2     async fetch(request, env, ctx): Promise<Response> {
3         return new Response('Hello World!');
4     },
5 } satisfies ExportedHandler<Env>;
```

@lalanikarim

Cloudflare Worker - request and env

request

Fetch API interface representing a resource request

env

Provides access to variables stored in wrangler.toml and secrets

Cloudflare Worker - additional configuration

wrangler.toml

```toml
name = "cf-worker"
main = "src/index.ts"
compatibility_date = "2024-09-03"
compatibility_flags = ["nodejs_compat"]

[vars]
MY_VARIABLE = "production_value"
```

@lalanikarim

Cloudflare Workers AI

- Cloudflare provides a host of models on their platform through their Workers AI service.
- Supported models include the llama family, mistral, phi-2, falcon, and qwen to list a few.
- Complete list of supported models can be found here: https://developers.cloudflare.com/workers-ai/models/#text-generation

LangChain + Cloudflare Workers AI

Partner Package @langchain/cloudflare

- CloudflareWorkersAI
- ChatCloudflareWorkersAI

Limitations:

- Tool use not natively supported. Look into **tool-calling-llm** npm package.
- Latest langgraph is not yet supported. Only upto v0.0.34 is supported.

Sample Project

langchain-js-worker

- Cloudflare Workers TS project that exposes a simple Q/A chain as an API
- https://github.com/lalanikarim/langchain-js-worker/

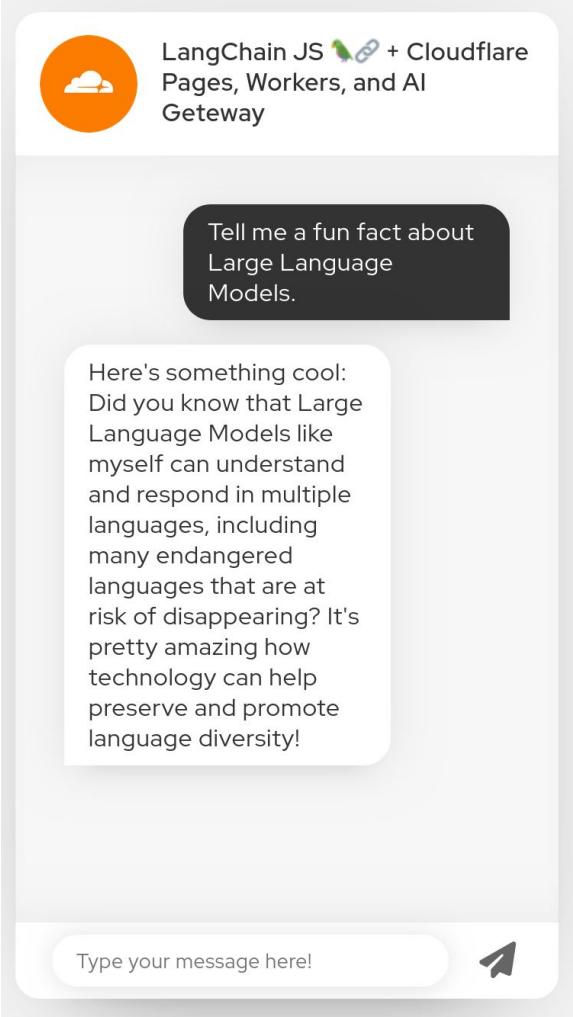
langchain-js-page

- A full-stack application front-end exposing a chat interface
- https://github.com/lalanikarim/langchain-js-page/

src/index.ts

```
1 import { CloudflareWorkersAI } from "@langchain/cloudflare"
2 import { PromptTemplate } from "@langchain/core/prompts"
3 import { StringOutputParser } from "@langchain/core/output_parsers"
4 export default {
5     async fetch(request, env, ctx): Promise<Response> {
6         const model = new CloudflareWorkersAI({
7             model: env.MODEL,
8             cloudflareAccountId: env.CLOUDFLARE_ACCOUNT_ID,
9             cloudflareApiToken: env.CLOUDFLARE_API_TOKEN,
10        });
11        const data = await request.json();
12        const prompt = data.prompt;
13        const messages = data.messages ?? [];
14        const promptTemplate = PromptTemplate.fromTemplate(
15            `You are a helpful AI companion. Keep your responses short and concise and keep the tone cheerful and positive.
16            {messages}
17            Human: {prompt}
18            AI: `);
19        const chain = promptTemplate.pipe(model).pipe(new StringOutputParser());
20        const response = await chain.invoke({messages, prompt});
21        return Response.json({response});
22    },
23 } satisfies ExportedHandler<Env>;
```

@lalanikarim


LangChain JS 🦜🔗 + Cloudflare Pages, Workers, and AI Geteway
Tell me a fun fact about Large Language Models.
Here's something cool: Did you know that Large Language Models like myself can understand and respond in multiple languages, including many endangered languages that are at risk of disappearing? It's pretty amazing how technology can help preserve and promote language diversity!
Type your message here!

Questions?