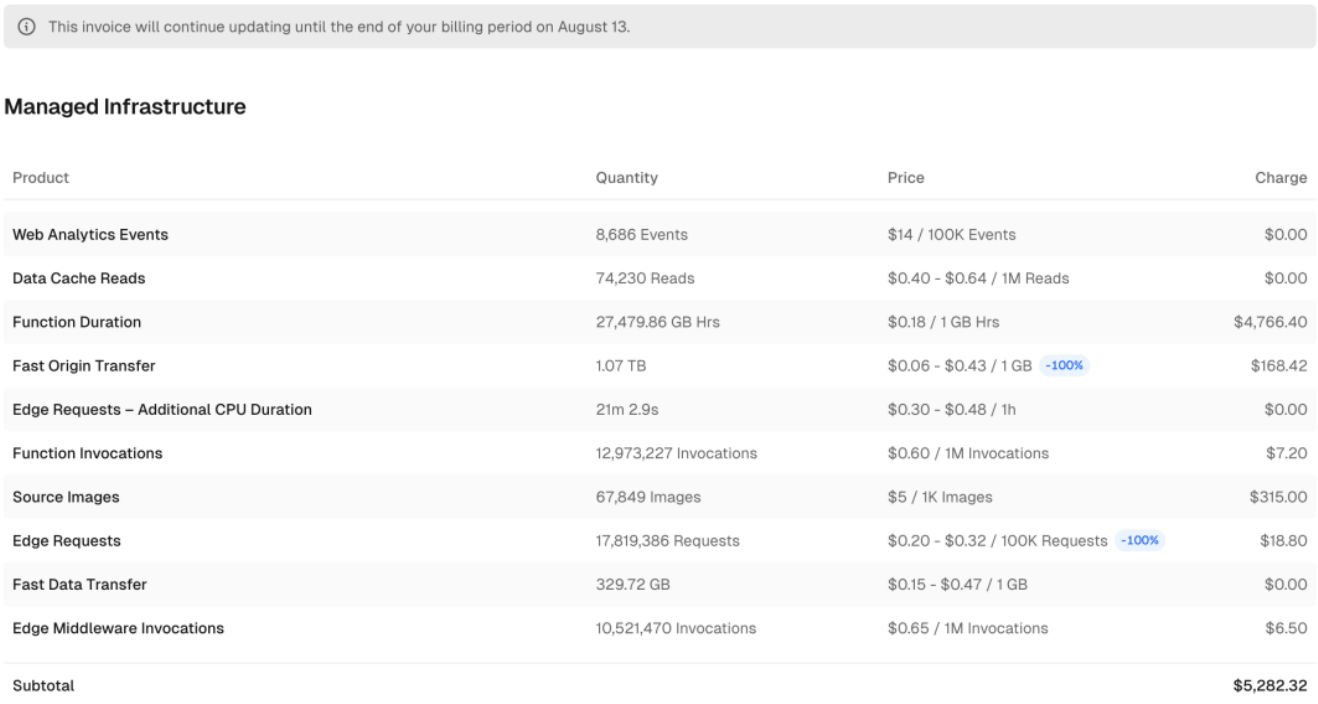
我把网站迁移到 cf，省了几万块

idoubi 哥飞 2024-08-14 21:16:35 广东

前言

在上一篇文章讲到，我做了一个 AI 搜索引擎，部署在 Vercel，每个月几十万访问量。

前几天登录 Vercel 看了一下账单，好家伙！！！一个月给我干到了 5000 多刀的支出，合计人民币 3 万 7 千多元。



This invoice will continue updating until the end of your billing period on August 13.

Managed Infrastructure

Product	Quantity	Price	Charge
Web Analytics Events	8,686 Events	$14 / 100K Events	$0.00
Data Cache Reads	74,230 Reads	$0.40 - $0.64 / 1M Reads	$0.00
Function Duration	27,479.86 GB Hrs	$0.18 / 1 GB Hrs	$4,766.40
Fast Origin Transfer	1.07 TB	$0.06 - $0.43 / 1 GB -100%	$168.42
Edge Requests – Additional CPU Duration	21m 2.9s	$0.30 - $0.48 / 1h	$0.00
Function Invocations	12,973,227 Invocations	$0.60 / 1M Invocations	$7.20
Source Images	67,849 Images	$5 / 1K Images	$315.00
Edge Requests	17,819,386 Requests	$0.20 - $0.32 / 100K Requests -100%	$18.80
Fast Data Transfer	329.72 GB	$0.15 - $0.47 / 1 GB	$0.00
Edge Middleware Invocations	10,521,470 Invocations	$0.65 / 1M Invocations	$6.50
Subtotal			$5,282.32

20240812200512

按理来说几十万的月访问量也不算大，这个服务器支出费用属实有点高了。

感觉很心疼，一下子对 Vercel 爱不起来了。研究了两天，把部署在 Vercel 的服务迁移到了 AWS 和 Cloudflare，世界变得美好了，谨以此文纪念之。

为什么要用 Vercel

在这篇文章：出海第一周，我的 GPTs 导航站□了 讲到，去年 11 月，我开始做独立出海项目，当时做的 GPTs 导航站，第一次用到了 Vercel 进行部署。

使用 Vercel 部署，主要图的是方便。

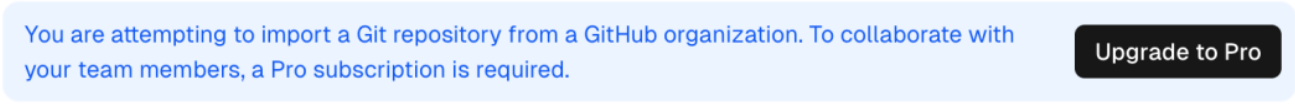
1. Vercel 集成了 Github 做 CI/CD，代码提交到 Github 自动发布上线，可以滚动更新，回退版本，DevOps 相当便利；
2. Vercel 内置域名服务，为你部署的每个项目，生成一个：xxx.vercel.app 的子域名，可以公网访问，方便新项目快速上线 Demo 版本；
3. Vercel 可以指定分支部署项目，为每一次部署生成一个独立的访问地址，方便正式上线前进行调试和验证，还支持团队协作，划线评论；
4. Vercel 集成了很多常用的功能，比如网站访问统计 / 页面性能分析 / 服务运行日志 / 环境变量管理 / 防火墙等，还支持存储套件，比如文件存储 / Postgres 数据存储 / KV 等；

最最关键的一点，Vercel 和流行全栈开发框架 nextjs 同属于一家公司，对 nextjs 项目支持非常到位。Vercel 模板中心有大量使用 nextjs 开发的模板，生态做的很完善：开发框架 + 模板组件 + 运维部署一条龙服务，刚接触全栈开发的朋友，简直不要太爱。

Vercel 有哪些坑

Vercel 最大的坑，是收费太贵了。

首先，免费版本只能部署 Github 个人项目，如果你的项目是放在某个组织下面，要在 Vercel 部署，就必须要升级成团队版，20 美金 / 月。



20240812210107

普通账户，在云函数的响应时间方面也有所限制。默认 10s 超时，最大可配置 60s，升级到 Pro 版本，默认 15s 超时，最大可配置 300s。如果不升级，请求 OpenAI 的 dall-e-3 生成图片，很容易就超时了。

```
export const maxDuration = 60; // This function can run for a maximum of 60 seconds
export const dynamic = 'force-dynamic';

export function GET(request: Request) {
  return new Response('Vercel', {
    status: 200,
  });
}
```

就算花 20 美金升级到了 Pro 版本，很多功能依然受限制。Vercel 几乎每项功能都是单独计费的，比如网页统计 Analytics，给一定的免费额度，访问量大一点就得付钱。比如数据存储，也是按照存储空间和访问流量计费的，计费规则一大堆，给你搞得迷迷糊糊，到了月初再给你出个账单，“惊喜值”拉满。

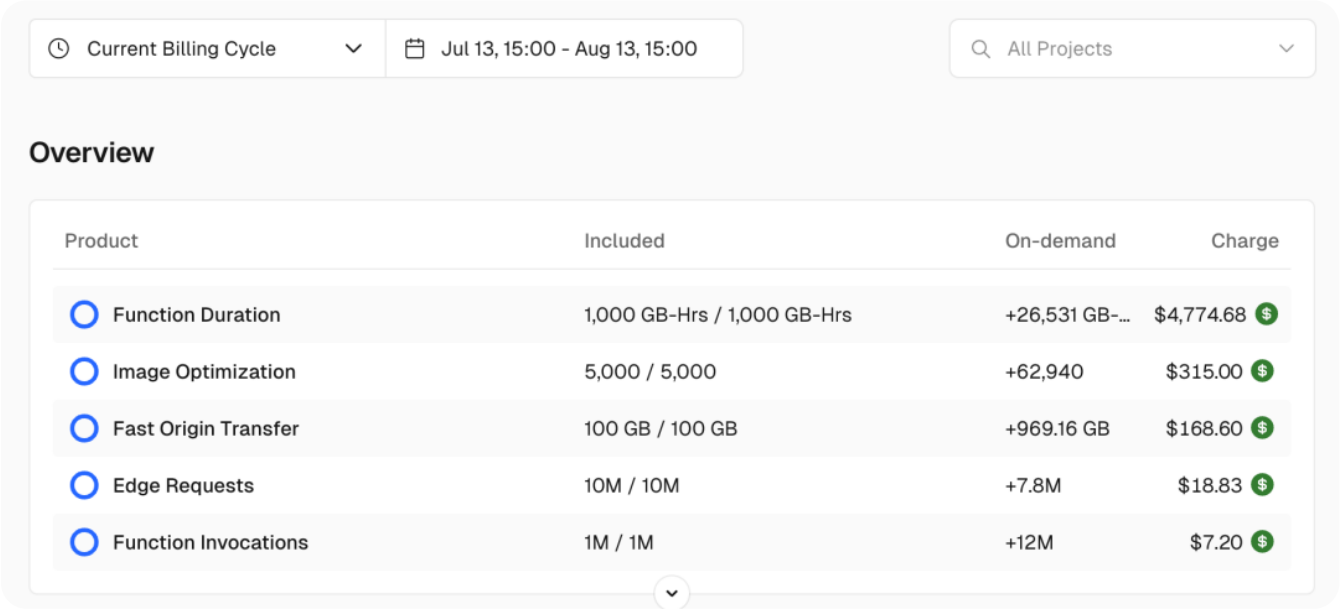
云函数调用是最大的支出，按流量计费：$0.18 / 1 GB Hrs。

如果你在 nextjs 项目中使用了 nextjs 的 Image 组件：

```tsx
import Image from 'next/image'

export default function Page() {
  return (
    <Image
      src="/profile.png"
      width={500}
      height={500}
      alt="Picture of the author"
    />
  )
}
```

Vercel 会帮你做图片裁剪，做 CDN 加速，对于图片型网站，体验是好了，但是这个费用，也是很大的一块支出，妥妥的”羊毛出在羊身上“。



20240812211314

别人给你提供了便利的服务，你就应该给别人交钱，这句话是没啥毛病。但是仔细想想，Vercel 本质上是 AWS 的一个套壳，你部署在 Vercel 上的网站，实际上是 Vercel 帮你部署在了 AWS 上，比起你自己在 AWS 买个 EC2 机器，部署若干个服务，Vercel 的费用高出不少，想想好像有点不太厚道。

天下苦 Vercel 久矣，是时候寻找其他的部署方案了。

有哪些可替代的部署方案

部署 nextjs 项目，除了 Vercel 之外，还有很多可替代的部署方案。包括跟 Vercel 类似的云部署平台，开源的部署方案，自建服务器部署以及使用 Cloudflare Pages 部署几种。

- 跟 Vercel 类似的云部署平台

1. Netlify：这是Vercel的直接竞争对手，提供类似的功能。Netlify还额外提供每个站点每月1000个已识别的活跃用户和站点分析。（月访问2.7M）

2. Railway：这个平台可以部署大部分项目，包括Docker容器。它支持项目内Dockerfile和公开打包好的docker镜像，但不

支持docker-compose。(月访问1.5M)

3. Zeabur：与Railway类似，可以部署多种项目类型，包括Docker容器。（月访问59.9K）
4. Render：另一个流行的云部署平台，提供类似Vercel的服务。（月访问1.6M）
5. Firebase：这是Google提供的一个平台，可用于部署和托管web应用。（月访问16.4M）
6. Heroku：虽然它的定位略有不同，但Heroku也是一个受欢迎的应用部署平台。（月访问1.9M）

- 开源部署方案

1. Coolify：这是一个引人注目的开源项目，旨在成为Heroku、Netlify和Vercel等流行平台的自托管替代方案。Coolify提供了一系列功能来简化应用程序部署过程。（27.7k star）
2. Dokku：Dokku是一个轻量级的开源PaaS（平台即服务）实现。它可以让你在自己的服务器上创建类似Heroku的环境，支持多种编程语言和框架。（26.5k star）
3. SST：SST是一个用于构建全栈无服务器应用的框架，专注于无服务器架构和AWS生态系统。（21.2k star）
4. Dokploy：Dokploy是一个开源的部署平台，旨在成为Vercel、Netlify和Heroku的替代方案。（5.2k star）

- 自建服务器部署

1. 买台服务器，安装宝塔面板，部署 nextjs 项目
2. 买台服务器，安装 pm2 做进程管理，部署 nextjs 项目
3. 买台服务器，安装 docker，使用容器部署 nextjs 项目
4. 买台服务器，使用 minikube 或 k3s 自建 K8S 集群，部署 nextjs 项目
5. 在 xx 云上面买托管的 K8S 集群，部署 nextjs 项目

- 使用 Cloudflare Pages 部署

Cloudflare 的定位为一家全球性的互联网基础设施提供商，提供了一系列的网络安全和性能优化服务，包括内容分发网络(CDN)、DDoS防护、SSL/TLS加密、DNS管理等。Cloudflare Workers（serverless计算平台）和Cloudflare Pages 可以用来部署 nextjs 应用。

以上几种方案，都可以用来替代 Vercel 部署 nextjs 项目，至于要选择哪一个，关键要考虑两点：服务费用和运维复杂度。

我选择从 Vercel 迁移，主要是为了降低成本，最关心的是费用问题，所以不会再选择其他云部署平台和托管的 K8S 集群，而是选择成本相当较低的自建服务器部署和 Cloudflare 的托管方案。

我实践了以下 3 种部署方案，分享一下具体的部署步骤。

在云服务器上用 pm2 部署 nextjs 项目

在 AWS 上购买一台 4c8g 的 EC2 服务器，选择 Ubuntu 操作系统，使用 pm2 做进程管理，部署 nextjs 项目。

1. 先确保安装了 nodejs 和 npm，并使用 pnpm 作为 nextjs 项目的依赖管理工具

```
npm install -g pnpm
```

2. 安装 pm2 做进程管理

```
npm install -g pm2
```

3. 在项目根目录下安装依赖，构建输出产物

```
pnpm install
pnpm build
```

4. 使用 pm2 启动服务

```
pm2 start pnpm --name sorafm -- start --port 8015
```

5. 使用 nginx 做反向代理

先确保安装和启动了 nginx：

```
sudo apt install nginx
sudo systemctl start nginx
```

为 nextjs 项目创建 nginx 配置：

`sudo vi /etc/nginx/conf/sorafm.conf`

```
server {
    listen 80;

    location / {
        proxy_pass http://127.0.0.1:8015/;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    error_log /var/log/nginx/sorafm.error;
}
```

nginx 重载新的网站配置：

```
sudo nginx -s reload
```

6. DNS 解析域名到服务器的公网 ip

在 DNS 控制台添加一条 A 记录，指向服务器的公网 ip，比如我这里使用的是子域名：sorafm.trys.ai

20240814152142

等解析生效后，就可以通过：

http://sorafm.trys.ai

访问 nextjs 项目了。

7. 配置 https 访问

可以在 Ubuntu 安装 certbot 生成域名证书：

```
sudo apt update
sudo apt install certbot python3-certbot-nginx
```

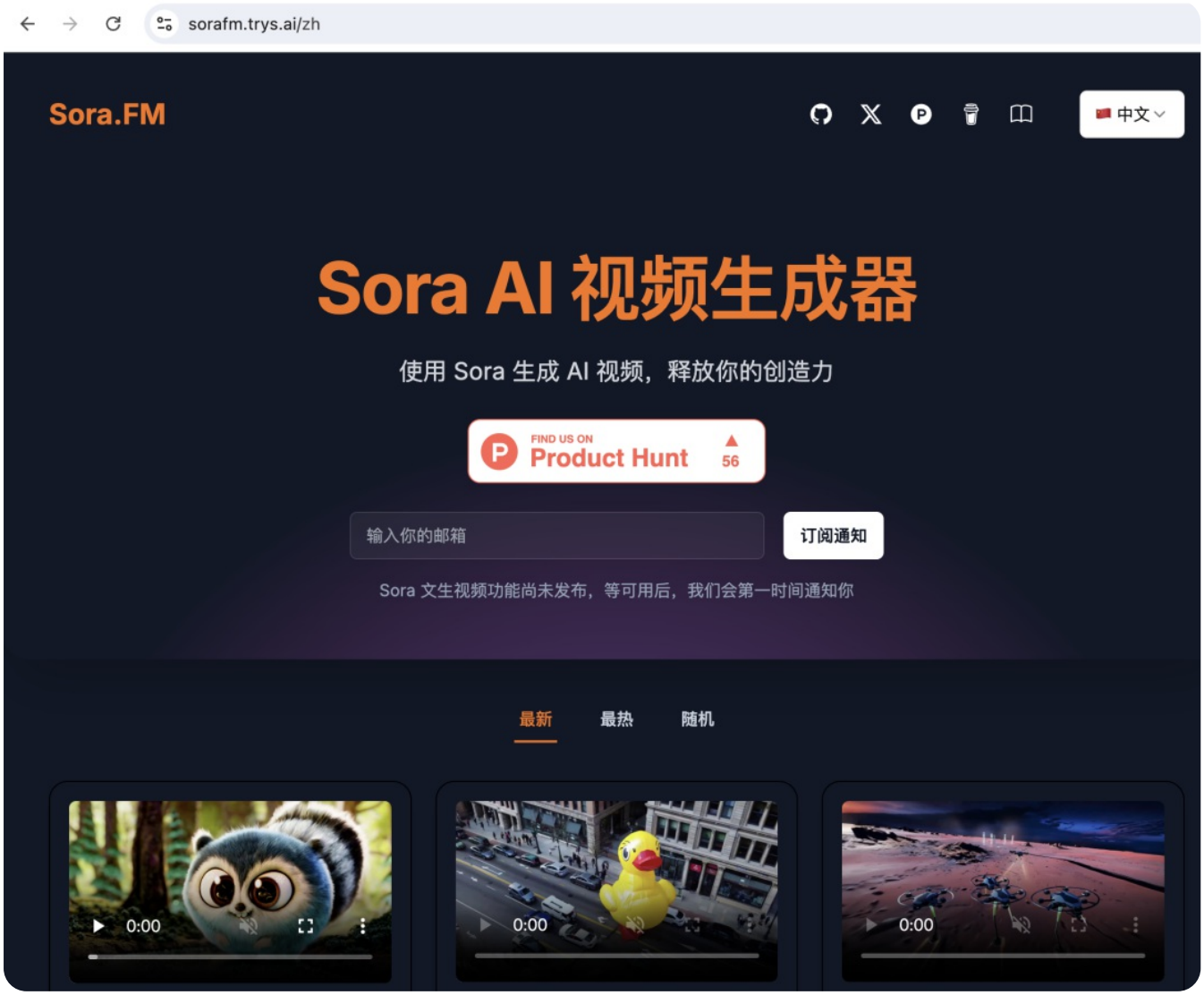
为新域名生成新的证书，并使用 https 访问

```
sudo certbot --nginx -d sorafm.trys.ai
```

配置完成后，就可以通过：

https://sorafm.trys.ai

安全访问 nextjs 项目了。



20240814152647

在云服务器上用 Docker 部署 nextjs 项目

同样是使用 Ubuntu 云服务器，使用 pm2 部署 nextjs 更简单直接，轻量级部署，服务资源占用少。使用 docker 部署，系统隔离性更好，更方便移植，适用于微服务架构。

要使用 docker 部署 nextjs 应用，先确保在服务器安装好了 docker，再来改造 nextjs 项目

1. 修改项目根目录下的 `next.config.mjs`，使用 `standalone` 模式输出编译产物

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: "standalone",
};

export default nextConfig;
```

2. 在 deploy 文件夹下新建

   `Dockerfile`

   文件，写入 docker 镜像构建内容

```
FROM node:18-alpine AS base

# Install dependencies only when needed
FROM base AS deps
RUN apk add --no-cache libc6-compat && yarn global add pnpm

WORKDIR /app

# Install dependencies based on the preferred package manager
COPY package.json pnpm-lock.yaml* ./
RUN pnpm i --frozen-lockfile

# Rebuild the source code only when needed
FROM deps AS builder

WORKDIR /app

# Install dependencies based on the preferred package manager
COPY . .
RUN pnpm build

# Production image, copy all the files and run next
FROM base AS runner
WORKDIR /app

RUN addgroup --system --gid 1001 nodejs && \
    adduser --system --uid 1001 nextjs && \
    mkdir .next && \
    chown nextjs:nodejs .next

COPY --from=builder /app/public ./public
COPY --from=builder --chown=nextjs:nodejs /app/.next/standalone ./
COPY --from=builder --chown=nextjs:nodejs /app/.next/static ./.next/static

USER nextjs

EXPOSE 8080

ENV NODE_ENV production

ENV PORT 8080
ENV HOSTNAME "0.0.0.0"

# server.js is created by next build from the standalone output
CMD ["node", "server.js"]
```

3. 在项目根目录下新建

   `.dockerignore`

文件，写入构建镜像时要忽略的内容

```
.next
.vercel
.vscode
data
debug
node_modules
```

4. 开始构建 docker 镜像

```
sudo docker build -f deploy/Dockerfile -t sorafm:latest .
```

5. 使用 docker 运行服务

```
sudo docker run -itd -p 127.0.0.1:8014:8080 --restart=always sorafm:latest
```

服务运行成功后，再通过 nginx 配置公网访问，DNS 解析公网域名，certbot 配置 https 证书，这三个步骤跟上面使用 pm2 部署 nextjs 的方案一致。

> 具体改造点参考：https://github.com/all-in-aigc/sorafm/commit/63802c832bf9a26dfe93e0964876c918c7132af2

使用 Cloudflare 部署 nextjs

上述两种方案：使用 pm2 和使用 docker 部署 nextjs 应用，需要先有一台服务器。

如果不想买服务器，而是通过托管的方式部署 nextjs 项目，可以选择 Cloudflare Pages，几乎免费的云部署方案。

按照 Cloudflare 的官方文档，使用 Cloudflare Pages 部署 nextjs 项目，主要的步骤：

1. 安装部署依赖

```
pnpm add -D @cloudflare/next-on-pages
```

2. 在项目根目录创建一个配置文件

   `wrangler.toml`

```
name = "sorafm"
compatibility_date = "2024-07-29"
compatibility_flags = ["nodejs_compat"]
pages_build_output_dir = ".vercel/output/static"
```

3. 更新

`next.config.mjs` 文件

```
import { setupDevPlatform } from "@cloudflare/next-on-pages/next-dev";

/** @type {import('next').NextConfig} */
const nextConfig = {};

if (process.env.NODE_ENV === "development") {
  await setupDevPlatform();
}

export default nextConfig;
```

4. 修改服务端路由运行时

对所有的 api 路由文件 route.ts 和所有的页面路由文件 page.tsx 都添加一行代码，指定使用 edge 运行时：

```
export const runtime = "edge";
```

5. 修改 `package.json` 文件，添加编译指令

```
"pages:build": "npx @cloudflare/next-on-pages",
"preview": "pnpmb pages:build && wrangler pages dev",
"deploy": "pnpm pages:build && wrangler pages deploy"
```

6. 在项目根目录通过命令行部署项目到 Cloudflare Pages

```
npm run deploy
```

在第一次运行 deploy 命令时，需要填写项目名称，跳转 Cloudflare 进行授权验证等。

发布完成后，就可以在 Cloudflare Workers and Pages 管理后台看到项目了：

20240814155355

跟 Vercel 一样，Cloudflare 也为发布的项目生成一个子域名：

`xxx.pages.dev`

，部署成功可直接公网访问，方便项目快速上线验证。

使用 Cloudflare 部署 nextjs 项目需要注意的点

1. Cloudflare 部署 nextjs 项目，只支持 edge 运行时

首先，我们使用

`thinkany.ai`

来搜一下什么是 nodejs 的 edge 运行时：

nodejs 的 edge 运行时是什么

搜索过程 4 步骤

来源 搜索

1. Rendering: Runtim... The Node.js runtime i... nextjs.org

2. js运行时| NextJs | ... Edge运行时. 在Next.js... www.yuansudong...

3. edge-runtime - NPM edge-runtime: Run an... npmjs.com

查看全部 » 10 来源

答案

Node.js的Edge运行时是一种轻量级的执行环境，它是Node.js API的一个子集，专为在边缘网络上运行而设计。以下是Edge运行时的主要特点和功能：

1. 轻量级设计：Edge运行时比完整的Node.js运行时更加精简，适合在分布式网络的边缘节点上快速部署和执行 2 。
2. 低延迟：由于其轻量级特性，Edge运行时能够以极低的延迟交付动态和个性化的内容，非常适合需要快速响应的应用场景 2 。
3. 有限的API支持：Edge运行时只支持Node.js API的一个子集，这意味着某些Node.js模块和功能在Edge环境中可能无法使用 5 。
4. 适用于简单功能：Edge运行时特别适合运行小型且简单的功能，如路由规则、重定向、重写和设置请求头等 1 。
5. 中间件支持：在Next.js等框架中，Edge运行时常用于执行中间件功能 1 。
6. 广泛部署：Edge函数可以部署在全球的边缘节点上，类似于Serverless架构，但更接近用户，从而提供更快的响应时间 6 。

20240814155740

要把 nextjs 项目部署到 Cloudflare，我们需要在项目中修改每一个 route.ts 和 page.tsx 文件，加一行代码显式指定使用 edge 运行时：

```
export const runtime = "edge";
```

2. Cloudflare 部署 nextjs 项目，需要改造 pg 客户端

如果 nextjs 项目用到了 postgres 数据库，客户端连接使用的是 pg 这个包，是不支持 edge 运行时的

```
import { Pool } from "pg";
```

我们需要更换一个支持 edge 运行时的 postgres 客户端，比如 neon，把操作 postgres 数据库的逻辑改成这样：

```
import { neon } from '@neondatabase/serverless';

async function getData() {
  const sql = neon(process.env.DATABASE_URL);
  const response = await sql`SELECT version()`;
  return response[0].version;
}
```

如果是原生的 postgres 数据库，这种改造是 OK 的，部署到 Cloudflare 也没问题。

如果数据库用的是 supabase，这种方案就行不通，原因是 neon 并不兼容 supabase。我们需要使用 supabase 官方客户端进行改造：

```
import { createClient } from "@supabase/supabase-js";

export function getSupabaseClient() {
  const client = createClient(
    process.env.SUPABASE_URL || "",
    process.env.SUPABASE_ANON_KEY || ""
  );

  return client;
}
```

`@supabase/supabase-js`
是支持 edge 运行时的，改造完可以部署到 Cloudflare，但是我遇到了一个问题，就是这个客户端不支持
`select * from xxx order by random()`
随机排序操作。

3. Cloudflare 部署 nextjs 项目，需要改造对 fs / http 等 nodejs API 的依赖

因为 edge 运行时不支持 fs / http 这种 nodejs API，所有依赖这两个 API 的逻辑都需要改造：

比如读取本地文件的逻辑：

```
const data = fs.readFileSync(dataFile, "utf8");
```

依赖了 fs API，edge 运行时不支持，可以改成用 fetch API 读取远端文件。

比如
`google-one-tap`
这个 npm 包，依赖了 axios 请求 google 登录 API，而 axios 依赖 http API，不兼容 edge 运行时，我们只能把这个 npm 拉到本地，再把对应的 http 请求换成 fetch API。

“

具体改造点可以参考：https://github.com/all-in-aigc/sorafm/compare/main...feature/deploy-to-cloudflare

Cloudflare 有哪些可白嫖的真香服务

Cloudflare 提供了很多常用的网站管理服务，其中大部分功能都有免费的使用额度，对于小网站而言，白嫖就够了。

1. DNS 解析

我们对 Cloudflare 使用最多的场景，可能就是 DNS 解析了。

Cloudflare 的 DNS 服务被认为是世界上最快的 DNS 解析器之一，平均响应时间为 11 毫秒，覆盖超过 330 座城市。这种快速的 DNS 解析可以提高网站的整体访问速度。

比如在 namecheap / godaddy 这些平台注册的域名，我们可以选择接入到 Cloudflare 做 DNS 解析。

20240814162414

2. 安全防护

Cloudflare 可以帮助提升网站的安全性。

比如可以给网站开启交互式验证，拒绝机器模拟请求，防止网站 API 被盗刷。

20240814172528

还可以配置 DDoS 防护，配置防火墙，拦截非法请求。

可以对指定访问路径配置速率限制，设置 ip 黑/白名单等。

20240814163241

3. Workers and Pages

可以把一些静态网站，比如个人博客 / landing page 之类的站点，托管到 Cloudflare Pages，

可以使用 Cloudflare Workers 部署定时程序 / 脚本 / API Proxy 等。

甚至可以托管 nextjs 之类的全栈应用，非常方便，且费用很低。

20240814162558

4. D1

可以使用 Cloudflare D1 来替代 supabase，做云数据库托管，比起 Vercel Postgres 之类的数据库方案，费用要低不少。

就是不清楚，Cloudflare 有没有提供数据迁移服务，如果能一键把 supabase 的数据迁移到 D1 就好了。

20240814162823

5. R2

可以使用 Cloudflare R2 代替 AWS S3 做文件存储，搭建自己的图床，存储网站的图片文件。

Cloudflare 兼容 AWS S3 的存储 API，还提供了一键迁移工具，使用起来非常方便，存储费用也很低。

我已经把 aiwallpaper.shop / aicover.design / gpts.works 几个项目的图片都迁移到 R2 了。

20240814162622

Cloudflare 上面还有很多实用的服务，我还没有探索完全，就不一一列举了。难怪大家都说 Cloudflare（cf）是赛博菩萨，服务好用，又让白嫖，真的太香。

总结

本文用实际的例子介绍了把 nextjs 项目从 Vercel 迁移到 Cloudflare 的全过程。

拥抱赛博菩萨 Cloudflare（cf），节省好几万的服务成本。

感谢 「1024 全栈开发社群」成员 @Deniffer 和 @7 提供的建议，我们这个群的技术讨论氛围真好。

> 欢迎新朋友加入「1024 全栈开发社群」，跟艾逗笔一起学全栈开发，让你的创意更快落地成产品。

阅读原文