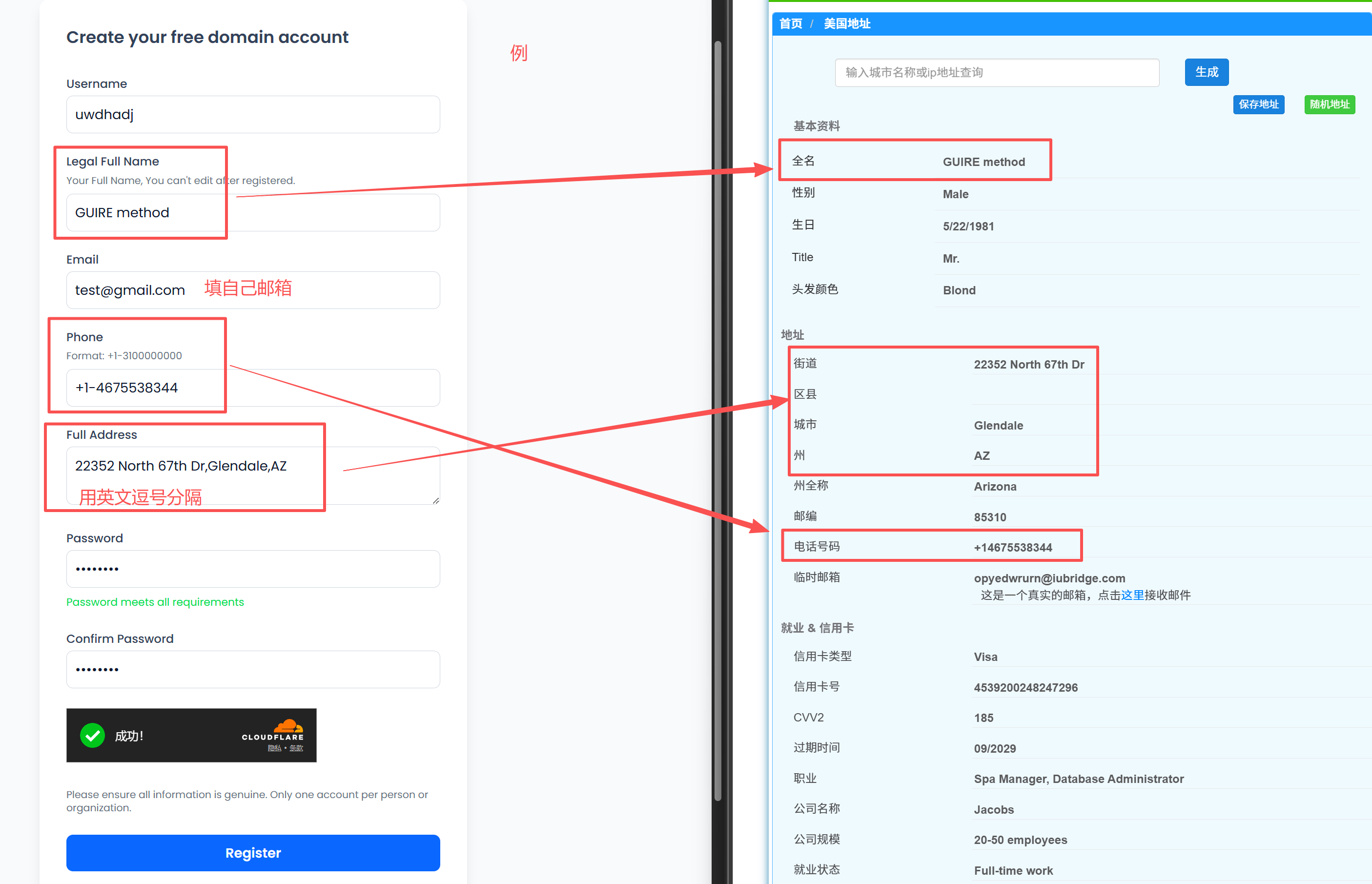
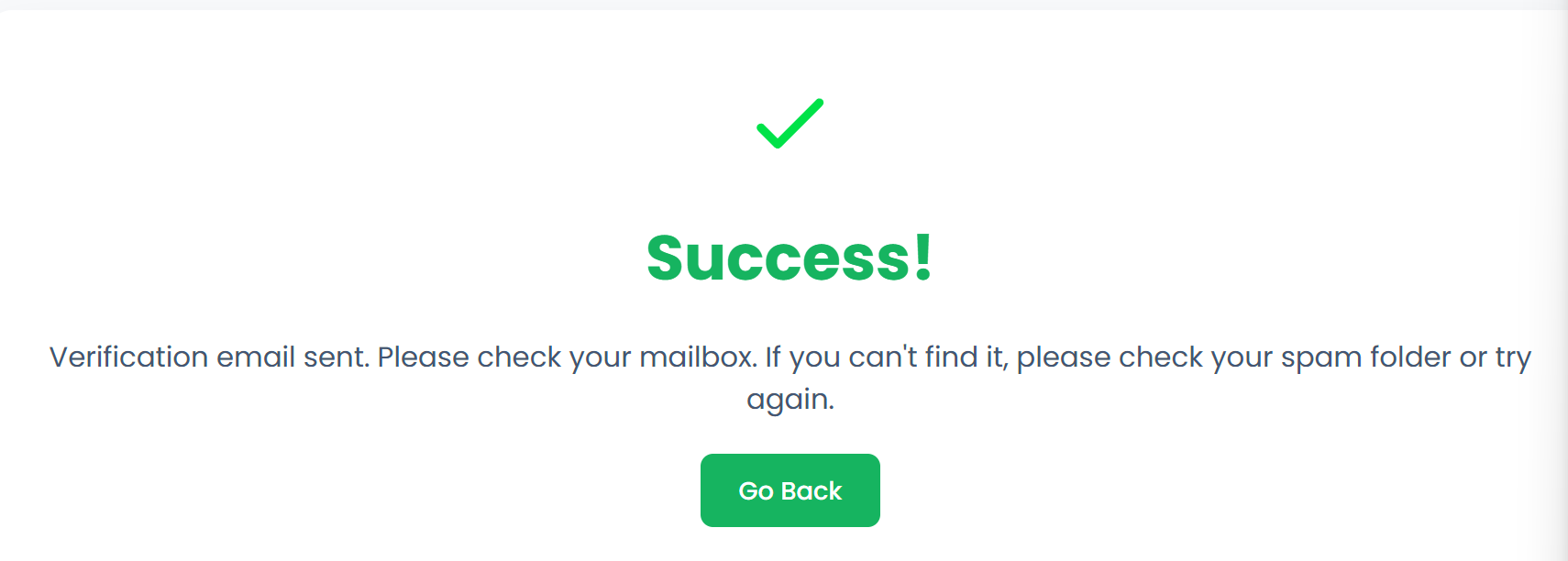
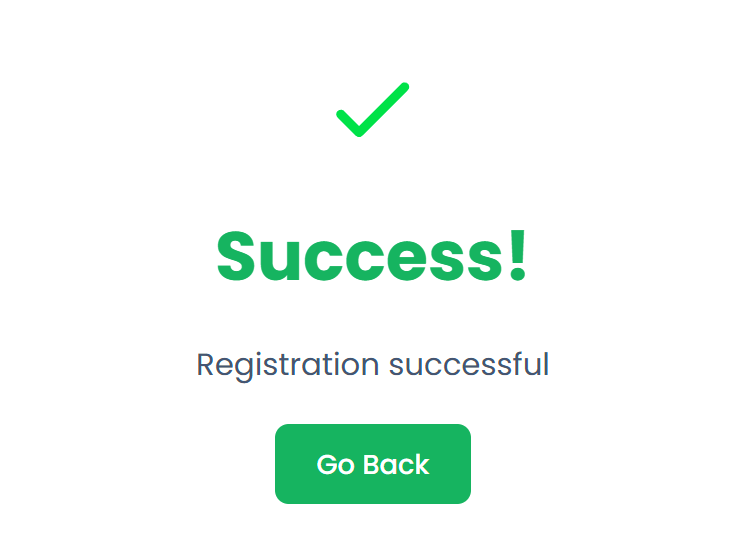
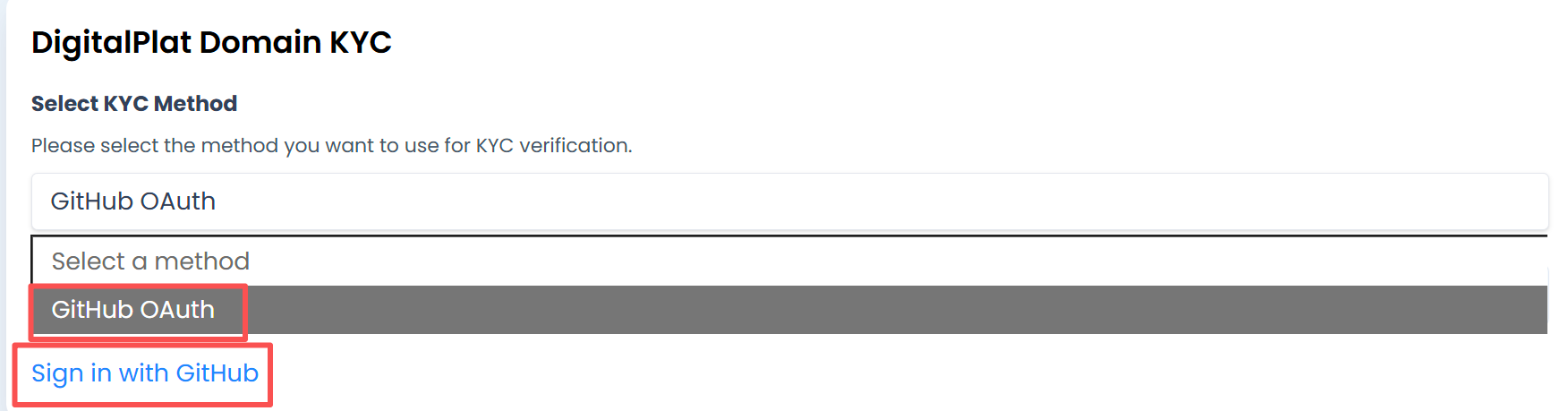
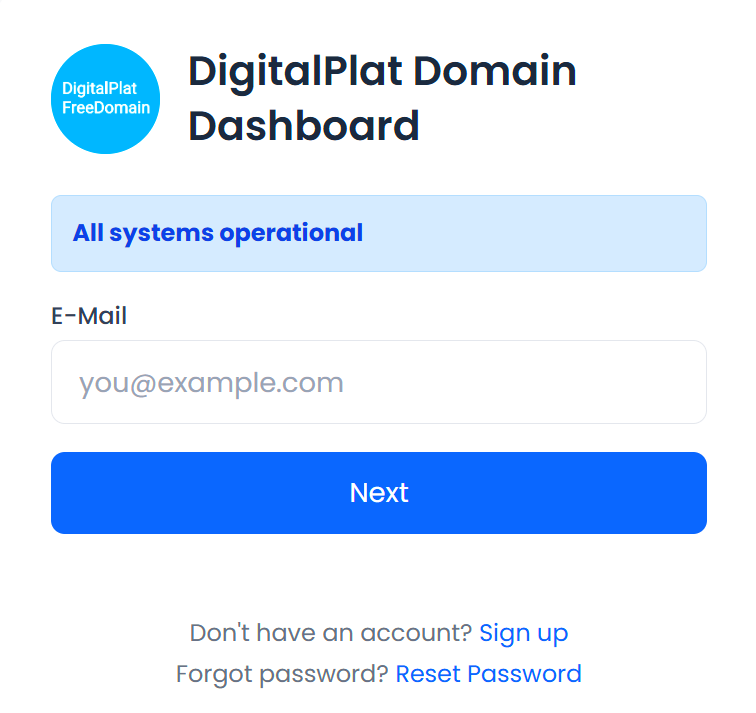
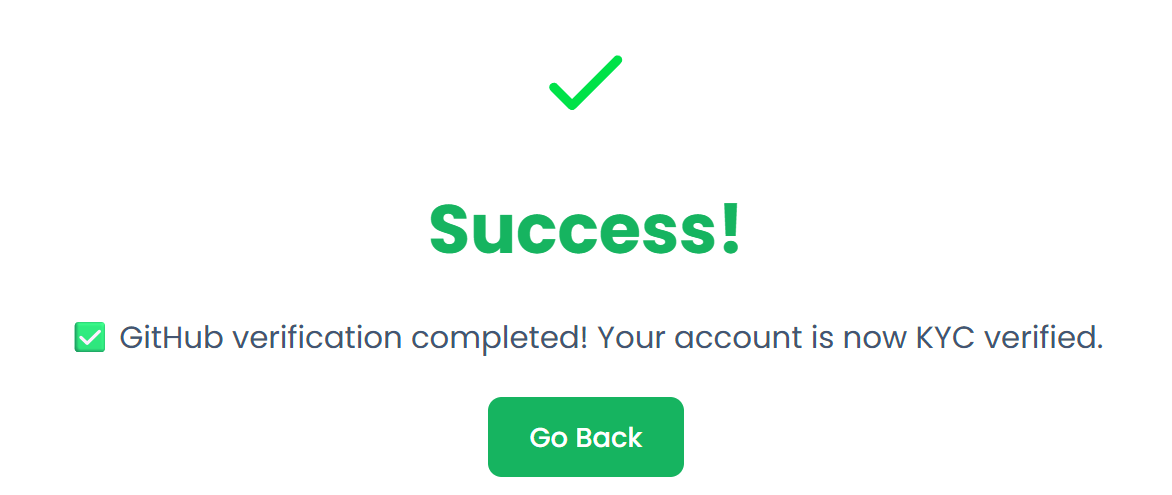
1. 注册一个outlook邮箱  
   2、用outlook邮箱注册一个github账号

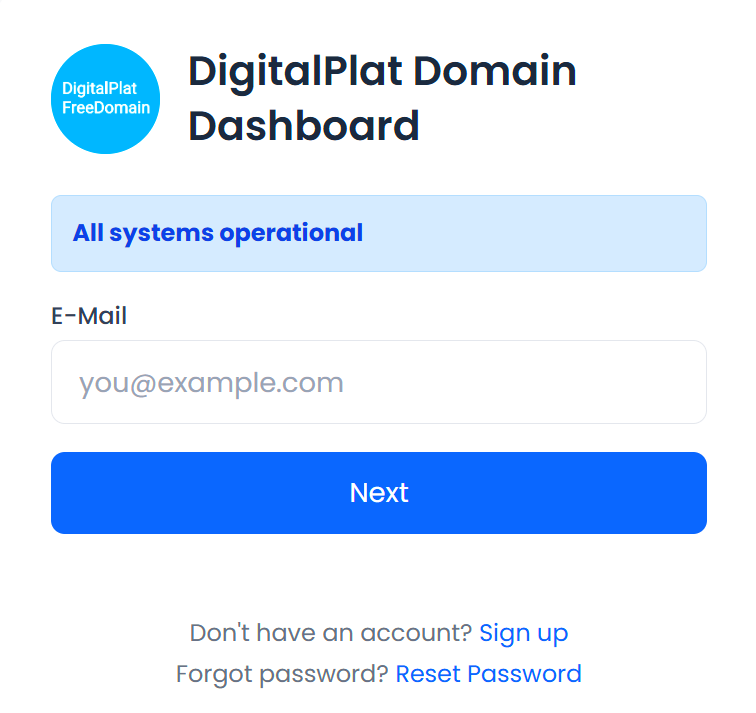
3、注册一个cloudflare账号  
4、使用美国地址生成器生成信息www.meiguodizhi.com  
5、https://dash.domain.digitalplat.org注册账号  




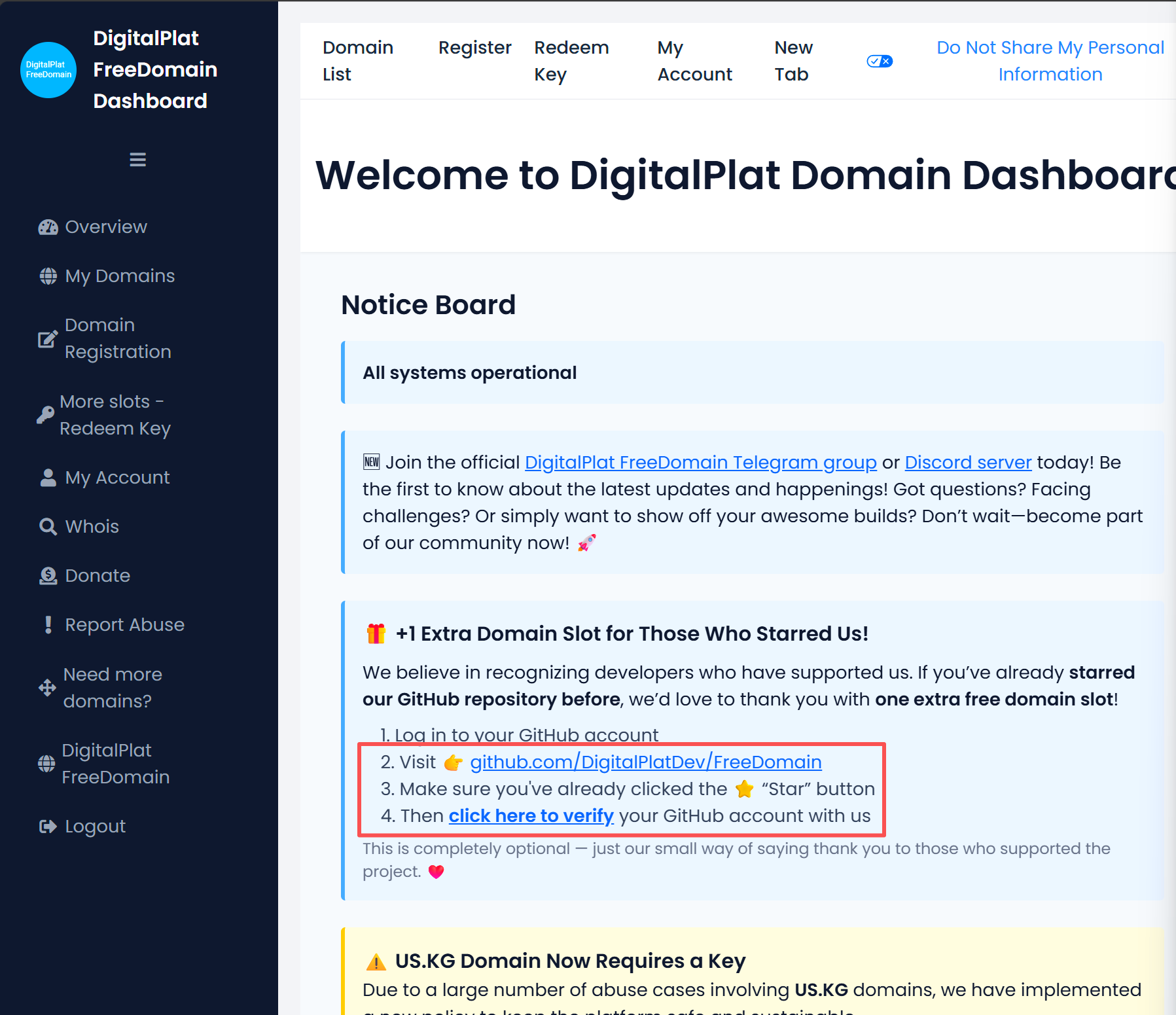
注册完后邮箱验证，信息可能在垃圾箱里，验证完有下图信息  


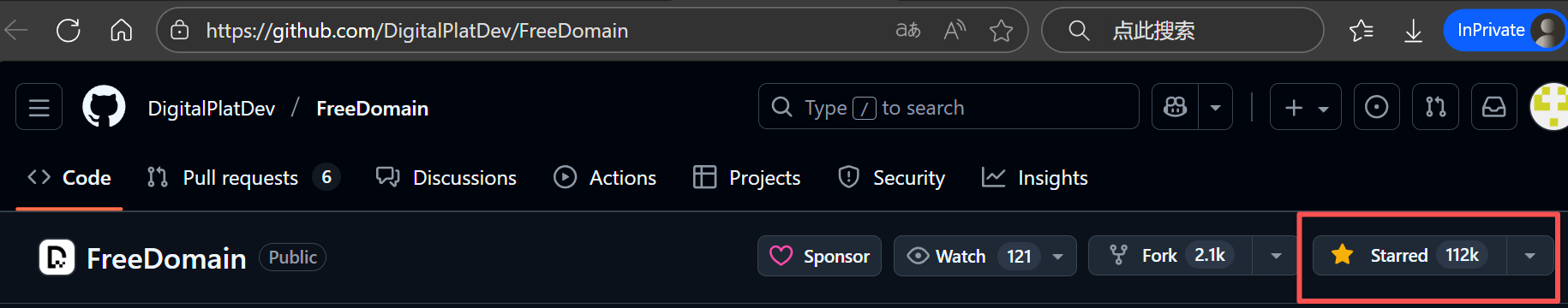
6、github账号绑定域名网站  
注册完先返回登录域名网站  


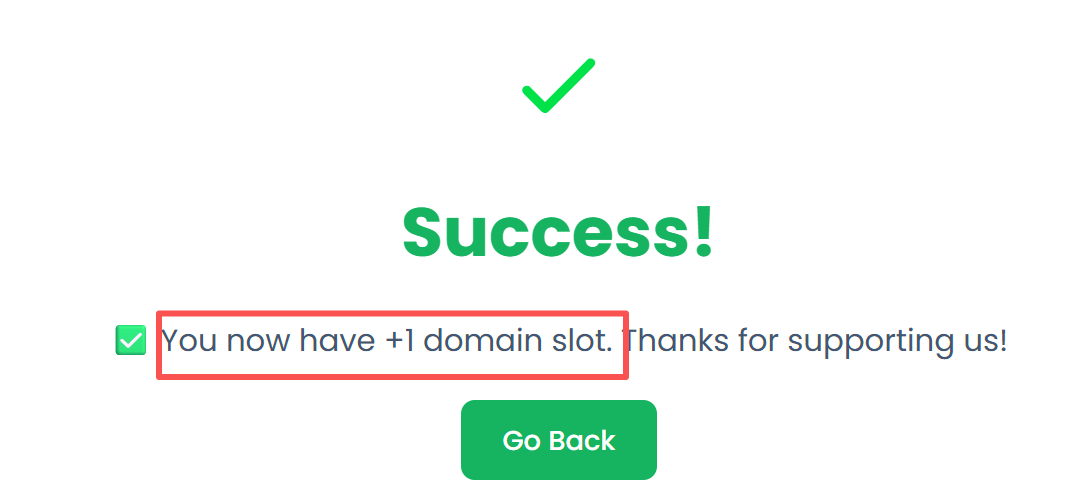
点击后授权绑定，绑定成功如下图  


重新登录域名网站  


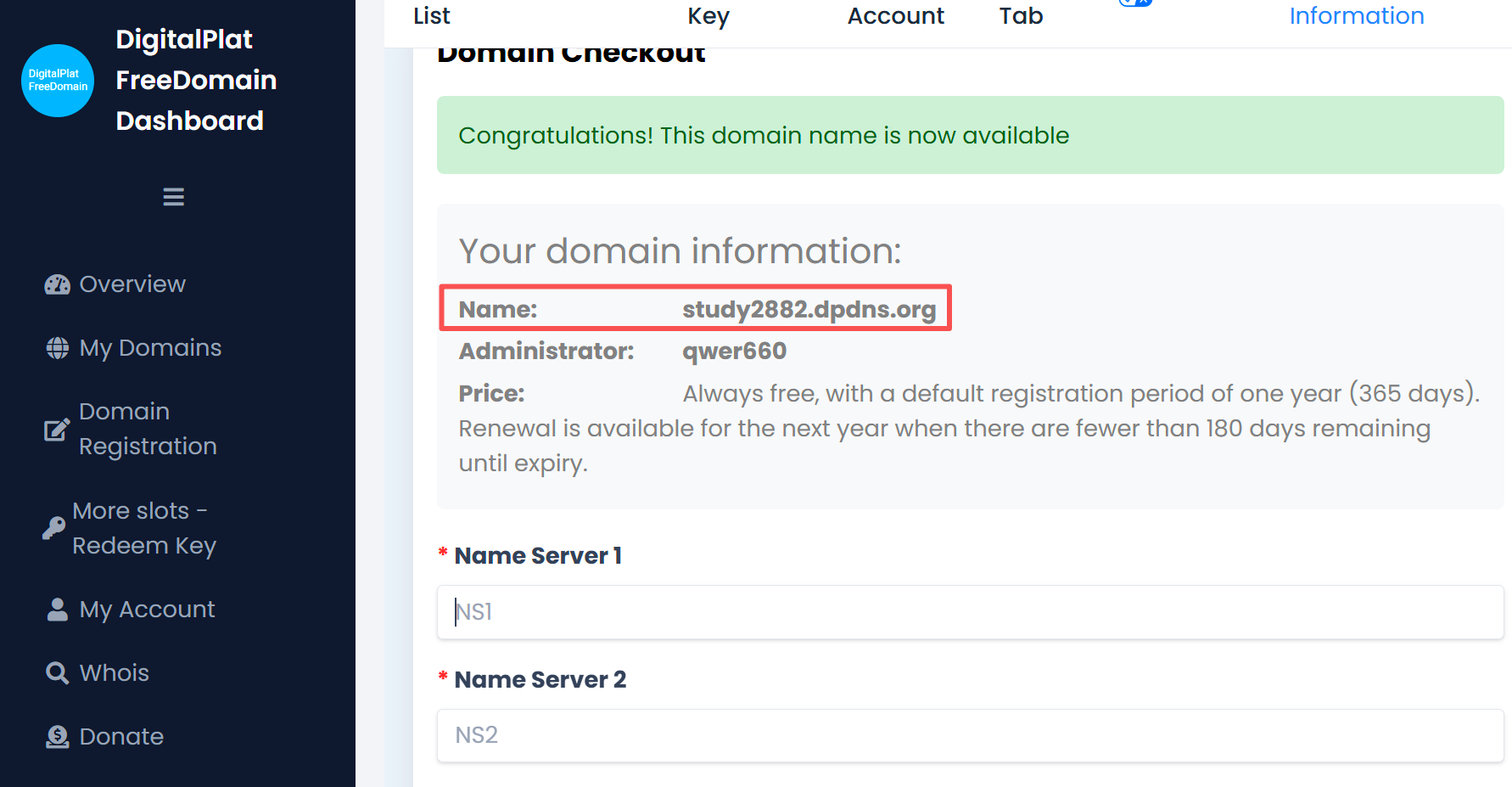
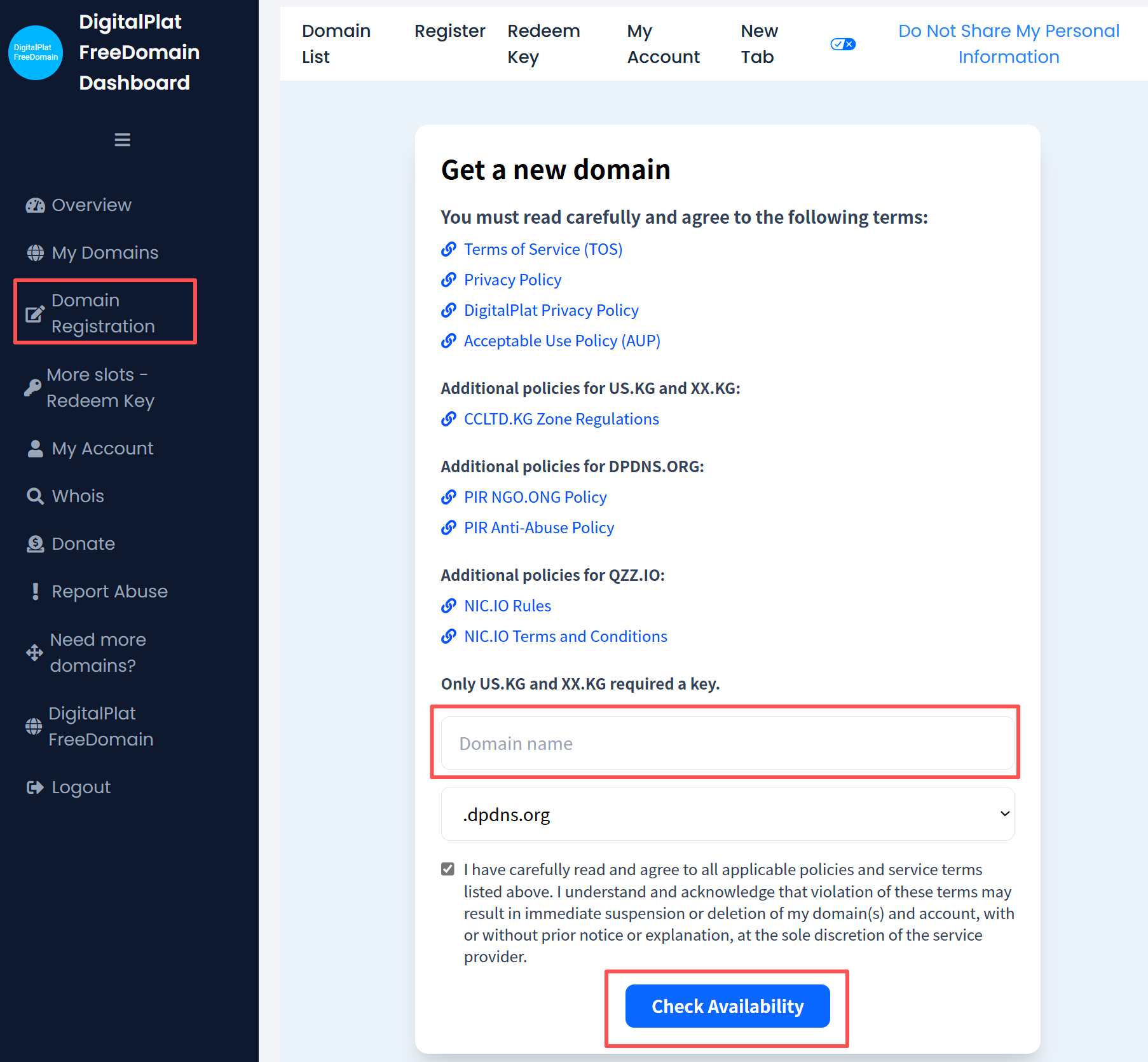
进入操作面板页面，点击2.链接star该项目然后点击4.验证链接可以多获得一个域名





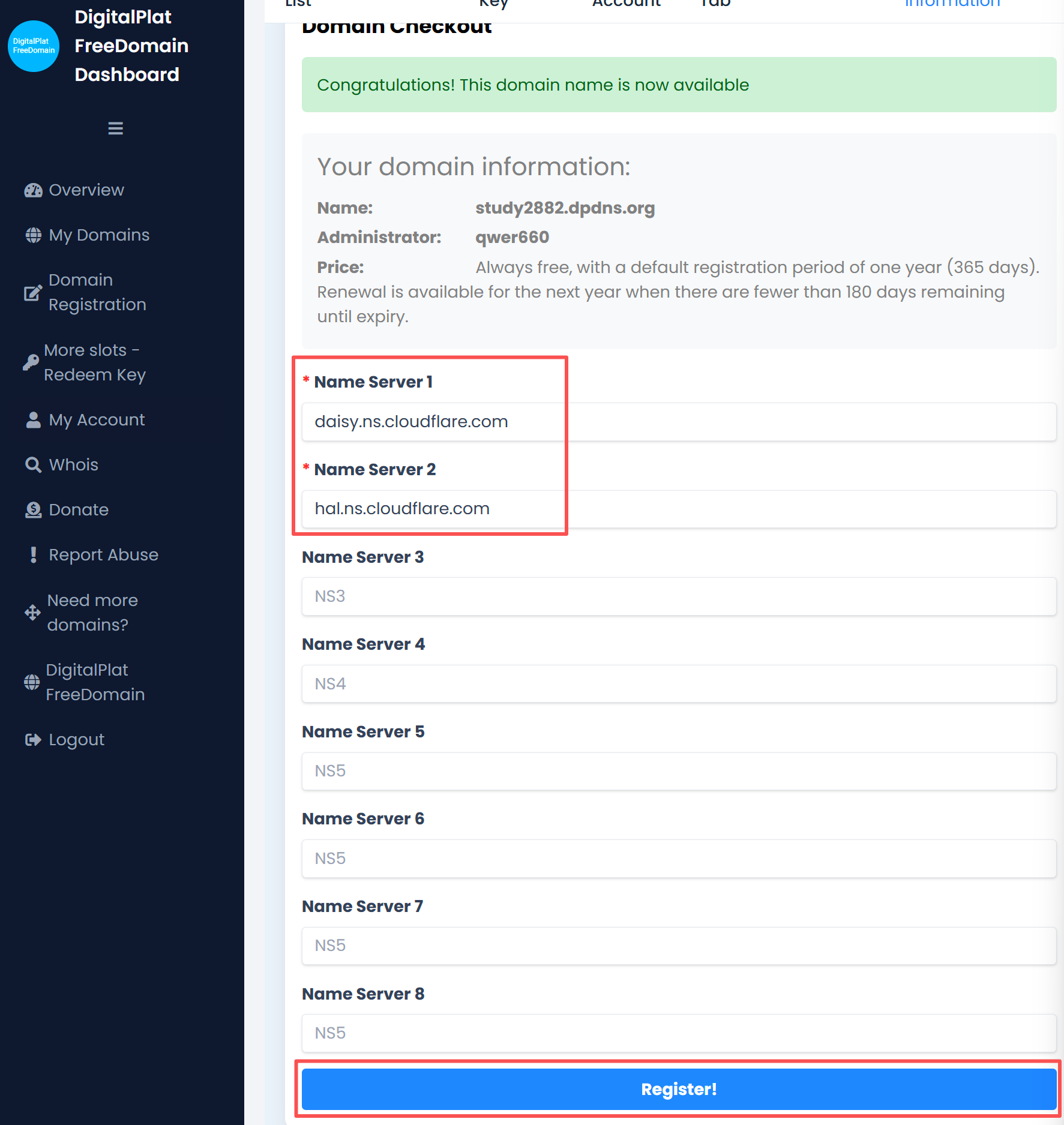
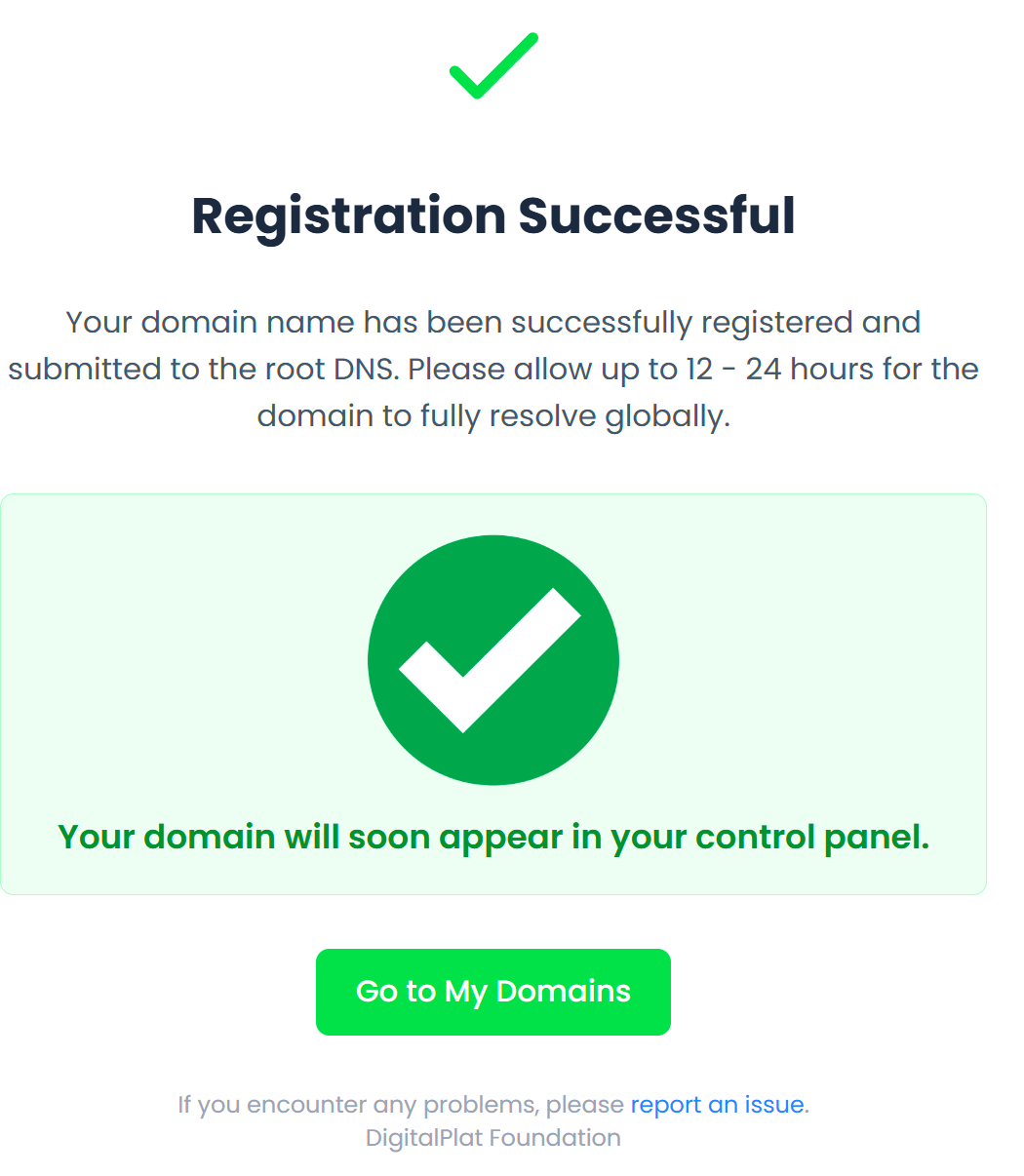


1. 创建域名

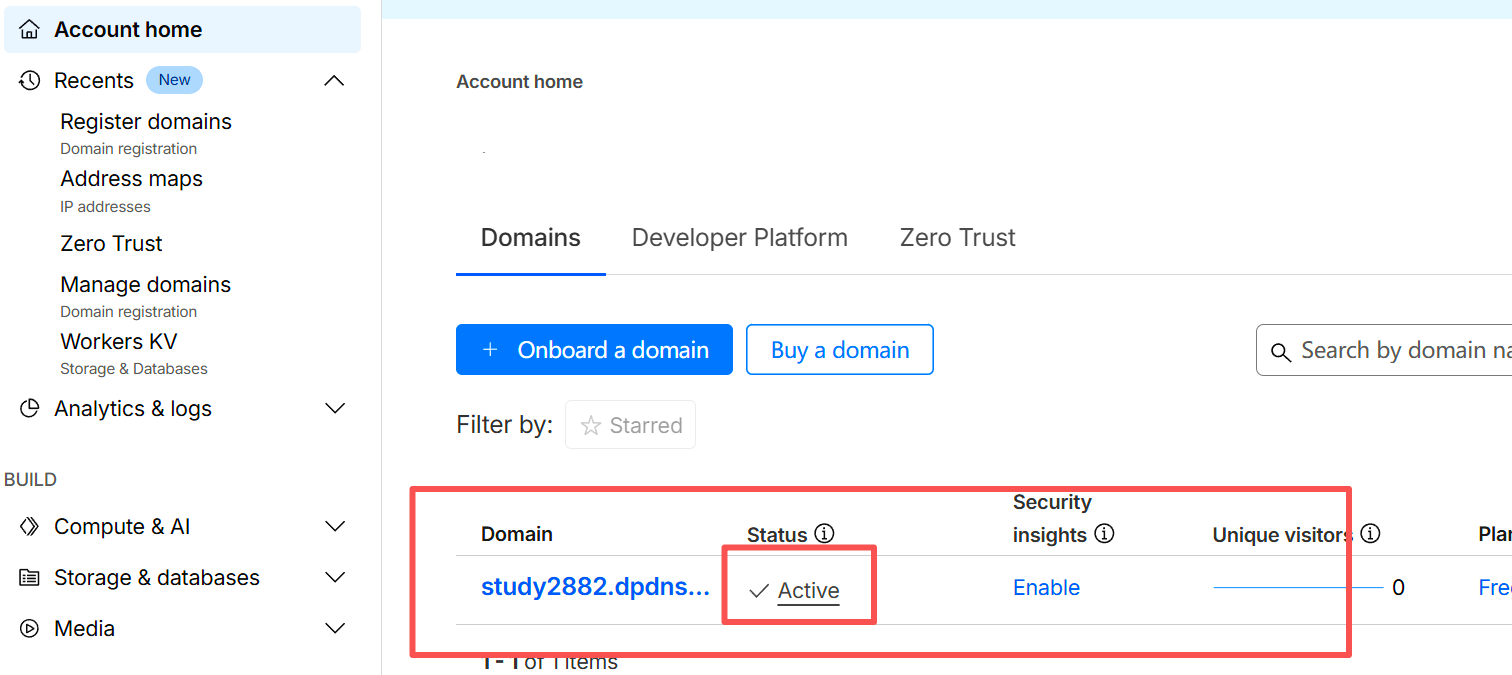
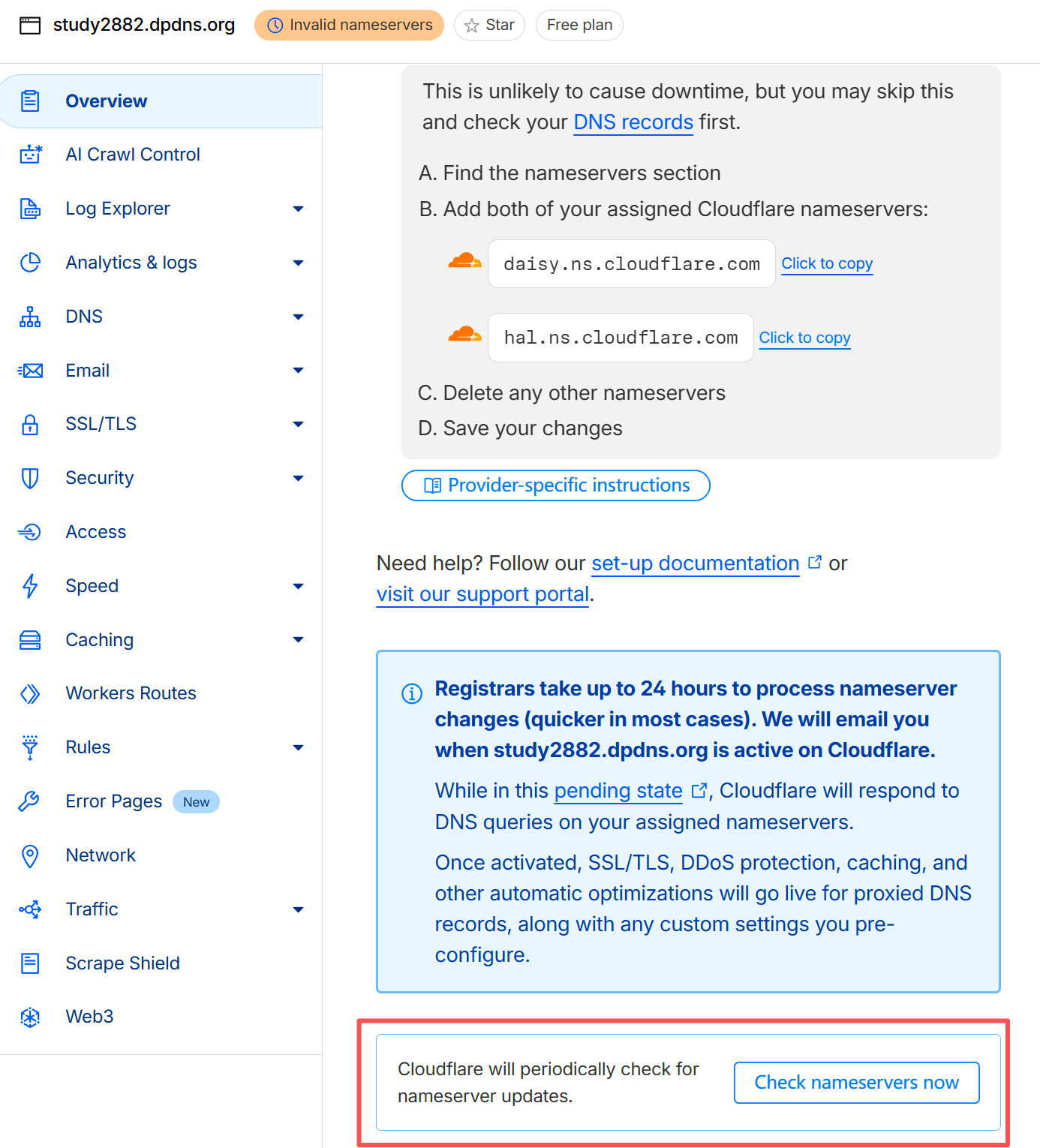


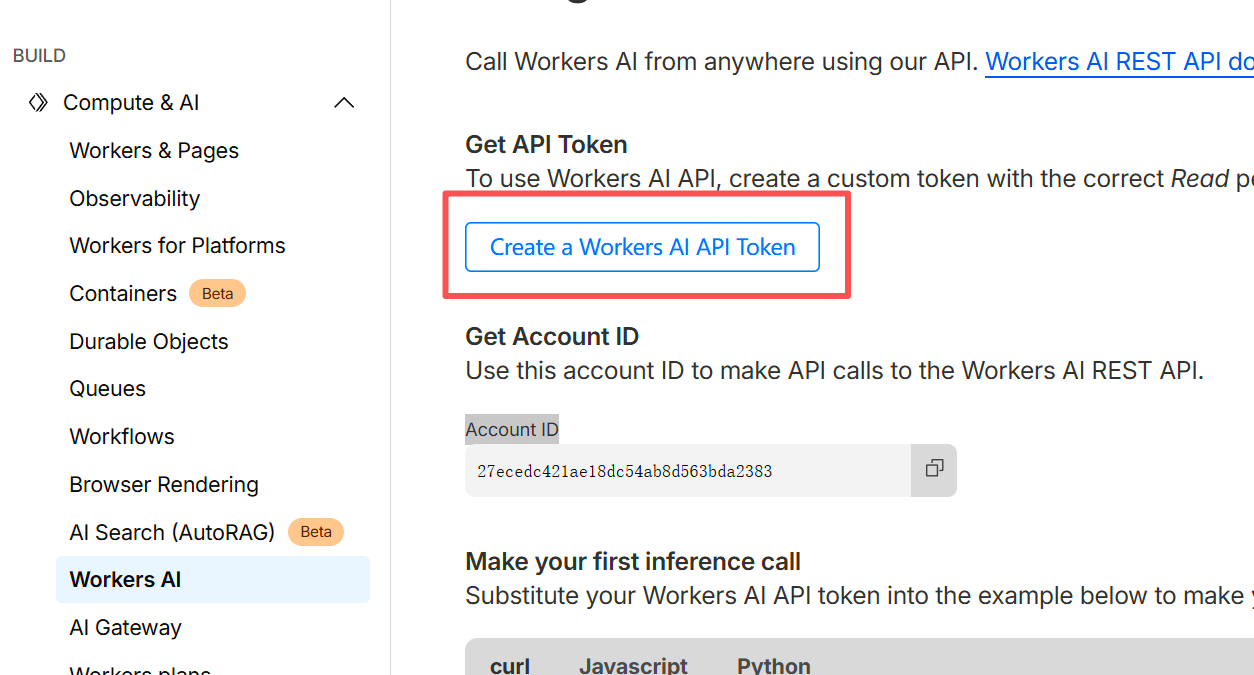
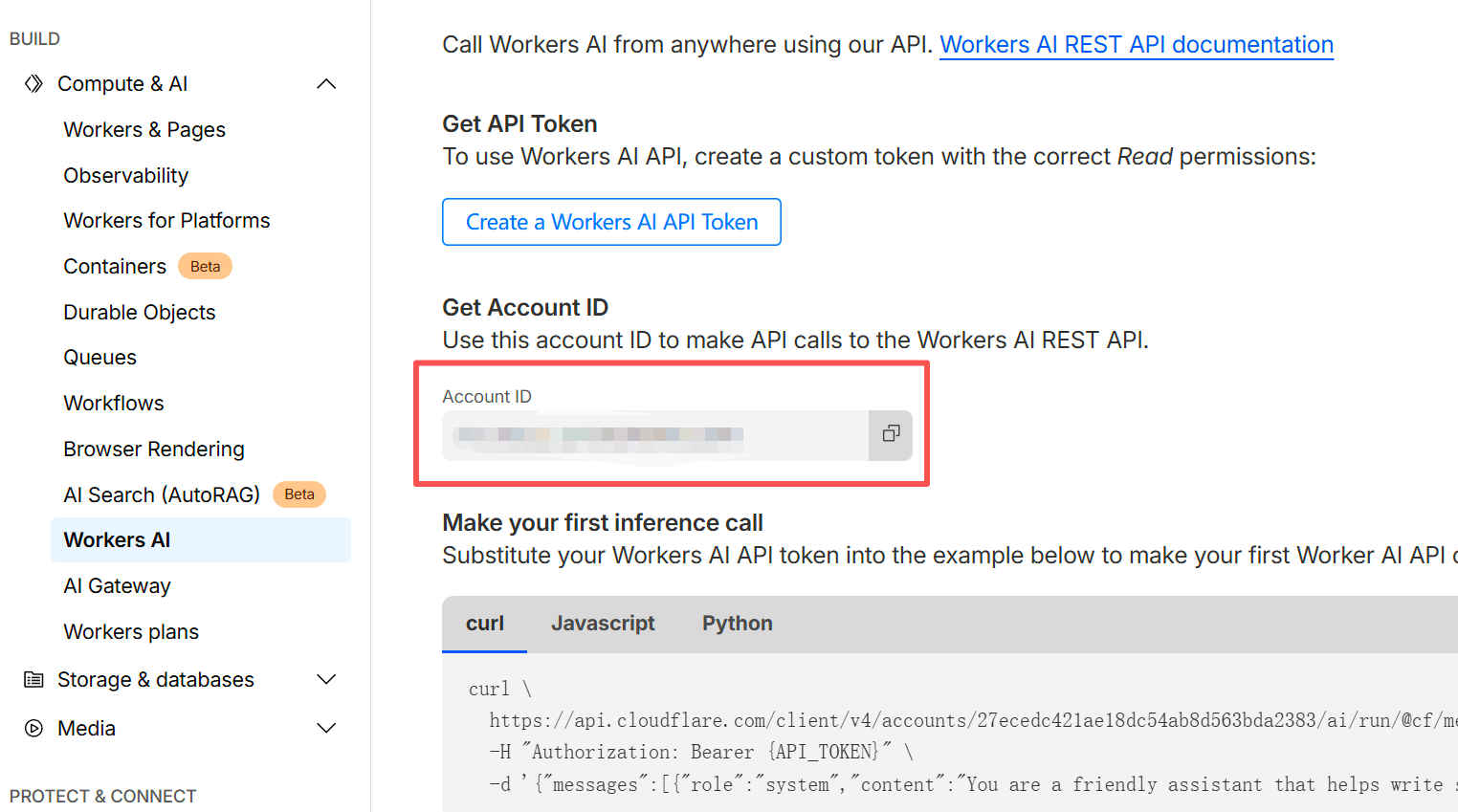
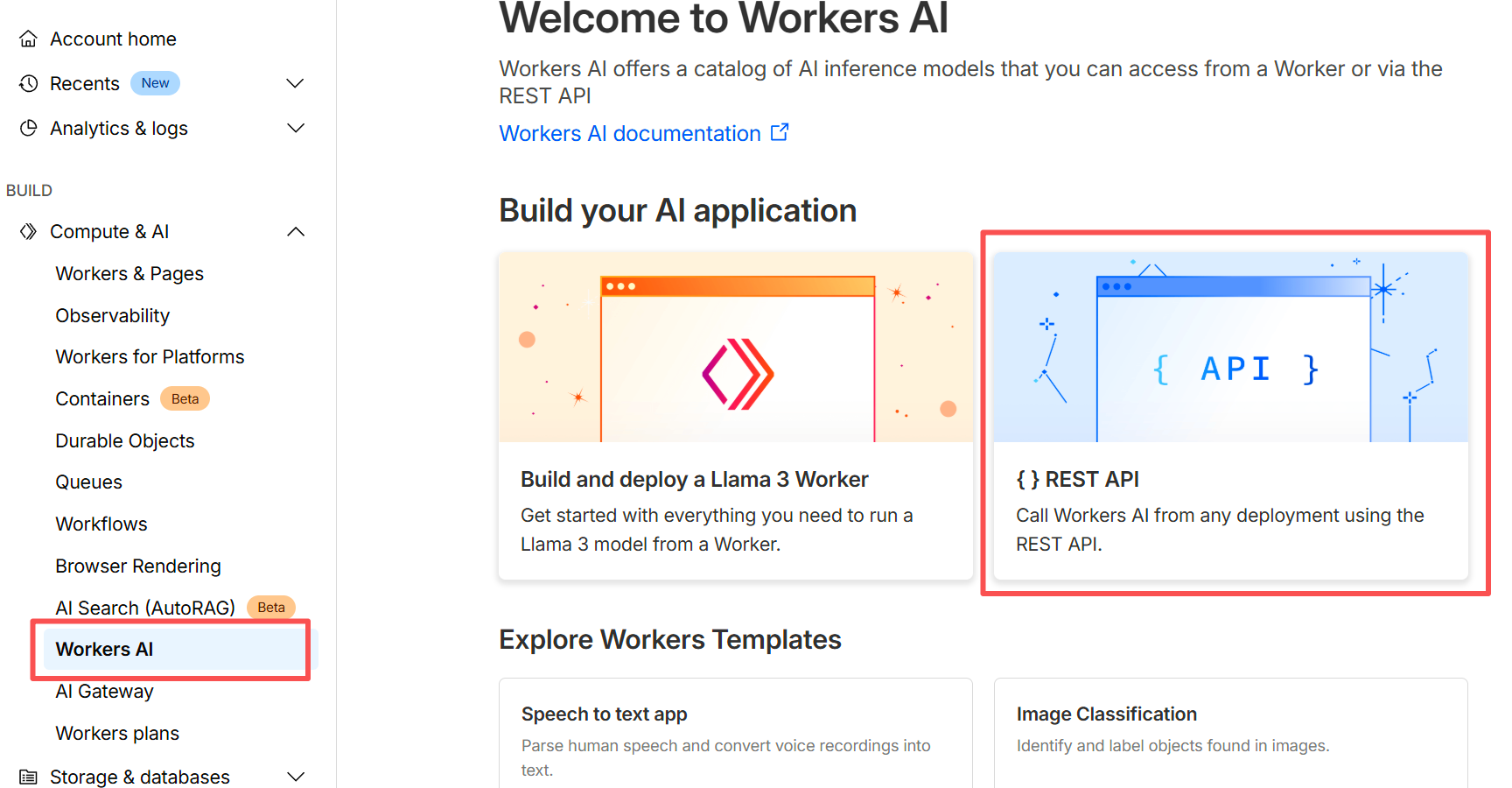
8、cloudflare托管域名  
来到cloudflare，输入刚创建的域名

  
将名称服务器粘贴过去，点击注册

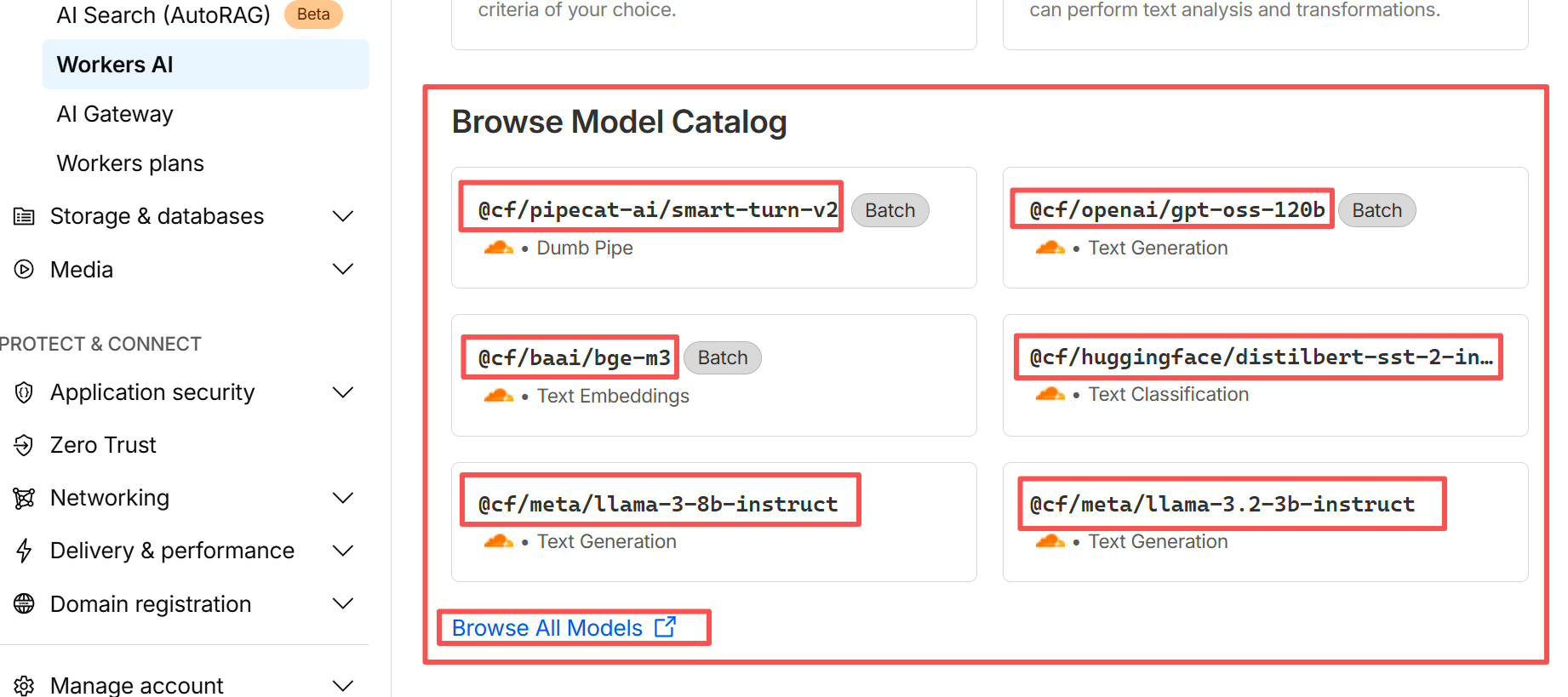
  


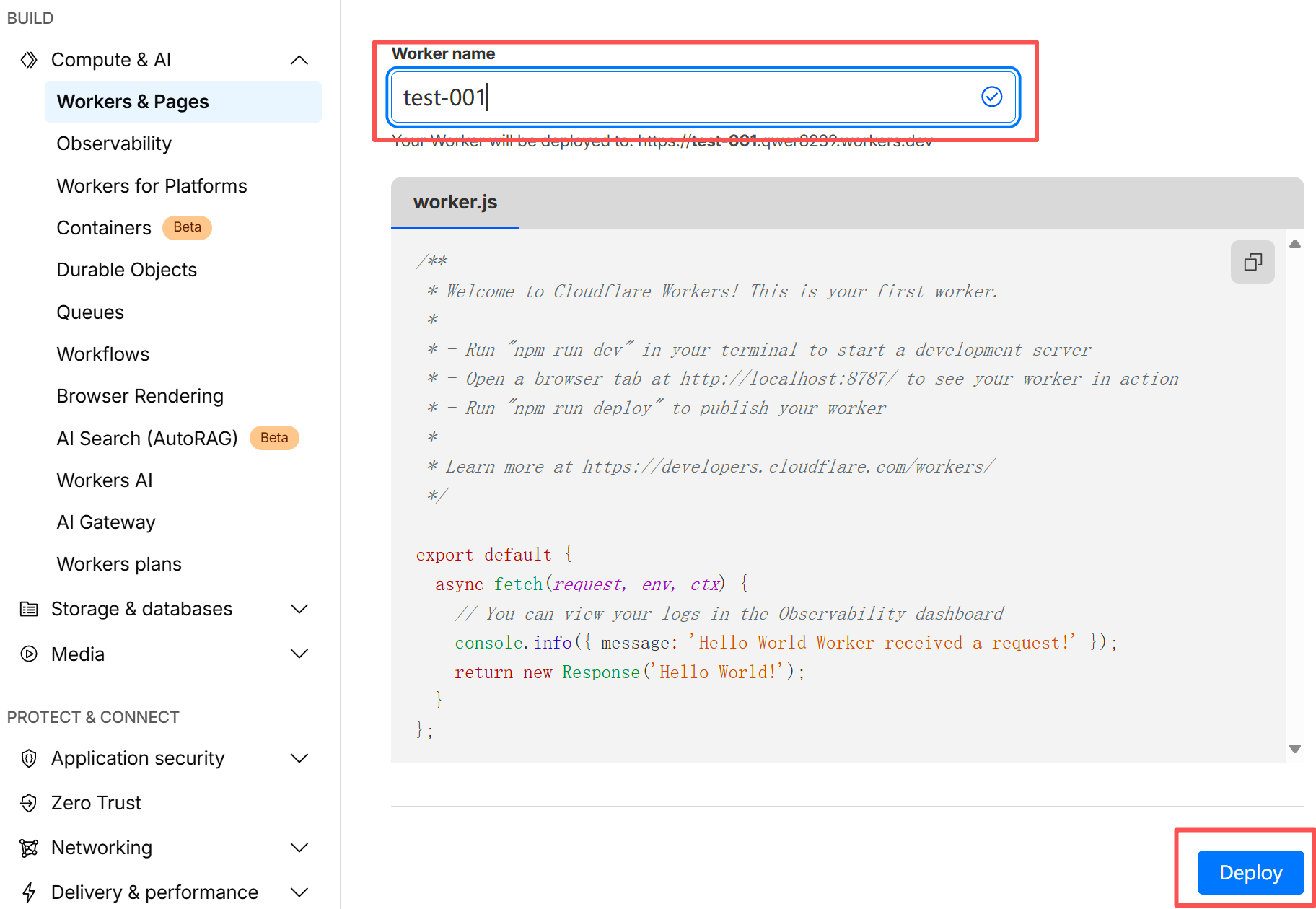
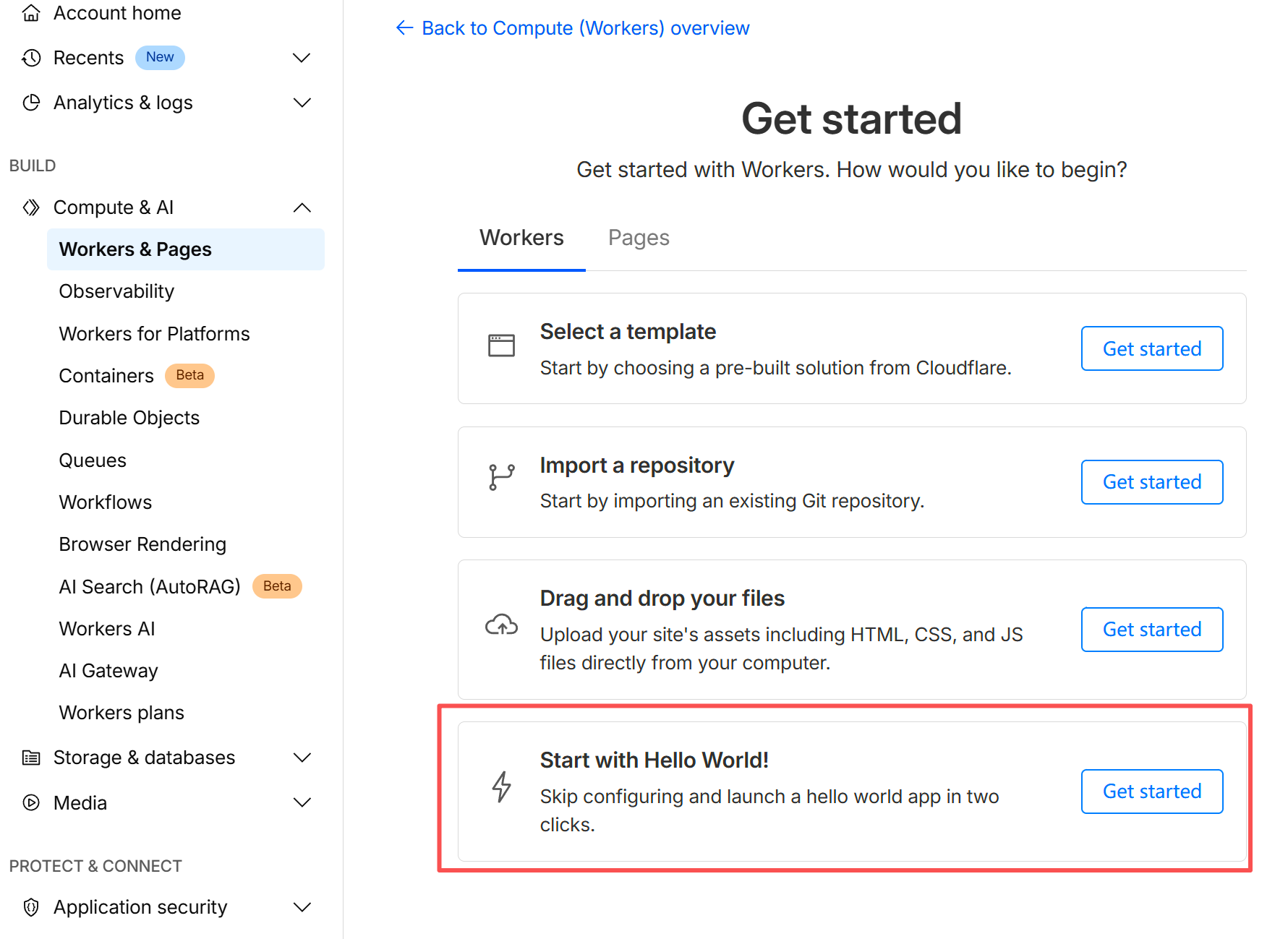
可以点击立即检查。域名活动状态即可

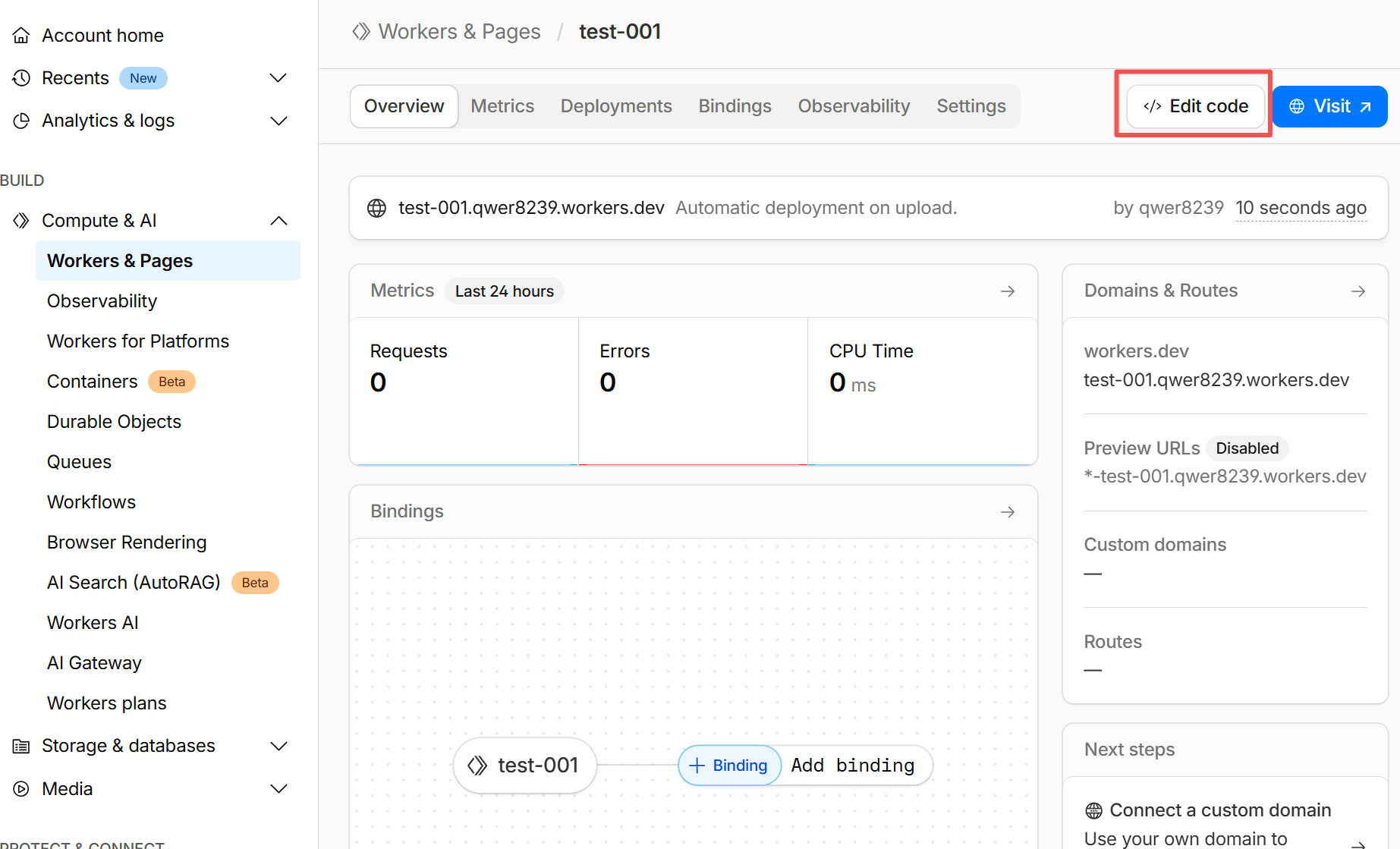
  
第二个域名也是一样的步骤

9、获取3个参数”Account ID”,”API Token”,”MODEL NAME”，剪贴板记录好  


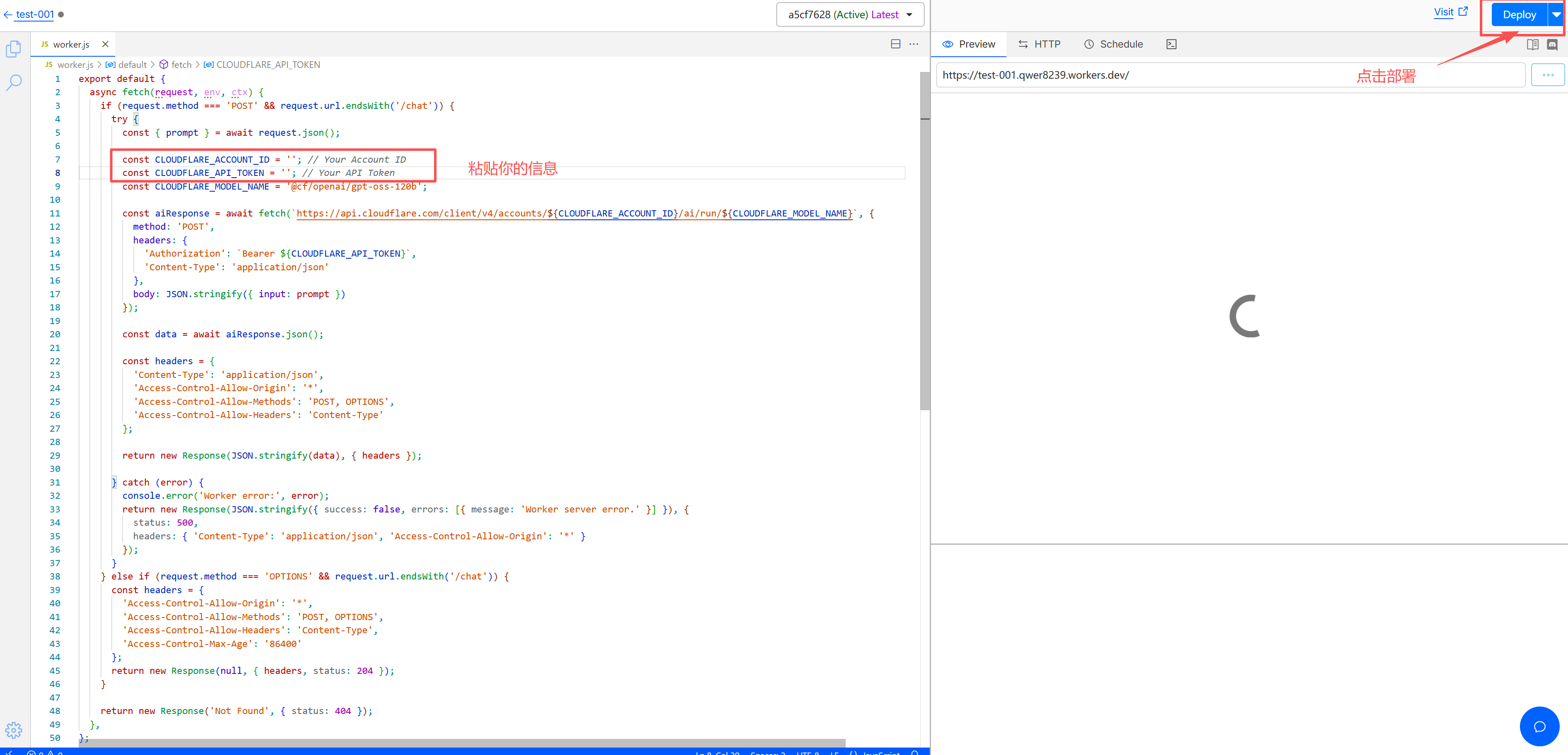




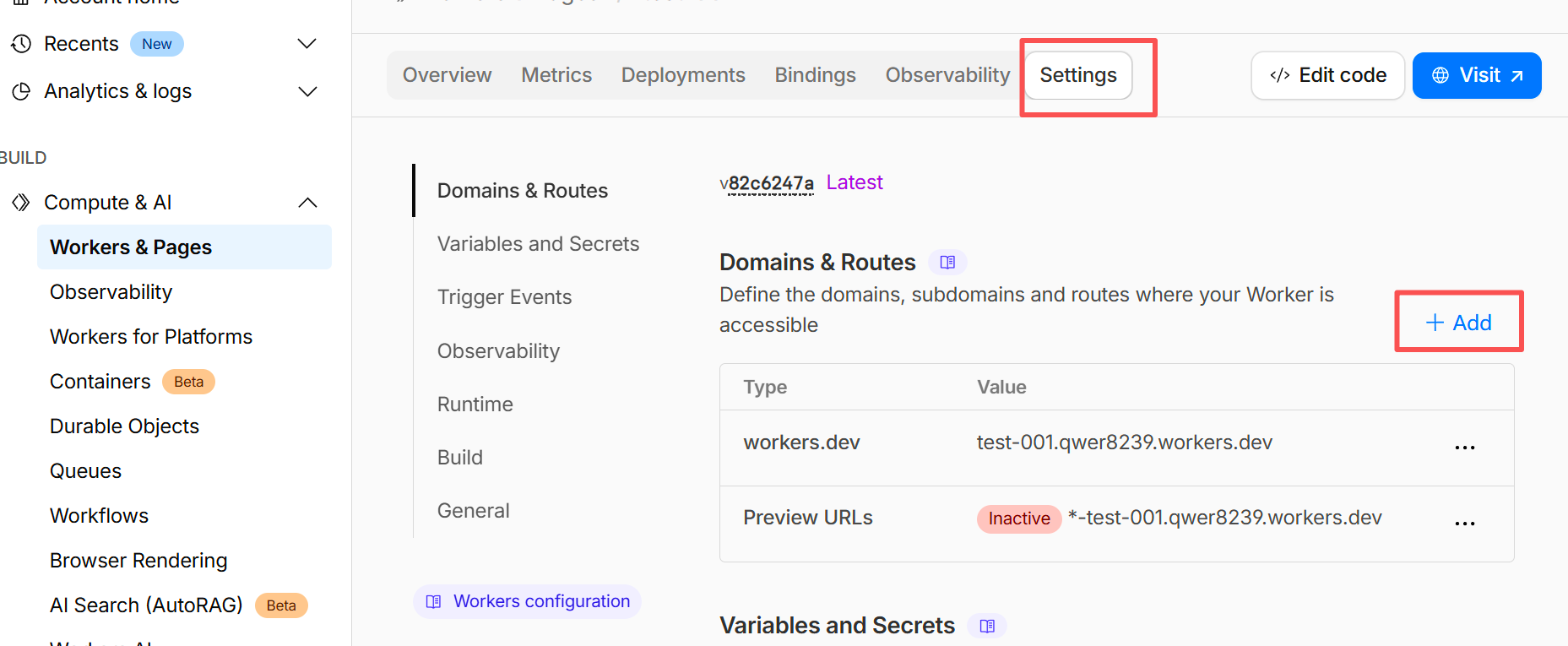
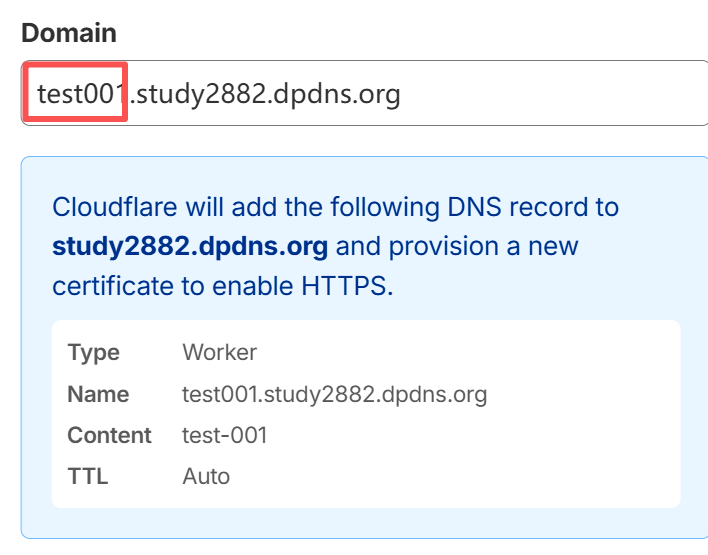
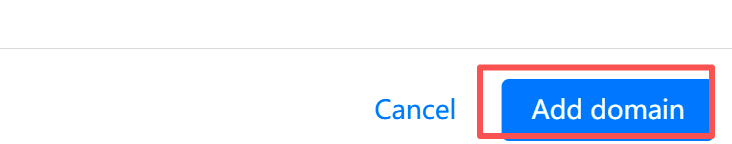
10、部署  


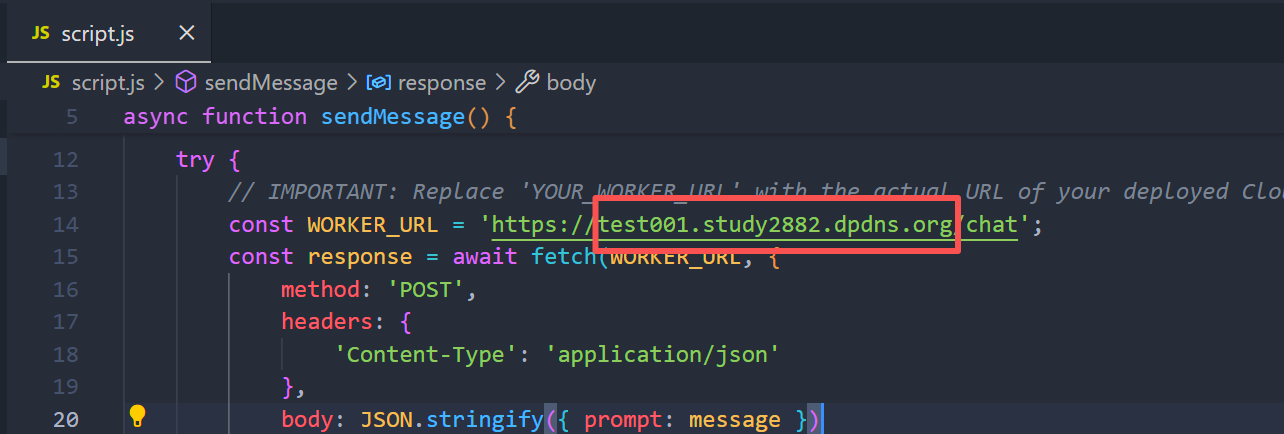
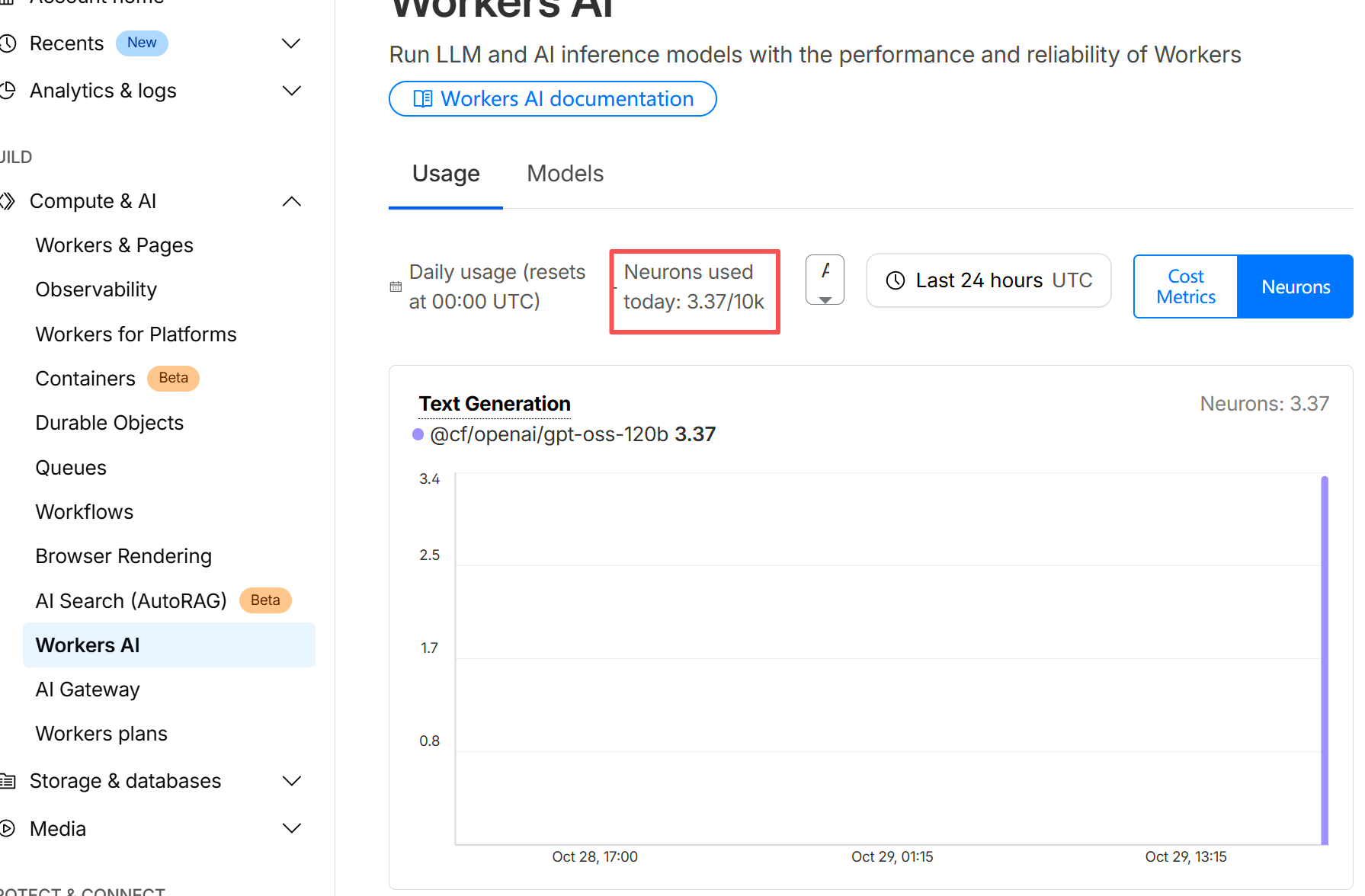


worker.js代码在文档底部



添加创建好的域名

  
域名前加上子域名  
  
点击添加  


填入你的域名  
  
打开index.html页面测试  
  
  
注意：  
1、github打不开的可以用Watt Toolkit加速  
2、添加自定义域名后不需要科学上网即可成功调用  
3、每日有一定限制，可以创建多个账号使用  


测试代码  
index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AI Chat</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="chat-container">

        <div class="chat-box" id="chat-box">

*<!-- Chat messages will be displayed here -->*

        </div>

        <div class="chat-input">

            <input type="text" id="user-input" placeholder="Type your message...">

            <button id="send-button">Send</button>

        </div>

    </div>

    <script src="script.js"></script>

</body>

</html>

style.css

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

    background-color: #f0f2f5;

}

.chat-container {

    width: 100%;

    max-width: 600px;

    height: 80vh;

    background-color: #fff;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    display: flex;

    flex-direction: column;

    border-radius: 8px;

    overflow: hidden;

}

.chat-box {

    flex-grow: 1;

    padding: 20px;

    overflow-y: auto;

    border-bottom: 1px solid #eee;

}

.message {

    margin-bottom: 15px;

    display: flex;

}

.message.user {

    justify-content: flex-end;

}

.message.ai {

    justify-content: flex-start;

}

.message-content {

    max-width: 70%;

    padding: 10px 15px;

    border-radius: 20px;

    line-height: 1.4;

}

.message.user .message-content {

    background-color: #007bff;

    color: #fff;

    border-bottom-right-radius: 5px;

}

.message.ai .message-content {

    background-color: #e2e6ea;

    color: #333;

    border-bottom-left-radius: 5px;

}

.chat-input {

    display: flex;

    padding: 15px;

    border-top: 1px solid #eee;

}

.chat-input input {

    flex-grow: 1;

    padding: 10px 15px;

    border: 1px solid #ddd;

    border-radius: 20px;

    margin-right: 10px;

    font-size: 16px;

}

.chat-input button {

    background-color: #007bff;

    color: #fff;

    border: none;

    border-radius: 20px;

    padding: 10px 20px;

    cursor: pointer;

    font-size: 16px;

    transition: background-color 0.2s;

}

.chat-input button:hover {

    background-color: #0056b3;

}

script.js

const chatBox = document.getElementById('chat-box');

const userInput = document.getElementById('user-input');

const sendButton = document.getElementById('send-button');

async function sendMessage() {

    const message = userInput.value.trim();

    if (message === '') return;

    appendMessage(message, 'user');

    userInput.value = '';

    try {

*// IMPORTANT: Replace 'YOUR\_WORKER\_URL' with the actual URL of your deployed Cloudflare Worker*

        const WORKER\_URL = 'https:// /chat';

        const response = await fetch(WORKER\_URL, {

            method: 'POST',

            headers: {

                'Content-Type': 'application/json'

            },

            body: JSON.stringify({ prompt: message })

        });

        const data = await response.json();

        if (data.success) {

*// The response structure for gpt-oss-120b is complex, extract the actual message*

            const aiResponse = data.result.output.find(*item* => *item*.type === 'message')?.content[0]?.text;

            if (aiResponse) {

                appendMessage(aiResponse, 'ai');

            } else {

                appendMessage('Error: Could not parse AI response.', 'ai');

                console.error('AI response structure:', data);

            }

        } else {

            appendMessage(`Error: ${data.errors[0].message}`, 'ai');

            console.error('API Error:', data.errors);

        }

    } catch (error) {

        appendMessage('Error: Could not connect to the AI service.', 'ai');

        console.error('Fetch error:', error);

    }

    chatBox.scrollTop = chatBox.scrollHeight; *// Scroll to bottom*

}

function appendMessage(*text*, *sender*) {

    const messageElement = document.createElement('div');

    messageElement.classList.add('message', *sender*);

    const contentElement = document.createElement('div');

    contentElement.classList.add('message-content');

    contentElement.textContent = *text*;

    messageElement.appendChild(contentElement);

    chatBox.appendChild(messageElement);

}

sendButton.addEventListener('click', sendMessage);

userInput.addEventListener('keypress', (*event*) => {

    if (*event*.key === 'Enter') {

        sendMessage();

    }

});

*// Initial welcome message*

appendMessage('Hello! How can I help you today?', 'ai');

worker.js(部署到cloudflare)

export default {

  async fetch(*request*, *env*, *ctx*) {

    if (*request*.method === 'POST' && *request*.url.endsWith('/chat')) {

      try {

        const { prompt } = await *request*.json();

        const CLOUDFLARE\_ACCOUNT\_ID = ''; *// Your Account ID*

        const CLOUDFLARE\_API\_TOKEN = ''; *// Your API Token*

        const CLOUDFLARE\_MODEL\_NAME = '@cf/openai/gpt-oss-120b';

        const aiResponse = await fetch(`https://api.cloudflare.com/client/v4/accounts/${CLOUDFLARE\_ACCOUNT\_ID}/ai/run/${CLOUDFLARE\_MODEL\_NAME}`, {

          method: 'POST',

          headers: {

            'Authorization': `Bearer ${CLOUDFLARE\_API\_TOKEN}`,

            'Content-Type': 'application/json'

          },

          body: JSON.stringify({ input: prompt })

        });

        const data = await aiResponse.json();

        const headers = {

          'Content-Type': 'application/json',

          'Access-Control-Allow-Origin': '\*',

          'Access-Control-Allow-Methods': 'POST, OPTIONS',

          'Access-Control-Allow-Headers': 'Content-Type'

        };

        return new Response(JSON.stringify(data), { headers });

      } catch (error) {

        console.error('Worker error:', error);

        return new Response(JSON.stringify({ success: false, errors: [{ message: 'Worker server error.' }] }), {

          status: 500,

          headers: { 'Content-Type': 'application/json', 'Access-Control-Allow-Origin': '\*' }

        });

      }

    } else if (*request*.method === 'OPTIONS' && *request*.url.endsWith('/chat')) {

      const headers = {

        'Access-Control-Allow-Origin': '\*',

        'Access-Control-Allow-Methods': 'POST, OPTIONS',

        'Access-Control-Allow-Headers': 'Content-Type',

        'Access-Control-Max-Age': '86400'

      };

      return new Response(null, { headers, status: 204 });

    }

    return new Response('Not Found', { status: 404 });

  },

};