# 把cloudflare变成你的专属免费图床 - wlnxing的小窝

### 前言

cloudflare作为一个免费的cdn,一直受到大家的喜爱.作为一个国外的cdn,使用他家的cdn服务域名不用备案,并且在国内使用的速度也不错,基本上是国内一些域名没有备案的博主的cdn首选.他的免费计划基本上也够用了.这么优秀的cdn,如果能够作为自己的私人图床该多好啊,经过多方查找解决方案,基本上已经可以完美使用了.

## 原理介绍

介绍原理之前,先了解一下backblaze B2吧,我们图片的原始文件就是存在这里的.

### backblaze B2介绍

backblaze B2是一个云存储方案,有点像各个云服务商提供的对象存储服务,说白了就是用来存东西的,一个免费账号可以用10G的存储空间,超出后的费用也非常便宜.

### 使用原理

通常像backblaze B2这种云存储方案存东西不是很贵,但是他的带宽费用通常是非常昂贵的,就是说如果你在里面存东西不贵,但是如果你要下载,上传内容的话,所使用的流量费用非常贵.但是,cloudflare组织了一个带宽联盟,里面就有backblaze.带宽联盟里面的成员传输数据到cloudflare是免费的.我们可以好好利用这个特性,由cloudflare来传输我们的图片数据.这样我们的图片就一分钱都不会花了.

## 各个部分免费额度介绍

### cloudflare

cloudflare的免费计划可以免费cloudflare的免费节点,只是不能自选节点,可以用三个防火墙规则,三个页面规则等等,个人使用基本上够了.

### cloudflare workers

1. 一天十万次请求
2. 每次请求最大cpu工作时间10ms(作图床根本不会超)
3. 详情:点击查看works计划详情:点击查看

### backblaze B2存储

1. 前10G存储免费
2. 超出后 每GB 0.005$一个月
3. 详情查看:https://www.backblaze.com/b2/cloud-storage-pricing.html

### backblaze B2 带宽费用

1. 每天前1GB下载免费
2. 由于cloudflare带宽联盟,这个费用我们可以不计.

### backblaze API调用

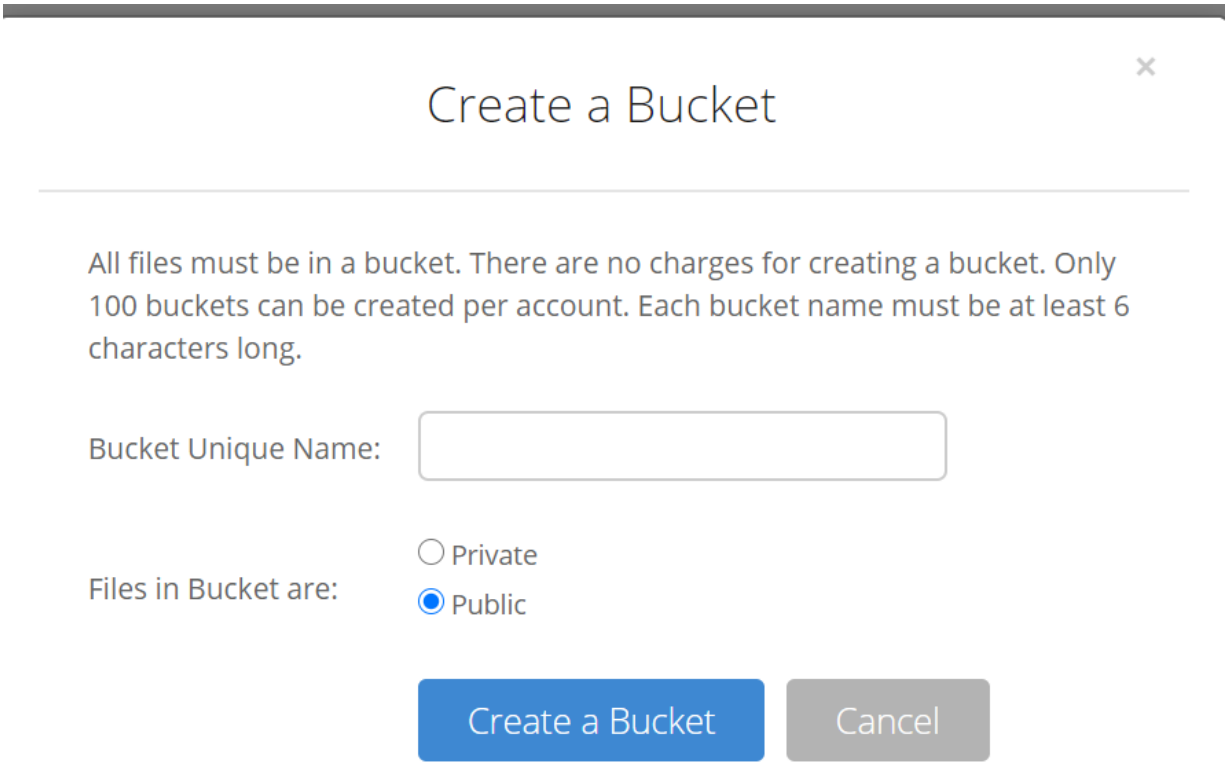
1. 具体详情点击查看
2. 解释一下,cloudflare用到的传输属于B类传输,B类传输我们每天前2500次是免费的.通过我们的cloudflare缓存策略,第一次访问后再次访问就不会消耗额度了,因为cloudflare已经将图片缓存了,所以就是直接访问的cloudflare的缓存了.因此,这个额度对于个人的私人图床来说绝对是绰绰有余.

## 开始配置

### backblaze B2储存桶设置

#### 创建存储桶

首先，你需要一个backblaze账号，如果你没有这个账号，可以免费去注册一个，地址：backblaze账号注册 注册过程我就不做示范了
账号注册完毕后,登录刚刚注册的账号,然后去这里,点击"Create a Bucket"创建一个B2存储桶

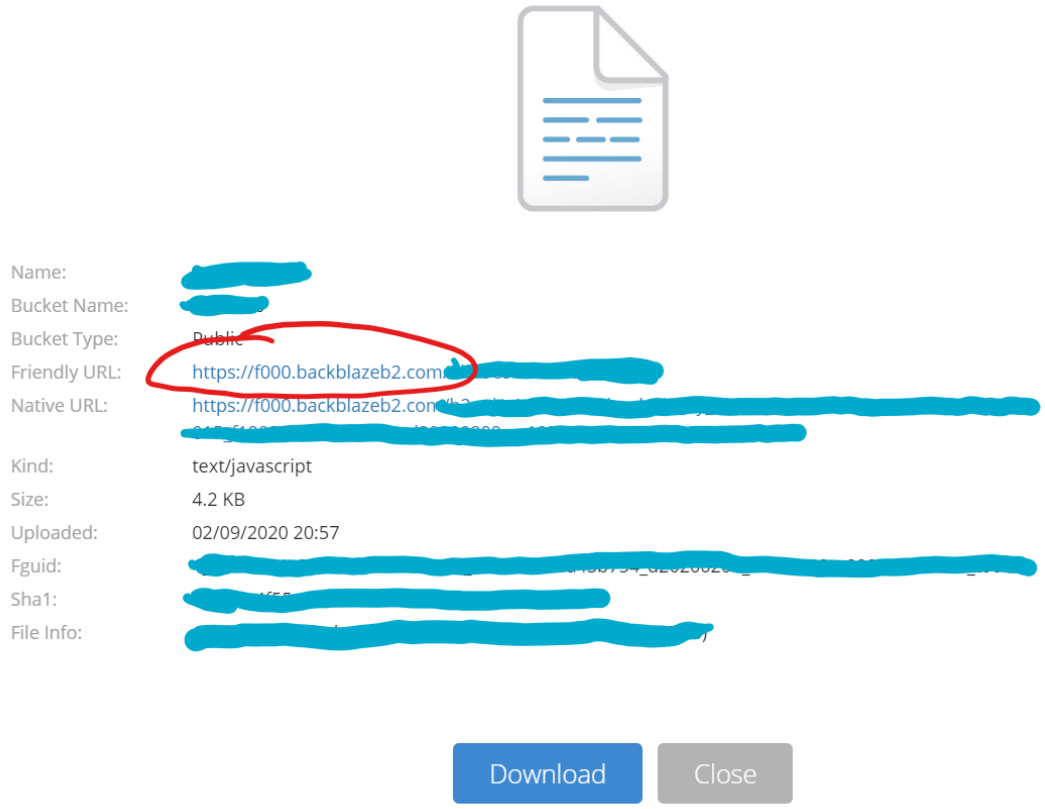



> 注意事项:
>
> 1. 要确保桶的类型是public,因为你的图片是要放出来看的.

> 2. 为了安全,请为储存桶起一个别人不容易猜测到的名字,如果名字暴露了的话,别人就可以直接对着你的backblaze刷你的免费额度了.

### 获取存储桶位置

创建完成后,我们进入这个储存桶,先点击"upload"顺便上传一个文件,然后点击那个文件,这时会弹出这个文件的详细信息



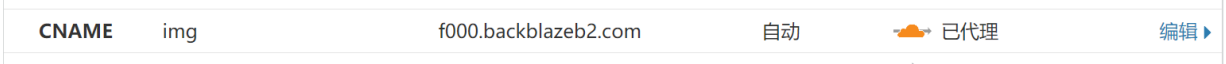

从我红线画出的位置可以看到我们的文件是保存在https://f000.backblazeb2.com/ 这个域名里的(个别人的位置可能会不一样),记住这个域名,这个就是我们桶的储存位置.

## cloudflare设置

### 解析cloudflare DNS

> 首先你得有一个cloudflare的账号和一个解析到cloudflare的域名,如果没有的话请注册一个.

登录cloudflare,进入你打算的作图床的主域名里面,点击DNS,进入dns解析界面,为你的域名添加一条cname记录,名称自定,目标输入之前让你记下来的那个域名.然后最重要的一点,确保那个云朵是点亮的,这代表这个域名的流量都会由cloudflare代理.点保存生效




如图,我使用的我的子域名img.wlnxing.com来作为我的图床域名,并且使用了cloudflare的CDN(云朵是点亮状态)来代理了我的请求.
这时,如果通过这种链接:https://你的图床域名/file/桶名/文件名 能够访问你的文件了的话,说明已经设置成功了.下面只需要对安全性,便捷性进行改进了.

## cloudflare进阶设置

### 缓存设置

> 如果你存的图片不会经常变动的话,我强烈建议你通过页面规则调大cloudflare的缓存等级和缓存时间,这样的话当我们访问图片时cloudflare就不会频繁的来backblazeB2获取文件了,只要有缓存,cloudflare就直接从它自身的缓存获取文件.这样即减少了backblaze的调用,也可以加快速度.

点击进入页面规则,点击创建页面规则,输入你的图床域名,注意后面要有一个星号,即:`你的图床域名/*` ,因为这个页面规则是按照你的url匹配的,所以需要一个通配符来匹配(你也可以自行设置).然后在下面点击添加设置,然后选取设置,将`缓存级别设置为缓存所有内容`, 将`边缘缓存设置为7天`其他的设置可以自行按需使用.以下是我的设置,供大家参考:


### 链接优化

默认的链接有点长,并且会暴露backblaze的原始链接,不安全,我们可以利用cloudflare的works功能做一些优化

> works是cloudflare新推出的一项功能,它允许你在cloudflare的服务器上运行你的JavaScript代码(限制等详见前文介绍)

我们的目标是要省去/file/桶名称/ 那部分,使链接更友好.
我们点击cloudflare上方的workers,然后点击管理路由,点击创建worker,将下面的代码粘贴在左边的代码栏里.右边的那些不用改.

(代码来自于驱蚊器喵blog.meow.page的修改,感谢)

> 使用前，注意修改 b2Domain 和 b2Bucket 这两个变量的值.

b2Domain，是你图床域名。
b2Bucket，是你存储桶的名字

```
'use strict';
const b2Domain = '使用前此处改为你的图床域名'; // configure this as per instructions above
const b2Bucket = '使用前此处改为你的存储桶的名字'; // configure this as per instructions above
const b2UrlPath = `/file/${b2Bucket}/`;
addEventListener('fetch', event => {
    return event.respondWith(fileReq(event));
});

// define the file extensions we wish to add basic access control headers to
const corsFileTypes = ['png', 'jpg', 'gif', 'jpeg', 'webp'];

// backblaze returns some additional headers that are useful for debugging, but unnecessary in production. We can remove these to save some size
const removeHeaders = [
    'x-bz-content-sha1',
    'x-bz-file-id',
    'x-bz-file-name',
    'x-bz-info-src_last_modified_millis',
    'X-Bz-Upload-Timestamp',
    'Expires'
];
const expiration = 31536000; // override browser cache for images - 1 year

// define a function we can re-use to fix headers
const fixHeaders = function(url, status, headers){
    let newHdrs = new Headers(headers);
    // add basic cors headers for images
    if(corsFileTypes.includes(url.pathname.split('.').pop())){
        newHdrs.set('Access-Control-Allow-Origin', '*');
    }
    // override browser cache for files when 200
    if(status === 200){
        newHdrs.set('Cache-Control', "public, max-age=" + expiration);
    }else{
        // only cache other things for 5 minutes
        newHdrs.set('Cache-Control', 'public, max-age=300');
```

```
    }
    // set ETag for efficient caching where possible
    const ETag = newHdrs.get('x-bz-content-sha1') || newHdrs.get('x-bz-info-
src_last_modified_millis') || newHdrs.get('x-bz-file-id');
    if(ETag){
        newHdrs.set('ETag', ETag);
    }
    // remove unnecessary headers
    removeHeaders.forEach(header => {
        newHdrs.delete(header);
    });
    return newHdrs;
};
async function fileReq(event){
    const cache = caches.default; // Cloudflare edge caching
    const url = new URL(event.request.url);
    if(url.host === b2Domain && !url.pathname.startsWith(b2UrlPath)){
        url.pathname = b2UrlPath + url.pathname;
    }
    let response = await cache.match(url); // try to find match for this request in
the edge cache
    if(response){
        // use cache found on Cloudflare edge. Set X-Worker-Cache header for helpful
debug
        let newHdrs = fixHeaders(url, response.status, response.headers);
        newHdrs.set('X-Worker-Cache', "true");
        return new Response(response.body, {
            status: response.status,
            statusText: response.statusText,
            headers: newHdrs
        });
    }
    // no cache, fetch image, apply Cloudflare lossless compression
    response = await fetch(url, {cf: {polish: "lossless"}});
    let newHdrs = fixHeaders(url, response.status, response.headers);

  if(response.status === 200){

    response = new Response(response.body, {
      status: response.status,
      statusText: response.statusText,
      headers: newHdrs
    });
```

```
  }else{
    response = new Response('File not found!', { status: 404 })
  }

    event.waitUntil(cache.put(url, response.clone()));
    return response;
}
```

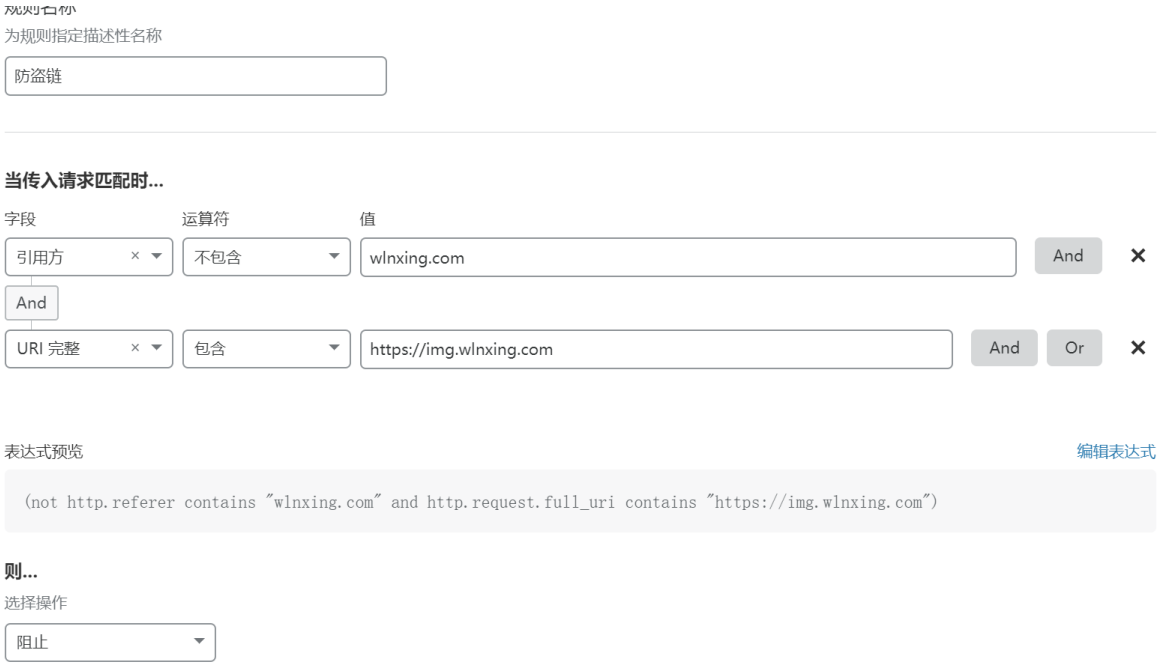
粘贴进去后,点击保存并部署之后点击"<"返回上一页,然后点击重命名,起一个你方便辨认的名字.然后点击最上方的workers选择你的域名,进入域名管理,再次点击workers图标,点击添加路由,路由框输入你的图床域名加一个星号(图床域名/*),worker选择你刚刚创建的workers,然后保存即可.


现在,你就可以直接你由 你的图床域名/文件在桶中的路径/文件名 这种方式来访问你的文件了.

**由cloudflare来提供防盗链**

我们可以利用cloudflare的防火墙判断referer来提供简单的防盗链功能,具体设置如下:
点击cloudflare上方防火墙,点击防火墙规则,点击创建防火墙规则.

名称随便即可,然后字段选择引用方(referer),运算符选择不包含,值填入你的博客域名(白名单域名),然后再点击"and",字段选择url完整(url full),运算符选择包含,值填入你的图床域名.最后下面选择"阻止",保存即可.




解释一下,这样配置的意思是
如果 请求头(referer)里面没有包含有我博客的域名(白名单域名),并且(and),完整请求(url full)里包含有我的图床域名,那么就阻止这个请求.
意思就是,不是因为访问的博客(白名单域名)而加载的图片,那么就阻止请求.