Alibaba Cloud CDN with GitHub Pages: Quick Guide

A short reference for the static site in this project. It explains the delivery path, the planned cache rules and how to check whether a response came from the CDN edge.

1. Delivery path

User browser → **accelerated domain** (CNAME to the Alibaba Cloud CDN hostname) → **CDN edge node** → on a cache miss, back to the **GitHub Pages origin** (*username*.github.io, HTTPS port 443, Host header = *username*.github.io).

2. Planned cache rules

Priority	Match	TTL
Highest	/api/news.json	1 minute
High	/index.html	10 minutes
Medium	jpg, jpeg, png	7 days
Medium	css, js	1 day
Medium	pdf	3 days

Specific path rules must outrank extension rules so that frequently changing files stay fresh.

3. Checking a response

Compare the origin and the CDN with header-only requests:

```
curl -I https://username.github.io/images/logo.png
curl -sS -D - -o /dev/null https://cdn.example.com/images/logo.png
```

Look at **Age**, **Cache-Control**, **Expires**, **Via** and any cache-status header the CDN adds. An increasing Age or an edge HIT marker suggests an edge cache hit. Repeated requests alone are not proof.

4. Updating content

After pushing a new version to GitHub, the origin changes immediately but the edge keeps its copy until the TTL expires. Submit the exact URLs (/index.html, /api/news.json) to **Refresh/Purge** in the CDN console, wait for the task to finish, then request the files through the CDN again.

5. Safety

Never commit credentials. Keep traffic low, review pricing before enabling paid features, set a billing alert, and disable or delete the CDN domain when the exercise is finished.

Original educational material created for an individual academic assignment.