

Konfigurasi CDN

Penjelasan Konfigurasi edge.conf

Berikut isi `edge.conf` — penjelasan baris per baris / blok per blok

```
proxy_cache_path /var/cache/nginx/cdn_cache levels=1:2 keys_zone=cdn_cache:200m
max_size=10g inactive=7d use_temp_path=off;

log_format cachelog '$remote_addr - [$time_local] "$request" '
'$status $body_bytes_sent "$http_referer" "$http_user_agent" '
'upstream_cache_status=$upstream_cache_status request_time=$request_time';

access_log /var/log/nginx/edge_access.log cachelog;
error_log /var/log/nginx/edge_error.log;

server {
    listen 80;
    server_name _;

    location / {
        proxy_pass http://192.168.1.10;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_cache cdn_cache;
        proxy_cache_key "$scheme$host$request_uri";
        proxy_cache_valid 200 302 6h;
        proxy_cache_valid 404 1m;
        proxy_cache_use_stale error timeout updating http_500 http_502 http_503;
        proxy_cache_bypass $http_cache_control;
        proxy_cache_lock on;

        add_header X-Cache-Status $upstream_cache_status;
        expires 6h;
    }
}
```

Penjelasan detail (baris / blok penting)

`proxy_cache_path /var/cache/nginx/cdn_cache ...` 

- **Fungsi:** mendefinisikan tempat dan behavior cache global.
- `/var/cache/nginx/cdn_cache` → folder filesystem untuk menyimpan file cache.

- `levels=1:2` → struktur subdirektori (mengurangi jumlah file dalam satu folder; performa filesystem).
- `keys_zone=cdn_cache:200m` → area memori (shared) bernama `cdn_cache` sebesar 200MB untuk menyimpan metadata kunci cache.
- `max_size=10g` → batas total ukuran cache (10 gigabyte).
- `inactive=7d` → bila objek tidak diakses selama 7 hari, dihapus.
- `use_temp_path=off` → tulis langsung ke direktori final (mengurangi I/O overhead pada beberapa setup).

`log_format cachelog '...' & access_log ... cachelog` 

- **Fungsi:** buat format log custom bernama `cachelog`.
- Isi log menyertakan: IP client, waktu, request, status HTTP, bytes, referer, user-agent, `upstream_cache_status` (HIT/MISS/...), dan `request_time`.
- Berguna untuk menghitung hit ratio dan analisis performa.



`error_log /var/log/nginx/edge_error.log;` 

- Tempat menyimpan log error Nginx untuk troubleshooting.

`server { listen 80; server_name _; }` 

- `listen 80` → terima request HTTP biasa.
- `server_name _;` → wildcard / default server block (berguna saat tidak pakai nama domain khusus).

`location / { ... }` — blok utama proxy & cache 

- `proxy_pass http://192.168.1.10;`
→ semua request diteruskan ke **origin** (IP origin). Kalau cache MISS, Nginx akan mengambil dari sini.
- `proxy_set_header Host $host; & proxy_set_header X-Real-IP $remote_addr;`
→ meneruskan header Host (penting untuk origin yang butuh host header) dan IP asli client (berguna di origin logs / security).
- `proxy_cache cdn_cache;`
→ mengaktifkan penggunaan zone cache `cdn_cache` (yang didefinisikan di `proxy_cache_path`).
- `proxy_cache_key "$scheme$host$request_uri";` 
→ kunci unik untuk setiap objek cache. Disusun dari scheme (http/https) + host + path+query.
→ Penting: kalau kamu ingin mengabaikan query string atau menambah header ke key, ubah di sini.
- `proxy_cache_valid 200 302 6h; & proxy_cache_valid 404 1m;` 
→ berapa lama response dengan status tertentu disimpan di cache. Contoh: response

200/302 disimpan 6 jam; 404 hanya 1 menit.

- `proxy_cache_use_stale error timeout updating http_500 http_502 http_503;` 🛡️
→ jika origin bermasalah, Nginx boleh menyajikan konten *stale* (kadaluarsa) untuk menjaga ketersediaan. Berguna saat origin sedang down atau lambat.
 - `proxy_cache_bypass $http_cache_control;` 🚫
→ jika client mengirim header `Cache-Control` (mis. `no-cache`), Nginx akan melewati cache dan minta langsung ke origin. Berguna untuk debugging atau force-refresh dari client.
 - `proxy_cache_lock on;` 🔒
→ mencegah *cache stampede*: saat banyak request bersamaan untuk resource yang belum ada di cache (MISS), hanya satu request yang diteruskan ke origin sementara request lain menunggu hasil (mengurangi beban origin).
 - `add_header X-Cache-Status $upstream_cache_status;` 🔊
→ menambahkan header `X-Cache-Status` (HIT / MISS / BYPASS / STALE) ke response. Sangat berguna untuk verifikasi manual dan debugging.
 - `expires 6h;` 💡
→ memberi header `Expires` / `Cache-Control` ke browser/client sebagai fallback (memberi tahu browser untuk menyimpan response sampai 6 jam). Perlu disesuaikan tergantung jenis asset.
-

1. `use_temp_path=off`

- Artinya: Nginx menulis file cache **langsung ke lokasi final** (`/var/cache/...`) tanpa menulis dulu ke temp dir.
- Efek: mengurangi I/O (lebih cepat, kurang operasi tulis ganda).
- Trade-off: jika proses terhenti tiba-tiba bisa meninggalkan file sebagian; jadi pastikan disk/storagenya andal.

2. `levels=1:2`

- Artinya: susunan subfolder untuk file cache — tiap cache disimpan di subdirektori ber-hash.
- Contoh: `1:2` → struktur `/<1char>/<2chars>/file` (mis. `/a/bc/file`).
- Tujuan: menghindari terlalu banyak file dalam satu folder → meningkatkan performa filesystem.

3. `proxy_cache_key "$scheme$host$request_uri";`

- Artinya: kunci unik untuk tiap resource di cache dibuat dari `scheme` (`http/https`) + `host` + `request_uri` (path + **query string**).
- Implikasi: request dengan query berbeda dianggap objek berbeda.

- Tip: kalau mau **abaikan query string**, pakai `"$scheme$host$uri"` ; kalau cache harus berbeda berdasarkan header tertentu, tambahkan variabel header (contoh: `$http_accept_encoding` atau `$http_cookie`).

4. `proxy_cache_bypass $http_cache_control;`

- Artinya: jika client mengirim header `Cache-Control` (mis. `no-cache`), variabel `$http_cache_control` berisi nilai → kondisi terpenuhi → **cache tidak dipakai** dan request diteruskan ke origin.
- Fungsi: memungkinkan client/teknisi memaksa refresh dari origin.
- Catatan: kamu bisa menambah kondisi lain (mis. `cookie`, `query param`) untuk bypass.

Tambahan — `add_header X-Origin "origin-server";`

- Artinya: tambahkan HTTP header `X-Origin: origin-server` pada response.
- Fungsi: berguna untuk debugging/identifikasi sumber response (mis. tahu response datang dari edge/atau origin tertentu).
- Kalau mau agar header selalu muncul (termasuk error), gunakan `add_header X-Origin "origin-server" always;` .