🍏 Nginx cache as CDN

menerapkan cdn menggunakan cache nginx - by raka and chatpgt😋
Client → Edge → Origin. 🎯

IP yang aku gunakan dalam panduan ini (siap dipakai langsung):

- **ORIGIN_IP = 192.168.1.10** 💻 (Debian)
- **EDGE_IP = 192.168.1.11** 🌐 (Debian)
- **CLIENT_IP = 192.168.1.12** 💻 (Debian)

Gambaran topologi (tiga mesin) 🧭

```
Client (192.168.1.12)  ---> hosts/DNS -> Edge Server (192.168.1.11, Nginx sebagai CDN/cache)
                                              |
                                              v (jika cache MISS)
                                    Origin Server (192.168.1.10, Nginx / app)
```

- **Edge Server** = CDN node (Nginx dengan `proxy_cache`).
- **Origin Server** = sumber konten (file HTML/JS/CSS atau API).
- **Client** = mesin uji (mengunjungi URL via Edge untuk melihat efek cache dan benchmarking).

Ringkasan singkat 🔎

- Tujuan: belajar caching, hit/miss, purge, perbandingan performa antara akses lewat Edge (CDN) dan akses langsung ke Origin, semuanya diuji dari **Client** (192.168.1.12).
- OS: semua mesin memakai **Debian/Ubuntu**

Prasyarat ✅

- 3 mesin (Origin, Edge, Client) dengan IP di atas.
- `sudo` access di Origin & Edge (Client hanya perlu akses untuk pengujian).
- Port default menggunakan **80** (kalau sudah dipakai, sesuaikan).
- Pastikan firewall internal mengizinkan koneksi HTTP antar mesin (192.168.1.10 ↔

192.168.1.11 ↔ 192.168.1.12).

---

1) Setup Origin (192.168.1.10) — server sumber konten 🗂️

Jalankan di **ORIGIN**:

1. Install nginx:

```
sudo apt update
sudo apt install -y nginx openssh-server
```

2. Buat folder konten dan file contoh:

```
sudo mkdir -p /var/www/origin/html
echo "<h1>Origin: versi 1</h1>" | sudo tee /var/www/origin/html/index.html
```

3. (Opsional) buat site config `/etc/nginx/sites-available/origin.conf`:

```
server {
    listen 80;
    server_name _;

    root /var/www/origin/html;
    index index.html;

    access_log /var/log/nginx/origin_access.log;
    error_log  /var/log/nginx/origin_error.log;

    add_header X-Origin "origin-server";
}
```

Enable & reload:

```
sudo ln -s /etc/nginx/sites-available/origin.conf /etc/nginx/sites-enabled/ ||
true
sudo systemctl restart nginx
```

4. Tes dari **EDGE** atau **CLIENT**:

```
curl -I http://192.168.1.10/
```

```
# harus mengembalikan HTTP/1.1 200 OK
```

---

2) Setup Edge (192.168.1.11) — Nginx sebagai CDN / proxy cache 🧰

Jalankan di **EDGE**:

1. Install nginx:

```
sudo apt update
sudo apt install -y nginx openssh-server
```

2. Buat konfigurasi cache `/etc/nginx/conf.d/edge.conf` (sudah mengganti ORIGIN_IP dengan 192.168.1.10):

```
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;             # origin IP
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_cache cdn_cache;
        proxy_cache_key "$scheme$host$request_uri";  # termasuk query string
        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;
```

```
    }


}
```

3. Reload nginx:

```
sudo systemctl restart nginx
```

4. Pastikan edge dapat akses origin:

```
curl -I http://192.168.1.10/
curl -I http://192.168.1.11/
```

---

3) Setup Client (192.168.1.12) — mesin uji 💻

Client ini akan kamu gunakan untuk test hit/miss, latency.

1. Install utils

```
sudo apt update
sudo apt install -y curl apache2-utils build-essential
```

2. Tambahkan entri `/etc/hosts` pada **CLIENT** agar mudah referensi (opsional):

```
sudo sh -c 'echo "192.168.1.10 origin.local" >> /etc/hosts'
sudo sh -c 'echo "192.168.1.11 cdn.local" >> /etc/hosts'
```

Sekarang dari client kamu bisa mengakses `http://192.168.1.11/` atau `http://cdn.local/`.

---

4) Cara tes dasar — cek HIT / MISS (dijalankan dari CLIENT) 🧪

Di **CLIENT (192.168.1.12)**:

1. Tes pertama kali (kemungkinan MISS):

```
curl -I http://192.168.1.11/ | grep -i X-Cache-Status || true
```

```
# harus -> X-Cache-Status: MISS (pada request pertama)
```

2. Tes ulang (seharusnya HIT):

```
curl -I http://192.168.1.11/ | grep -i X-Cache-Status || true
# sekarang -> X-Cache-Status: HIT
```

3. Cek origin langsung (untuk perbandingan):

```
curl -I http://192.168.1.10/
# origin biasanya tidak punya X-Cache-Status kecuali kamu tambahkan X-Origin
```

---

5) Ukur waktu & bandingkan latency ⏱️ (dijalankan dari CLIENT)

Di **CLIENT**:

Single-request time compare:

```
curl -s -o /dev/null -w "EDGE time: %{time_total}\n" http://192.168.1.11/
curl -s -o /dev/null -w "ORIGIN time: %{time_total}\n" http://192.168.1.10/
```

Loop contoh (10x):

```
for i in {1..10}; do curl -s -o /dev/null -w "EDGE %{time_total}\n"
http://192.168.1.11/; done
for i in {1..10}; do curl -s -o /dev/null -w "ORIGIN %{time_total}\n"
http://192.168.1.10/; done
```

Ekspektasi: EDGE (pada cache HIT) jauh lebih cepat daripada ORIGIN.

---

6) Memantau hit ratio dari log (bukti kuantitatif) 📊 (di EDGE)

Pastikan `edge.conf` sudah memiliki `log_format cachelog`.

Hit dan miss count (di **EDGE**):

```
HIT=$(grep -o 'upstream_cache_status=HIT' /var/log/nginx/edge_access.log | wc -l)
MISS=$(grep -o 'upstream_cache_status=MISS' /var/log/nginx/edge_access.log | wc -
l)
TOTAL=$((HIT+MISS))
echo "HIT: $HIT, MISS: $MISS, TOTAL: $TOTAL"
if [ $TOTAL -gt 0 ]; then
  awk -v h=$HIT -v t=$TOTAL 'BEGIN{ printf "Hit ratio: %.2f%%\n", (h/t)*100 }'
fi
```

Target: untuk static assets, hit ratio > 70–80% adalah tanda baik.

Untuk melihat beban pada origin, cek access log origin:

```
wc -l /var/log/nginx/origin_access.log
```

(angka request ke origin harus turun ketika Edge melayani banyak request).

---

7) Tes invalidasi / purge — otomatis & manual 🔁

Kamu ingin tahu bagaimana Edge otomatis memperbarui cache ketika Origin berubah. Berikut beberapa opsi (pilih sesuai kebutuhan). Untuk lab sederhana, aku sertakan cara manual & SSH purge yang mudah.

A — Manual update & purge (langsung)

Di **ORIGIN**:

```
echo "<h1>Origin: versi 2</h1>" | sudo tee /var/www/origin/html/index.html
```

Di **EDGE** (manual purge):

```
sudo rm -rf /var/cache/nginx/cdn_cache/*
sudo systemctl reload nginx
```

Di **CLIENT**:

```
curl -I http://192.168.1.11/ | grep -i X-Cache-Status || true
# seharusnya sekarang MISS, lalu HIT dan menampilkan versi baru
```

---

8) Perbandingan nyata: dengan CDN vs tanpa CDN VS

- **Tanpa CDN (langsung ke origin):**
  - Latency lebih tinggi, origin menangani semua request → CPU & bandwidth tinggi.
- **Dengan Edge CDN (Nginx proxy_cache):**
  - Latency rendah untuk cached assets; origin traffic berkurang; butuh mekanisme purge & ops.
