



**A client's CloudFront bill rose from \$500 to \$2,000. Which metrics would you check first, and what are the likely causes?**

**What could cause increased data transfer or cache misses?**

1. **High Data Transfer Out (DTO)**
  - Serving large media files (videos, images, software downloads).
  - Global traffic without effective caching → more origin fetches.
  - No compression (Gzip/Brotli disabled).
2. **Cache Misses (Low Cache Hit Ratio)**
  - Cache policy too strict (forwarding all headers, cookies, query strings → unique cache keys).
  - Short TTLs → frequent revalidation with origin.
3. **Invalidations**
  - Frequent **wildcard invalidations** (/\*) → forces CloudFront to re-fetch objects.
4. **Unoptimized Routing**
  - Serving traffic across expensive regions (e.g., lots of traffic in Asia, where DTO rates are higher).

---

**How do you monitor, analyze, and optimize CloudFront costs?**

◆ **Monitor**

- **CloudFront Metrics (CloudWatch):**
  - BytesDownloaded, BytesUploaded → check DTO patterns.
  - CacheHitRate → low % means higher origin costs.
- **CloudFront Access Logs** → analyze traffic by object size, region, cache status.
- **AWS Cost Explorer / CUR** → drill down by service & region.

◆ **Analyze**

- Identify **top URLs** or file types contributing to DTO.
- Look at **cache hit vs miss patterns**.
- Check if some clients send requests with varying query params/headers causing cache fragmentation.

**Optimize**

### 1. Improve Caching

- Use appropriate **cache policies** (limit headers/cookies forwarding).
- Set **longer TTLs** for static assets + versioning (style.v2.css).

### 2. Reduce Data Transfer

- Enable **compression (Gzip/Brotli)**.
- Optimize images/video (WebP, adaptive streaming).
- Move large static files to S3 with signed URLs (only deliver to authorized users).

### 3. Geographic Optimization

- Use **Regional Edge Caches** for repetitive global traffic.
- Consider **Origin Shield** (single point for cache fill, reduces cross-region fetches).

### 4. Avoid Expensive Invalidations

- Use **object versioning** instead of invalidating /\*.

### 5. Security/Access

- Ensure only CloudFront can access your origin (no direct costly hits).
- Use WAF to block abusive/bot traffic.

---

#### **In short:**

Unexpected CloudFront costs often come from **low cache hit ratio** or **large DTO**. Monitor via CloudWatch + Access Logs, analyze top contributors, and optimize with **better cache policies, compression, versioning, and regional caching**.

#### **Which of the following could increase Data Transfer Out (DTO) costs in CloudFront?**

- a) Serving large video or software files
- b) Enabling Gzip/Brotli compression
- c) Traffic surge from Asia-Pacific region
- d) Cache misses due to low hit ratio

**Answer:** a, c, d

---

#### **How can frequent wildcard invalidations (/\*) affect CloudFront cost?**

- a) Improve cache hit ratio
- b) Reduce origin fetches
- c) Force CloudFront to re-fetch objects → higher DTO
- d) Lower 4xx errors

**Answer:** c

---

#### **Which metric is the primary driver of CloudFront costs?**

- a) RequestCount
- b) BytesDownloaded

- c) CacheHitRate
- d) OriginLatency

**Answer: b**

---

Which tool helps **break down CloudFront costs by region and usage type?**

- a) CloudFront Popular Objects Report
- b) AWS Cost Explorer
- c) CloudWatch Logs Insights
- d) AWS WAF

**Answer: b**

---

Which of the following are **best practices to optimize CloudFront costs?**

- a) Use longer TTLs + asset versioning
- b) Enable compression (Gzip/Brotli)
- c) Block direct origin access (via OAC)
- d) Use signed URLs for large static files
- e) Always forward all headers and cookies

**Answer: a, b, c, d**