CDN (Content Delivery Network) Guide + System Design Q&A

CDN Cheatsheet for Developers

1. What is a CDN?

A Content Delivery Network (CDN) is a globally distributed network of servers that deliver web content to users based on their geographic location, improving speed and performance.

2. Key Benefits

- Faster page load times
- Reduces server load
- Protects against DDoS attacks
- Improves global accessibility
- Optimizes bandwidth usage

3. What Does It Cache?

- Images
- CSS / JavaScript
- Videos
- Fonts
- Static HTML

4. Popular CDN Providers

- Cloudflare
- Amazon CloudFront
- Akamai
- Google Cloud CDN
- Fastly

5. When to Use a CDN

- You serve users globally
- Your website has heavy traffic
- You use a lot of static or media-heavy content
- You want high uptime and performance

6. Bonus Dev Tip

Use HTTP headers like Cache-Control, ETag, and Expires to fine-tune CDN behavior.

System Design Questions – CDN Focused

1. How would you design a global video streaming service like Netflix using a CDN?

Use a CDN to cache and serve videos from edge locations near users. Ensure adaptive bitrate streaming and multi-region failover.

2. Explain how CDN helps with scalability and latency reduction in a high-traffic web application.

CDNs offload traffic from the origin server and serve cached content from nearby locations, reducing server strain and user latency.

3. What are the trade-offs of using a CDN vs. relying only on your origin server?

CDNs improve performance and availability but introduce complexity in cache invalidation, cost, and dependency on third-party infrastructure.

4. How would you handle cache invalidation and versioning in a CDN setup?

Use versioned URLs (e.g., /style.v2.css), and configure short TTLs or purge APIs to control cache freshness.

5. Describe how a CDN can be integrated with load balancers and origin servers.

CDN sits in front of the origin server and load balancer. It handles static assets, while dynamic requests go to the load balancer/origin.

6. How does a CDN contribute to security in a distributed architecture?

CDNs offer DDoS protection, rate limiting, bot filtering, and WAF features, helping secure the application edge.

7. Design a multi-region web application. Where and how would you place CDN nodes?

Use CDN edge locations in each target region, with origin servers and DBs replicated or sharded regionally.

8. How would you monitor and troubleshoot issues in a CDN-powered system?

Use CDN logs, performance monitoring tools, status pages, and real-user monitoring (RUM) to detect and resolve issues.

9. What would you do if some regions experience CDN cache miss frequently?

Optimize cache TTL, pre-warm caches, and use origin shield nodes to reduce direct origin hits.

10. How would you ensure data consistency across CDN nodes for near real-time content?

Reduce TTLs, use purge/invalidation APIs, or use dynamic edge caching where appropriate with validation headers.