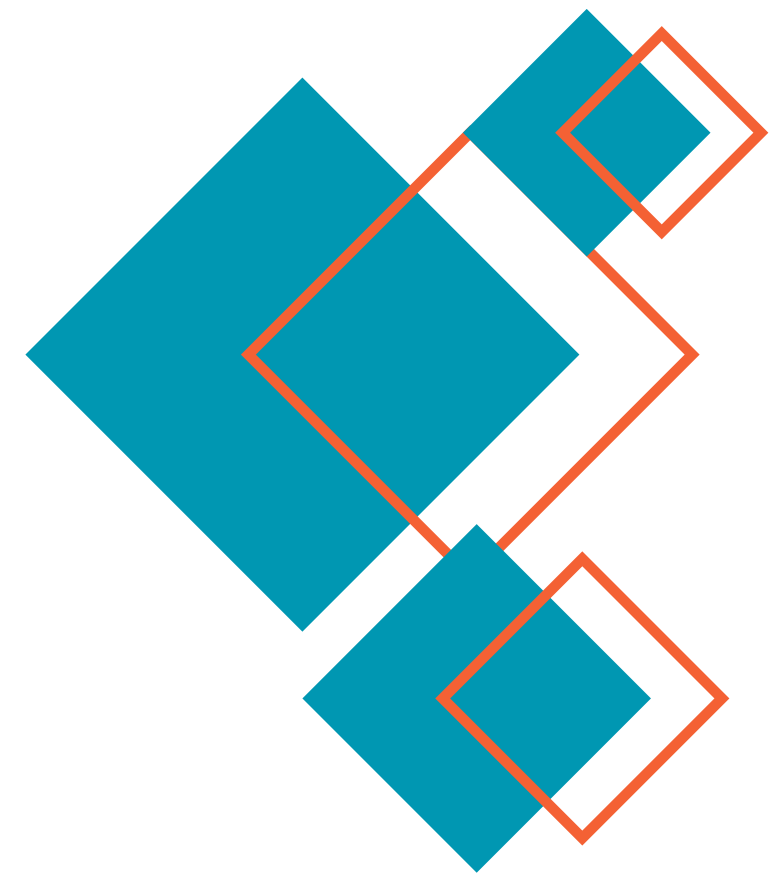




# SYSTEM DESIGN



Basic Networking  
DNS & Proxy Server



## DNS কী?

DNS কে ইন্টারনেটের ফোনবই বলতে পারেন। যেমন আপনার ফোনে “Sohel” নামটি দিয়ে সার্চ করলে নাম্বার খুঁজে পাই, ঠিক তেমনি, DNS একটি ডোমেইন নাম (যেমন: google.com) কে একটি IP তে (যেমন: 142.250.192.14) ইন্ডিকেট করে।

ব্রাউজার আসলে আইপি ছাড়া কাজ করতে পারে না – আপনি যেকোনো ওয়েবসাইট ই সার্চ করেন না কেন বিহাইন্ড দা সীন DNS কাজ করে।

## DNS কিভাবে কাজ করে?

ধরেন আপনি ব্রাউজারে লিখলেন: `www.google.com`

ধাপসমূহ:

প্রথমে এটি ব্রাউজার cache চেক করে – যদি IP জানা থাকে তাহলে সেখান থেকেই serve করে।

না থাকলে OS cache → এরপর ISP DNS resolver এ query পাঠায়

যদি resolver না জানে, তাহলে:

Root DNS Server → .com এর info দেয়

TLD DNS Server (.com) → google.com এর Authoritative server-এ পাঠায়

Authoritative DNS Server → IP address দিয়ে দেয়

এখন সেই IP দিয়ে ব্রাউজার TCP/HTTP দিয়ে request পাঠায়

## EXAMPLE FLOW

You → DNS Resolver → Root DNS → .com DNS → Google DNS → IP Found

## DNS RECORDS (TYPES)

Record Type	কাজ
A	ডোমেইন → IPv4 Address
AAAA	ডোমেইন → IPv6 Address
CNAME	Alias/another name (e.g. www → root domain)
MX	Mail Exchange (ইমেইল সার্ভার)
TXT	Text info (SPF, verification, etc.)
NS	Name Server info

## DNS , SYSTEM DESIGN-এ কেন গুরুত্বপূর্ণ

Performance boost — কাস্টম DNS + cache দিয়ে দ্রুত IP রেসল্বে হয়

Failover & redundancy — Multiple DNS সার্ভার configure করলে সার্ভার ডাউন হলেও fallback হয়

Load balancing — Round-robin DNS দিয়ে বিভিন্ন IP-তে রিকোয়েস্ট পাঠানো যায়

Global distribution — Geo-DNS বা Anycast DNS ব্যবহার করে নিকটস্থ সার্ভারে পাঠানো যায়

### সংক্ষেপে দেখে নেই

বিষয়	ব্যাখ্যা
DNS কি?	Domain নাম কে IP address এ রূপান্তর
কাজের ধাপ	Browser → Resolver → Root → TLD → Auth DNS
System Design-এ ভূমিকা	Performance, Redundancy, Load Balance
Real tools	Cloudflare DNS, Google DNS (8.8.8.8), Route53

# PROXY SERVER – (FORWARD PROXY এবং REVERSE PROXY)

PROXY কী?

PROXY মানে হলো মধ্যস্থতাকারী – এটি CLIENT ও SERVER এর মাঝখানে বসে এবং তাদের মধ্যে যোগাযোগ নিয়ন্ত্রণ করে। এটি দুইভাবে কাজ করতে পারে:

1. FORWARD PROXY → CLIENT-এর পক্ষ হয়ে সার্ভারে অনুরোধ পাঠায়
2. REVERSE PROXY → সার্ভারের সামনে বসে, CLIENT-এর অনুরোধকে পেছনে থাকা বিভিন্ন সার্ভারে পাঠায়

## FORWARD PROXY

এটি কিভাবে কাজ করে:

- ক্লায়েন্ট সরাসরি ইন্টারনেট অ্যাক্সেস করতে পারে না
- তার বদলে, ক্লায়েন্ট প্রক্সি সার্ভারকে রিকোয়েস্ট পাঠায়
- প্রক্সি সেটা ইন্টারনেটে পাঠায়, রেসপন্স নিয়ে ক্লায়েন্টকে দেয়

ব্যবহারের ক্ষেত্র:

- কোম্পানিতে ওয়েব ফিল্টারিং
- GEO-BLOCKED CONTENT BYPASS (VPN)
- ANONYMOUS BROWSING

উদাহরণ:

- TOR, SQUID PROXY, VPN সার্ভার

# Reverse Proxy

এটি কিভাবে কাজ করে:

- ক্লায়েন্ট বুঝতেই পারে না পেছনে একাধিক সার্ভার আছে
- Reverse Proxy request নেয়, আর পেছনে থাকা appropriate সার্ভারে forward করে
- Load balancing, SSL termination, caching করতে পারে

ব্যবহারের ক্ষেত্র:

- Web server protection
- Load balancing
- Static content caching
- SSL termination

উদাহরণ:

- Nginx, HAProxy, Apache mod\_proxy
- Cloudflare, AWS ELB, Fastly (CDN+Reverse Proxy)

## System Design-এ Reverse Proxy এর ভূমিকা


প্রধান কাজগুলো:

1. Load balancing – একাধিক সার্ভারে অনুরোধ ভাগ করে দেয়
2. SSL Termination – HTTPS encryption প্রক্রিয়া-তে শেষ হয়ে যায়
3. Caching – স্ট্যাটিক ফাইল প্রক্রিয়া থেকে সার্ভ করে
4. Security – প্রকৃত সার্ভার hide থাকে, DDoS থেকে সুরক্ষা রাখে
5. Compression – Content গুলো compress করে পাঠায়

# FORWARD PROXY VS REVERSE PROXY তুলনা

বৈশিষ্ট্য	Forward Proxy	Reverse Proxy
বসে কোথায়?	Client-এর সামনে	Server-এর সামনে
উদ্দেশ্য	ক্লায়েন্ট হাইড করা	সার্ভার হাইড করা
ব্যবহারকারীর কে?	ক্লায়েন্ট	সার্ভার
ব্যবহারের জায়গা	ফিল্টারিং, অ্যানোনিমিটি	Load balancing, security, SSL
User দেখতে পারে?	হ্যাঁ, বুঝে	না, বুঝে না





# শেষ কথা:

সুতরাং এই ছিল আজকের ছোট আলোচনা আমরা DNS এবং Proxy Server খুব ভালো করে জানলাম এগুলো কি? কেন দরকার উপকারিতা গুলো কি কি। পরবর্তীতে আমরা Load Balancer and Firewall and CDN নিয়ে আলোচনা করবো সাথেই থাকুন