

# Backend & Proxy Mastery Roadmap

## Stage 1: Core Proxy Features

- Implement full HTTP/1.1 parsing (headers, body, chunked encoding)
- Support persistent connections (keep-alive)
- Handle request/response streaming with proper buffering
- Implement connection timeout handling
- Support logging of raw requests/responses (tcpdump style)

## Stage 2: Advanced Proxy Features

- HTTP pipelining support
- Connection pooling & reuse
- TLS termination (basic OpenSSL integration)
- Basic caching layer
- Support for configurable buffer sizes
- Graceful error handling & retries

## Stage 3: System-Level Mastery

- Implement non-blocking I/O with epoll/kqueue
- Master edge-triggered vs level-triggered event loops
- Efficient memory handling (arena allocators, mmap)
- Understand socket options (SO\_REUSEPORT, TCP\_NODELAY, etc.)
- Implement a load test harness (wrk/ab style)

## Stage 4: Distributed Systems Mastery

- Implement reverse proxy with load balancing (round-robin, least connections)
- Introduce caching with LRU eviction
- Support clustering and multi-process architecture
- Experiment with consistent hashing for distributed caches
- Learn basics of Raft/Paxos consensus

## Stage 5: Godlike Tier

- Implement zero-copy proxying (splice, sendfile)
- Kernel bypass networking (DPDK, io\_uring, eBPF)
- Write your own lightweight HTTP library (no dependencies)
- Custom binary protocol design & parsing
- Hybrid architecture using C++/Go for critical paths, Python for orchestration
- Performance tuning: benchmark up to 1M+ connections