

CLOUDFLARE

AND THE MODERN CDN

HOW THE INTERNET EDGE
ACTUALLY WORKS



ROMAN LYSENKO

Cloudflare and the Modern CDN

How the Internet Edge Actually Works

Steve Publications

This book is available at <https://leanpub.com/cloudflareandthemoderncdn>

This version was published on 2026-09-08



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Steve Publications

Contents

- How the Internet Edge Actually Works 1**
- Introduction 2**
- Chapter 1: The Shape of the Modern Internet Edge 6**
 - The Latency Problem: Why Distance Still Matters 6
 - A Request’s Journey: From Click to Response 7
 - What Is an Edge Network? 8
 - Centralized Origins vs. Distributed Edges 9
 - Why CDNs Changed Everything 10
 - Chapter Summary 11
- Chapter 2: How DNS Actually Works 13**
 - The DNS Hierarchy: Root, TLD, and Authoritative 13
 - Recursive Resolution and the Stub Resolver 14
 - Authoritative DNS and Zone Files 15
 - Caching, TTLs, and Propagation Myths 17
 - Key Record Types: A, AAAA, CNAME, MX, TXT, SRV, and Beyond 18
 - DNSSEC: Trust in the Name System 20
 - Chapter Summary 21
- Chapter 3: Routing the Internet: BGP, Anycast, and IP Transit 23**
 - Autonomous Systems and the BGP Protocol 23
 - Route Announcement, Prefixes, and Path Selection 23
 - What Anycast Is and Why CDNs Depend on It 23
 - IP Transit, Peering, and Internet Exchanges 23
 - When Routing Goes Wrong: Leaks, Hijacks, and Outages 23
 - Chapter Summary 23
- Chapter 4: Transport and Application Protocols at the Edge 25**
 - TCP Connections: Handshake, Retransmit, and Congestion Control . . 25

CONTENTS

TLS: Encryption, Handshake Costs, and Session Resumption 25

HTTP/1.1: Pipelining, Keep-Alive, and Head-of-Line Blocking 25

HTTP/2: Multiplexing, Frames, and Server Push 25

HTTP/3 and QUIC: UDP-Based Transport with Built-In TLS 25

Protocol Negotiation: ALPN and the Actual User Experience 26

Chapter Summary 26

Chapter 5: The CDN Architecture 27

 The Reverse Proxy Pattern 27

 Edge Nodes, Points of Presence, and Topology 27

 The Cache: Hitting, Missing, Stale, and Bypassing 27

 Origin Shielding and Tiered Caching 27

 CDN Request Lifecycle: End User to Origin and Back 27

 What a CDN Can and Cannot Accelerate 27

 Chapter Summary 28

Chapter 6: Cloudflare’s Architecture and Network 29

 Cloudflare’s Global Network: Data Centers and POPs 29

 Cloudflare’s Anycast IP Space and How Traffic Lands 29

 Request Flow Through Cloudflare: Edge to Origin 29

 How Cloudflare Differs from Akamai, Fastly, and Others 29

 What Is and Is Not Visible to the End User 29

 Chapter Summary 29

Chapter 7: Getting Started: Accounts, Zones, and Nameservers 31

 Accounts, Memberships, and Plan Tiers 31

 Adding a Site and Understanding Zone Status 31

 Transferring DNS: Nameservers and Cutover Strategy 31

 Proxied vs. DNS-Only: The Orange Cloud 31

 Verifying Your Site Is Live and Troubleshooting the Initial Setup 31

 Configuration Examples: Initial DNS Records for a Real Site 31

 Chapter Summary 32

Chapter 8: DNS on Cloudflare 33

 Cloudflare DNS: Authoritative, Managed, Fast 33

 Managing Records: Dashboard, API, and Terraform 33

 Split-Horizon DNS and Internal Resolvers 33

 Load Balancing DNS Records and Traffic Management 33

 DNS Automation: API Patterns and Terraform Modules 33

 Operational Considerations: Propagation, Cutover, and Rollback 33

Chapter Summary	34
Chapter 9: SSL/TLS with Cloudflare	35
The Cloudflare SSL/TLS Problem Space	35
SSL/TLS Modes: Off, Flexible, Full, Full (Strict)	35
Certificate Types: Universal, Custom, Dedicated, Origin CA	35
Certificate Management and Renewal	35
Security Trade-offs and Common Mistakes	35
Configuration Example: Hardened TLS Setup for Production	35
Chapter Summary	36
Chapter 10: Web Application Firewall and Security Rules	37
The WAF Rule Engine: How Cloudflare Inspects Requests	37
Managed WAF Rulesets: OWASP, Category-Based, and Threat Intelli- gence	37
Custom WAF Rules and Field Operators	37
Firewall Rules: The Legacy System and When to Use It	37
Security Rules and Traffic Filtering	37
Avoiding False Positives and Rule Conflicts	38
Chapter Summary	38
Chapter 11: DDoS Protection, Rate Limiting, and Bot Management	39
DDoS at the Edge: How Cloudflare Absorbs Attacks	39
Rate Limiting Rules: Tokens, Windows, and Responses	39
Bot Fight Mode and Advanced Bot Management	39
Challenge Pages: JavaScript, Turnstile, and Under Attack Mode	39
API Shield and Automated Threat Mitigation	39
Configuration Examples: Rate Limiting and Bot Policies for Real Ap- plications	39
Chapter Summary	40
Chapter 12: Zero Trust, Access, and Private Networks	41
Zero Trust Architecture and Cloudflare's Approach	41
Cloudflare Access: Identity-Based Application Access	41
Zero Trust Gateway: Secure Web and DNS Gateways	41
Cloudflare Tunnel: Exposing Origins Without Open Ports	41
Integrating Identity Providers and Device Posture	41
Architecture Example: Secure Internal App Behind Access and Tunnel	41
Chapter Summary	42

Chapter 13: Caching Fundamentals in Cloudflare 43

HTTP Caching Headers: Cache-Control, ETag, Expires 43

How Cloudflare Decides What to Cache 43

Cache Keys, Variants, and Query String Handling 43

Edge TTLs and Origin Response Headers 43

Cache Hits, Misses, Bypasses, and Stale Responses 43

Reading and Understanding Cloudflare Cache Headers 43

Chapter Summary 44

Chapter 14: Cache Rules, Page Rules, and Advanced Caching 45

Cache Rules: The Modern Rule System 45

Page Rules: Legacy Behavior and Migration Path 45

Cache Reserve: Pay-For-Priority in Peak Times 45

Tiered Caching: Two-Tier and Smart Tiered 45

Cache Purging: Individual URLs, Tags, and Bulk 45

Configuration Examples: Caching Strategy for a Real Application . . . 45

Chapter Summary 46

Chapter 15: Optimization Features: Compression, HTTP/2, HTTP/3, and Argo 47

Compression: Gzip, Brotli, and Automatic Settings 47

HTTP/2 and HTTP/3: Enable, Disable, and When It Matters 47

Argo Smart Routing: Private Backbone and Latency Reduction 47

Rocket Loader, Minify, and Miscellaneous Optimizations 47

Measuring Impact: What Actually Moves the Needle 47

Configuration Example: Performance Stack for a Dynamic Application 48

Chapter Summary 48

Chapter 16: Cloudflare Workers 49

What Workers Are: JavaScript at the Edge 49

The Workers Runtime: V8 Isolates and Execution Model 49

Routing: Hostnames, Patterns, and Dispatch 49

Workers Examples: Rewrites, A/B Testing, Auth, and Transformations 49

Limits, Quotas, and Cold Starts (Or Lack Thereof) 49

Configuration: Wrangler, Deploy, and Environment Variables 49

Chapter Summary 50

Chapter 17: Workers Data Products: KV, Durable Objects, D1, and Queues 51

Workers KV: Global Key-Value Storage 51

Durable Objects: Stateful Co-located Compute 51

CONTENTS

D1: SQLite at the Edge 51

Queues: At-Least-Once Delivery for Background Work 51

Choosing the Right Data Product: Comparison and Patterns 51

Architecture Example: Worker-Powered API with KV and D1 51

Chapter Summary 52

Chapter 18: Pages, R2, Stream, Images, and WebSockets 53

Cloudflare Pages: Serverless-Style Static and Full-Stack Hosting . . . 53

R2: Object Storage Without Egress Fees 53

Stream: Video Transcoding and Delivery 53

Images: On-the-Fly Transformations 53

WebSockets: Persistent Connections Through Cloudflare 53

Architecture Example: Media-Rich Application on Cloudflare Platform 53

Chapter Summary 54

Chapter 19: Cloudflare Load Balancing 55

What Load Balancing Solves: Global Affinity and Failover 55

Pools, Monitors, and Load Balancers 55

Health Checks: HTTP, TCP, and Custom 55

Steering Policies: Least Connections, Geolocation, and Custom 55

Failover Behavior and Graceful Degradation 55

Configuration Example: Multi-Region Application with Load Balancing 55

Chapter Summary 56

Chapter 20: Advanced Architectures and Patterns 57

Static Site Architecture on Cloudflare 57

Dynamic Web Application Behind Cloudflare 57

API-First Architecture with Edge Security 57

Multi-Region Origins with Load Balancing and Tiered Cache 57

Kubernetes and Containerized Backends Behind Cloudflare 57

Private Origin Architectures with Tunnel and Access 57

Chapter Summary 58

Chapter 21: Logging, Analytics, and Observability 59

The Analytics Dashboard: Traffic, Performance, and Security 59

Logpush: Exporting Logs to Your Systems 59

Building Observability Pipelines with Cloudflare Data 59

Metrics That Matter: Latency, Cache Ratio, Errors, and Threats 59

Alerts and Notifications: Proactive Monitoring 59

Example: Setting Up Full Observability for a Production Site 60

Chapter Summary	60
Chapter 22: Troubleshooting Cloudflare Deployments	61
The Troubleshooting Methodology: Isolate the Layer	61
DNS and Resolution Issues	61
SSL/TLS Errors and Misconfigurations	61
Caching Problems: Stale Content and Unexpected Behavior	61
Redirect Loops and Proxy Issues	61
Worker Debugging and Routing Failures	61
Common Configuration Mistakes and How to Avoid Them	62
Chapter Summary	62
Chapter 23: Automation and Infrastructure as Code	63
Cloudflare API: Authentication, Rate Limits, and Patterns	63
Terraform Provider: State, Resources, and Modules	63
CI/CD for Workers and Pages	63
Automating Security Rules and DNS Changes	63
GitOps and Configuration Management	63
Example: Terraform Module for a Production Zone	63
Chapter Summary	64
Chapter 24: Production Operations and Migration Strategies	65
Migrating to Cloudflare: Strategy and Execution	65
Cutover Strategies: Big Bang vs. Gradual	65
Cost Management and Plan Selection	65
Capacity, Limits, and Scaling Considerations	65
Change Management and Rollback Procedures	65
Running Cloudflare in Mission-Critical Environments	65
Chapter Summary	66
Conclusion: The Edge as a Platform	67
What You Know Now: The Edge Architecture Stack	67
Choosing Cloudflare: When It Makes Sense and When It Does Not	67
The Future of the Edge: Compute Everywhere	67
A Practical Decision Framework	67
Final Thoughts on Mastering the Edge	67
References	68

How the Internet Edge Actually Works

This book explains how the modern Internet edge works and how Cloudflare fits into it. You will learn the underlying protocols and architectures that power content delivery, from DNS and BGP routing to TLS and HTTP/3. Then you will learn how to configure, operate, secure and optimize Cloudflare deployments in real production environments. The book is written for engineers who need to understand not just which buttons to click, but why the edge works the way it does and how to make sound architectural decisions. No marketing fluff, no hand-holding tutorials. Just engineering depth.

Introduction

You type a URL into your browser and hit enter. Within a fraction of a second, the page appears. It feels effortless, instantaneous, invisible. In reality, your request just crossed the entire Internet, navigating through dozens of networks, jumping between cities, and passing through security filters before reaching a server that might be thousands of miles from where you sit. And if that server is also in another hemisphere, it just had to send data back across the same chaotic, global path.

Now imagine that same request going through Cloudflare. The journey is still complex, but the dynamics are different. Cloudflare does not own the Internet. It does not lay fiber or operate autonomous systems the way telecom carriers do. What it does is sit in the middle and intercept requests before they travel the full distance. It has servers in more than 348 cities around the world, each advertising the same set of anycast IP addresses via BGP. When your request lands, it arrives at the nearest Cloudflare data center. Cloudflare inspects it, applies security rules, checks its cache, runs edge code if necessary, and either responds directly or forwards the request to your origin server on your behalf.

This book is about what happens in that intercept. It is about the Internet edge, what it means for you as an operator, and how Cloudflare implements it.

When you use Cloudflare, you are making an architectural choice. You are deciding to put a massive distributed system between your users and your origin. That system can reduce latency, absorb attacks, serve cached content from three hundred and forty eight cities instead of your single origin, and terminate TLS connections for millions of requests. It can also introduce new failure modes, obscure problems with its error pages, and cause subtle bugs when caching behaves unexpectedly or security rules block legitimate traffic. Understanding Cloudflare means understanding both the power it gives you and the complexity it introduces.

This book has three goals. First, it explains the foundational protocols and concepts that make the Internet edge work: DNS resolution, BGP routing, anycast, TCP connections, TLS handshakes, HTTP versions and caching behavior. These are not specific to Cloudflare. They are the bedrock of how

content delivery operates across any CDN or reverse proxy deployment. If you understand them, you can reason about what is happening when your application experiences a problem, even when you are not using Cloudflare at all.

Second, the book walks through Cloudflare's entire platform in practical depth. You will learn how to set up an account, transfer DNS, configure SSL/TLS modes, create cache rules, deploy Workers, set up load balancing, protect against attacks, build Zero Trust access and automate configuration at scale. Each topic is covered with realistic examples, not toy setups. You will see complete DNS record configurations, full Workers code, Terraform modules, WAF rule expressions, and the specific Cloudflare dashboard or API paths that make things happen. The examples assume you are building something you might actually ship, not running through a tutorial.

Third, the book distinguishes between what is fundamental and what is Cloudflare-specific. HTTP caching headers work the same way regardless of which CDN you use. TLS modes like Flexible and Full (Strict) are Cloudflare's framing of the problem of end-to-end encryption. Some concepts are universal standards documented in RFCs; others are Cloudflare's implementation choices, product naming and opinionated defaults. A good engineer needs to know which is which, because portability matters and vendor lock-in is a real concern.

The book is organized into four main movements. The first section covers Internet and CDN fundamentals: how DNS actually resolves, how BGP and anycast route packets, what TCP and TLS handshakes cost in latency, how HTTP evolved from version 1.1 to HTTP/3 and QUIC, and how generic CDNs cache and serve content. This foundation takes the longest to read but will repay itself many times over as you encounter Cloudflare-specific features.

The second section dives into Cloudflare's architecture and core services. You will learn how Cloudflare's global network is structured, how traffic flows through its edge, how to set up zones and DNS records, how SSL/TLS modes work and what security each one provides, and how the Web Application Firewall, security rules, rate limiting, and bot management products protect your applications. This is where the book shifts from theory to practice.

The third section covers performance, caching, and edge computing. Caching is the heart of what a CDN does, and Cloudflare's caching behavior is nuanced enough that it deserves its own deep dive. You will learn about

cache control headers, cache keys, cache rules, tiered caching, cache reserve and purge strategies. Then the book moves into Cloudflare Workers and the developer platform: V8 isolates, KV, Durable Objects, D1, Queues, Pages, R2, and other products that let you run compute and store data at the edge itself, not just pass traffic through it.

The final section covers advanced architectures, observability, troubleshooting, and operations. This is where the book ties everything together. You will see how to design real production systems behind Cloudflare, from static sites to multi-region applications to private origin architectures protected by Tunnel and Access. You will learn how to observe what is happening through logs and analytics, how to troubleshoot problems systematically, how to automate configuration with Terraform and CI/CD, and how to run Cloudflare reliably in mission-critical environments.

Who this book is for

The intended reader is technically capable. You are likely a software engineer, DevOps engineer, SRE, platform engineer, or technically literate founder who has deployed internet-facing services and wants to do it more effectively. You know what an HTTP request looks like, what a domain name is, and what TLS does at a conceptual level. You may or may not have used Cloudflare before. The book does not assume deep networking knowledge, but it does assume you are comfortable reading technical documentation and working with configuration files.

If you are looking for a marketing overview of Cloudflare, this book is not it. If you are looking for simple step-by-step screenshots of every button to click, this book is not it either. What it is, is a reference and explanation of the actual mechanics behind Cloudflare deployments and the Internet edge more broadly. You will find it useful when you are choosing a TLS mode and need to understand the security trade-offs, when you are debugging a strange cache miss pattern, when you are deciding between Workers KV and D1, when you are designing a global application that uses Load Balancing and Tiered Cache or when you are trying to explain to a team why their security rules are conflicting.

A note on accuracy and currency

Cloudflare updates its product constantly. Features move between plan tiers, limits change, naming conventions shift, and legacy products are deprecated in favor of newer ones. The material in this book is based on Cloudflare's official documentation, engineering blog posts, and public statements available

as of 2026. Specific plan-dependent features and pricing are subject to change. When a feature is version-sensitive or plan-dependent, the book calls it out explicitly. Always consult Cloudflare's current documentation for the latest details on limits, pricing and availability.

How to use this book

Read it in order if you want to build a complete understanding. The chapters build on each other, so skipping ahead can mean encountering terms and concepts before they are explained. That said, the book is also designed as a reference. You can jump to the caching chapter when you need to understand cache rules, to the WAF chapter when you are configuring security or to the troubleshooting chapter when something is broken. The chapters are self-contained enough that you can read them individually, but reading the fundamentals first will make the later chapters easier to understand.

This book will not make you a Cloudflare expert overnight. But if you read it carefully and use it as a reference while you build and operate systems, you will develop a working understanding of how the edge actually works and how to use Cloudflare in ways that are secure, performant and reliable. Let us get started.

Chapter 1: The Shape of the Modern Internet Edge

Every application that talks to the Internet faces the same fundamental constraint: the physical universe has limits. Data moves at the speed of light through fiber optic cables. It bounces between routers, switches and servers. Each hop adds delay. The further your user is from your server, the longer their requests take to complete.

This is the problem CDNs solve, and it is the problem Cloudflare exists to solve. Understanding it is the first step to understanding everything that follows.

The Latency Problem: Why Distance Still Matters

Latency is the delay between when a user initiates an action and when they see a response. For web applications, latency comes from several sources. The dominant source is propagation delay: the time it takes for a packet to travel through fiber optic cables between your user and your server.

Light in a vacuum travels at approximately 300,000 kilometers per second. In fiber optic cable, it travels at about two thirds of that, roughly 200,000 kilometers per second. That sounds fast, but round-trip time matters more than one-way time for most web protocols, because the request has to go out and the response has to come back.

Consider a user in Tokyo making a request to a server in New York. The great-circle distance is approximately 10,800 kilometers. A round trip is roughly 21,600 kilometers. At 200,000 kilometers per second, the pure propagation delay is about 108 milliseconds. That is before accounting for router processing delays, queueing delays, TCP handshake overhead, TLS handshake overhead, and server processing time. By the time you add those up, a single round-trip can easily take 150 to 250 milliseconds.

This matters because most web pages and applications require multiple round-trips before they display anything. The initial DNS lookup requires

a round-trip. The TCP connection requires at least one round-trip for the three-way handshake. The TLS handshake requires one or more additional round-trips depending on the version and whether session resumption is used. Then the actual HTTP request and response require at least one more round-trip. A simple page load can easily involve four to six round-trips before any meaningful content appears on screen.

Multiply that by 200 milliseconds per round-trip and you are looking at over a second before the user sees anything. A second feels instantaneous in some contexts but is painfully slow for interactive applications. Mobile networks add more delay and packet loss, making the problem worse.

CDNs address this by putting servers closer to users. If a CDN has a data center in Tokyo, that same request might only travel a few hundred kilometers instead of 10,800. The propagation delay drops from 108 milliseconds to perhaps 5 milliseconds. That is the fundamental latency benefit: physical proximity.

But CDNs do more than reduce distance. Cloudflare, for example, advertises that its network reaches 95 percent of internet users within 50 milliseconds [1]. It operates in 348 cities across more than 120 countries and connects directly to over 13,000 autonomous systems [2]. When your application sits behind Cloudflare, users around the world connect to the nearest Cloudflare data center instead of your single origin server. The latency improvement is not just incremental. For users far from your origin, it can be dramatic.

A Request's Journey: From Click to Response

To understand how CDNs reduce latency, it helps to trace what happens when a user requests a URL. This journey is more complex than most people realize, and understanding each step is crucial for understanding how Cloudflare intercepts and optimizes it.

Consider a user in São Paulo visiting `https://www.example.com`. Behind the scenes, Cloudflare is already in the path, but the user never needs to know that. Here is what actually happens:

The user's browser sends a DNS query asking for the IP address of `www.example.com`. The browser's stub resolver forwards this query to a recursive resolver, typically operated by the user's ISP or a public resolver like Google's 8.8.8.8 or Cloudflare's 1.1.1.1. The recursive resolver walks the

DNS hierarchy, querying root servers, then the .com TLD servers, then the authoritative DNS servers for example.com. Cloudflare operates those authoritative servers for zones delegated to it. When the recursive resolver finally receives the answer, it caches the result and returns the IP address to the browser.

That IP address belongs to Cloudflare, not the user's origin server. Cloudflare uses anycast routing to advertise the same IP addresses from hundreds of data centers around the world. The BGP routing protocols in the Internet direct the user's traffic to the topologically nearest Cloudflare data center, which in this case is likely one in São Paulo or nearby.

The browser initiates a TCP connection to that Cloudflare IP address, completing the three-way handshake. Then it performs a TLS handshake, negotiating encryption parameters and verifying Cloudflare's certificate. The browser then sends an HTTP request over the encrypted connection.

Cloudflare's edge server receives the request. It checks its security rules. It checks whether the content is cached. If it is cached and valid, Cloudflare returns the response directly from its edge server in São Paulo. The user's request never reaches the origin server at all. The total latency is just the round-trip between São Paulo and the user's device, plus minimal processing time.

If the content is not cached or has expired, Cloudflare forwards the request to the origin server. This might be a server in another country or another cloud region. Cloudflare fetches the response from the origin, caches it if appropriate, and returns it to the user. The user experiences the latency of their connection to Cloudflare plus the additional latency of Cloudflare's connection to the origin. On subsequent requests, the cached copy serves the user directly.

Throughout this journey, Cloudflare is doing several things simultaneously: terminating TLS connections, inspecting traffic for attacks, applying WAF rules, checking bot scores, enforcing rate limits, managing cache, and potentially running edge code via Workers. All of this happens before the request reaches the origin, which means the origin is protected from attacks, shielded from excess traffic, and only handles requests that legitimately need it.

What Is an Edge Network?

The term “edge” refers to the periphery of a network, the points closest to end users. In traditional networking, the “core” is the high-capacity backbone that connects data centers and cities. The “edge” is where users connect, through home routers, mobile towers, or corporate networks.

A CDN inverts this model. It takes application logic and content and moves them from centralized data centers to the edge itself. Instead of all users connecting to a single origin, content is replicated to hundreds or thousands of edge servers around the world. Each edge server can serve users nearby with low latency.

Cloudflare describes its approach as running every service in every data center. This is distinct from other CDNs that separate services geographically. When Cloudflare adds a new data center in a city, it does not just add caching servers. It adds full-featured edge nodes that can run Workers, perform security inspection, handle DNS resolution, serve as load balancers, and participate in anycast routing. This uniformity means users get the same capabilities regardless of which data center they connect to.

An edge network is not just about caching static files. Modern edge networks like Cloudflare support compute, storage, security, and networking services at the edge itself. Cloudflare Workers let you run JavaScript code in every data center. Cloudflare D1 and KV provide edge-optimized storage. Cloudflare Load Balancing distributes traffic globally based on health and latency. These capabilities make the edge not just a distribution layer but an application layer in its own right.

Centralized Origins vs. Distributed Edges

Before CDNs were common, applications typically ran from centralized origins. An e-commerce platform might host its application in a single cloud region, perhaps US East. Users from around the world would connect directly to that region. Users nearby experienced acceptable latency. Users far away experienced slow load times, especially on poor network connections.

The problems with centralized origins are straightforward:

- Latency: Users far from the origin experience higher round-trip times, degrading user experience and conversion rates.
- Single points of failure: If the origin region experiences an outage, the entire application is unavailable globally.
- Limited capacity: A single region has finite resources. Traffic spikes can overwhelm it.
- Attack surface: All traffic arrives at the same endpoints, making the origin an easy target for DDoS attacks.
- Cost: Egress bandwidth from cloud providers can be expensive, especially for globally distributed users.

Distributed edges solve these problems by creating many points of presence instead of one. Traffic is naturally load-balanced by geography. Outages in one region do not affect users who connect to other regions. Capacity scales horizontally across the network. Attacks are absorbed and mitigated at the edge before they reach the origin. Bandwidth is optimized because content is served from local caches instead of being fetched from the origin on every request.

But distributing an application adds complexity. You need to ensure consistency across edge locations. You need to handle cache invalidation correctly. You need to decide what content is appropriate to cache and what must always hit the origin. You need to monitor many locations instead of one. Cloudflare abstracts much of this complexity, but the trade-offs are real and understanding them is essential for production deployments.

Why CDNs Changed Everything

CDNs did not just make websites faster. They fundamentally changed how applications are architected and operated.

The first major change was performance normalization. Before CDNs, application performance depended heavily on where you chose to deploy your origin. If you deployed in Europe, European users got fast performance and Asian users got slow. CDNs decouple performance from origin location. You can deploy your origin almost anywhere and still provide good performance globally because the edge serves users locally.

The second major change was security at scale. Before CDNs, if you wanted DDoS protection, you needed to provision extra capacity at your origin or buy

mitigation services that were often slow to activate. CDNs absorb attacks at the edge because they already have massive capacity. Cloudflare, for example, has absorbed attacks exceeding 25 million requests per second [3]. When an attack arrives, Cloudflare's edge servers distribute the load across its entire network, apply WAF rules to filter malicious traffic, and never let most of it reach the origin.

The third major change was edge computing. Early CDNs were passive caches. They stored content and served it. They could not run application logic. Modern CDNs like Cloudflare blur the line between edge and application. Cloudflare Workers let you run authentication, authorization, A/B testing, response modification, and routing logic at the edge itself. This means you can offload work from your origin, reduce latency, and build globally distributed applications without managing servers in multiple regions.

CDNs also changed the economics of hosting. Cloud providers charge for egress bandwidth. If your application serves a lot of content to global users, those charges add up quickly. CDNs cache content at the edge, reducing the amount of data fetched from the origin. Some CDNs, including Cloudflare, charge less for bandwidth than cloud egress fees or offer generous free tiers. Over time, the savings can be significant.

The final change is architectural. Applications designed for CDNs are different from applications designed for direct access. They use HTTP caching headers intentionally. They consider edge routing and geographic distribution in their design. They treat the CDN not as an optional add-on but as a fundamental part of the architecture. This is the mindset this book helps you develop.

Chapter Summary

- Latency is dominated by physical distance. The further users are from your origin, the slower their experience.
- CDNs reduce latency by serving content from edge servers close to users instead of distant origins.
- A single HTTP request involves multiple steps: DNS resolution, TCP handshake, TLS handshake, and the request itself. Each step adds latency.
- Cloudflare intercepts requests at its edge, applies security and caching, and either responds directly or forwards to the origin.

- Edge networks run application logic, storage, and security at the periphery of the network, not just in centralized data centers.
- CDNs normalized global performance, provided security at scale, introduced edge computing, changed hosting economics, and influenced how applications are architected.

In the next chapters, we will look at the individual protocols and mechanisms that make this work, starting with DNS, the system that translates human-readable domain names into IP addresses and is the entry point for every request.

Chapter 2: How DNS Actually Works

Before any packet can reach your server, something has to translate the domain name you type into an IP address. That translation is handled by the Domain Name System, or DNS. DNS is the most fundamental infrastructure service on the Internet. Without it, you would have to memorize IP addresses instead of using names.

DNS is also the entry point for Cloudflare. When you use Cloudflare, the DNS resolution step is where your traffic first encounters Cloudflare's infrastructure. Understanding how DNS works is essential for understanding how Cloudflare intercepts traffic, how DNS-based load balancing operates, and how to configure zones correctly.

The DNS Hierarchy: Root, TLD, and Authoritative

DNS is hierarchical and distributed. There is no single DNS server that knows every domain name. Instead, the system is organized as a tree, with different levels of authority.

At the top of the tree are the root name servers. There are 13 root server clusters, labeled A through M, and each cluster is implemented using anycast routing to serve queries from multiple locations worldwide [4]. When a DNS resolver needs to look up a domain name and does not have the answer cached, it starts by asking a root server. Root servers do not know the IP addresses of individual websites. They only know where to find the authoritative servers for each top-level domain.

Below the root servers are the top-level domain, or TLD, name servers. Each TLD has its own set of authoritative servers. For `.com`, the TLD servers are operated by Verisign. For `.org`, they are operated by Public Interest Registry. Country-code TLDs like `.uk`, `.de`, and `.jp` have their own operators. When a resolver queries a root server for `example.com`, the root server responds with a referral to the `.com` TLD servers. The resolver then queries a `.com` TLD server, which responds with a referral to the authoritative name servers for `example.com`.

At the bottom of the tree are the authoritative name servers. These are the servers that hold the actual DNS records for a domain. They know the IP addresses, mail server records, and other information for that domain. When you register a domain and point it at Cloudflare, Cloudflare becomes the authoritative name server for your domain. Cloudflare operates these authoritative servers from its anycast network, which means DNS queries for your domain are answered from the nearest Cloudflare data center.

This hierarchy is designed for scalability. No single server needs to know everything. Root servers only know TLD locations. TLD servers only know authoritative server locations for domains in their TLD. Authoritative servers only know records for their own zones. This means the system can grow indefinitely without requiring any single component to store the entire database.

The hierarchy is also a security model. Each level delegates authority to the level below. The root servers delegate `.com` to Verisign. Verisign delegates `example.com` to the domain owner's chosen DNS provider. This chain of delegation can be cryptographically verified using DNSSEC, which we will cover later. Without DNSSEC, the delegation chain can be spoofed through cache poisoning attacks, but the hierarchical structure still provides a logical framework for understanding who is authoritative for what.

Recursive Resolution and the Stub Resolver

When you enter a URL in your browser, the translation from domain name to IP address is not performed by your computer directly. It is performed by a recursive resolver on your behalf.

The process starts with a stub resolver, which is the minimal DNS client built into your operating system. When your browser needs to resolve `example.com`, it calls the stub resolver. The stub resolver does not perform the actual lookup. Instead, it forwards the query to a recursive resolver configured on your system. This is typically the DNS server provided by your ISP, or a public resolver like Google's `8.8.8.8`, Quad9's `9.9.9.9`, or Cloudflare's `1.1.1.1`.

The recursive resolver is where the real work happens. It receives your query and performs the full resolution process on your behalf. If it has the answer cached from a previous query, it returns the cached result immediately. If not, it walks the DNS hierarchy: it queries a root server, receives a

referral to the TLD servers, queries the TLD servers, receives a referral to the authoritative servers, and finally queries the authoritative servers for the actual answer.

This is called recursive resolution because the recursive resolver performs all the iterative lookups required to get a final answer. From the stub resolver's perspective, it sends a single query and receives a single answer. The complexity is hidden behind the recursive resolver.

Recursive resolution has several important properties:

- **Caching:** Recursive resolvers cache answers to avoid repeating the full lookup chain for every query. When you resolve `example.com` and the answer is cached, the next query for `example.com` is answered instantly from the cache. This dramatically reduces latency for repeat queries and reduces load on authoritative servers.
- **Time-To-Live:** Cached answers expire based on the TTL value set by the authoritative server. The TTL tells resolvers how long they can cache an answer before they need to fetch a fresh one. Shorter TTLs mean fresher data but more queries. Longer TTLs mean faster repeat queries but slower updates.
- **Authority:** Recursive resolvers are not authoritative. They do not own the data they return. They only cache it. If you need to be authoritative for a domain, you run authoritative name servers or use a DNS hosting service like Cloudflare.

The separation between recursive and authoritative DNS is deliberate and important. Recursive resolvers are client-facing. They accept queries from anyone and perform lookups on their behalf. Authoritative servers are data-facing. They hold the records for specific zones and answer queries only for those zones. This separation prevents any single server from doing both jobs, which would make the system vulnerable to denial of service attacks.

Authoritative DNS and Zone Files

Authoritative DNS servers hold the records for specific domains. The collection of records for a domain is called a zone. The zone apex is the highest-level domain in that zone. For `example.com`, the zone apex is `example.com`.

All records for `example.com` and its subdomains like `www.example.com` and `api.example.com` are part of the same zone.

A zone is traditionally stored in a zone file, a plain text file with a specific format. Each line in the zone file defines a DNS record with a name, TTL, type, and value. Here is a simplified example:

```
1 $ORIGIN example.com.
2 $TTL 3600
3 @      IN      SOA      ns1.example.com. admin.example.com. (
4                               2026090801 ; serial
5                               7200       ; refresh
6                               3600       ; retry
7                               1209600    ; expire
8                               300 )      ; minimum TTL
9
10 ; Name servers
11 @      IN      NS       ns1.example.com.
12 @      IN      NS       ns2.example.com.
13
14 ; A records
15 @      IN      A        192.0.2.10
16 www    IN      A        192.0.2.10
17 api    IN      A        192.0.2.20
18
19 ; CNAME
20 mail   IN      CNAME    mail.google.com.
```

When you use Cloudflare DNS, you are not editing zone files directly. You are using Cloudflare’s dashboard, API, or Terraform provider to manage records. Cloudflare stores the zone data internally and serves it from its anycast network. But the underlying structure is the same: each zone has records with names, types, values, and TTLs.

Cloudflare’s authoritative DNS is fast and reliable. It operates from the same anycast network that serves HTTP traffic, so DNS queries are answered from the nearest data center. Cloudflare also provides advanced features like split-horizon DNS, where internal DNS resolvers receive different answers than public resolvers, and load balancing DNS records that distribute traffic across multiple origins.

When you add a domain to Cloudflare, you are delegating authority for that domain to Cloudflare’s nameservers. This means Cloudflare’s authoritative servers become the source of truth for your domain’s DNS records. The

delegation happens when you update your domain's nameservers at your registrar to point to Cloudflare's nameservers.

Caching, TTLs, and Propagation Myths

DNS caching is one of the most misunderstood aspects of the system. Understanding how caching works is essential for predicting how quickly DNS changes take effect and for configuring TTLs appropriately.

When a recursive resolver answers a query, it caches the result for the duration of the TTL. During that time, subsequent queries for the same name are answered from the cache without contacting the authoritative server. When the TTL expires, the resolver fetches a fresh answer from the authoritative server.

This means DNS changes take effect at different times for different users, depending on when their local recursive resolver's cache expires. If you change an A record from one IP address to another, users whose resolver has the old record cached will continue to see the old IP address until the TTL expires and the cache is refreshed. This is not a failure of DNS. It is exactly how caching is designed to work.

The myth of DNS propagation comes from this behavior. When people say "DNS takes up to 48 hours to propagate," they are describing the worst-case scenario where some resolvers cached an old answer with a very long TTL and are holding onto it. In practice, most recursive resolvers use reasonable TTL values, and most DNS changes are visible globally within a few hours at most.

The key control is the TTL value you set on your records. If you know you will be making a change, you can lower the TTL beforehand to a short value like 300 seconds (5 minutes). When you make the change, resolvers will refresh the cache quickly because the TTL is short. After the change is stable, you can increase the TTL again to reduce query load on your authoritative servers.

Cloudflare handles TTL differently depending on whether the record is proxied or DNS-only. When a record is proxied through Cloudflare (the orange cloud is enabled), Cloudflare uses anycast IP addresses instead of your origin IP. The TTL on those records is controlled by Cloudflare and is typically very short, allowing Cloudflare to move traffic between data centers if necessary. When a record is DNS-only (the orange cloud is disabled), Cloudflare respects the TTL you set, and queries are answered directly from your origin IP.

There is another caching layer to consider: browser caching. Modern browsers cache DNS results for a short period, typically a few minutes, to avoid sending DNS queries for every request. This cache is independent of the recursive resolver cache and can cause additional delay in seeing DNS changes on a specific machine. Clearing the browser cache or restarting the browser forces it to re-resolve the domain name.

TTLs are a trade-off between freshness and performance. Shorter TTLs mean faster updates but more queries to authoritative servers, which increases latency and load. Longer TTLs mean faster repeat queries and less load on authoritative servers, but slower updates. Choosing the right TTL depends on how often you expect to change records and how critical it is that changes take effect quickly.

For most production applications, a TTL of 300 to 3600 seconds is reasonable. Critical records like A and CNAME records for web applications should use shorter TTLs, especially if you need the ability to failover quickly. Mail records like MX can use longer TTLs because they change less frequently. Records you expect to change during maintenance or deployments should have their TTLs lowered before the change and raised again afterward.

Key Record Types: A, AAAA, CNAME, MX, TXT, SRV, and Beyond

DNS has many record types, each serving a different purpose. Here are the most common ones you will encounter when working with Cloudflare:

A records map a domain name to an IPv4 address. For example, `www.example.com` might have an A record pointing to `192.0.2.10`. This is the most common record type for web applications. When you add an A record in Cloudflare with proxying enabled, the address you set is your origin server's IP. Cloudflare does not return this IP to users. Instead, it returns one of its own anycast IPs, and traffic is routed through Cloudflare to your origin.

AAAA records are the IPv6 equivalent of A records. They map a domain name to an IPv6 address. Modern applications should have both A and AAAA records to support both IPv4 and IPv6 users. Cloudflare supports AAAA records for both proxied and DNS-only configurations.

CNAME records map a domain name to another domain name. For example, `www.example.com` might have a CNAME record pointing to `example.com`.

CNAMEs are commonly used for aliasing, load balancing, and pointing subdomains to services provided by other vendors. Cloudflare supports CNAME records at the zone apex through CNAME flattening, which means you can use a CNAME for your root domain even though traditional DNS does not allow CNAMEs at the apex.

MX records specify the mail servers for a domain. They include a priority value that determines the order in which mail servers are contacted. Lower priority values are preferred. MX records cannot be proxied through Cloudflare because email does not use the HTTP proxy. Cloudflare DNS handles MX records as DNS-only records that point to your mail servers.

TXT records store arbitrary text data. They are used for various purposes, including SPF records for email authentication, DKIM records for email signing, domain verification for services like Google and Microsoft, and custom metadata. TXT records cannot be proxied and are served as DNS-only.

NS records specify the authoritative name servers for a zone. When you add a domain to Cloudflare, Cloudflare provides you with a set of NS records to configure at your registrar. Those NS records point to Cloudflare's authoritative servers. You can also use NS records within a zone to delegate subdomains to different DNS providers.

SRV records specify the location of services. They include priority, weight, port, and target fields. SRV records are used for protocols like SIP, XMPP, and LDAP. Cloudflare supports SRV records as DNS-only.

PTR records map IP addresses to domain names, the reverse of A records. They are used for reverse DNS lookups. PTR records are typically managed by the organization that owns the IP address range, not by the domain owner. Cloudflare does not manage PTR records for proxied traffic.

CAA records specify which certificate authorities are allowed to issue certificates for a domain. They provide an additional layer of security for certificate issuance.

Cloudflare also supports newer record types like HTTPS and SVCB, which are part of the HTTP Alternative Services framework and allow domains to specify alternative protocols and endpoints for clients to use.

When configuring DNS in Cloudflare, understanding which record types can be proxied and which must be DNS-only is important. Only A, AAAA, CNAME, and certain other web-related records can use the orange cloud proxy.

MX, TXT, NS, SRV, and PTR records are always DNS-only.

DNSSEC: Trust in the Name System

DNS was designed without security. The original protocols did not include any cryptographic verification, which means responses can be spoofed. An attacker who can intercept or inject DNS responses can redirect traffic to a malicious server, a technique known as DNS cache poisoning.

DNSSEC adds cryptographic signatures to DNS records. With DNSSEC enabled, authoritative servers sign their records with a private key. Resolvers verify the signatures using the corresponding public key. If the signature is valid, the resolver knows the response came from the authoritative server and was not tampered with. If the signature is invalid, the resolver rejects the response.

DNSSEC is not encryption. The DNS records themselves are still readable by anyone. DNSSEC provides authenticity and integrity, not confidentiality. It ensures that the records you receive are the records that the authoritative server actually published.

DNSSEC works as a chain of trust. The root zone is signed, and the root key is distributed to all resolvers. Each TLD signs its zone and provides a DS record to the parent zone. Each authoritative zone signs its records and provides a DS record to the TLD. When a resolver queries a domain, it follows the chain of signatures from the root down to the authoritative zone, verifying each step. If any signature in the chain is invalid, the entire response is rejected.

Cloudflare supports DNSSEC for all zones. When you enable DNSSEC on a zone in Cloudflare, Cloudflare automatically generates keys, signs the zone, and publishes the DS record to your registrar (if you use Cloudflare Registrar) or provides the DS record for you to publish at your registrar. The signing process is transparent and automated. You do not need to manage keys manually.

Enabling DNSSEC adds minimal overhead. The cryptographic signatures increase the size of DNS responses slightly, but modern resolvers handle them without issue. The benefits are significant: protection against cache poisoning, protection against certain types of DDoS attacks, and compliance with security requirements in regulated industries.

There are some operational considerations. Once DNSSEC is enabled, you must ensure the signing continues to work correctly. If a zone is signed but the key expires or is rotated incorrectly, the zone becomes unsigned from the resolver's perspective, and queries will fail with a validation error. This can cause a complete outage. Cloudflare automates most of this management, but it is still important to understand the dependency.

If you are using DNSSEC, you should also consider using encrypted DNS protocols like DNS over HTTPS, DNS over TLS, or DNS over QUIC. These protocols encrypt the DNS query itself, preventing eavesdropping and manipulation on the network path between the client and the resolver. Cloudflare's 1.1.1.1 resolver supports all three encrypted protocols.

Chapter Summary

- DNS is a hierarchical system with root servers, TLD servers, and authoritative servers. Each level delegates authority to the level below.
- Recursive resolvers perform the full lookup process on behalf of clients, caching results for the duration of the TTL.
- Stub resolvers in operating systems forward queries to recursive resolvers and do not perform lookups themselves.
- Zones are collections of DNS records for a domain. Cloudflare manages zones and serves them from its anycast network.
- TTLs control how long cached records are valid. Shorter TTLs allow faster updates but increase query load. Longer TTLs improve performance but slow down updates.
- DNS changes do not “propagate” in the traditional sense. They take effect gradually as cached records expire, based on TTL values.
- Common record types include A (IPv4), AAAA (IPv6), CNAME (alias), MX (mail), TXT (text), NS (name servers), and SRV (services).
- Proxyable records (A, AAAA, CNAME) can use Cloudflare's orange cloud. Other records are DNS-only.
- DNSSEC provides cryptographic verification of DNS records, protecting against cache poisoning and tampering.
- Understanding DNS is essential for configuring Cloudflare zones correctly and troubleshooting resolution issues.

In the next chapter, we will look at how packets actually travel across the Internet, covering BGP, anycast routing, and the network infrastructure that makes CDNs possible.

Chapter 3: Routing the Internet: BGP, Anycast, and IP Transit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Autonomous Systems and the BGP Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Route Announcement, Prefixes, and Path Selection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

What Anycast Is and Why CDNs Depend on It

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

IP Transit, Peering, and Internet Exchanges

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

When Routing Goes Wrong: Leaks, Hijacks, and Outages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 4: Transport and Application Protocols at the Edge

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

TCP Connections: Handshake, Retransmit, and Congestion Control

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

TLS: Encryption, Handshake Costs, and Session Resumption

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

HTTP/1.1: Pipelining, Keep-Alive, and Head-of-Line Blocking

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

HTTP/2: Multiplexing, Frames, and Server Push

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

HTTP/3 and QUIC: UDP-Based Transport with Built-In TLS

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Protocol Negotiation: ALPN and the Actual User Experience

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 5: The CDN Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Reverse Proxy Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Edge Nodes, Points of Presence, and Topology

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Cache: Hitting, Missing, Stale, and Bypassing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Origin Shielding and Tiered Caching

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

CDN Request Lifecycle: End User to Origin and Back

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

What a CDN Can and Cannot Accelerate

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 6: Cloudflare's Architecture and Network

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare's Global Network: Data Centers and POPs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare's Anycast IP Space and How Traffic Lands

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Request Flow Through Cloudflare: Edge to Origin

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

How Cloudflare Differs from Akamai, Fastly, and Others

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

What Is and Is Not Visible to the End User

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 7: Getting Started: Accounts, Zones, and Nameservers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Accounts, Memberships, and Plan Tiers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Adding a Site and Understanding Zone Status

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Transferring DNS: Nameservers and Cutover Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Proxied vs. DNS-Only: The Orange Cloud

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Verifying Your Site Is Live and Troubleshooting the Initial Setup

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration Examples: Initial DNS Records for a Real Site

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 8: DNS on Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare DNS: Authoritative, Managed, Fast

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Managing Records: Dashboard, API, and Terraform

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Split-Horizon DNS and Internal Resolvers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Load Balancing DNS Records and Traffic Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

DNS Automation: API Patterns and Terraform Modules

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Operational Considerations: Propagation, Cutover, and Rollback

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 9: SSL/TLS with Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Cloudflare SSL/TLS Problem Space

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

SSL/TLS Modes: Off, Flexible, Full, Full (Strict)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Certificate Types: Universal, Custom, Dedicated, Origin CA

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Certificate Management and Renewal

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Security Trade-offs and Common Mistakes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration Example: Hardened TLS Setup for Production

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 10: Web Application Firewall and Security Rules

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The WAF Rule Engine: How Cloudflare Inspects Requests

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Managed WAF Rulesets: OWASP, Category-Based, and Threat Intelligence

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Custom WAF Rules and Field Operators

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Firewall Rules: The Legacy System and When to Use It

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Security Rules and Traffic Filtering

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Avoiding False Positives and Rule Conflicts

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 11: DDoS Protection, Rate Limiting, and Bot Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

DDoS at the Edge: How Cloudflare Absorbs Attacks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Rate Limiting Rules: Tokens, Windows, and Responses

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Bot Fight Mode and Advanced Bot Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Challenge Pages: JavaScript, Turnstile, and Under Attack Mode

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

API Shield and Automated Threat Mitigation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration Examples: Rate Limiting and Bot Policies for Real Applications

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 12: Zero Trust, Access, and Private Networks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Zero Trust Architecture and Cloudflare's Approach

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare Access: Identity-Based Application Access

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Zero Trust Gateway: Secure Web and DNS Gateways

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare Tunnel: Exposing Origins Without Open Ports

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Integrating Identity Providers and Device Posture

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Architecture Example: Secure Internal App Behind Access and Tunnel

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 13: Caching Fundamentals in Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

HTTP Caching Headers: Cache-Control, ETag, Expires

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

How Cloudflare Decides What to Cache

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cache Keys, Variants, and Query String Handling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Edge TTLs and Origin Response Headers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cache Hits, Misses, Bypasses, and Stale Responses

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Reading and Understanding Cloudflare Cache Headers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 14: Cache Rules, Page Rules, and Advanced Caching

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cache Rules: The Modern Rule System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Page Rules: Legacy Behavior and Migration Path

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cache Reserve: Pay-For-Priority in Peak Times

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Tiered Caching: Two-Tier and Smart Tiered

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cache Purging: Individual URLs, Tags, and Bulk

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration Examples: Caching Strategy for a Real Application

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 15: Optimization Features: Compression, HTTP/2, HTTP/3, and Argo

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Compression: Gzip, Brotli, and Automatic Settings

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

HTTP/2 and HTTP/3: Enable, Disable, and When It Matters

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Argo Smart Routing: Private Backbone and Latency Reduction

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Rocket Loader, Minify, and Miscellaneous Optimizations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Measuring Impact: What Actually Moves the Needle

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration Example: Performance Stack for a Dynamic Application

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 16: Cloudflare Workers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

What Workers Are: JavaScript at the Edge

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Workers Runtime: V8 Isolates and Execution Model

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Routing: Hostnames, Patterns, and Dispatch

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Workers Examples: Rewrites, A/B Testing, Auth, and Transformations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Limits, Quotas, and Cold Starts (Or Lack Thereof)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration: Wrangler, Deploy, and Environment Variables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 17: Workers Data Products: KV, Durable Objects, D1, and Queues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Workers KV: Global Key-Value Storage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Durable Objects: Stateful Co-located Compute

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

D1: SQLite at the Edge

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Queues: At-Least-Once Delivery for Background Work

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Choosing the Right Data Product: Comparison and Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Architecture Example: Worker-Powered API with KV and D1

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 18: Pages, R2, Stream, Images, and WebSockets

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare Pages: Serverless-Style Static and Full-Stack Hosting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

R2: Object Storage Without Egress Fees

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Stream: Video Transcoding and Delivery

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Images: On-the-Fly Transformations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

WebSockets: Persistent Connections Through Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Architecture Example: Media-Rich Application on Cloudflare Platform

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 19: Cloudflare Load Balancing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

What Load Balancing Solves: Global Affinity and Failover

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Pools, Monitors, and Load Balancers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Health Checks: HTTP, TCP, and Custom

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Steering Policies: Least Connections, Geolocation, and Custom

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Failover Behavior and Graceful Degradation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Configuration Example: Multi-Region Application with Load Balancing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 20: Advanced Architectures and Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Static Site Architecture on Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Dynamic Web Application Behind Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

API-First Architecture with Edge Security

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Multi-Region Origins with Load Balancing and Tiered Cache

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Kubernetes and Containerized Backends Behind Cloudflare

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Private Origin Architectures with Tunnel and Access

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 21: Logging, Analytics, and Observability

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Analytics Dashboard: Traffic, Performance, and Security

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Logpush: Exporting Logs to Your Systems

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Building Observability Pipelines with Cloudflare Data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Metrics That Matter: Latency, Cache Ratio, Errors, and Threats

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Alerts and Notifications: Proactive Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Example: Setting Up Full Observability for a Production Site

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 22: Troubleshooting Cloudflare Deployments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Troubleshooting Methodology: Isolate the Layer

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

DNS and Resolution Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

SSL/TLS Errors and Misconfigurations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Caching Problems: Stale Content and Unexpected Behavior

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Redirect Loops and Proxy Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Worker Debugging and Routing Failures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Common Configuration Mistakes and How to Avoid Them

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 23: Automation and Infrastructure as Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cloudflare API: Authentication, Rate Limits, and Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Terraform Provider: State, Resources, and Modules

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

CI/CD for Workers and Pages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Automating Security Rules and DNS Changes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

GitOps and Configuration Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Example: Terraform Module for a Production Zone

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter 24: Production Operations and Migration Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Migrating to Cloudflare: Strategy and Execution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cutover Strategies: Big Bang vs. Gradual

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Cost Management and Plan Selection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Capacity, Limits, and Scaling Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Change Management and Rollback Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Running Cloudflare in Mission-Critical Environments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Conclusion: The Edge as a Platform

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

What You Know Now: The Edge Architecture Stack

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Choosing Cloudflare: When It Makes Sense and When It Does Not

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

The Future of the Edge: Compute Everywhere

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

A Practical Decision Framework

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

Final Thoughts on Mastering the Edge

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.

References

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/cloudflareandthemoderncdn>.