

# Cloudflare Training Plan - Beginner to Advanced

This training plan is designed for engineers who are new to Cloudflare. It provides a **structured, hands-on progression** from foundational knowledge to advanced engineering and security practices. Each stage covers **technical concepts, processes, labs, troubleshooting, and best practices**.

---

## 1. Beginner Level - Foundations

### Technical Concepts

- **CDN Fundamentals** – Cloudflare as a reverse proxy & CDN; comparison with traditional hosting.
- **DNS Basics** – Authoritative vs. Recursive DNS, record types (A, CNAME, MX, TXT, NS, etc.), Orange-cloud vs Grey-cloud.
- **SSL/TLS Fundamentals** – Flexible, Full, Full (Strict) modes; Origin vs. Cloudflare-managed certificates.
- **Caching Basics** – Edge caching, Browser vs. CDN cache, default behaviors.
- **Cloudflare Rules Overview** – Page Rules vs. Ruleset Engine; redirects (301 vs. 302).
- **Firewall Basics** – WAF, IP blocking, Country blocking, Managed Rulesets.

**Best Practice:** Start with DNS hygiene and SSL/TLS configuration before enabling caching or firewall rules.

### Process Concepts

- Onboarding a domain into Cloudflare.
- Changing nameservers to Cloudflare.
- Identifying which DNS records to proxy.
- Enabling HTTPS using Universal SSL.

### Labs

Task	Description
Domain Setup	Add a test domain and configure A/CNAME records.
SSL Enablement	Enable Universal SSL and verify HTTPS connectivity.
Redirect Rule	Create a redirect rule (www → non-www).
Cache Test	Test caching using <code>CF-Cache-Status</code> headers.

### Troubleshooting

Issue	Guidance
DNS Misconfiguration	Check for missing A records, propagation delays.

Issue	Guidance
SSL Handshake Errors	Ensure correct SSL mode alignment.
Cache Bypass	Verify Development Mode or bypass rules.

## 2. Intermediate Level - Operations & Optimization

### Technical Concepts

- **Advanced DNS** – CNAME Flattening, Load Balancer basics.
- **Rules Engine** – Redirects with expressions, URL rewrites, Origin rules.
- **Caching Controls** – Cache Everything, Custom cache keys, Tiered caching.
- **Firewall Rules & Bot Management** – IP/ASN/Headers filtering, Bot Fight Modes.
- **Rate Limiting** – Path-based and method-specific controls.
- **Workers Basics** – Edge compute, simple redirects, header modifications.

**Best Practice:** Document rule lifecycles (staging → validation → production) to avoid downtime.

### Process Concepts

- Redirect migration strategy (e.g., Akamai → Cloudflare).
- Rule lifecycle management.
- Caching strategies for static vs. dynamic content.
- Documenting Firewall/Ruleset policies.

### Labs

Task	Description
Country Redirects	Create rules for country-based redirection.
Origin Rule	Deploy rule for custom port mapping.
Cache Everything	Apply rule for <code>/static/*</code> .
Firewall IP Allowlist	Configure firewall rule to allowlist internal IPs.
Worker Deployment	Deploy Worker for <code>/promo → /new-promo</code> .

### Troubleshooting

Issue	Guidance
Firewall False Positives	Analyze logs, use preview mode.
Redirect Loops	Validate rules and redirects.

Issue	Guidance
Dynamic Content Caching	Exclude APIs or dynamic paths.
Worker Errors	Debug using <code>wrangler logs</code> .

---

### 3. Advanced Level – Engineering & Security

#### Technical Concepts

- **Workers & KV/Durable Objects** – State management, config storage.
- **API Shield** – API discovery, schema validation, JWT, mTLS.
- **Zero Trust (ZTNA)** – WARP, Cloudflare Tunnel, IDP integration.
- **Traffic Steering & Load Balancing** – Session affinity, Geo-routing.
- **Advanced WAF** – Attack Score, Bot Score rules.
- **Logs & Monitoring** – Logpush, Real-time logs.
- **Enterprise Automation** – Multi-zone Terraform automation.

**Best Practice:** Automate configurations with Terraform for consistency, rollback capability, and scalability.

#### Process Concepts

- Secure onboarding checklist (DNSSEC, TLS 1.3, HSTS, WAF tuning).
- Automated rule management via Terraform.
- Incident response using Cloudflare logs.
- Scaling from POC to enterprise rollout.

#### Labs

Task	Description
JWT Worker	Deploy Worker for JWT validation.
API Shield	Implement schema validation.
Tunnel Setup	Configure Cloudflare Tunnel for internal apps.
Geo Load Balancing	Create geo-based load balancing pools.
Terraform Automation	Automate DNS & WAF via Terraform.

#### Troubleshooting

Issue	Guidance
Worker Limitations	Monitor CPU/memory thresholds.

Issue	Guidance
API Schema Errors	Validate schema definitions.
Bot Detection Issues	Fine-tune false positives/negatives.
Load Balancer Stickiness	Check session persistence configs.
Debugging	Use <code>cf-ray</code> headers, Firewall Events, Logpush.

---

## Appendix – Cloudflare Troubleshooting Tools

### DNS & Network

- **dig** – Query DNS servers.  
Example: `dig A cloudflare.com`
- **WHOIS** – Verify registrar, nameservers, domain status.

### HTTP/S Analysis

- **cURL** – Test HTTP/S requests.  
Example: `curl -sv -o /dev/null https://cloudflare.com`
- **Browser DevTools & HAR files** – Inspect and export HTTP flows.

### Connectivity

- **Traceroute & MTR** – Identify network latency and routing issues.  
Example: `mtr cloudflare.com -rwc 30 -4`
- **Wireshark** – Deep packet capture and analysis.

**Best Practice:** Always capture HAR or packet-level data before escalating issues to Cloudflare support.