Modernising Application Access

Replacing F5 with Cloudflare Tunnel & Zero Trust

Solutions Engineering

Today's Agenda

Current State

Architecture review & the challenge with F5 in a Cloudflare world

The Solution

Cloudflare Tunnel, Access & Load Balancing — end state design

Migration & Business Case

Three-phase migration plan, benefits, risks & next steps

Cloudflare Tunnel — outbound-only, no public IP required

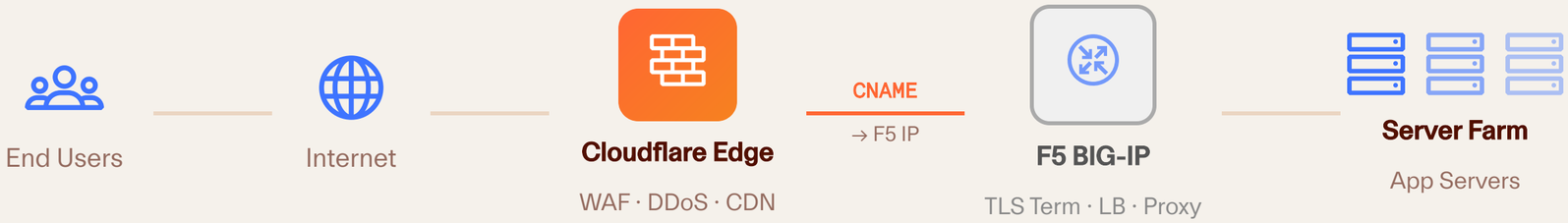
Cloudflare Access — identity-aware proxy, Zero Trust policies

Cloudflare Load Balancing — replaces F5 LB with active health checks

Cloudflare SSL/TLS — automated cert management, Universal & Total TLS

Current State

Current Architecture



Public inbound
F5 has a public IP — origin can be targeted directly if Cloudflare is bypassed

Cert duplication
TLS managed separately on both Cloudflare edge and F5 — two renewal lifecycles

No app-layer auth
No identity-aware access control — anyone reaching F5 hits the app directly

The Challenge with F5 in a Cloudflare World

SECURITY GAPS

- ⊘ **Origin exposure** — F5's public IP is discoverable; origin bypass attacks are a real risk
- ⊘ **No Zero Trust enforcement** — network perimeter model with no per-user, per-device policy
- ⊘ **Inbound firewall rules required** — open ports expose the server farm to potential attack

OPERATIONAL OVERHEAD

- ⊘ **Dual TLS lifecycle** — certificates managed separately on CF edge and F5
- ⊘ **F5 licensing cost** — ongoing subscription, hardware refresh cycles, specialised skills
- ⊘ **Redundant functionality** — Cloudflare already proxies traffic; F5 adds latency and complexity for capabilities CF can provide natively

> *You are already paying for Cloudflare to proxy and protect your traffic — F5 as a middle layer duplicates cost and creates security gaps rather than closing them.*

The Solution

Cloudflare Replaces F5 — Component by Component

Cloudflare Tunnel

Replaces F5 reverse proxy & inbound connectivity

```
cloudflared daemon
outbound-only
no public IP needed
```

CF Load Balancer

Replaces F5 load balancing & health checks

```
active-active pools
geo-steering
health monitors
```

Cloudflare Access

Adds identity-aware auth (replaces F5 APM)

```
IdP integration
per-app policies
Zero Trust JWT
```

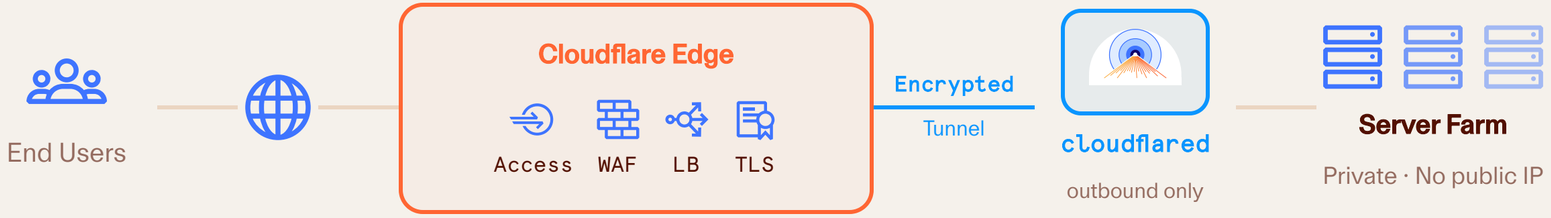
CF SSL / TLS

Replaces F5 certificate management

```
Universal SSL (free)
Total TLS
auto-renewal
```

Net result: F5 is entirely removed from the ingress path. Your server farm is no longer reachable from the public internet — **only Cloudflare's network can reach your servers, via outbound tunnel connections your servers initiate.**

End State Architecture

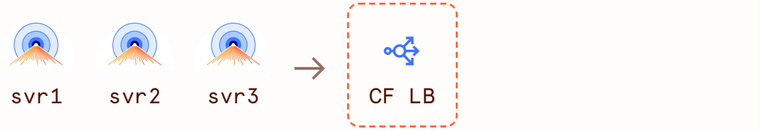


- **F5 removed** — no public IP, no inbound holes, attack surface eliminated
- **Access enforced at edge** — identity checks before traffic reaches your network
- **Single TLS layer** — cert lifecycle fully managed by Cloudflare, auto-renewed

Tunnel Topology: Two Deployment Models

Option A

cloudflared on Every Server



- Fine-grained per-server health checks
- Session affinity per-server possible
- N tunnels to manage — higher ops overhead
- Separate UUID per server required for LB affinity

Option B ★ Recommended

Dedicated Proxy Server(s)

- **Simplest migration path** — mirrors existing F5 topology
- Run 2+ replicas for HA — each makes 4 connections to CF
- Single tunnel config — all app routes in one place
- Replicas on same UUID = single LB endpoint (no per-replica affinity)

Each `cloudflared` instance creates **4 persistent connections** to at least **2 distinct Cloudflare data centres** — redundancy is built in.

Load Balancing: Replacing F5 LTM

HOW IT WORKS WITH TUNNEL

Tunnel endpoint address
`<UUID>.cfargotunnel.com`

Host header override — tells cloudflared which app route to serve
`Header: Host: app.example.com`

DNS CNAME update
app.example.com → lb.example.com (LB hostname)
Previously pointed to F5 IP — now points to CF LB

CAPABILITIES

- **Active-Active** — multiple tunnel UUIDs per pool, traffic distributed across all
- **Active-Passive failover** — primary pool + fallback pool, automatic failover
- **Health monitors** — HTTPS monitors with host-header override per app
- **Geo-steering & latency routing** — route users to nearest healthy origin
- **TCP monitors not supported** for tunnel endpoints — use HTTPS monitor with `HTTP_STATUS 200` health-check route

TLS Management & Zero Trust Access

SSL / TLS Management

Universal SSL — free, auto-issued for your zone. Covers apex + one wildcard level

Total TLS — automatically issues certificates for all subdomain levels. Eliminates manual cert work entirely

Origin certs optional — tunnel traffic is encrypted end-to-end by cloudflared. Origin servers don't need public certificates

Custom cipher suites & min TLS version — configurable per zone for compliance requirements

Cloudflare Access

Identity-aware proxy — policies enforced at the Cloudflare edge before traffic reaches your network

IdP integration — Azure AD, Okta, SAML, OIDC, Google. Single sign-on for all apps

Per-app policies — group membership, device posture, geo, MFA enforcement per application

Enable "Protect with Access" in Tunnel settings — cloudflared validates the JWT so even misconfigured bypasses are blocked at origin

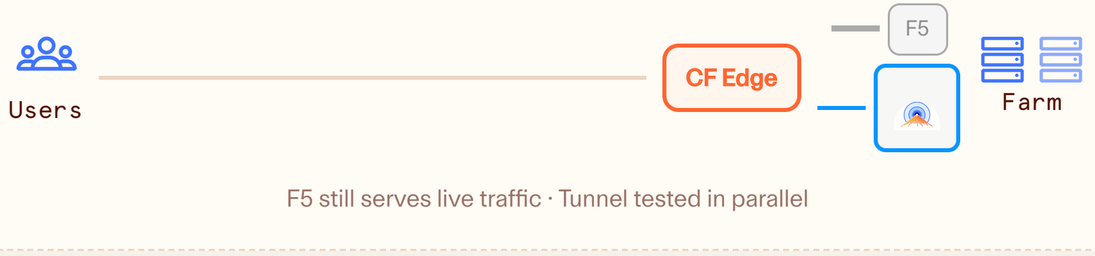
Migration Plan

Phase 1 — Parallel Deployment Zero Risk

ACTIONS

1. Deploy `cloudflared` on dedicated proxy host(s)
2. Create tunnel with published routes for each application
3. Configure Cloudflare Load Balancer pools using tunnel UUIDs
4. Create Access applications and connect IdP
5. Verify firewall allows outbound port 7844 from `cloudflared` hosts
6. **Test each app via tunnel UUID directly** — `<UUID>.cfargotunnel.com`

STATE



F5 still serves live traffic · Tunnel tested in parallel

Rollback: Nothing to roll back — F5 is untouched. Stop cloudflared and remove test DNS entries.

Phase 2 — DNS Cutover

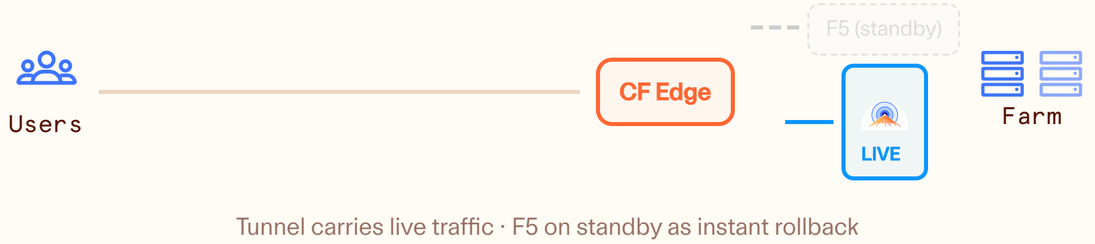
Low Risk

ACTIONS

1. Update DNS CNAMEs — change from F5 IP to CF Load Balancer hostname
2. Monitor application logs and CF analytics for any errors
3. Validate Access policies — confirm users authenticate correctly
4. Test all application endpoints and critical user journeys
5. Keep F5 live and accessible — it is the rollback target
6. Run in this state for a defined soak period (e.g. 1–2 weeks)

```
Before: app.example.com → 10.x.x.x (F5 IP)
After:  app.example.com → lb.example.com (CF LB)
```

STATE



Tunnel carries live traffic · F5 on standby as instant rollback

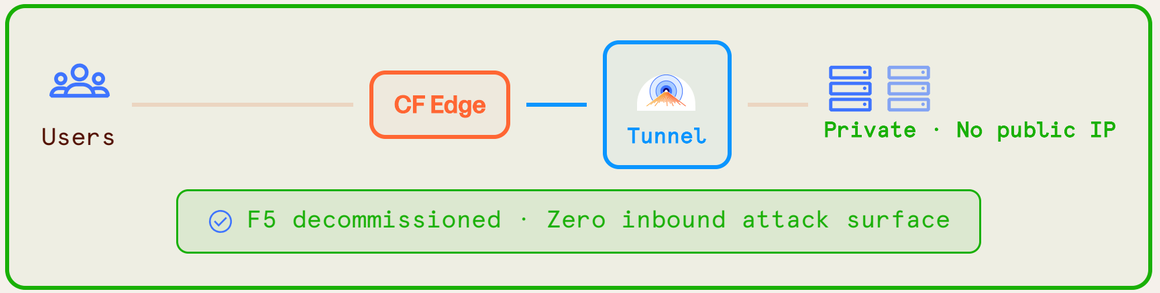
Rollback: Single DNS change — revert CNAMEs back to F5 IP. <60 second TTL recommended during cutover window.

Phase 3 — Decommission F5 Final State

ACTIONS

1. Confirm zero residual traffic hitting F5 — check F5 access logs
2. Tighten server farm firewall — **block ALL inbound**, allow only outbound 7844
3. Remove F5 from DNS entirely and revoke public IP allocation
4. Cancel F5 licensing / maintenance contracts
5. Remove F5 TLS certificates and certificate renewals from runbooks
6. Repurpose or decommission F5 hardware / VM

FINAL STATE



F5 licensing
✓ Cancelled

Manual cert renewals
✓ Eliminated

Business Case

Pros & Cons

Advantages

Zero inbound attack surface — servers are not reachable from the internet at all. Eliminates origin bypass attacks.

Identity-aware access control — Zero Trust policies enforced at edge, per-app, per-user, per-device

Reduced operational complexity — one platform to manage (Cloudflare), no F5 iRules, no dual cert lifecycle

Cost reduction — F5 licensing, hardware refresh and specialist skills eliminated

Safe, reversible migration — three phases, F5 live as fallback, DNS rollback <60 seconds

Considerations

CF Load Balancer is a paid add-on — cost must be evaluated against F5 licensing savings (almost always net positive)

Complex F5 iRules need review — most standard rules map to CF Rules, but bespoke logic requires individual assessment

Session affinity constraints — per-host affinity requires separate tunnel UUIDs, adds management overhead

Multi-level subdomain certs — require Advanced Certificates (not covered by Universal SSL free tier)

cloudflared dependency — a new daemon to operate and keep updated on cloudflared hosts

Business Benefits

Reduced Security Risk

Origin servers become unreachable from the internet. Zero Trust policies enforce identity and device trust on every request. Eliminates a category of attack entirely.

Cost Savings

Eliminate F5 licensing, maintenance contracts, hardware refresh cycles, and specialist F5 engineering costs. Consolidate spend into Cloudflare.

Operational Simplicity

One platform, one dashboard, one support contract. TLS certificates auto-renew. No iRule maintenance. Faster onboarding of new applications.

Compliance & Audit

Cloudflare Access creates a full audit trail of who accessed which application, when, from which device. Strengthens posture for SOC 2, ISO 27001, and similar frameworks.

Performance

Remove a hop from the traffic path. Cloudflare's Anycast network routes users to the nearest PoP. Argo Smart Routing available for further optimisation.

Scalability

Cloudflare's global network absorbs traffic growth and DDoS attacks at scale. No hardware capacity planning for load balancers. Add new applications in minutes.

F5 → Cloudflare: Feature Replacement Matrix

F5 Capability	Cloudflare Equivalent	Coverage
L7 Load Balancing	`Cloudflare Load Balancing`	✓ Full
SSL/TLS Termination	`Universal SSL / Advanced Certs`	✓ Full
Reverse Proxy / Ingress	`Cloudflare Tunnel (cloudflared)`	✓ Full + Better
Application Auth (APM)	`Cloudflare Access`	✓ Full + Better
WAF / DDoS	`CF WAF / DDoS (already active)`	✓ Already in place
Health Checks	`CF LB Monitors (HTTPS)`	~ HTTPS only*
Session Persistence	`CF LB Session Affinity`	~ Separate UUIDs req'd
Geo / Latency Routing	`CF LB Traffic Steering`	✓ Full
Certificate Management	`Total TLS / Cert Manager`	✓ Full + Auto-renew
iRules / Traffic Scripting	`CF Rules / Transform Rules`	~ Review required

* TCP monitors not supported for tunnel endpoints — use HTTPS health-check route in cloudflared

Key Risks & Gotchas

Session Affinity & Replicas

Replicas sharing a tunnel UUID = single LB endpoint. Separate UUIDs required per server for per-host affinity. Plan this upfront.

Port 7844 Firewall Rule

cloudflared requires outbound TCP 7844. Verify connectivity before Phase 1 using CF's pre-check tool. Blocking this port = no tunnel.

Multi-Level Subdomain Certs

dev.app.example.com style hostnames require Advanced Certificates (paid). Universal SSL covers only one subdomain level.

Access Token Bypass

Always enable "Protect with Access" in Tunnel settings. This makes cloudflared validate the CF JWT — blocking requests that bypass Access via network misconfiguration.

Anycast Local Preference

cloudflared prefers serving requests via PoPs in the same region. With geographically distributed endpoints, traffic imbalances can occur — review LB steering policies.

Create Access App Before Tunnel Route

If you publish a tunnel route before creating the Access application, the app is publicly accessible to anyone. Always configure Access policies first.

Recommended Next Steps

1 iRules Discovery Workshop

Audit all F5 iRules and Virtual Server configs. Map each to a Cloudflare Rules equivalent. Identify any requiring custom Workers logic.

2 Application Inventory

List all applications behind F5, their hostnames, subdomain structure, and any session affinity requirements. This drives tunnel and LB design.

3 IdP Integration Planning

Identify your identity provider (Azure AD, Okta, etc.) and plan the Cloudflare Access integration. Define initial access policies per application.

4 Firewall & Connectivity Verification

Verify outbound TCP 7844 is permitted from the intended cloudflared hosts. Run Cloudflare's connectivity pre-check tool on each host.

5 Proof of Concept

Deploy cloudflared in a non-production environment. Validate tunnel connectivity, LB health checks, Access authentication, and application routing end-to-end.

6 Cloudflare SE-Led PoC

We can run a structured PoC — Cloudflare provides technical resources to validate all requirements before any production commitment.