

Certificate lifecycle management for F5 BIG-IP

PKIFactor discovers what your **F5® BIG-IP® Local Traffic Manager™ (LTM)** appliances actually serve, decides whether a write is safe, and deploys certificates — with the private key born on the appliance or its HSM — across **client-ssl**, **server-ssl** and **forward-proxy CA** profiles, from any issuing CA behind one policy. Agentless and REST-first, it complements the **F5® Application Delivery and Security Platform (ADSP)**: F5 ADSP delivers and secures applications; PKIFactor governs the certificate lifecycle behind them.

1 · WHAT PKIFACTOR DOES — AND DOES NOT DO — ON BIG-IP

DOES
Reads virtual servers, client-ssl / server-ssl profiles and served certificates (serial, SPKI) — read-only discovery
Installs certificate / key / chain objects and repoints the SSL profile, inside an iControl® transaction
Generates the key and CSR on the appliance (software, on-board FIPS or network HSM custody)
Saves the configuration (<code>save sys config</code>) and triggers config-sync to the device group
Reads TLS options and cipher strings of client-ssl profiles for posture

DOES NOT
Does not create, modify or delete virtual servers
Does not write the trust store (ca-file) of server-ssl profiles — inventory and grading only
Does not touch iRules , AS3 declarations or iApps
Never overwrites a profile owned by AS3 / iApp (BLOCK) or one serving a foreign or drifted certificate (BLOCK / HOLD) without an explicit, audited adopt
Never writes on a standby unit : follows the active member of the pair automatically (same credentials, no reconfiguration after a failover); refuses when the active peer is unreachable, unless explicitly allowed for a lab (<code>allow_standby_writes</code>)

2 · THREE TLS ROLES ON THE APPLIANCE

ROLE (PROFILE_TYPE)	WHAT THE CERTIFICATE IS	CUSTODY OPTIONS	HITLESS ROTATION	GUARDRAIL
client-ssl	Server certificate presented to inbound clients (<code>certKeyChain</code>)	Key on device (<code>csr_on_device</code>) or key pushed (<code>push_both</code>)	Yes — versioned objects, repoint, cleanup	Foreign / drifted incumbent protected (BLOCK / HOLD); greenfield <code>/Common/default.crt</code> recognised
server-ssl	Client certificate the BIG-IP presents for outbound mTLS to a backend or external service (<code>cert</code> / <code>key</code> / <code>chain</code> set directly)	Key on device or key pushed; the appliance signs the handshake, so the key must live on it	Yes — same engine as client-ssl	EKU check: a template that provably excludes clientAuth is refused; trust store read-only
forward-proxy-ca	Inspection CA the BIG-IP uses to forge per-site leaves for outbound TLS inspection (<code>certKeyChain usage:CA</code> + <code>sslForwardProxy</code>)	Key on device (recommended) or key pushed	Yes — SERVER entry preserved	Refuses any certificate that is not a CA (<code>basicConstraints CA:TRUE</code> required)

■ 3 · USER MODEL — AUTOMATION × CUSTODY

The operator sees one axis, the **Automation level**, combined with where the key lives. The API keeps two internal fields, `delivery_mode` and `auto_renew`, shown in small text below.

AUTOMATION	KEY CUSTODY	WHAT PKIFACTOR DOES	RENEWES AT D-X AUTOMATICALLY?	API PAIR
Full auto	Key on device	Re-provisions the key + CSR on the appliance at D-X, gets it signed, installs and repoints — the re-provision <i>is</i> the go-live	Yes	<code>provide</code> · <code>auto_renew=true</code>
	Key pushed	Issues, pushes cert + key + chain, repoints, saves and syncs — hands-off	Yes	<code>full_auto</code> · <code>auto_renew=true</code>
Install	Key on device	Prepares; the operator triggers <i>Provision on appliance</i> — provisioning is manual	No	<code>provide</code> · <code>auto_renew=false</code>
	Key pushed	Issues and installs the renewed objects as staged (not active); the operator activates	Yes — staged, operator activates	<code>push_staged</code> · <code>auto_renew=true</code>
ACME (advanced)	Key on device (by nature)	The appliance renews itself against PKIFactor's ACME endpoint; PKIFactor supervises (inventory, drift, digest)	The box renews itself	<code>acme_pull</code> · <code>auto_renew=false</code>
Provide (record-only)	Key pushed	Nothing is written on the appliance; a renewed certificate, if renewal is enabled, is only made <i>available</i> for pickup	No write	<code>provide</code> · <code>auto_renew</code> as set

■ 4 · SAFE ORCHESTRATOR — SITUATION → DECISION

PKIFactor never deploys blindly. Before writing it reads what the profile really serves and decides: deploy, stage, hold, or refuse. Decisions are taken by a pure policy engine from a live risk preflight (blast radius, SNI-shared VIP, inherited certificate, certificate owner, key custody, RBAC scope, config ownership).

SITUATION READ ON THE APPLIANCE	DECISION	WHY
Profile serves a foreign certificate (another CA)	BLOCK	a partner identity is never overwritten
Out-of-band drift (appliance ≠ what PKIFactor deployed)	HOLD	deliberate manual change or incident: alert, do not overwrite
FIPS / HSM key on the profile + key-pushed custody	BLOCK	key not exportable → custody must be key-on-device
Service account outside the partition's RBAC scope	BLOCK	the write would return 403 mid-transaction; said upfront
Profile owned by laC (AS3 / iApp)	BLOCK	an out-of-band write would be erased at the tool's next deployment
Shared VIP / SNI / inherited certificate	STAGE	supervised cutover (blast radius wider than one profile)
Fresh profile serving the placeholder <code>/Common/default.crt</code>	DEPLOY	greenfield recognised → first deployment allowed

Adopt / reclaim — a deliberate, audited, one-shot action that lifts *only* the incumbent protection (BLOCK foreign / HOLD drift). Hard incompatibilities (FIPS/HSM, RBAC, laC) and topology safeties stay in force. The UI only offers the actions the backend would accept.

■ 5 · DISCOVERY & ONBOARDING

- **Estate discovery** by listing: virtual server → SSL profile → served certificate, plus served TLS options — read-only, pure REST
- **Onboarding by issuing CA**: certificates grouped in five buckets — **managed** (issuer is one of our CAs, adopt in place) / **foreign** (reissue from a template) / **unverified** (issuer not checkable) / **self-signed** / **unknown** (unreadable, human look)
- **Reconcile** (one binding) and **reconcile-all** (per bucket) — **dry-run by default**, full preview and zero writes until confirmed; matching by normalised serial, then SPKI
- **Adopt / reclaim** a protected incumbent; **greenfield** profiles still on `/Common/default.crt` deploy without forcing

■ 6 · RENEWAL

- **Hourly D-X worker**; default threshold **D-30** (per endpoint, overridable per binding); the threshold is clamped to a third of a short-lived certificate's lifetime
- **Issuing-CA latency budget**: public CAs (Sectigo, DigiCert) fire **5 days earlier**; local CAs sign synchronously
- **Maintenance hour** (UTC) honoured, with a **2-day near-expiry grace** that overrides the window
- **Stuck-renewal reaper**: a binding claimed for renewal or activation for more than **24 h** is released and flagged
- **Failure memory** with exponential backoff **1 h → 24 h**; an already-issued renewal is **re-deployed**, not re-issued
- **Persisted deployment jobs**: an orphaned job (container restart, lock loss) is replayed at the next tick, up to 5 attempts
- **One-shot schedule** per binding (consumed once); **Renew + deploy now** on demand
- **Digest** of what will move within **7 days** (**90 days** for foreign-CA certificates), sent daily to the owner team
- **Notice recipients** (pinned peers, mTLS partners) warned ahead; they can **veto** by pausing the binding — a reason is mandatory — and never receive a private key

■ 7 · KEY CUSTODY & HSM

- Per-appliance key custody: **normal** (software key, the path validated live), **fips** (on-board FIPS HSM) or **nethsm** (external network HSM, **partition required**); the HSM brand is the appliance's `external-hsm` configuration, not PKIFactor's
- Anything but *normal* makes the key non-exportable: the endpoint **refuses any push of key material**; such bindings use key-on-device custody
- **HA netHSM preflight**: every device-group member must be enrolled with the external HSM — a member provably not enrolled **blocks**; an unreachable member is **reported, not blocking**
- **Same-key rotation** (opt-in per binding): keeps the SPKI stable for pinned peers and does not consume an HSM slot
- On-device algorithms **RSA-2048 / 3072 / 4096** and **ECC P-256 / P-384 / P-521**, validated against the certificate template

■ 8 · OPERATIONS SECURITY

- **resource-admin** service account with **SOAP install** (on by default): the full cycle without file transfer and without the Administrator role on that path — verified live on 17.5.1.8. With SOAP off, the REST file-transfer path needs Administrator and key pushes are refused
- **Partition-scope probe** at test-connection: an account broader than the partition it writes to (all-partitions) is flagged; managed-partition allowlist
- **Follows the active member** when the configured host is standby (announced at test-connection), and **write refusal** when the active peer is unreachable or the group is not *In Sync* (no automatic from-group; a human arbitrates)
- **AS3 not readable** → warning (the IaC-ownership check could not run); AS3-owned profiles are **BLOCKED**
- **iControl transaction** with rollback on the REST path; SOAP-imported objects rolled back if the reposit fails
- `save sys config` after every commit, then config-sync and a convergence wait of **~30 s**; non-convergence is warned, never silently ignored
- **Token cache and revocation**: iControl tokens reused until expiry and revoked on close (BIG-IP caps active tokens per account)
- **Audit trail on every write**: actor, resource, action, outcome — and the reason where one is given (pause, refusal); no secret is ever logged

■ 9 · POSTURE

- **CBOM per appliance** (CycloneDX 1.7): algorithms, key sizes, issuing CAs, expiry, post-quantum exposure, unreferenced certificates
- **Minimum TLS version per client-ssl profile** and a weak-cipher verdict from explicit cipher strings; profiles using a **cipher-group** are reported as *not evaluable*
- **Signed posture attestation** (ML-DSA-65) produced on the CLM side

■ 10 · PREREQUISITES, VERSIONS, MATURITY

- **Validated on BIG-IP VE 17.5.1.8**; CSR-on-device over REST needs **BIG-IP $\geq$ 14.1**
- Management-plane reachability (HTTPS 443); management certificate verified or pinned (`ca_cert`)
- **iControl SOAP** enabled for least-privilege installs, the CSR read-back and the served-cert SPKI
- **fips / nethsm** custody requires the FIPS / External-HSM module licensed *and* active on the appliance
- **Post-quantum**: PKIFactor is CLM-ready (CBOM, dual-certificate strategy); serving PQC certificates on BIG-IP depends on platform support
- Edition: the F5 BIG-IP deployment engine is included in **all PKIFactor editions**
- Platform features (KMS, SSO, HA, DCV / DNS providers, licensing): **see the platform datasheet**

MATURITY — STATED PLAINLY

Proven live on F5: estate discovery, dry-run reconciliation and foreign-certificate onboarding; key-on-device provisioning with software (*normal*) custody and REST CSR; resource-admin + SOAP install; server-ssl (outbound mTLS) and forward-proxy CA deployments; hitless rotation; partition probe; posture (minimum TLS, weak ciphers, CBOM).

Written and unit-tested, gated on hardware for live validation: HA config-sync and failover across a device group (a single VE and a lab pair so far), on-board FIPS and network-HSM custody (netHSM partition seen from the VE; end-to-end key generation pending).

Roadmap-adjacent: post-quantum *certificate* issuance is generated and orchestrated by PKIFactor with signing delegated to a post-quantum-native CA; serving PQC certificates on F5 depends on platform support.

About ZetaCert

ZetaCert builds sovereign, European certificate-lifecycle infrastructure: PKIFactor, a multi-CA certificate lifecycle management (CLM) orchestrator, and ZetaCA, a post-quantum-native certificate authority. zetacert.com

F5, BIG-IP, Local Traffic Manager, and iControl are trademarks, service marks, or trade names of F5, Inc., in the U.S. and other countries. PKIFactor is a product of ZetaCert.

Document 2026-08-30 · PKIFactor v1.5.144.