EZRMF · WHITEPAPER · ARCHITECTURE

FIPS + Chainguard: Securing IL-5 Cloud Deployments

Reference architecture for running ezRMF in AWS GovCloud (IL-5) with FIPS-validated cryptography, zero-CVE Chainguard images, and STIG-hardened workloads.

SHE BASH LLC · EZRMF 2026 · V4.2 DISTRIBUTION: UNCLASSIFIED

Threat model

> A compliance platform is a high-value target. It holds the inventory map, the policy gaps, and the in-flight POA&M for the systems it tracks. The deployment must assume an adversary that already operates inside the perimeter and is looking for the cheapest pivot to a higher-value system.

Concretely, ezRMF defends against:

- **Container escape from a malicious upload.** Untrusted documents and AI agent outputs run only inside a network-restricted bubblewrap sandbox.
- **Credential theft from the running pod.** Static credentials are eliminated; IRSA / OIDC supply scoped role assumptions.
- **Cryptographic substitution.** All TLS terminates against FIPS 140-2/3 validated modules. Algorithm negotiation is constrained at the policy layer.
- **Supply-chain attacks via base images.** Chainguard images carry a minimal package set with continuous CVE backfill. Builds fail-closed on known CVE > High.

Reference architecture

Compute

- EKS in AWS GovCloud (us-gov-west-1), private subnets only.
- Two deployments: `agenticrmf` (frontend + API + agent runtime) and `postgres` (PostgreSQL 16 with pgvector).
- Pod security context: `runAsNonRoot`, `runAsUser: 65532`, all capabilities dropped, `allowPrivilegeEscalation: false`, read-only root filesystem except `/tmp/agenticrmf-sessions` (tmpfs).

Storage

- MinIO (S3-compatible) for object storage of project artifacts, evidence chunks, and conversation transcripts.

- Three buckets: `agenticrmf-private` (project + user data), `agenticrmf-shared` (cross-tenant artifacts), `agenticrmf-public` (signed download surface).
- Server-side encryption with KMS keys; bucket policies enforce TLS-only access.

Network

- Ingress through ALB with TLS 1.2+ (FIPS suites only), client certificates optional for high-side environments.
- Egress restricted to: AWS APIs (via VPC endpoints), an inference endpoint (Bedrock or on-prem), and configured GitHub Enterprise instances for source ingest. Default-deny otherwise.
- Pod-to-pod traffic mediated by Cilium network policies; the agent sandbox has no direct cluster egress.

Identity

- OIDC via the enterprise IdP; SAML supported for legacy environments.
- Role-based access at the project level: PM, ISSM, Engineer, ISSO, SCA, SCAR, AO.
- Agent / MCP tokens are project-scoped, time-bound, and recorded in the audit log on creation.

FIPS posture

SURFACE	MODULE	VALIDATION
TLS termination (ALB / nginx)	AWS-LC / OpenSSL FIPS	FIPS 140-2 Level 1
Node.js runtime	Chainguard node-fips	FIPS-validated OpenSSL provider
PostgreSQL TLS	OpenSSL FIPS	FIPS 140-2
MinIO server	boringssl FIPS	FIPS 140-2
JWT / session crypto	Node FIPS provider (SHA-256, RSA-PSS)	Inherits FIPS provider

> FIPS enforcement is provider-level, not application-level. ezRMF does not implement its own cryptography. If the provider is in FIPS mode, the application is in FIPS mode.

Chainguard zero-CVE base

Every base image used in ezRMF — Node FIPS, PostgreSQL, MinIO, init containers — comes from cgr.dev with the Chainguard zero-CVE policy. Image SBOMs are signed; build pipelines verify signatures before pushing to private ECR.

- **Build-time:** the release pipeline runs Trivy with a fail-open threshold (table output, exit-0) for visibility but tracks any Critical / High in the release notes.
- **Runtime:** the deployed images target zero known vulnerabilities at release; rebuilds happen on every Chainguard upstream advisory.

STIG compliance

ezRMF passes the Application Security and Development (ASD) STIG V6R4 — 286 of 286 rules at the V6R4 release. STIG hardening is encoded in the Dockerfile and in the application:

- Security headers (CSP, HSTS, X-Frame-Options, Referrer-Policy) are applied at the nginx layer.
- Session cookies are HttpOnly, Secure, SameSite=Strict.
- OWASP top-10 mitigations: parameterized queries, output encoding, file upload validation, deserialization guards.
- Audit log captures authentication, authorization, and CUI-relevant data access.

Continuous attestation

The asd-stig skill in the ezRMF tooling automates the assessment loop: scan the source tree, evaluate ASD rules, emit a CKL file. The same skill runs in pre-release to confirm regression coverage.

For a copy of the deployment Terraform reference or to discuss high-side adaptations, contact **info@shebash.io**.