iTmethods · Forge

FORGE HARDENING SHEET

Managed GitLab Self-Managed on Forge. The hardening sheet.

Twelve named controls. Inside your trust boundary. Sovereign-friendly. Reign-ready.

SECTION 2

Why this matters.

Source code is the largest single attack surface in a regulated enterprise. It contains every secret, every architecture decision, every business logic rule, and every coding-assistant prompt the bank has ever issued. The 2024-2026 wave of source-code-platform incidents made one principle plain. The trust boundary matters more than the feature set. Source code that lives outside the customer's authorization boundary is exposed by the vendor's posture, not the customer's. Forge runs GitLab Ultimate or GitLab Self-Managed Premium inside the customer's authorization boundary. GitLab Self-Managed is the default choice for buyers whose regulator or sovereignty requirement is non-negotiable on air-gapped deployment. Forge operates GitLab inside that boundary. Reign governs every AI coding tool that touches the code. Continuous assurance of every read and every write.

SECTION 3

What Forge applies on day one. Twelve named controls.

01 **SAML / SCIM identity boundary, group and sub-group inheritance enforced.**

Group-based access from the IdP into GitLab. SCIM-driven provisioning and de-provisioning. MFA at the IdP. No GitLab-side users outside the SCIM-managed boundary.

02 **IP restrictions at the GitLab Ultimate group level and at the Forge edge.**

Two-layer enforcement. GitLab-side group-level allow-list. Forge-edge IP allow-list at the network boundary.

03 **GitLab Runners isolated per environment.**

Short-lived JWT tokens. No long-lived runner registration tokens. Runner topology configured for environment isolation. Tag-based runner targeting per project.

04 **Secure Code Scanning (SAST, DAST, IaC scanning) enabled by default.**

Dependency Scanning. Container Scanning. License Compliance. Custom rulesets supported.

05 **Push rules and protected branches enforced via policy-as-code.**

Required approvers, required status checks, signed commits, no force push. Policy-as-code stored in a separate governance repository. Drift detection against policy.

06 **Audit Events streamed to the customer's SIEM and to the Reign Audit Ledger (CAVR).**

Every push, merge request, merge, secret access, settings change, integration install. Tamper-evident replication.

01 **GitLab Duo and any third-party AI coding tools governed through the Reign AI Gateway.**
Every coding-agent call routed through the call-layer policy decision point.

02 **Approved-model registry for the coding-agent LLM fleet.**
Versioned. PCCP-aligned change records. Out-of-policy model invocations blocked at the gateway.

03 **Project and group classification labels propagated to Reign for policy enforcement.**
Customer-defined classification taxonomy propagated from project metadata.

04 **Private network egress only.**
No vendor-side telemetry without explicit customer approval. Service-by-service allow-list enforced.

05 **Backup and DR posture documented and audit-grade.**
Snapshot cadence, cross-region replication, RPO and RTO defined and tested.

06 **Quarterly FDE-pod hardening review.**
Continuous-remediation SLA. P0 within seven days, P1 within twenty-one days, P2 within sixty days.

SECTION 4

Where each control maps.

CONTROL	SR 26-2	OSFI E-23	EU AI Act	DORA	FINOS AIGF	ISO 42001	SOC 2	FedRAMP
01 Identity boundary	✓	✓	✓	✓	✓	✓	✓	✓
02 IP allow-list	✓	✓		✓		✓	✓	✓
03 Runner isolation	✓	✓		✓	✓	✓	✓	✓
04 Secure Code Scanning	✓	✓	✓	✓	✓	✓	✓	
05 Push rules / protected branches	✓	✓	✓	✓	✓	✓	✓	✓
06 Audit Events to SIEM+CAVR	✓	✓	✓	✓	✓	✓	✓	✓
07 Reign AI Gateway	✓	✓	✓	✓	✓	✓	✓	
08 Approved-model registry	✓		✓		✓	✓		
09 Project / group classification	✓	✓	✓	✓	✓	✓	✓	✓
10 Private egress	✓	✓		✓	✓	✓	✓	✓
11 Backup / DR	✓	✓		✓		✓	✓	✓
12 Quarterly FDE review	✓	✓	✓	✓			✓	✓

SECTION 5

Deployment options.

Three deployment topologies. Pick based on regulatory posture.

GitLab Self-Managed in Customer Cloud.

Your AWS, Azure, or GCP. Forge operates. Default posture for organizations standardizing on a single hyperscaler.

GitLab Self-Managed in dedicated sovereign on-prem or hybrid topology.

For buyers whose data sovereignty requirement is non-negotiable. Forge operates inside the customer's authorization boundary.

GitLab Self-Managed in fully air-gapped FIPS 140-2 Level 3 deployment.

Defense and classified programs. ITAR, NIST 800-171, CMMC alignment. No outbound calls to gitlab.com.

SECTION 6

The AI coding tool layer.

GitLab Duo named alongside Copilot, Cursor, Claude Code, Cortex. Same call-layer governance through the Reign AI Gateway. Every coding-agent call identity-bound, content-classified, policy-enforced before the LLM sees the code. Tamper-evident audit trail of every coding-agent decision in the Reign Audit Ledger (CAVR). Approved-model registry mapped to the customer's AI governance posture.

SECTION 7

What we run, what you run.

Path B shared-responsibility model, scoped to GitLab.

FORGE AUTOMATED	CUSTOMER AUTHORED	FDE INTERVENTION
Identity boundary, network policy, audit log streaming, AI Gateway policy decision point, CAVR ledger.	Project and group structure, protected branch policy-as-code, classification taxonomy, approved-model registry contents.	Initial hardening, examination snapshots, P0/P1/P2 remediation, quarterly hardening reviews, regulator-facing posture support.

SECTION 8

Reign tier alignment.

Most regulated GitLab on Forge customers are planning to utilize **Reign Assurance** or **Reign Continuous** for AI Governance requirements. The coding-agent governance is delivered by Reign at the call layer. The hardening and operations are delivered by Forge. Same FDE pod runs both.

SECTION 9

Operator credentials and next step.

SOC 2 Type II. AWS Advanced Tier Services Partner. Validated AWS MSP. Twenty-one years operating critical infrastructure for regulated enterprises.

NEXT STEP

Talk to Forge engineering.

itmethods.com/contact?topic=forge →