

Ghaymah Systems - Cloud Infrastructure Security Checklist

Comprehensive 15-Point Security Baseline based on CISA, NIST, CIS, and OWASP

Executive Summary

As a Cloud Security Architect designing the security posture for Ghaymah (ghaymah.systems), this framework is engineered specifically for our architecture as a leading cloud provider. Given our scale and the direct infrastructure access we provide, our threat landscape is complex. This 15-point master policy provides 90%+ coverage against the most critical cloud attack vectors by integrating strict guidelines from CISA, NIST, CIS, and OWASP.

1. Identity & Access Management (IAM)

The control plane is the perimeter of the cloud. If our IAM is compromised, the entire Ghaymah infrastructure falls.

1.1. Enforce Phishing-Resistant MFA & Context-Aware Access

- Rule: Mandate FIDO2/WebAuthn hardware security keys and block SMS-based MFA for administrative and production access.
- Why for Ghaymah: Ghaymah engineers possess the 'keys to the kingdom.' Standard MFA is vulnerable to SIM swapping and fatigue attacks.
- Source: CISA Zero Trust Maturity Model / NIST SP 800-63B

1.2. Implement Just-in-Time (JIT) Privileges and Zero Standing Access

- Rule: Eliminate persistent standing privileges. Access must be temporary, time-bound (e.g., 1-4 hours), and heavily logged.
- Why for Ghaymah: Drastically reduces the blast radius if an engineer's workstation is compromised.
- Source: CISA Cloud Security Technical Reference Architecture

1.3. Centralize Secrets Management & Prohibit Hardcoded Credentials

- Rule: Use a centralized secrets vault with auto-rotation (30-90 days). Never hardcode API keys or credentials in code or containers.

- Why for Ghaymah: Protects our CI/CD pipelines and internal repositories from supply-chain credential leaks.
- Source: NIST SP 800-53 / CIS Controls v8

2. Container & Kubernetes Security

*Ghaymah provides container orchestration and rapid deployments.
We must secure the container lifecycle from build to runtime.*

2.1. Enforce Immutable, Signed Container Images

- Rule: Mandate image scanning in CI/CD (block critical CVEs) and enforce cryptographic signatures (e.g., Cosign) before deployment.
- Why for Ghaymah: Prevents software supply chain attacks and ensures only trusted code runs on our multi-tenant nodes.
- Source: NIST SP 800-190 / CISA Supply Chain Guidelines

2.2. Isolate Workloads (Rootless & Read-Only)

- Rule: Run containers as non-root users, enforce read-only filesystems, and drop unnecessary Linux capabilities.
- Why for Ghaymah: Our primary defense against 'container escape' vulnerabilities, preventing tenants from compromising the underlying host node.
- Source: CIS Kubernetes & Docker Benchmarks

2.3. Continuous Runtime Security & eBPF

- Rule: Deploy eBPF-based runtime monitoring (e.g., Falco/Cilium) to detect anomalous behavior (unexpected shells, outbound connections) in real-time.
- Why for Ghaymah: Pre-deployment scanning misses zero-days. Runtime monitoring catches active exploitation.
- Source: CISA Cloud Security TRA / CIS Benchmarks

3. Network Security & Cloud Edge

3.1. Zero Trust Micro-segmentation

- Rule: Isolate environments (Prod, Staging, Mgmt) in separate VPCs. Enforce strict egress filtering and default-deny network policies.
- Why for Ghaymah: Prevents lateral movement. If a tenant application is breached, the attacker cannot pivot to internal Ghaymah management planes.
- Source: NIST SP 800-207 / CISA Zero Trust Maturity Model

3.2. DDoS Mitigation and WAF at the Edge

- Rule: Route all external traffic through edge protection featuring L3/L4 DDoS mitigation and a WAF configured with OWASP Core Rule Sets.
- Why for Ghaymah: Ensures 99.9% uptime and protects our infrastructure and clients from volumetric and application-layer attacks.
- Source: CISA Shields Up / OWASP Framework

3.3. Enforce mTLS & ZTNA

- Rule: Use mTLS for all inter-service communication. Replace traditional VPNs with Zero Trust Network Access (ZTNA) for administrative access.
- Why for Ghaymah: Secures internal APIs and ensures administrative access is verified continuously at the identity and device level, not just network location.
- Source: DoD Zero Trust Architecture / NIST SP 800-52

4. Data Protection & Governance

4.1. Envelope Encryption with Customer-Managed Keys (CMK)

- Rule: Enforce AES-256 encryption at rest. Provide CMK options so clients control their cryptographic keys.
- Why for Ghaymah: Guarantees data sovereignty and privacy. Even compromised Ghaymah admin accounts cannot read client plaintext data.
- Source: CSA Cloud Controls Matrix / CIS Control 3

4.2. Immutable Backups (WORM) & DR

- Rule: Store critical backups in Write-Once-Read-Many (WORM) storage with multi-region replication to prevent deletion or alteration.
- Why for Ghaymah: Ensures total recoverability in the event of a catastrophic ransomware attack targeting cloud backup systems.
- Source: CISA Ransomware Readiness Guide / NIST SP 800-34

4.3. Automated Data Exposure Guardrails (DSPM)

- Rule: Deploy automated Data Security Posture Management (DSPM) to enforce public access blocks on object storage and scan for exposed sensitive data.
- Why for Ghaymah: Misconfigured public buckets are a leading cause of breaches. We must prevent accidental data exposure programmatically.
- Source: NIST Privacy Framework

5. Application Security (OWASP Top 5 Focus)

5.1. Strict Resource Authorization (Mitigate Broken Access Control - A01)

- Rule: Enforce strict server-side object-level authorization (BOLA) checks on all API requests. Mandate IMDSv2 to prevent SSRF.
- Why for Ghaymah: Prevents tenants from manipulating API requests to access or modify resources belonging to other tenants.
- Source: OWASP Top 10 A01 / A10

5.2. Prevent Injection via Parameterized Execution (A03)

- Rule: Utilize parameterized queries/ORMs exclusively. Strictly validate and sanitize all inputs to the Ghaymah CLI and APIs before execution.
- Why for Ghaymah: Protects the orchestration backend from Remote Code Execution (RCE) via command or SQL injection.
- Source: OWASP Top 10 A03

5.3. Centralized Logging & SIEM Integration (A09)

- Rule: Centralize all API, CloudTrail, and Kubernetes audit logs into an immutable SIEM/SOAR platform with automated alerting for anomalies.
- Why for Ghaymah: Reduces Mean Time To Detect (MTTD) breaches and ensures we have forensically sound data for incident response.
- Source: OWASP Top 10 A09 / CISA Logging Playbook