CloudForge Architecture Overview

Document ID: PRD-CF-002 **Last Updated:** 2024-02-15 **Owner:** CloudForge Engineering **Classification:** Internal

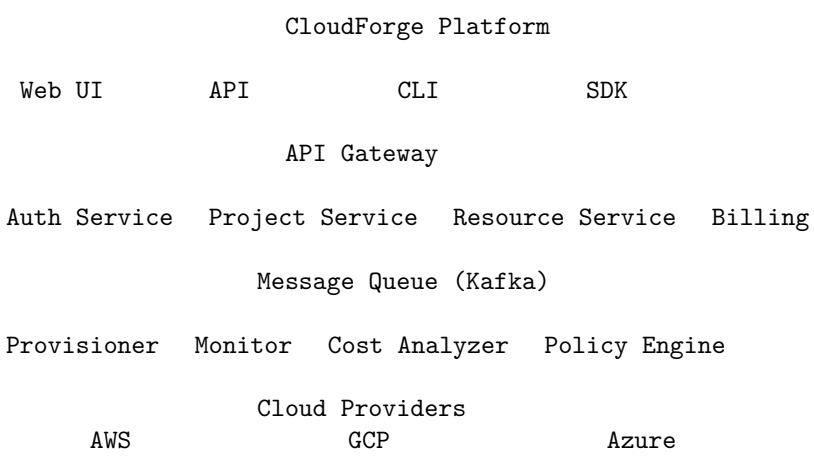
---

Introduction

CloudForge is NovaTech's flagship cloud infrastructure management platform. This document provides an architectural overview for internal teams, partners, and customers seeking to understand the system's design.

System Architecture

High-Level Components



Core Services

API Gateway

- Entry point for all client requests

- Rate limiting and throttling
- Request routing and load balancing
- API versioning support (v1, v2)
- Technology: Kong + custom plugins

Authentication Service

- User authentication via Okta integration
- API key management
- Service-to-service authentication (mTLS)
- RBAC policy enforcement
- Technology: Go, PostgreSQL

Project Service

- Multi-tenant project management
- Organization hierarchy
- Team membership and permissions
- Audit logging
- Technology: Go, PostgreSQL

Resource Service

- Cloud resource lifecycle management
- State tracking and reconciliation
- Dependency resolution
- Drift detection
- Technology: Go, PostgreSQL, Redis

Provisioner Service

- Infrastructure provisioning engine
- Terraform execution runtime
- Multi-cloud provider adapters
- Parallel resource creation
- Technology: Go, Terraform

Monitor Service

- Real-time resource monitoring
- Health checks and alerting
- Metrics collection and aggregation

- Integration with external monitoring
- Technology: Go, InfluxDB, Prometheus

Cost Analyzer Service

- Cloud cost aggregation
- Budget tracking and alerts
- Cost allocation by project/team
- Optimization recommendations
- Technology: Python, PostgreSQL, BigQuery

Policy Engine

- Infrastructure policy enforcement
- Compliance rule evaluation
- Pre-deployment validation
- Real-time policy monitoring
- Technology: Go, Open Policy Agent

Data Architecture

Primary Database (PostgreSQL)

- User and organization data
- Project configurations
- Resource state and metadata
- Audit logs

Cluster Configuration: - Primary + 2 read replicas - Automated failover - Point-in-time recovery (30 days) - Cross-region replication

Cache Layer (Redis)

- Session data
- API response caching
- Rate limiting counters
- Resource state cache

Cluster Configuration: - 6-node cluster with sentinel - Automatic failover - Cache invalidation on writes

Time-Series Database (InfluxDB)

- Resource metrics
- Performance data
- Cost time-series data

Message Queue (Kafka)

- Async task processing
- Event streaming
- Service communication
- Audit event stream

Topics: - `resource.events` - Resource lifecycle events - `provisioner.tasks` - Provisioning jobs - `monitor.metrics` - Monitoring data - `audit.logs` - Audit trail

---

Infrastructure

Cloud Provider: AWS (Primary)

- Region: us-west-2 (primary), us-east-1 (DR)
- EKS for container orchestration
- RDS for PostgreSQL
- ElastiCache for Redis
- MSK for Kafka

Kubernetes Architecture

- Production cluster: 50+ nodes
- Node pools by workload type
- Horizontal Pod Autoscaler
- Pod Disruption Budgets

CDN and Edge

- CloudFront for static assets
- Global edge for API acceleration
- DDoS protection via AWS Shield

Security Architecture

Network Security

- VPC isolation per environment
- Private subnets for all services
- WAF for API protection
- Network policies (Calico)

Data Security

- Encryption at rest (AES-256)
- Encryption in transit (TLS 1.3)
- Customer-managed keys (BYOK) option
- Data residency controls

Access Control

- RBAC with custom roles
- Organization-level policies
- Project-level permissions
- Audit logging for all access

Scalability

Auto-Scaling Policies

Service	Min	Max	Scale Metric
API Gateway	5	50	Request rate
Resource Service	3	30	CPU
Provisioner	10	100	Queue depth
Monitor	5	25	Memory

Performance Targets

Metric	Target	Current
API Latency (p99)	<200ms	150ms

Metric	Target	Current
Provisioning Time	<5min	3.2min
UI Load Time	<2s	1.5s
Availability	99.9%	99.95%

Disaster Recovery

RTO/RPO

- RTO (Recovery Time Objective): 4 hours
- RPO (Recovery Point Objective): 1 hour

Backup Strategy

- Database: Continuous replication + daily snapshots
- Configuration: Version controlled (Git)
- Secrets: Vault with HA backend

Failover Process

1. Automated health checks detect failure
2. DNS failover to DR region (Route 53)
3. DR cluster promoted to primary
4. Data sync verification
5. Traffic shift complete

Integration Points

Inbound Integrations

System	Protocol	Purpose
Customer Apps	REST API	Resource management
CI/CD Systems	REST API / Webhooks	Automated deployments
Monitoring Tools	Prometheus	Metrics export
SIEM	Syslog / API	Security logs

Outbound Integrations

System	Protocol	Purpose
AWS	AWS SDK	Cloud provisioning
GCP	Google Cloud SDK	Cloud provisioning
Azure	Azure SDK	Cloud provisioning
Slack	Webhooks	Notifications
PagerDuty	API	Incident alerts

Future Architecture

Planned Improvements

- Multi-region active-active (Q3 2024)
- GraphQL API support (Q4 2024)
- Edge computing integration (2025)
- AI-powered optimization engine (2025)

Related Documents: Getting Started Guide (PRD-CF-001), API Reference (PRD-CF-010), Security Whitepaper (PRD-CF-020)