CLOUD PLATFORM ENGINEERING

Cloud Infrastructure Architecture Design

Asteria Commerce Platform

Production Reference Architecture • AWS • Kubernetes

Document ID	ARC-CLD-001
Version	1.0
Status	APPROVED BASELINE
Owner	Cloud Platform Engineering
Classification	Confidential — Internal Use
Effective date	02 August 2026
Next review	02 February 2027

This document defines the deployable baseline for the production platform. Deviations require an approved architecture decision record.

Document Control

Field	Value	Field	Value
Sponsor	VP Engineering	Architecture owner	Principal Cloud Architect
Service owner	Director, Platform Engineering	Security owner	Director, Security Engineering
System criticality	Tier 1 / customer-facing	Data classification	Confidential
Review cadence	Six months or material change	Repository	`platform/architecture`

Revision history

Ver.	Date	Change	Author
0.1	10 Jun 2026	Initial architecture and requirement capture	Cloud Architecture
0.7	08 Jul 2026	Security, resilience, and cost review incorporated	Design Working Group
0.9	24 Jul 2026	Production-readiness review candidate	Platform Engineering
1.0	02 Aug 2026	Approved implementation baseline	Architecture Review Board

Approvals

Role	Accountability	Decision	Date
Principal Cloud Architect	Technical integrity	APPROVED	30 Jul 2026
Director, Security Engineering	Security and compliance	APPROVED	31 Jul 2026
Director, Platform Engineering	Operability and delivery	APPROVED	01 Aug 2026
VP Engineering	Funding and risk acceptance	APPROVED	02 Aug 2026

Printed or exported copies are uncontrolled. The repository version is authoritative. Secrets, credentials, production endpoints, customer identifiers, and complete account numbers are intentionally excluded.

Contents

1 Executive Summary

This design establishes a secure, highly available AWS foundation for Asteria, a business-to-business commerce platform. The target supports public APIs, a web application, asynchronous order processing, and analytics while keeping operational ownership practical for a platform team of eight engineers.

The architecture uses separate AWS accounts for production, non-production, security, logging, and shared services. Production workloads run on Amazon EKS across three Availability Zones. Amazon CloudFront, AWS WAF, and an Application Load Balancer provide the public edge. Amazon Aurora PostgreSQL is the system of record; ElastiCache for Redis supports ephemeral caching; S3 stores objects and immutable audit exports; and managed queues decouple background work. Infrastructure and policy are delivered through reviewed, immutable pipelines.

Architecture decision

Adopt a multi-account, single-region / multi-AZ production architecture with warm recovery in a second region. Prefer managed services for undifferentiated operations, private networking by default, short-lived identity, encrypted data paths, and observable failure domains.

1.1 Outcome targets

Quality	Target	Evidence
Availability	99.95% monthly for public API and checkout	SLO dashboard and synthetic probes
Performance	API p95 < 250 ms and p99 < 600 ms at 2,500 requests/s	Distributed load test
Recovery	Tier 1 RTO ≤ 60 min; RPO ≤ 5 min	Quarterly recovery exercise
Security	No public workloads or databases; all privileged access is just-in-time	Config rules and access reports
Delivery	Routine production release in < 20 min; rollback in < 10 min	Pipeline telemetry
Cost	Monthly variance within 10% of forecast	Tagged cost allocation report

1.2 Key decisions and trade-offs

Decision	Why	Accepted trade-off
EKS for services	Consistent scheduling, policy, autoscaling, and deployment patterns	Kubernetes operating complexity
Aurora PostgreSQL	Transactional integrity, mature tooling, managed failover	Higher baseline cost than a single instance
Multi-AZ primary	Contains common infrastructure failures without operator action	Does not cover regional failure
Warm regional recovery	Meets one-hour RTO at sustainable cost	Capacity scale-up is required during failover
Private egress	Restricts network paths and centralizes inspection	NAT and endpoint charges

2 Scope and Context

2.1 Business context

Asteria processes merchant catalogs, quotes, orders, and payment-provider tokens for customers in North America and Europe. It has a predictable daily baseline with campaign-driven bursts up to five times normal traffic. The platform must deploy frequently without compromising transaction integrity or auditability.

2.2 In scope

- AWS organization, accounts, identity federation, guardrails, and centralized audit logging.
- DNS, edge protection, regional networking, compute, data services, and service-to-service connectivity.
- Continuous delivery, infrastructure as code, secrets, observability, backup, recovery, and cost controls.
- Production and structurally equivalent non-production environments.

2.3 Out of scope

Application-domain design, payment-provider implementation, end-user device security, corporate office networking, detailed data-retention schedules, and business continuity outside the technology platform are governed by separate documents.

2.4 Assumptions and constraints

ID	Statement	Design effect
A-01	Stateless services can run concurrently in multiple Availability Zones.	Horizontal scaling
A-02	Card data is tokenized by the payment provider and is not persisted.	Reduced sensitive-data scope
A-03	The primary region is `us-east-1`; recovery is `us-west-2`.	Regional topology
C-01	Production cannot depend on inbound SSH, shared users, or static cloud keys.	Federated, audited access
C-02	New managed-service classes require architecture and security review.	Controlled technology growth
C-03	Recovery-region recurring cost should remain below 35% of primary cost.	Warm, not active-active, recovery

2.5 Architecture principles

1. **Design for failure.** Components are replaceable; recovery paths are automated and exercised.
2. **Minimize trust.** Authenticate every workload, authorize the narrow action, and record the decision.
3. **Keep data authoritative.** A single system owns each mutable record; integration is event-driven where practical.
4. **Automate repeatable change.** Consoles are diagnostic tools, not a deployment mechanism.
5. **Expose operational truth.** Every service publishes health, golden signals, ownership, and runbook links.
6. **Cost is a design dimension.** Capacity, retention, and service tiers have explicit value hypotheses.

3 Requirements

3.1 Functional infrastructure requirements

ID	Requirement	Owner	Priority
FR-01	Route public HTTPS traffic to healthy application instances with bot and application-layer protection.	Platform	Must
FR-02	Schedule independently deployable web, API, and worker containers.	Platform	Must
FR-03	Provide transactional, cache, object, queue, and event persistence.	Data	Must
FR-04	Support private connectivity to approved external partners and SaaS endpoints.	Network	Should
FR-05	Centralize logs, metrics, traces, audit events, alerts, and service ownership.	SRE	Must
FR-06	Recreate environments and recover data using versioned automation.	Platform	Must

3.2 Non-functional requirements

ID	Area	Acceptance threshold	Validation
NFR-01	Availability	99.95% monthly service SLO	Error-budget report
NFR-02	Latency	p95 < 250 ms at nominal load	Load test and APM
NFR-03	Elasticity	5x API load in ≤ 10 min	Scheduled stress test
NFR-04	Recovery	RTO ≤ 60 min, RPO ≤ 5 min for Tier 1	Regional game day
NFR-05	Confidentiality	TLS 1.2+ in transit; approved KMS keys at rest	Automated configuration tests
NFR-06	Traceability	100% of control-plane events retained centrally	Audit-log reconciliation
NFR-07	Change safety	Progressive delivery; automated rollback signal	Release simulation
NFR-08	Maintainability	Every production service has owner, SLO, dashboard, and runbook	Service-catalog check

4 Target Architecture

4.1 System context

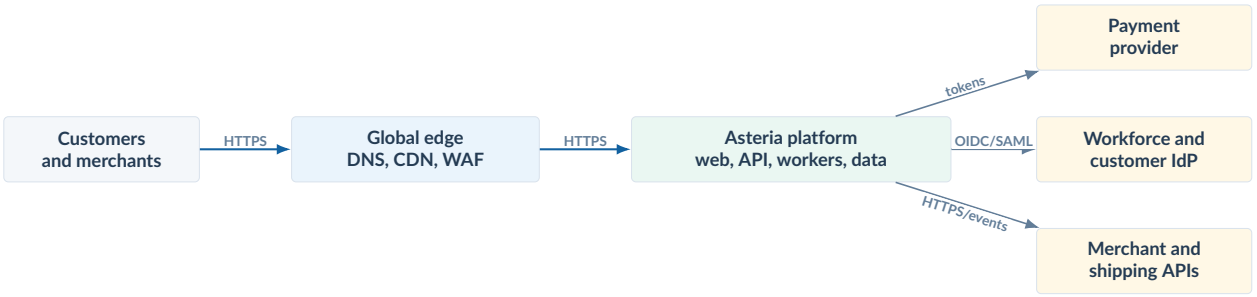

Figure 1: System context and external trust boundaries.

4.2 Logical component architecture

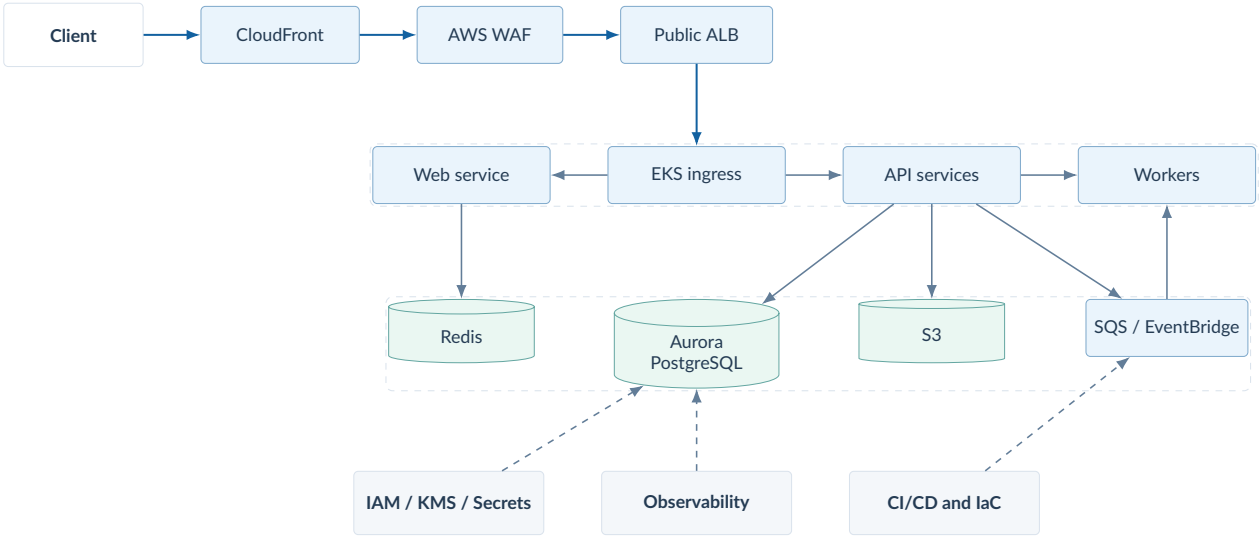

Figure 2: Logical service architecture. Solid arrows carry application traffic; dashed arrows represent management or telemetry relationships.

4.3 AWS account topology

AWS Organizations applies service control policies and consolidated billing. Workloads cannot create unapproved regions, disable security services, leave the organization, or make protected log stores public.

Organizational unit	Accounts	Purpose
Security	Security Tooling, Log Archive	Delegated security administration and immutable evidence
Infrastructure	Network, Shared Services	Transit, endpoints, DNS, image and package services
Workloads / Prod	Production	Customer workloads and production data
Workloads / Non-prod	Development, Test, Staging	Isolated delivery stages with production-equivalent controls
Sandbox	Engineering sandboxes	Time-limited experimentation with restricted spend and data

Root users have hardware-backed MFA and no access keys. Workforce access is federated through IAM Identity Center. Organization CloudTrail, AWS Config, GuardDuty, Security Hub, and access findings are delegated to the security account; workload administrators cannot alter the evidence trail.

5 Network Architecture

5.1 Production VPC topology

The primary production VPC uses a non-overlapping `10.20.0.0/16` block across three Availability Zones. Subnets are separated by function so routes, network ACLs, capacity, and ownership remain clear. There are no public IP addresses on application nodes or data services.

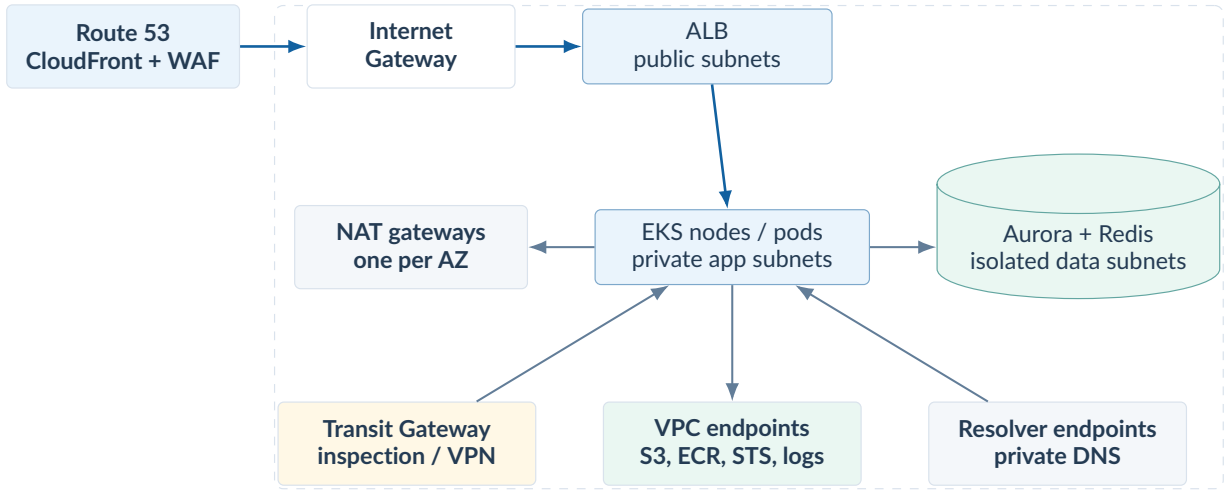

Figure 3: Production network zones and principal traffic paths.

5.2 Subnet and address plan

Tier	CIDRs by AZ	Contents	Route
Public edge	`10.20.0.0/24–2.0/24`	ALB, NAT gateways	IGW
Private application	`10.20.16.0/20–48.0/20`	EKS nodes and pods	NAT / endpoints
Isolated data	`10.20.64.0/21–80.0/21`	Aurora, Redis	Local only
Private endpoints	`10.20.96.0/24–98.0/24`	Interface endpoints, resolvers	Local / TGW
Reserved	`10.20.112.0/20` and above	Growth and migration	None

5.3 Traffic policy

Security groups are the primary stateful control. Network policies enforce namespace and workload boundaries inside EKS. Network ACLs remain coarse safeguards, not an application authorization layer.

Source	Destination	Port	Control
CloudFront origin	Public ALB	TCP 443	Managed prefix list and secret origin header
ALB	Ingress pods	TCP 8443	Dedicated security groups
Application pods	Aurora proxy	TCP 5432	Workload SG and namespace policy
Application pods	Redis	TCP 6379	TLS, workload SG, auth token
Application pods	AWS APIs	TCP 443	VPC endpoint policy
Application pods	Internet allowlist	TCP 443	Egress proxy / NAT and DNS policy
Administrators	Workloads	None inbound	SSM or Kubernetes API via approved access path

6 Compute and Application Platform

6.1 EKS baseline

The cluster control plane exposes a private endpoint and a restricted public endpoint for approved CI runners only. Managed node groups span three Availability Zones. System add-ons run on a small on-demand group; application services use mixed on-demand and Spot capacity. Critical services maintain enough on-demand replicas to survive loss of any one zone.

Concern	Baseline	Guardrail
Scheduling	Topology spread across zone and host; anti-affinity for critical replicas	Admission policy rejects single-replica Tier 1 workloads
Capacity	Cluster Autoscaler / Karpenter with approved instance families	Defined CPU, memory, and ephemeral-storage requests
Pod identity	EKS workload identity mapped to one IAM role per service	No node-role credential use by applications
Images	Private ECR, digest-pinned, signed, scanned before promotion	Critical vulnerabilities block release
Runtime	Read-only root filesystem, non-root UID, dropped capabilities	Policy exceptions expire automatically
Availability	PDBs, readiness/startup probes, graceful termination	Planned disruption respects error budget

6.2 Workload deployment pattern

Each service is a versioned Helm release promoted across environments. A deployment contains resource requests and limits, health probes, a service account, pod disruption budget, network policy, autoscaling rules, telemetry annotations, and ownership metadata. Configuration is externalized; secrets are mounted at runtime from Secrets Manager and never stored in Git or container images.

6.3 Scaling model

API pods scale on CPU and requests per second, workers on queue age and depth, and node capacity on pending pods. Scale-out begins at 60% target utilization, leaving headroom for a zone failure. Scheduled scaling precedes known campaigns. Scale-in uses a ten-minute stabilization window to avoid oscillation.

Capacity invariant

At steady state, the remaining two Availability Zones must absorb the loss of one zone without exceeding 75% allocatable CPU or memory. The invariant is tested monthly and before every major campaign.

7 Data and Integration Architecture

7.1 Data services

Data class	Service	Configuration	Recovery
Transactions	Aurora PostgreSQL	Multi-AZ writer, two readers, RDS Proxy, encrypted storage	PITR + cross-region replica
Ephemeral cache	ElastiCache Redis	Multi-AZ replication group, TLS, no authoritative records	Rebuild from source
Objects	Amazon S3	Versioning, SSE-KMS, lifecycle tiers, block public access	Cross-region replication
Work queues	Amazon SQS	Standard queues, DLQ, encryption, age alarms	Replay from source / DLQ
Domain events	EventBridge	Versioned schemas, archive for critical buses	Archive replay
Search	OpenSearch Serverless	Private collection, encrypted indexes	Reindex from S3 / database

7.2 Database design controls

Applications connect through RDS Proxy using short-lived credentials. Schema changes use backward-compatible expand/migrate/contract steps. Automated backups retain 35 days; monthly recovery points are copied to the recovery region and protected by logically isolated backup vault controls. Production exports are encrypted, access-logged, time-limited, and prohibited in lower environments unless irreversibly masked.

7.3 Event delivery contract

Producers use the transactional outbox pattern where a database mutation and event publication must agree. Consumers are idempotent and record the event identifier. Messages exceeding retry policy move to a dead-letter queue and page only when age or business impact crosses the service threshold.

SQS and EventBridge provide at-least-once delivery in this design. Consumers must tolerate duplicates, late arrival, and out-of-order events. Exactly-once business outcomes are achieved with idempotency keys and database constraints, not transport assumptions.

7.4 Data lifecycle

Data owners assign classification, residency, retention, and deletion rules in the data catalog. S3 lifecycle policies transition eligible objects to lower-cost tiers and expire them when retention completes. Database deletion requests propagate through authoritative tables, derived indexes, caches, exports, and backup-expiry procedures with auditable completion.

8 Security Architecture

8.1 Identity and access

Humans authenticate to the corporate identity provider with phishing-resistant MFA and assume permission-set roles through IAM Identity Center. Elevated roles require ticket linkage and time-bound approval. Workloads receive service-specific IAM roles through EKS workload identity. CI uses OIDC federation; no long-lived AWS keys are stored in the CI platform.

Principal	Access model	Maximum session
Developer	Read non-sensitive telemetry; deploy through pipeline	8 hours
Production operator	Just-in-time operational role with ticket	1 hour
Security analyst	Organization-wide read and investigation tooling	4 hours
CI pipeline	OIDC trust restricted by repository, branch, and environment	1 hour
Application pod	Namespace-bound service account to narrow IAM role	1 hour, rotated
Break-glass operator	Dual approval, isolated credential, full session capture	30 minutes

8.2 Encryption and key management

TLS 1.2 or later protects external and internal application traffic. AWS Certificate Manager manages public certificates. KMS customer-managed keys are separated by data class and environment, use annual rotation where supported, and deny direct administrative decrypt operations. Key policies grant services through conditions such as account, VPC endpoint, encryption context, and resource ARN.

8.3 Secrets management

Secrets Manager stores database, partner, and signing credentials. Rotation is automated where the target supports it; otherwise the owner receives a rotation task before expiry. Applications fetch secrets at runtime through workload identity. Secret values are redacted from logs, traces, crash dumps, command histories, and deployment outputs.

8.4 Detection and response

CloudTrail, Config, VPC Flow Logs, WAF logs, EKS audit logs, GuardDuty, Inspector, ECR scanning, Security Hub, IAM Access Analyzer, and application security events feed the centralized security account and SIEM. High-confidence findings invoke an on-call route and containment runbook; evidence is written to a retention-locked log bucket.

No internet-addressable database or node; no wildcard action on wildcard resource in workload roles; no plaintext secret in code or pipeline variables; no unscanned image in production; no mutable audit-log retention; and no production change without a traceable identity and reviewed artifact.

8.5 Threat summary

Threat	Prevent / limit	Detect / recover
Account takeover	MFA, conditional access, short sessions, least privilege	Identity alerts, session revocation, credential rotation
Web exploitation	WAF, input validation, patched images, rate limits	WAF/APM correlation, isolate workload, deploy fix
Supply-chain compromise	Pinned dependencies, signed artifacts, protected branches	Provenance checks, image quarantine, known-good rollback
Data exfiltration	Private paths, endpoint policies, egress allowlist, KMS	Flow/data events, revoke access, preserve evidence
Destructive operator	Separation of duties, protected backups, approval gates	CloudTrail alert, account containment, restore
Denial of service	CDN, Shield Standard, WAF rate rules, autoscaling	Edge telemetry, load shedding, provider escalation

9 Reliability and Disaster Recovery

9.1 Failure-domain design

The normal operating mode tolerates loss of a pod, node, or one Availability Zone without human intervention. Health checks remove failed endpoints; Kubernetes reschedules workloads; Aurora promotes a healthy replica; and queues buffer transient downstream loss. Timeouts, bounded retries with jitter, circuit breakers, concurrency limits, and load shedding prevent cascading failure.

9.2 Recovery tiers

Tier	Examples	RTO	RPO	Strategy
1	Checkout, API, orders	≤ 60 min	≤ 5 min	Warm recovery stack; replicated data
2	Catalog admin, integrations	≤ 4 h	≤ 1 h	Pilot light; restore and scale
3	Analytics, internal tools	≤ 24 h	≤ 24 h	Backup and rebuild

9.3 Regional recovery sequence

1. Incident commander declares regional recovery after impact and primary-region safety checks.
2. Freeze releases, preserve evidence, validate last replicated recovery points, and record the recovery timestamp.
3. Apply reviewed Terraform to scale recovery EKS nodes, workers, endpoints, and observability collectors.
4. Promote the Aurora cross-region secondary; validate object replication and the required event-replay position.
5. Run synthetic and transaction-integrity tests through the recovery endpoint.
6. Change Route 53 failover health state and progressively restore traffic (5%, 25%, 50%, 100%).
7. Reconcile asynchronous work, announce service restoration, and start controlled failback planning.

Database promotion and DNS failover are gated by a single incident commander. The primary writer is fenced before recovery writes are enabled. Failback is a planned migration with reconciliation; DNS is never switched merely because the old region appears healthy again.

9.4 Backup controls and exercises

AWS Backup policies enforce schedules, cross-account copies, vault lock, and restore-point monitoring. Backup success does not prove recoverability: a sample Aurora cluster and S3 object set are restored monthly; a service recovery exercise runs quarterly; and a full regional exercise runs twice each year. Results, actual RTO/RPO, data exceptions, and corrective actions are retained.

10 Observability and Operations

10.1 Telemetry model

Applications emit structured logs, Prometheus-compatible metrics, and OpenTelemetry traces with consistent service, environment, version, region, zone, and correlation fields. Collection agents buffer and forward telemetry to the managed observability platform. Sensitive fields are filtered before export.

Signal	Required coverage	Primary use
Metrics	Rate, errors, duration, saturation; business success counters	Alerting, SLOs, capacity
Logs	Structured event, request ID, severity, safe context	Diagnosis and audit
Traces	Edge-to-database spans with sampled errors and slow paths	Dependency and latency analysis
Synthetics	Public journey and private dependency probes	User-visible availability
Events	Deployments, scaling, configuration, and incident annotations	Change correlation

10.2 SLO and alert policy

Availability and latency SLIs are measured at the edge from eligible requests. Fast-burn and slow-burn error-budget alerts page only for actionable user impact. Resource alerts create tickets unless immediate exhaustion threatens the SLO. Every page identifies the service, symptom, current impact, dashboard, runbook, and escalation owner.

10.3 Operational ownership

The platform team owns cloud foundations, clusters, base images, and shared observability. Product teams own service code, dependency behavior, SLOs, runbooks, and first-line response. Security Engineering owns detection policy and leads security incidents. Major incidents use a named incident commander, operations lead, communications lead, and scribe.

Activity	Platform	Product	Security	SRE
EKS lifecycle	A/R	C	C	C
Service deployment	C	A/R	C	C
SLO and alerting	C	A	C	R
Vulnerability response	R	R	A	C
Regional recovery	A/R	R	C	R

11 Delivery and Infrastructure as Code

11.1 Source-to-production flow

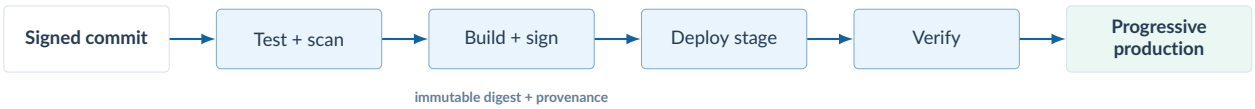

Figure 4: Artifact promotion flow. The same signed artifact advances through every stage.

11.2 Pipeline controls

Pull requests require code ownership, passing tests, policy checks, secret scanning, dependency scanning, and an approved plan for infrastructure changes. Production promotion requires environment approval and is blocked during an active major incident. The pipeline assumes a narrow AWS role through OIDC, records artifact provenance, and writes deployment events to the audit trail.

Deployments use canary steps of 5%, 25%, 50%, and 100%, advancing only while SLO and application checks remain healthy. Automated rollback returns traffic to the prior signed image. Database migrations are forward-compatible and are not treated as reversible container deployments.

11.3 Infrastructure repositories and state

Terraform modules define organization, account, network, cluster, data, security, and observability layers. State is encrypted in a dedicated administrative account with locking, versioning, and access logging. Each environment has independent state and roles. Drift detection runs daily; critical drift alerts the platform team and cannot be silently adopted.

12 Cost and Sustainability

Costs are allocated by account, environment, service, owner, and cost center. Mandatory tags are enforced where the service supports them. Budgets alert at forecast thresholds of 80%, 100%, and 120%; anomaly detection routes unusual daily spend to the owning team.

Lever	Policy	Review signal
Compute	Rightsize quarterly; use Spot for interruptible workers; commit stable baseline only	Cost per 1,000 requests
Database	Match replicas and I/O mode to measured load; stop non-prod on schedule	Cost per order / connection
Storage	Lifecycle logs and objects; bound snapshots and trace retention	Growth and retrieval rate
Network	Prefer VPC endpoints where total cost and security justify them; monitor cross-AZ transfer	Cost per GB by path
Recovery	Maintain warm minimums; verify scale-up time rather than overprovisioning	Recovery cost ratio

Workloads publish utilization and unit-cost metrics. Carbon and cost generally align through higher utilization, appropriate data retention, and fewer unnecessary transfers; resilience capacity is retained where it protects a defined reliability target.

13 Risks, Decisions, and Open Items

13.1 Risk register

ID	Risk	Likelihood	Impact	Treatment / owner
R-01	EKS expertise concentrated in two engineers	Medium	High	Training, pairing, managed runbooks / Platform Director
R-02	Cross-region database lag breaches RPO during burst	Low	High	Lag alarm, capacity test, snapshot fallback / Data Lead
R-03	Third-party API saturation cascades into checkout	Medium	High	Bulkhead, circuit breaker, queue, load shed / Product Lead
R-04	NAT and cross-AZ transfer exceed forecast	Medium	Medium	Path-level cost dashboard, endpoint review / FinOps
R-05	Recovery automation drifts from primary	Medium	High	Scheduled plan, parity test, semiannual game day / SRE Lead

13.2 Architecture decision records

ADR	Decision	Status
ADR-001	Use AWS Organizations with dedicated security and log-archive accounts	Accepted
ADR-002	Standardize container services on EKS	Accepted
ADR-003	Use Aurora PostgreSQL as transactional system of record	Accepted
ADR-004	Adopt warm cross-region recovery for Tier 1	Accepted
ADR-005	Evaluate service mesh only when measured requirements justify it	Deferred

13.3 Open items

ID	Decision needed	Owner	Due
O-01	Confirm European data residency boundary before EU launch	Data Protection Officer	30 Sep 2026
O-02	Select managed observability vendor after volume trial	SRE Lead	15 Sep 2026
O-03	Validate payment provider private-connectivity option and cost	Network Architect	31 Aug 2026

14 Implementation and Validation

14.1 Delivery roadmap

Phase	Window	Exit outcome	Gate owner
1. Foundation	Weeks 1–4	Organization, accounts, identity, logs, KMS, baseline policies	Security Architect
2. Network	Weeks 3–7	VPCs, DNS, endpoints, edge, egress, connectivity tests	Network Architect
3. Platform	Weeks 6–11	EKS, registries, secrets, policy, observability, golden path	Platform Director
4. Data	Weeks 9–13	Aurora, Redis, S3, queues, backup and restore evidence	Data Lead
5. Migration	Weeks 12–17	Staging soak, performance test, production canary	Product Director
6. Recovery	Weeks 15–19	Recovery-region deployment and successful game day	SRE Lead

14.2 Production readiness checklist

	Required evidence	Approver
☐	Threat model closed or explicitly accepted; external test has no unresolved critical/high finding	Security Engineering
☐	Load, endurance, zone-loss, dependency-failure, and rollback tests meet targets	SRE
☐	Backup restore and regional recovery demonstrate recorded RTO and RPO	Service Owner
☐	Dashboards, SLOs, alerts, runbooks, escalation, and ownership are registered	Operations Review
☐	Cost forecast, budgets, anomaly alerts, tags, and retention policies are active	FinOps
☐	Data classification, residency, retention, and deletion flows are approved	Data Governance
☐	Architecture drift scan and policy suite pass from a clean deployment	Principal Architect

14.3 Acceptance criteria

The design is accepted for general production use when all Must requirements pass, no unaccepted critical or high risk remains, the production-readiness checklist is signed, a clean environment can be created from version control, and the recovery exercise completes inside Tier 1 objectives. Any exception has an owner, compensating control, expiry date, and recorded risk acceptance.

A Configuration Baseline

Domain	Baseline	Implementation note
Regions	`us-east-1`, `us-west-2`	All other regions denied except explicitly required global services.
Availability Zones	Three per region	Resolve account-specific AZ IDs before assigning subnets.
Kubernetes	Supported EKS release	Upgrade at least one minor version before end of standard support.
Node operating system	Minimal managed image	Immutable replacement; no in-place package management.
Container registry	Private ECR	Tag for humans; deploy by digest; enable scan-on-push.

Domain	Baseline	Implementation note
Database	Aurora PostgreSQL	Multi-AZ, deletion protection, performance telemetry, TLS required.
Object storage	S3	Block public access at organization and account; versioning enabled.
Encryption	Customer-managed KMS keys	Separate production keys; key administrators cannot decrypt data.
Log retention	30–90 days hot; archive per policy	Security evidence retained in a separate protected account.
Time	Amazon Time Sync Service	Alert on host or application clock drift.
DNS	Route 53 public/private zones	Changes through code; query logging for protected zones.
Certificates	AWS Certificate Manager	Automated renewal alarms and ownership metadata.
Patching	Rebuild within risk SLA	Critical internet-facing: 72 hours; critical internal: 7 days.

B Control Traceability

Control objective	Design mechanism	Evidence
Authorized access	Federation, MFA, least privilege, JIT elevation, workload identity	Identity Center and CloudTrail reports
Secure configuration	Terraform modules, SCPs, admission policy, Config rules	Plans, policy results, drift reports
Change traceability	Protected branches, approvals, signed artifacts, deployment events	Git and pipeline audit records
Data protection	TLS, KMS, private networks, Secrets Manager, DLP rules	Configuration snapshots and key logs
Vulnerability management	Dependency/image scanning, immutable rebuild, patch SLA	Scanner and remediation records
Resilience	Multi-AZ, autoscaling, backups, cross-region recovery	Load, restore, and game-day reports
Security monitoring	Central telemetry, managed detections, SIEM, on-call routing	Alert tests and incident records

C Glossary

Term	Meaning
ALB	Application Load Balancer; regional Layer 7 load balancer.
AZ	Availability Zone; an isolated location within an AWS region.
DLQ	Dead-letter queue holding messages that exceeded retry policy.
EKS	Amazon Elastic Kubernetes Service.
IaC	Infrastructure as code; versioned, reviewable infrastructure definitions.
JIT	Just-in-time access granted for a bounded task and duration.
KMS	AWS Key Management Service.
RPO	Recovery point objective; maximum tolerable data-loss interval.
RTO	Recovery time objective; maximum target time to restore service.
SCP	Service control policy applied through AWS Organizations.
SLI / SLO	Service level indicator / service level objective.
TGW	AWS Transit Gateway.

D Review Record

Review trigger	Required participants	Cadence
Scheduled architecture review	Architecture, Platform, Security, SRE, Product	Six months
Material service or region change	Architecture, affected owner, Security	Before change
Tier 1 recovery exercise	SRE, Platform, Data, Product, Communications	Twice yearly
Major incident follow-up	Incident roles and accountable executives	Within 10 business days
Regulatory or policy change	Security, Legal, Data Governance, Architecture	As triggered

End of controlled document • ARC-CLD-001 v1.0