Arquitectura del Sistema de Reservas en la Nube

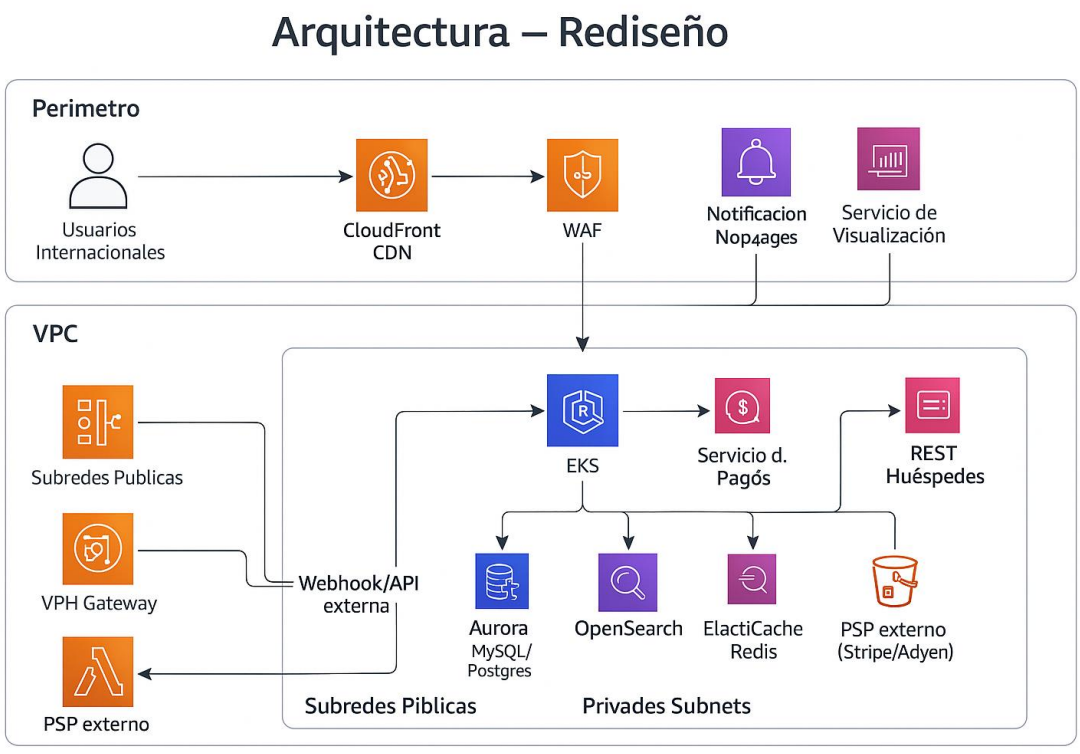
Fecha: 2025-08-19
Propósito: Proveer una visión gerencial de la arquitectura objetivo para migrar el sitio de reservas a la nube con foco en disponibilidad, seguridad, desempeño global y escalabilidad.

Resumen

- **Objetivo:** Migrar el sitio de reservas del hotel a una arquitectura **cloud** global con **alta disponibilidad**, **baja latencia** y **seguridad end-to-end**.
- **Alcance:** Servicios de **Búsqueda**, **Visualización**, **Huéspedes** (favoritos/recientes) y **Pagos** integrados con un **PSP externo** (tokenización/3DS, webhooks firmados).
- **Arquitectura:** Perímetro **CDN + WAF + ALB**, microservicios en **contenedores** (EKS/ECS), datos **especializados por propósito** (OLTP, NoSQL, búsqueda, caché), y **procesos asíncronos** basados en eventos.
- **Beneficios:** Experiencia del huésped más **rápida y segura**, resiliencia a picos, **time-to-market** mejorado, trazabilidad y gobierno.

Riesgos mitigados: Caídas por picos, fraudes de pago, pérdida de datos, tiempos de recuperación altos, exposición de secretos.

Diagrama de arquitectura (alto nivel)



Componentes y responsabilidades

Perímetro
- **CDN (CloudFront):** Entrega global, menor latencia/costo para estáticos e imágenes.
- **WAF:** Reglas OWASP administradas, **rate limiting** y bloqueo de IPs maliciosas.
- **ALB/NLB:** Balanceo L7/L4, health checks, enrutamiento a servicios.

Cómputo
- **EKS/ECS Fargate:** Contenedores para microservicios: **Pagos**, **Visualización**, **Huéspedes**, **Búsqueda**; autoescalado; despliegues.

Datos
- **Aurora (RDS):** OLTP de reservas/ocupación con consistencia transaccional y réplicas de lectura.
- **DynamoDB:** Preferencias/favoritos por usuario con baja latencia (TTL para elementos efímeros, DAX opcional).
- **OpenSearch:** Búsqueda facetada, relevancia, sinónimos.
- **Redis:** Caché de sesiones y resultados calientes para absorber picos y bajar latencia.
- **S3:** Almacenamiento de imágenes/activos; origen de CDN.
- **Secrets Manager / KMS:** Gestión y rotación de credenciales; cifrado de datos en reposo.

Integración/Asíncronos
- **EventBridge / SNS+SQS:** Orquestación por eventos; colas para indexación y jobs.
- **Workers/Lambda:** Indexación a OpenSearch, tareas batch y conciliaciones.

Pagos
- **PSP externo (Stripe/Adyen):** Tokenización y **3DS**; webhooks firmados; **idempotencia** en operaciones; conciliación de estado en OLTP.

Seguridad y cumplimiento

- **Cifrado** en tránsito (TLS) y en reposo (KMS en RDS, S3, DynamoDB, OpenSearch, Redis).
- **IAM** con privilegios mínimos, segmentación con **Security Groups/NACLs** y **aislamiento del servicio de pagos**.
- **WAF** con reglas administradas, protección DDoS; **headers** de seguridad (CSP, HSTS, etc.).
- **Gestión de secretos** con rotación; no exponer credenciales en imágenes ni ConfigMaps.
- **Auditoría/Observabilidad:** logs, métricas y trazas (OpenTelemetry); retención y alertas.
- **PCI-DSS:** delegar almacenamiento de tarjetas al PSP; validación de firmas en webhooks.

Disponibilidad, resiliencia y DR

- **Multi-AZ** para cómputo y datos; auto-healing y health checks.
- **Backups automáticos** con **PITR**; pruebas periódicas de restauración.
- **RPO ≤ 5 min** y **RTO ≤ 30 min** (ajustable por negocio).
- **Multi-región**: activa-pasiva (failover) o activa-activa (latency routing).
- **Patrones de resiliencia**: timeouts, reintentos exponenciales, **circuit breakers**, bulkheads.

Escalabilidad, rendimiento y costos

- **HPA** por CPU/RAM/latencia y **autoscaling por colas** en asíncronos.
- **Cache/CDN** agresivo; paginación/batching en APIs.
- Dimensionamiento inicial **mínimo** + escalado progresivo de RDS/OpenSearch; **lifecycle policies** en S3.
- Optimización de consultas/índices; **spot/Graviton** donde aplique.
- **Ahorro**: apagar/habilitar **no-prod** fuera de horario.
