Caso de Estudio: Startup de Venta de Vinos

Descripción General

"Vinotech" es una startup de e-commerce especializada en la venta online de vinos premium en América Latina. Su plataforma permite a clientes:

- Navegar por un catálogo amplio de productos.
- Realizar pedidos y pagos en línea.
- Consultar recomendaciones personalizadas mediante un motor de sugerencias.
- Recibir notificaciones sobre promociones y entregas.

La plataforma está alojada en AWS y ha crecido rápidamente en los últimos 12 meses. Sin embargo, la empresa enfrenta serios problemas en dos áreas:

1. **Costos elevados** sin una visibilidad clara del consumo por servicio o entorno.
2. **Vulnerabilidades de seguridad** en la exposición pública de sus servicios y gestión de credenciales.

Arquitectura Actual

Servicios Usados

- Frontend: SPA Angular desplegado en un bucket S3 con CloudFront.
- Backend:
    - API REST en contenedores desplegados en Amazon ECS Fargate.
    - Motor de recomendaciones usando instancias EC2 tipo c5.xlarge.
- Base de datos: Amazon RDS (PostgreSQL), una sola AZ.
- Autenticación: manejo manual de usuarios con base de datos local.
- Notificaciones: SNS + correos transaccionales mediante Amazon SES.
- Almacenamiento de imágenes y archivos: S3 (sin políticas restrictivas).
- Monitoreo: CloudWatch limitado solo a errores críticos.

Infraestructura

- Sin tagging por ambiente (dev, staging, prod).
- ECS configurado para escalar con thresholds inadecuados (escala aún con baja carga).
- EC2 ejecuta procesos de recomendación 24/7.
- No se usan herramientas como Cost Explorer o Budgets.
- No hay WAF ni Shield en APIs públicas.
- IAM roles con privilegios excesivos (uso de políticas "AdministratorAccess").

Problemas Identificados

Costos

- EC2 y RDS en modo on-demand, sin instancias reservadas.
- Recursos corriendo sin apagarse en horas no laborales (EC2, ECS).
- Gran parte del tráfico proviene de bots (scraping) al catálogo sin ningún control.
- Sin monitoreo ni alertas de gasto.

Seguridad

- API pública sin WAF ni control de tráfico sospechoso.
- Bucket S3 configurado como público, con acceso sin autenticación.
- Secretos embebidos en el código fuente de contenedores.
- Gestión de usuarios propia, sin MFA ni OAuth.
- IAM mal configurado: roles con privilegios amplios y sin rotación de claves.

Actividad

1. **Identificar vulnerabilidades** en la arquitectura desde la perspectiva de seguridad.
2. **Detectar fuentes de gasto innecesarias** y evaluar su impacto.
3. **Proponer mejoras** que integren prácticas de seguridad desde el diseño.
4. **Recomendar herramientas y servicios de AWS** para lograr visibilidad y control de costos.
5. **Redactar una lista priorizada de acciones** para la startup.

Recursos a Entregar

- Documento con análisis de vulnerabilidades.
- Tabla con componentes costosos y alternativas.
- Diagrama de arquitectura mejorada (puede ser esquemático).