Parte 1: Diseño de la infraestructura de la nube

Arquitectura propuesta para la aplicación web escalable:

1. Instancias de cómputo (EC2):
   * Se utilizarán instancias de Amazon EC2 para alojar la aplicación web. Estas instancias estarán en un grupo de Auto Scaling para asegurar la escalabilidad horizontal, garantizando que el número de instancias se ajuste automáticamente según la demanda.
   * Se configurará un balanceador de carga elástico (ELB) para distribuir el tráfico entrante entre las instancias de EC2, mejorando la disponibilidad y la tolerancia a fallos.
2. Almacenamiento (S3):
   * Amazon S3 se usará para almacenar contenido estático como imágenes, CSS, JavaScript y otros activos. Esto reduce la carga en las instancias de EC2 y mejora el rendimiento general.
   * Los datos en S3 estarán protegidos mediante políticas de bucket que restringen el acceso a los recursos solo a roles específicos de IAM.
3. Configuraciones de red (VPC):
   * Se configurará una Virtual Private Cloud (VPC) personalizada con subredes públicas y privadas.
     + Subredes públicas: Para alojar el ELB, que redirigirá el tráfico a las instancias de EC2.
     + Subredes privadas: Para alojar las instancias de EC2 y otros recursos internos como bases de datos, mejorando la seguridad.
   * Se utilizará un gateway de internet para permitir el acceso público al ELB y una gateway NAT para que las instancias en subredes privadas puedan acceder a internet de manera controlada.

Parte 2: Configuración de IAM

Roles y políticas de IAM diseñados:

1. Rol para desarrolladores:
   * Permisos para interactuar con recursos en el entorno de desarrollo (instancias de EC2, S3, CloudWatch, etc.).
   * Política adherida al principio del mínimo privilegio: los desarrolladores solo pueden modificar y visualizar recursos en un entorno de prueba/desarrollo.
2. Rol para administradores:
   * Permisos completos para gestionar la infraestructura de producción, incluyendo configuración de VPC, escalamiento automático y ELB.
   * Restricciones de acceso a datos sensibles mediante etiquetas y políticas específicas.
3. Rol para servidores de aplicaciones (EC2):
   * Permisos para leer archivos en S3 (contenido estático) y enviar logs a CloudWatch.
   * No se les otorgará acceso directo a internet salvo a través de la gateway NAT.

Política de ejemplo para acceso mínimo a S3:

json

Copiar código

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::nombre-del-bucket/\*"

}

]

}

Parte 3: Estrategia de gestión de recursos

1. Escalamiento automático (Auto Scaling):
   * Configuración de un grupo de Auto Scaling para las instancias EC2 con reglas de ajuste basadas en métricas como el uso de CPU y la cantidad de solicitudes entrantes.
   * Umbrales de ejemplo:
     + Aumentar instancias cuando el uso de CPU supere el 70% durante 5 minutos consecutivos.
     + Reducir instancias cuando el uso de CPU sea menor al 30% durante 10 minutos.
2. Equilibrio de carga (ELB):
   * ELB manejará el tráfico entrante, distribuyéndolo de manera uniforme entre las instancias activas.
   * Habilitará comprobaciones de salud para garantizar que solo se enrute tráfico a instancias saludables.
3. Optimización de costos:
   * Configuración de presupuestos en AWS Budgets para monitorear y alertar sobre costos en tiempo real.
   * Uso de instancias reservadas para cargas estables y Spot Instances para procesos no críticos.
   * Implementación de políticas de apagado de recursos en horarios de baja demanda (p. ej., instancias en ambientes de desarrollo).

Parte 4: Implementación teórica

Flujo de interacción de los componentes:

1. Los usuarios acceden a la aplicación a través del ELB, que distribuye el tráfico a las instancias EC2 activas en el grupo de Auto Scaling.
2. Las instancias EC2 recuperan contenido estático desde S3 y lo sirven a los usuarios finales.
3. Los logs generados por las instancias EC2 se envían a CloudWatch para monitoreo.
4. Las métricas de CloudWatch activan el escalamiento automático según las reglas configuradas, garantizando que la aplicación maneje cambios en la demanda sin intervención manual.
5. AWS Budgets monitorea el uso de recursos para garantizar que los costos no excedan los límites presupuestados.

Parte 5: Discusión y evaluación

1. Elección de servicios y sus interacciones:

* La combinación de EC2, S3 y VPC proporciona una arquitectura segura, escalable y altamente disponible.
* ELB asegura la distribución del tráfico, mientras que Auto Scaling garantiza que la capacidad se ajuste dinámicamente.

2. Seguridad mediante políticas de IAM:

* Las políticas diseñadas minimizan riesgos al restringir los permisos a lo estrictamente necesario.
* La separación de roles para desarrolladores, administradores y servidores asegura que cada usuario o servicio tenga acceso solo a los recursos requeridos.

3. Estrategia de gestión de recursos:

* Auto Scaling y ELB garantizan la escalabilidad y el rendimiento óptimos.
* La integración de AWS Budgets y el uso de instancias reservadas y Spot Instances minimizan costos sin comprometer la disponibilidad.