UCLM Universidad de Castilla~La Mancha

Sistema de alertas CIAS

Memoria Técnica
Universidad Castilla La-Mancha

Erick Mesias Rosero Lesano
Matías Esteban Muñoz Palacios
Johann Ismael Ordóñez Echeverría

Tabla de contenido

Resumen

La arquitectura de CIAS se estructura mediante una separación clara de responsabilidades entre capas de presentación, procesamiento, persistencia y notificaciones. La interfaz de usuario se despliega como una aplicación estática en Amazon S3, mientras que la lógica de negocio se implementa mediante funciones AWS Lambda que procesan solicitudes a través de Amazon API Gateway. La persistencia de datos se realiza en Amazon DynamoDB, configurado con índices secundarios globales que permiten consultas eficientes por severidad y servicio. El procesamiento asíncrono de eventos mediante DynamoDB Streams y una función Lambda dedicada a notificaciones desacopla completamente la lógica de alertas del backend principal, mejorando significativamente el rendimiento y la escalabilidad del sistema. La totalidad de la infraestructura se define mediante código utilizando Terraform, garantizando despliegues reproducibles y consistentes.

El sistema proporciona baja latencia en operaciones de lectura y escritura, escalabilidad automática ante picos de tráfico, y alta disponibilidad mediante servicios gestionados por AWS. El análisis de costes, realizado mediante AWS Pricing Calculator para un escenario de uso académico moderado durante doce meses, estima un coste anual total de 359,28 USD, siendo Amazon DynamoDB el componente de mayor impacto económico. La validación operacional ha confirmado el funcionamiento correcto de todos los flujos principales, incluyendo registro de incidencias, consultas por criterios específicos, actualización de estados y generación automática de alertas. Los resultados demuestran que CIAS constituye una solución viable, escalable y económicamente accesible para la gestión de incidencias en entornos educativos e institucionales, validando la viabilidad de arquitecturas serverless para aplicaciones modernas en la nube.

Palabras Clave

Cloud Computing, AWS, Arquitectura Serverless, Gestión de Incidencias, Lambda, DynamoDB, Microservicios, Orientado a Eventos, Infrastructure as Code, Terraform, API Gateway, Notificaciones Automáticas, SNS, Alta Disponibilidad, Escalabilidad, Aplicaciones Distribuidas

Introducción

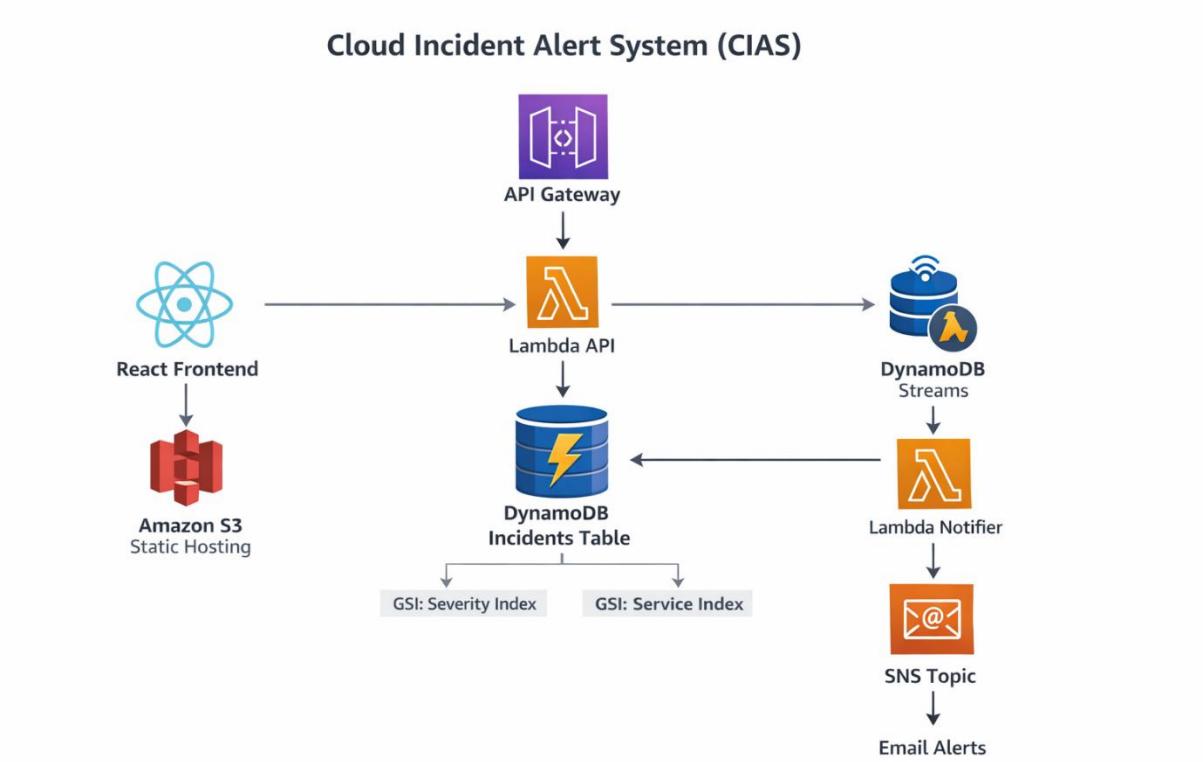
El Cloud Incident Alert System (CIAS) constituye una aplicación distribuida de naturaleza moderna, construida íntegramente sobre la infraestructura de Amazon Web Services. Esta solución ha sido desarrollada con el propósito fundamental de automatizar la gestión de incidencias técnicas y facilitar la notificación inmediata de eventos críticos que puedan afectar los entornos tecnológicos corporativos. El sistema responde a la necesidad contemporánea de contar con herramientas que permitan identificar, documentar y resolver problemas técnicos de manera ágil y coordinada.

La aplicación se ha concebido siguiendo un enfoque arquitectónico denominado cloud-native, el cual aprovecha de manera integral los servicios gestionados y las capacidades sin servidor que ofrece AWS. Esta estrategia de diseño tiene como propósitos principales reducir significativamente la complejidad operativa, permitiendo a los equipos dedicar recursos a aspectos de valor agregado en lugar de tareas administrativas de infraestructura, así como mejorar sustancialmente la escalabilidad del sistema para adaptarse a cambios en la demanda de recursos.

Desde una perspectiva funcional, CIAS proporciona a los usuarios la capacidad de registrar incidencias técnicas de manera sencilla e intuitiva, consultarlas aplicando diversos criterios de búsqueda como el nivel de severidad o el servicio afectado, y actualizar progresivamente el estado de cada incidencia a lo largo de su ciclo de vida completo, desde su identificación inicial hasta su resolución definitiva. Un aspecto particularmente relevante del sistema es su capacidad para generar y distribuir alertas automáticas mediante correo electrónico dirigidas a los responsables técnicos cuando se detectan incidencias de carácter crítico, posibilitando una respuesta organizacional rápida y coordinada ante fallos que pudieran impactar significativamente en los servicios.

La arquitectura de CIAS se despliega completamente en el entorno de nube pública, utilizando una estructura desacoplada y basada en eventos, integrando de manera coordinada múltiples servicios especializados de AWS que incluyen API Gateway para la exposición de interfaces, Lambda para la ejecución de código sin servidor, DynamoDB para la persistencia de datos, SNS para la distribución de notificaciones y S3 para el alojamiento de contenido estático. Un elemento distintivo del proyecto es que la totalidad de la infraestructura se define mediante código, específicamente a través de Terraform, lo que garantiza que el despliegue sea reproducible en múltiples ocasiones, completamente trazable para propósitos de auditoría, y fácilmente transferible entre diferentes cuentas de AWS, incluyendo entornos académicos con restricciones específicas.

Arquitectura



La arquitectura del sistema CIAS ha sido diseñada atendiendo a principios establecidos de separación de responsabilidades, asignando funciones claramente diferenciadas a cada componente del sistema. Esta estructura se organiza en capas distintas que incluyen la presentación al usuario, el procesamiento de lógica de negocio, la persistencia de datos y la generación de notificaciones. Tal configuración responde a patrones de diseño contemporáneos que buscan maximizar la mantenibilidad, la escalabilidad y la capacidad de evolución del software a lo largo de su ciclo de vida.

Interfaz de Usuario

La capa de presentación de CIAS ha sido desarrollada utilizando React, un marco de trabajo moderno para la construcción de interfaces de usuario interactivas, combinado con Vite, una herramienta de construcción que optimiza significativamente los tiempos de compilación y carga. Esta interfaz se despliega como un sitio web estático alojado en Amazon S3, que ofrece características de alta disponibilidad, escalabilidad automática y costos operacionales reducidos. La comunicación entre la interfaz de usuario y los servicios de backend se establece mediante peticiones HTTP estándar, permitiendo una separación clara entre la capa de presentación y la lógica de negocio. El acceso a estos servicios se controla de manera granular mediante políticas de intercambio de recursos entre orígenes (CORS) que se configuran en API Gateway, asegurando que solo las solicitudes provenientes de orígenes autorizados puedan acceder a los recursos backend.

API Gateway

Amazon API Gateway actúa como el punto de entrada único a los servicios backend, asumiendo la responsabilidad de exponer los endpoints REST necesarios para que la interfaz de usuario pueda crear nuevas incidencias, consultar incidencias existentes y actualizar el estado de las incidencias. Este servicio gestionado de AWS se encarga de gestionar aspectos técnicos complejos como el enrutamiento inteligente de peticiones hacia la función Lambda correspondiente, la validación de solicitudes, la limitación de velocidad para proteger contra abusos, y la transformación de solicitudes y respuestas cuando es necesario. API Gateway proporciona además características de seguridad integradas, monitoreo automático de métricas y generación de logs detallados para propósitos de auditoría y resolución de problemas.

Función Lambda para API

La lógica central del backend se implementa mediante una función AWS Lambda especializada, que recibe las solicitudes desde API Gateway y se encarga de ejecutar toda la lógica de negocio requerida. Esta función valida de manera exhaustiva los datos de entrada proporcionados por el usuario, verificando que cumplan con los formatos esperados y las restricciones de negocio establecidas. Una vez validados, realiza operaciones de lectura y escritura sobre DynamoDB de acuerdo con la naturaleza de la solicitud, ya sea para crear una nueva incidencia, recuperar incidencias existentes conforme a criterios específicos, o actualizar el estado de una incidencia particular. El uso de funciones Lambda elimina la necesidad de gestionar servidores dedicados, permitiendo que el sistema escale automáticamente en respuesta a cambios en la demanda sin intervención manual.

Base de Datos NoSQL (DynamoDB)

Amazon DynamoDB se utiliza como sistema de persistencia principal, proporcionando un almacenamiento de datos de alto rendimiento diseñado específicamente para aplicaciones que requieren baja latencia y alta disponibilidad. La tabla principal de incidencias utiliza una estructura de clave compuesta que incluye el identificador único de la incidencia y la marca temporal de creación, lo que permite recuperar registros de manera eficiente y mantener un historial ordenado cronológicamente. El sistema implementa además índices secundarios globales (Global Secondary Indexes, o GSI) que permiten que los usuarios ejecuten consultas eficientes basadas en otros atributos, como la severidad de la incidencia o el servicio tecnológico afectado, sin la necesidad de ejecutar costosos escaneos completos de la tabla. Esta estructura de indexación múltiple resulta fundamental para proporcionar una experiencia de usuario ágil al ejecutar búsquedas complejas sobre el conjunto de incidencias.

DynamoDB Streams

Una característica especializada de DynamoDB denominada Streams permite al sistema capturar automáticamente todos los cambios que ocurren en la tabla de incidencias, ya sean inserciones de nuevos registros, actualizaciones de registros existentes o eliminaciones. Estos eventos capturados se envían a un flujo ordenado que puede ser procesado por otros componentes del sistema. Este mecanismo es fundamental para implementar una arquitectura orientada a eventos, permitiendo que la lógica de notificaciones permanezca completamente

desacoplada del endpoint principal de API, lo que mejora significativamente tanto la escalabilidad del sistema como el tiempo de respuesta percibido por los usuarios del frontend.

Función Lambda para Notificaciones

Complementando la función Lambda de API existe una segunda función Lambda denominada Notifier, que se activa automáticamente siempre que se generan eventos en DynamoDB Streams. Esta función especializada implementa la lógica de evaluación de reglas de negocio, examinando atributos clave de cada incidencia como su nivel de severidad y su estado actual. Cuando detecta que una incidencia cumple con criterios predefinidos para alertas críticas, la función procesa esta información y genera un mensaje de notificación apropiado que será distribuido a través del servicio SNS. Esta arquitectura de procesamiento asíncrono permite que el sistema responda a nuevas incidencias sin impactar el tiempo de respuesta del endpoint principal.

Servicio de Notificaciones (SNS)

Amazon Simple Notification Service (SNS) gestiona de manera centralizada el envío de notificaciones por correo electrónico a los destinatarios especificados. Cuando la función Lambda Notifier determina que se debe generar una alerta, publica un mensaje en un tema SNS configurado. El servicio SNS se encarga automáticamente de entregar este mensaje a todas las direcciones de correo electrónico suscritas al tema, permitiendo que múltiples responsables técnicos reciban la notificación de manera simultánea. SNS también proporciona capacidades de reintentos automáticos y manejo de errores de entrega, asegurando que las notificaciones críticas se distribuyan de manera confiable.

Infraestructura como Código (Terraform)

La totalidad de los recursos de AWS necesarios para operar CIAS se define de manera declarativa mediante archivos de configuración Terraform. Esta aproximación proporciona múltiples ventajas significativas: permite que cualquier miembro del equipo de desarrollo pueda desplegar el sistema completo en su propia cuenta de AWS ejecutando un único comando, garantiza consistencia entre diferentes despliegues al eliminar variaciones causadas por procesos manuales, facilita el control de versiones de la infraestructura permitiendo rastrear cambios históricos, y simplifica tareas de auditoría y cumplimiento normativo. Los archivos de Terraform actúan como documentación viva del sistema, permitiendo que nuevos integrantes del equipo comprendan la arquitectura mediante lectura del código.

Implementación Técnica

Arquitectura de Backend Serverless

El backend de CIAS se ha implementado mediante una arquitectura completamente serverless, utilizando dos funciones AWS Lambda que actúan de manera independiente pero coordinada. Esta aproximación elimina la necesidad de gestionar máquinas virtuales o servidores dedicados, reduciendo significativamente la complejidad operativa y permitiendo que el sistema escale automáticamente en respuesta a cambios en la carga de trabajo.

La primera función Lambda, denominada Lambda API, se encarga de gestionar la totalidad de los endpoints REST que exponen la funcionalidad principal del sistema. El endpoint POST incidentes permite que usuarios autorizados registren nuevas incidencias técnicas, transmitiendo información detallada sobre el problema identificado. El endpoint GET incidentes facilita la consulta de incidencias existentes, soportando parámetros opcionales que permiten filtrar por severidad, por servicio tecnológico afectado, o por otros criterios relevantes, permitiendo a los usuarios localizar incidencias específicas entre potencialmente cientos de registros. El endpoint PATCH incidentes posibilita la actualización del estado de una incidencia particular, permitiendo que el personal técnico documente el progreso en la resolución de problemas. Finalmente, el endpoint GET health proporciona información sobre el estado operacional del sistema, permitiendo procesos de monitoreo externo verificar que el backend funciona correctamente.

La segunda función Lambda, denominada Lambda Notifier, opera de manera completamente asíncrona, siendo activada automáticamente mediante eventos generados por DynamoDB Streams. Esta función no es invocada directamente por usuarios, sino que se despierta cuando cambios ocurren en los registros de incidencias. Cada vez que se activa, la función evalúa las condiciones de negocio establecidas, examinando particularmente el nivel de severidad de la incidencia y su estado actual. Las reglas de negocio pueden ser configuradas para generar alertas cuando, por ejemplo, se registra una incidencia con severidad alta que permanece en estado abierto.

Estrategia de Persistencia en DynamoDB

El almacenamiento de datos en CIAS utiliza DynamoDB en modo PROVISIONED, una configuración que fue seleccionada considerando las restricciones operacionales y presupuestarias típicas de entornos académicos. En este modo, se asignan explícitamente capacidades mínimas de lectura y escritura para la tabla principal y para cada uno de sus índices secundarios, garantizando que el sistema cuente con recursos suficientes para operar de manera confiable.

La estructura de la tabla CIAS_Incidents utiliza una clave compuesta que combina el identificador único de la incidencia y la marca temporal de creación. Esta configuración de clave permite realizar búsquedas eficientes de incidencias específicas, mientras que la ordenación por timestamp facilita la recuperación de incidencias en orden cronológico. Los índices secundarios globales (GSI) implementados en la tabla permiten que se ejecuten consultas eficientes basadas en atributos alternativos, específicamente el nivel de severidad y

el servicio afectado, sin la necesidad de ejecutar escaneos completos de la tabla que serían computacionalmente costosos. Esta estructura de indexación múltiple es fundamental para proporcionar tiempos de respuesta aceptables cuando los usuarios buscan incidencias conforme a estos criterios.

Patrones de Notificaciones y Procesamiento de Eventos

La implementación de notificaciones en CIAS sigue un patrón orientado a eventos, una arquitectura moderna que proporciona múltiples ventajas en términos de escalabilidad y rendimiento. El proceso comienza cuando se crea o modifica una incidencia en DynamoDB, generando automáticamente un evento en el stream correspondiente. Este evento contiene toda la información relevante sobre el cambio ocurrido, permitiendo que componentes del sistema interesados en responder a este cambio puedan procesarlo sin acoplar la lógica de respuesta al componente que originó el cambio.

La función Lambda Notifier consume estos eventos de forma asíncrona, lo que significa que la creación de una incidencia se completa y se retorna una respuesta al usuario sin esperar a que finalice el procesamiento de la notificación. Esta arquitectura desacoplada proporciona dos beneficios significativos: primero, mejora el tiempo de respuesta percibido por los usuarios del frontend, ya que no deben esperar a que se completen procesos de notificación potencialmente lentos; segundo, permite que el sistema escale de manera independiente cada componente, ajustando recursos según la demanda específica de cada función en lugar de considerar el sistema como un todo monolítico.

Aproximación de Infraestructura como Código

La totalidad de la infraestructura de CIAS se define mediante archivos de configuración declarativos en Terraform. Esta herramienta permite especificar el estado deseado de todos los recursos de AWS de manera que es legible por humanos y procesable por máquinas. Cuando un desarrollador ejecuta el comando de aplicación de Terraform, la herramienta compara el estado actual de los recursos en AWS con el estado deseado especificado en los archivos de configuración, identificando automáticamente qué cambios deben realizarse y ejecutándolos de manera ordenada.

Esta aproximación proporciona múltiples beneficios operacionales significativos. Permite que cualquier miembro del equipo de desarrollo pueda desplegar la solución completa en una nueva cuenta de AWS ejecutando únicamente unos pocos comandos, sin necesidad de recordar procedimientos manuales complejos o de realizar clics manuales en la consola de AWS. Garantiza que la configuración sea completamente consistente entre diferentes despliegues, eliminando variaciones accidentales que podrían resultar de procesos manuales inconsistentes. Los archivos de Terraform pueden ser versionados mediante sistemas de control de código fuente, permitiendo rastrear todos los cambios en la infraestructura a lo largo del tiempo, facilitando auditorías y permitiendo revertir cambios cuando sea necesario. Finalmente, la infraestructura se convierte en documentación viva del sistema, permitiendo que nuevos integrantes del equipo comprendan la arquitectura y su justificación mediante lectura del código.

Análisis de Prestaciones y Costo

El análisis de costes se ha realizado utilizando AWS Pricing Calculator, considerando un escenario de uso académico moderado en la región us-east-1 durante un período de doce meses.

Servicio	Coste Mensual	Coste Inicial	Coste Anual	Descripción
Amazon API Gateway (HTTP API)	3,04 USD	—	36,50 USD	Basado en aproximadamente 100 solicitudes diarias con tamaño medio de 1,5 KB
Amazon DynamoDB (Provisioned)	11,41 USD	180,00 USD	316,92 USD	Capacidad mínima (1 RCU / 1 WCU) con compromiso anual, principal componente de coste
Amazon S3 (Frontend Estático)	0,03 USD	—	0,36 USD	Almacenamiento y servicio HTTP del frontend, impacto económico despreciable
Amazon SNS (Notificaciones)	0,45 USD	—	5,40 USD	Aproximadamente 200 notificaciones por correo electrónico mensual
AWS Lambda (API)	0,01 USD	—	0,12 USD	Bajo número de invocaciones y duración reducida de ejecuciones
AWS Lambda (Notifier)	0,00 USD	—	0,00 USD	Coste prácticamente nulo por baja actividad
TOTAL	**14,94 USD**	**180,00 USD**	**359,28 USD**	Coste total estimado a 12 meses

El coste total estimado para operar el sistema CIAS durante doce meses asciende a 359,28 USD, distribuido en un coste inicial de 180,00 USD más un promedio de 14,94 USD mensuales. Amazon DynamoDB representa el componente de mayor impacto económico, contribuyendo aproximadamente con 316,92 USD anuales, lo que corresponde a cerca del 88 por ciento del gasto total. Esta situación responde a la configuración de capacidad provisionada, necesaria debido a las restricciones operacionales de las cuentas académicas de AWS.

Los servicios complementarios presentan un impacto económico marginal. API Gateway contribuye con 36,50 USD anuales, mientras que SNS aporta 5,40 USD. Los costes asociados a Lambda, S3 y otros servicios resultan prácticamente despreciables, ubicándose por debajo del 1 por ciento del gasto total. Esta estructura de costes demuestra que es viable construir aplicaciones empresariales funcionales, escalables y orientadas a eventos en AWS utilizando presupuestos contenidos, particularmente en contextos académicos e institucionales donde los volúmenes de transacciones son moderados.

La relación coste-prestaciones ofrecida por el sistema CIAS resulta particularmente ventajosa cuando se considera que proporciona alta disponibilidad, escalabilidad automática, baja latencia y arquitectura completamente administrada sin requerir inversión en infraestructura física o personal operativo dedicado.

Conclusiones

El registro de incidencias desde la interfaz de usuario funciona correctamente, permitiendo a los usuarios crear nuevas incidencias con todos los atributos relevantes. El almacenamiento persistente en DynamoDB garantiza que estas incidencias se preservan de manera confiable y están disponibles para futuras consultas. Las consultas de incidencias producen resultados eficientes conforme a diversos criterios, permitiendo a los usuarios localizar incidencias específicas sin demoras significativas. La actualización del estado de incidencias permite que el personal técnico progrese en la resolución de problemas, documentando el avance en el ciclo de vida de cada incidencia. La generación automática de notificaciones para incidencias críticas funciona de manera confiable, asegurando que los responsables técnicos reciben alertas oportunas. Finalmente, el despliegue reproducible mediante Terraform permite que la totalidad del sistema pueda ser replicado en múltiples cuentas de AWS de manera consistente.

El proyecto CIAS constituye un caso de estudio especialmente valioso para propósitos educativos, demostrando de manera práctica cómo construir aplicaciones modernas en entornos cloud utilizando servicios gestionados y arquitecturas serverless. El sistema ejemplifica la aplicación de mejores prácticas reconocidas en diseño arquitectónico, incluyendo separación clara de responsabilidades, desacoplamiento mediante patrones orientados a eventos, y automatización integral de procesos de despliegue. Estos principios resultan fundamentales para desarrollar software escalable, mantenible y económicamente eficiente en la era contemporánea de computación en nube, siendo conocimientos altamente relevantes para profesionales que buscan desarrollar competencias en tecnologías cloud actuales.