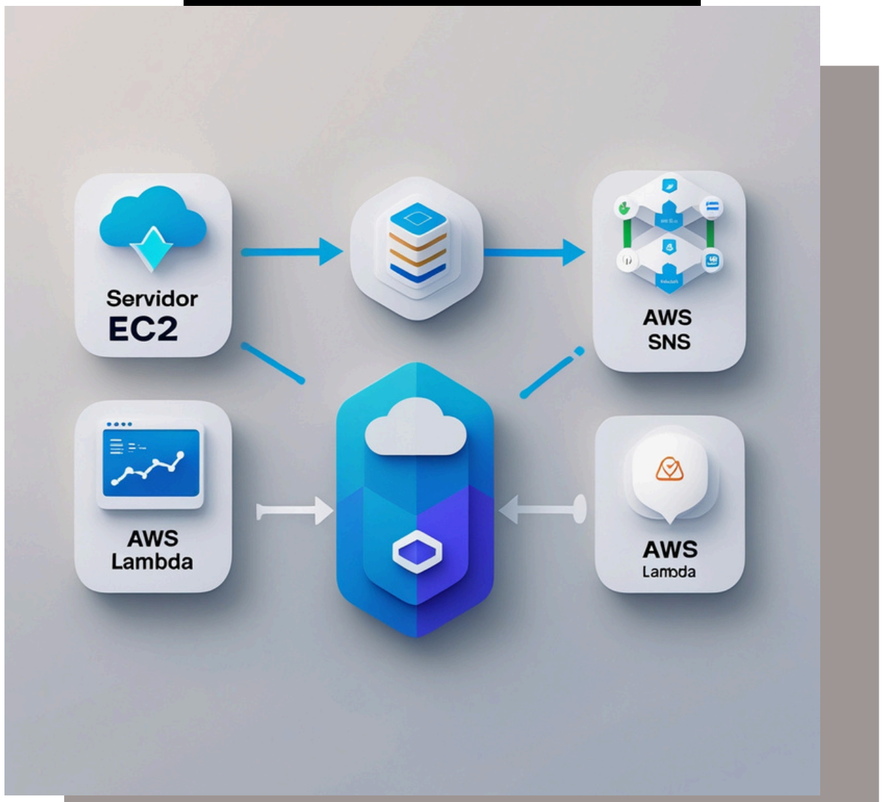
PRESENTACIÓN DE PROYECTO TFC

MONITORIZACIÓN AWS CON TELEGRAM

Presentado por Sergio González García

2º ASIR

ÍNDICE



INTRODUCCIÓN

- Descripción: Este proyecto implementa un sistema de monitorización para un servidor (instancia EC2) en Amazon Web Services (AWS). Se enfoca en supervisar métricas clave como CPU, RAM y disco.
- Propósito: Garantizar la eficiencia operativa y optimizar costes simulando las necesidades de una empresa.
  - Detectar problemas de rendimiento de forma proactiva mediante alertas automáticas.
- Alcance: Configuración de un servidor Linux con CloudWatch Agent.
  - Creación de alarmas en CloudWatch para umbrales críticos.
  - Integración de notificaciones vía email (SNS) y Telegram (SNS + Lambda + Bot).
  - Pruebas de carga y exportación de métricas para análisis.
  - Todo desarrollado dentro del Free Tier de AWS.

OBJETIVOS

1. Implementar Monitorización en EC2 (AWS)
2. Crear Sistema de Alertas Automatizadas
3. Integrar Notificaciones Multicanal
4. Validar y Analizar el Sistema
5. Exportación de métricas

¿POR QUÉ AWS?

- Madurez y Ecosistema Integrado
- Solución Nativa de Monitorización (CloudWatch)
- Free Tier Generoso

AWS VS AZURE

- Madurez y Amplitud del Ecosistema de Servicios
- Rendimiento y Menor Latencia
- Free Tier más Alineado con las Necesidades del Proyecto
- Mayor Cuota de Mercado y Ecosistema Comunitario

DESARROLLO DEL PROYECTO

1. Configuración Inicial en AWS
2. Recolección de Métricas con CloudWatch
3. Creación de Alarmas Críticas
4. Sistema de Notificación SNS
5. Integración con Telegram vía Lambda
6. Pruebas y Verificación del Flujo

PRUEBAS Y VALIDACIÓN

Se realizaron pruebas de estrés simulando condiciones críticas en la instancia EC2 para verificar la correcta detección de anomalías por CloudWatch, la activación de alarmas, y la recepción de notificaciones en tiempo real a través de email y Telegram, validando así todo el flujo del sistema de monitorización.

PRUEBA DE CARGA DE CPU

Se estresó la CPU con stress, validando la alarma de CPU y las notificaciones.

PRUEBA DE CONSUMO DE RAM

Se simuló alto uso de RAM con stress, verificando la alarma de RAM y alertas.

PRUEBA DE LLENADO DE DISCO

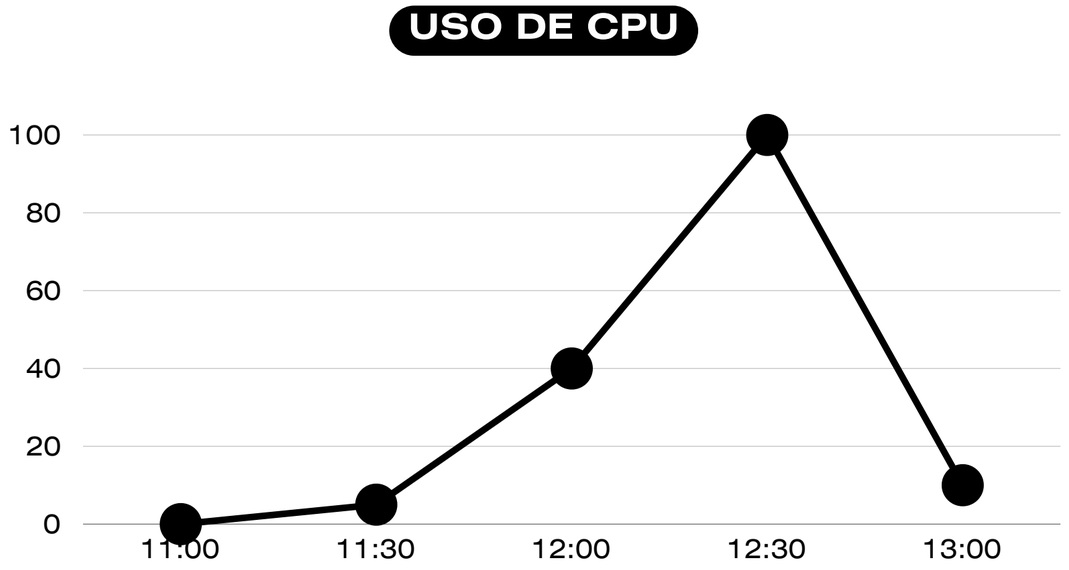
Se llenó el disco instalando zips con wget, comprobando la alarma de disco y notificaciones.

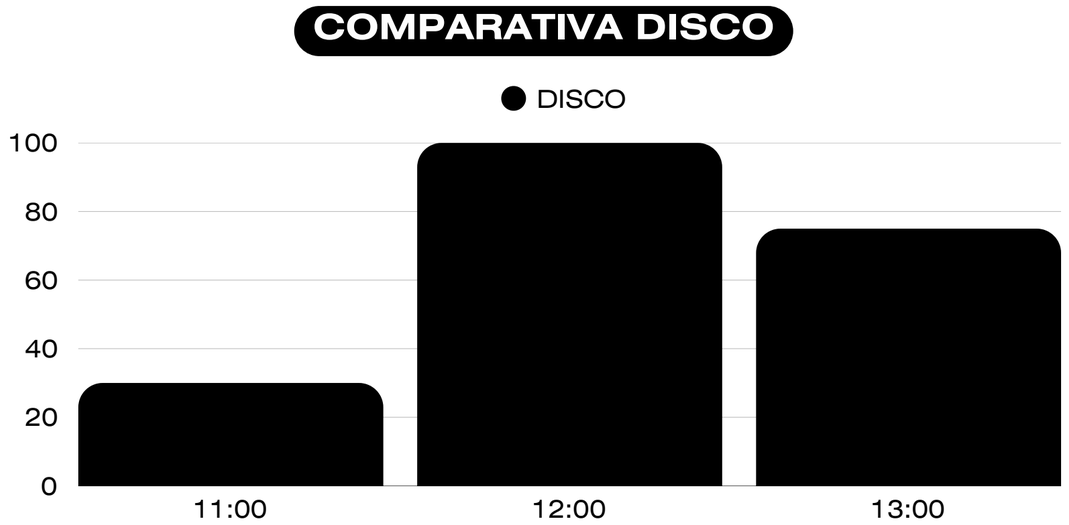
VALIDACIÓN INTEGRAL DEL FLUJO DE ALERTA

Se confirmó en todas las pruebas la correcta operación desde la métrica hasta la notificación en Telegram.

EXPORTACIÓN Y ANÁLISIS DE MÉTRICAS

Se exportaron las métricas de rendimiento (CPU, RAM, Disco) desde Amazon CloudWatch, principalmente en formato CSV, para permitir su análisis externo y la creación de visualizaciones, por ejemplo, con herramientas como Power BI.





CONCLUSIÓN

Se demostró con éxito la viabilidad de un sistema de monitorización funcional y escalable en AWS (EC2, CloudWatch, SNS, Lambda) con notificaciones en Telegram, utilizando exclusivamente recursos del Free Tier, permitiendo anticipar problemas y optimizar costes.

¡Gracias por su atención!