

# Observabilidad de infraestructuras y aplicaciones con *AWS*

---

UPC - Cloud Computing Architecture

# Índice - ¿Qué servicios vamos a ver?

- CloudWatch
  - Alarms
  - Logs
  - Metrics
- EventBridge
- CloudTrail
  - Athena
- X-Ray

# Monitorización vs Observabilidad

- **Monitorización**

- Supervisión de métricas predefinidas con el objetivo de detectar errores en servicios específicos
- Debemos conocer las métricas clave de antemano

- **Observabilidad**

- Indica la capacidad que tenemos de sacar conclusiones sobre el estado de un sistema a partir de los datos que devuelve
- Nos ayuda a comprender mejor cómo funciona nuestro sistema
- Nos permite anticiparnos a problemas desconocidos
- Significado más global de sistema
  - Sistemas complejos
  - Arquitectura de microservicios (+ Servicios)

# Datos a monitorizar

- Métricas

- Medidas periódicas de KPIs (Key Performance Indicators)
  - Uso de CPU
- Información agregable

- Logs

- Registros de eventos + Timestamp
- Ejemplos
  - Access Log de un servidor web

- Trazas

- Orientado a arquitecturas de microservicios
- Comportamiento de las peticiones
- Ejemplos
  - Latencias específicas

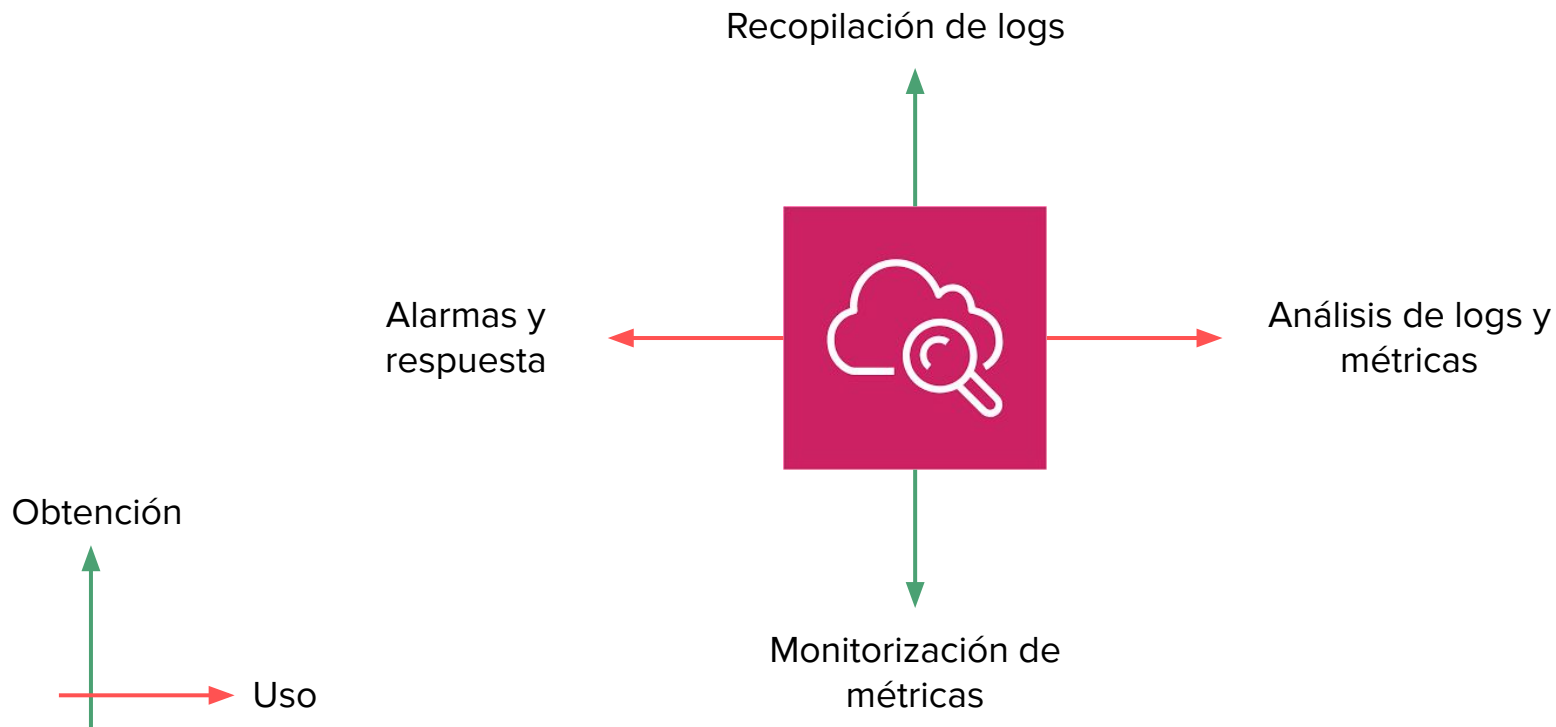
# ¿Qué métricas son importantes?

- Servicios centrados a usuarios (user-centric)
  - **Four Golden Signals**
    - **Latencia:** Tiempo que tarda en procesar una petición
    - **Tráfico:** ¿Cuánta demanda tiene el servicio? (req / s)
    - **Errores:** Ratio de peticiones que fallan (req / s)
    - **Saturación:** Carga del sistema (memoria en uso)
- Infraestructura
  - **USE**
    - **Utilización:** % CPU
    - **Saturación:** Trabajos (jobs) en cola
    - **Errores:** Número de eventos de error

# CloudWatch

- Nos permite monitorizar recursos y aplicaciones en tiempo real
- Los servicios de *AWS* disponen de métricas propias
  - EC2 uso de CPU
  - RDS espacio libre de almacenamiento / IOPS
- Casos de uso:
  - Controlar recursos y rendimiento de aplicaciones (métricas)
  - Agrupar y monitorizar ficheros de logs
  - Crear y disparar alarmas
    - Basadas en métricas y logs

# CloudWatch



# CloudWatch

- **Conceptos clave:**

- Métricas
  - Estándar
  - Personalizadas
- Alarmas
  - Notificaciones

- **Limitaciones**

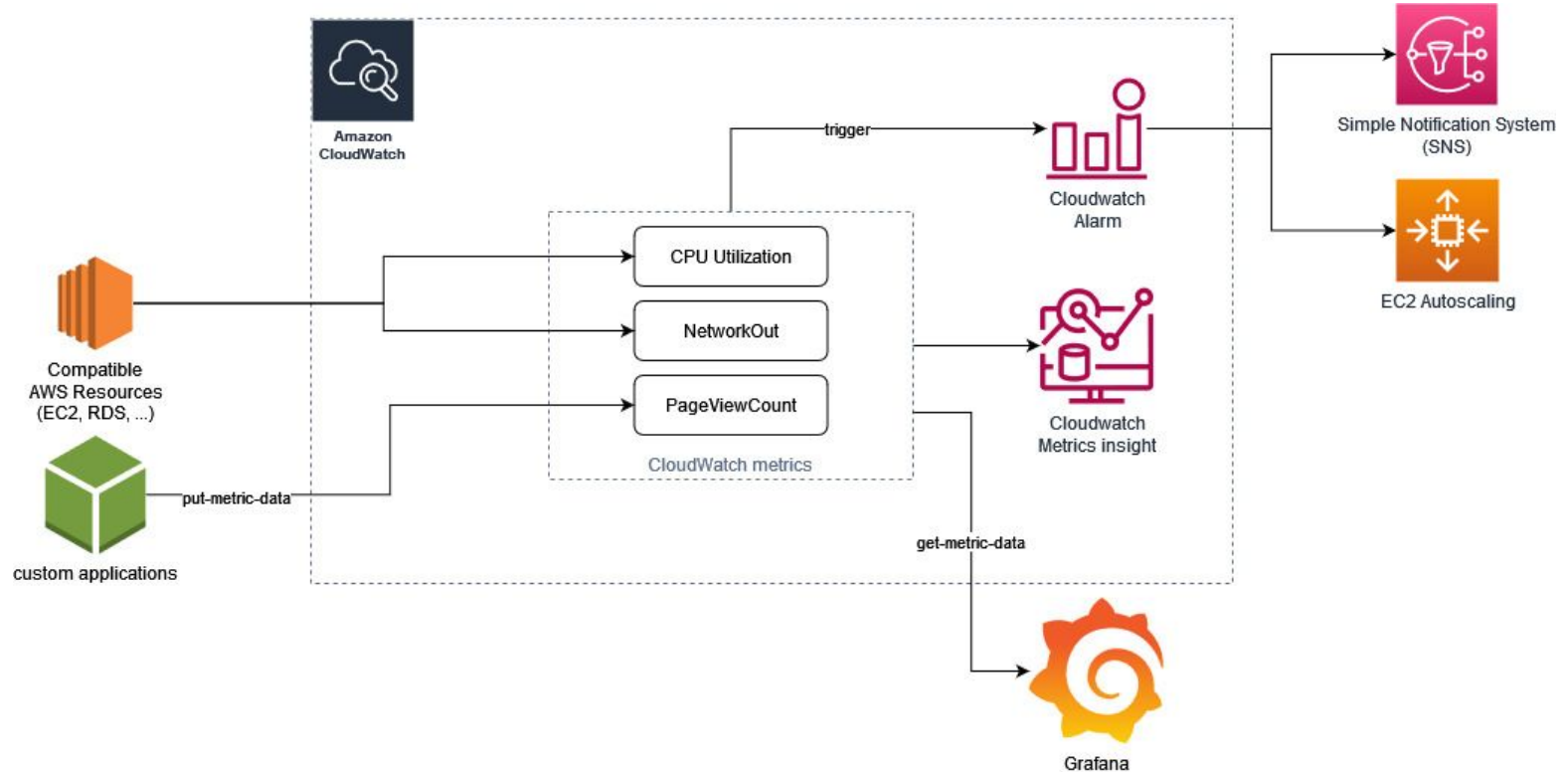
- CloudWatch por sí solo, no monitoriza a nivel de SO
  - Solo recolecta métricas estándar expuestas por los servicios (EC2, RDS, S3...)

- **CloudWatch agent:** recopila métricas a nivel de sistema

- Espacio de disco disponible
- Métricas custom
- etc...



# CloudWatch - Ejemplo - Caso de uso



# CloudWatch - Métricas

**Metrica:** Clave + Valor + Timestamp

**Namespace:** Grupo de métricas relacionadas

**Dimensión:** Par clave valor que categoriza la métrica

MetricName + Dimension = Nueva metrica!

```
1  {
2    "Metrics": [
3      {
4        "Namespace": "AWS/S3",
5        "MetricName": "BucketSizeBytes",
6        "Dimensions": [
7          {
8            "Name": "StorageType",
9            "Value": "StandardStorage"
10         },
11         {
12           "Name": "BucketName",
13           "Value": "mySuperBucket"
14         }
15       ]
16     }
17   ]
18 }
```

\$ aws cloudwatch list-metrics --namespace "AWS/S3"

# CloudWatch - Metrics - Resolución

- **Estándar:** 5 minutos
- **Detallada:** 1 minuto
- **Alta resolución personalizada:** < 1 minuto

# CloudWatch - Metrics - Retención de datos

A medida que pasa el tiempo, las métricas van perdiendo resolución, agregando los datos para ser almacenados:

- Periodo de < **60 segundos** (métricas custom de alta resolución)
  - Durante **3 horas**
- Periodo de **60 segundos**
  - Durante **15 días**
- Periodo de **300 segundos (5 min)**
  - Durante **63 días (2 meses)**
- Periodo de **3600 segundos (1 hora)**
  - Durante **455 días (15 meses)**

**Importante!** Las métricas que no han tenido ninguna actualización no aparecen en la consola



**get-metric-data / get-metric-statistics**

# CloudWatch - Metrics - Precio

- **Capa gratuita:**

- Todas las Métricas estándar (frecuencia de 5 minutos)
- 10 métricas de monitoreo detallado (frecuencia de 1 minuto)
- 1 millón de solicitudes API (no aplicable a GetMetricData ni GetMetricWidgetImage)

Capas	Costo (métrica/mes)
Primeras 10 000 métricas	0,30 USD
Siguientes 240 000 métricas	0,10 USD
Siguientes 750 000 métricas	0,05 USD
Más de 1 000 000 de métricas	0,02 USD

## API

GetMetricData, GetInsightRuleReport	0,01 USD por 1000 métricas solicitadas
GetMetricWidgetImage	0,02 USD por 1000 métricas solicitadas
Solicitudes de GetMetricStatistics, ListMetrics, PutMetricData, GetDashboard, ListDashboards, PutDashboard y DeleteDashboards	0,01 USD por cada 1000 solicitudes

# CloudWatch - Metrics - Precio

- **Ejemplo 1 - Monitorización detallada:**

- Suponemos 10 VM EC2
- Cada VM EC2 dispone de 7 métricas
- Métricas totales =  $7 * 10 = 70$
- $0,3 \text{ USD} * 70 = \mathbf{21 \text{ USD al mes!}}$

# CloudWatch - Metrics - Precio

- **Ejemplo 2 - Métricas personalizadas:**

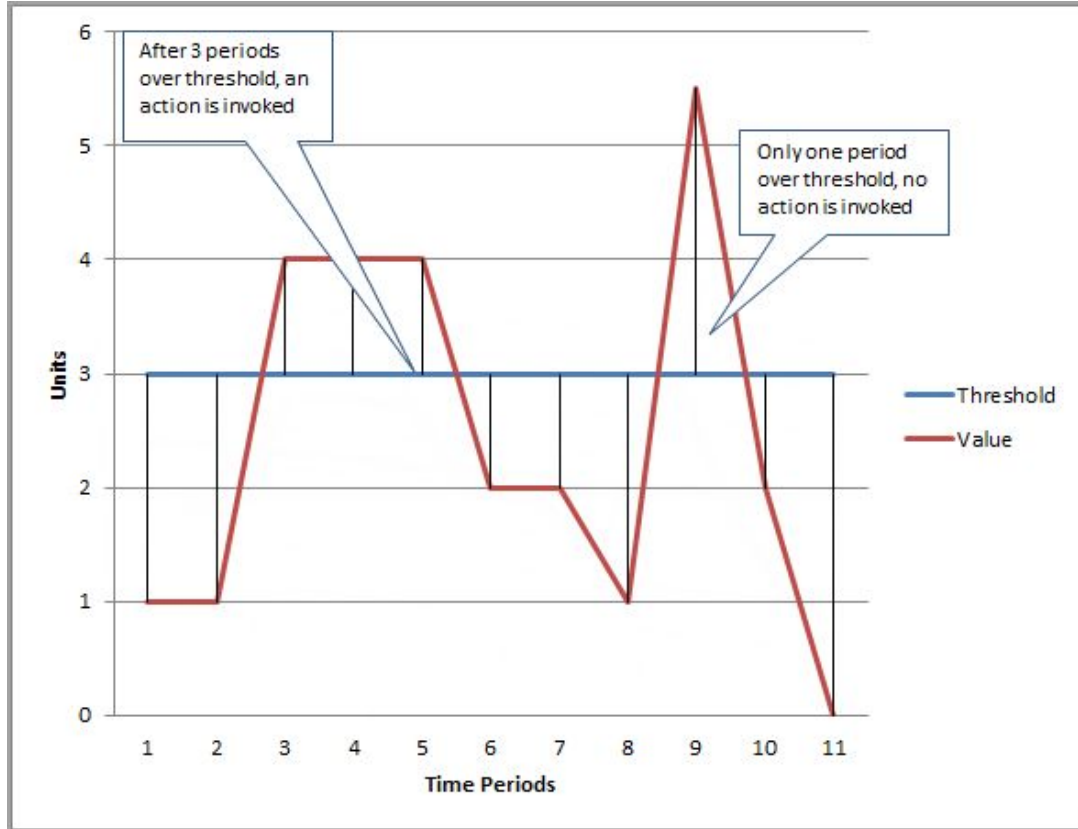
- Suponemos 100 VM EC2
- Publican 5 métricas personalizadas con el Agente CloudWatch cada minuto
- **Métricas**
  - $5 * 100 = 500$  métricas
  - $500 * 0,3 \text{ USD} = \mathbf{150\text{USD}}$
- **Peticiones**
  - $100 * (43200 \text{ minutos} / 1 \text{ minuto}) = \mathbf{4.320.000 \text{ peticiones al mes}}$
  - 1.000.000 entran dentro de la capa gratuita
  - $3.320.000 / 1000 * 0.01 = \mathbf{33 \text{ USD}}$
- **Total = 33 + 150 USD = 183 USD**

# CloudWatch - Alarms

- 3 estados:
  - OK
    - No se ha excedido el umbral
  - Alarma
    - Se ha excedido el umbral
  - Datos insuficientes
    - La alarma se acaba de crear, o aún no hay datos para hacer una evaluación



# Cloudwatch - Alarms - Ejemplo



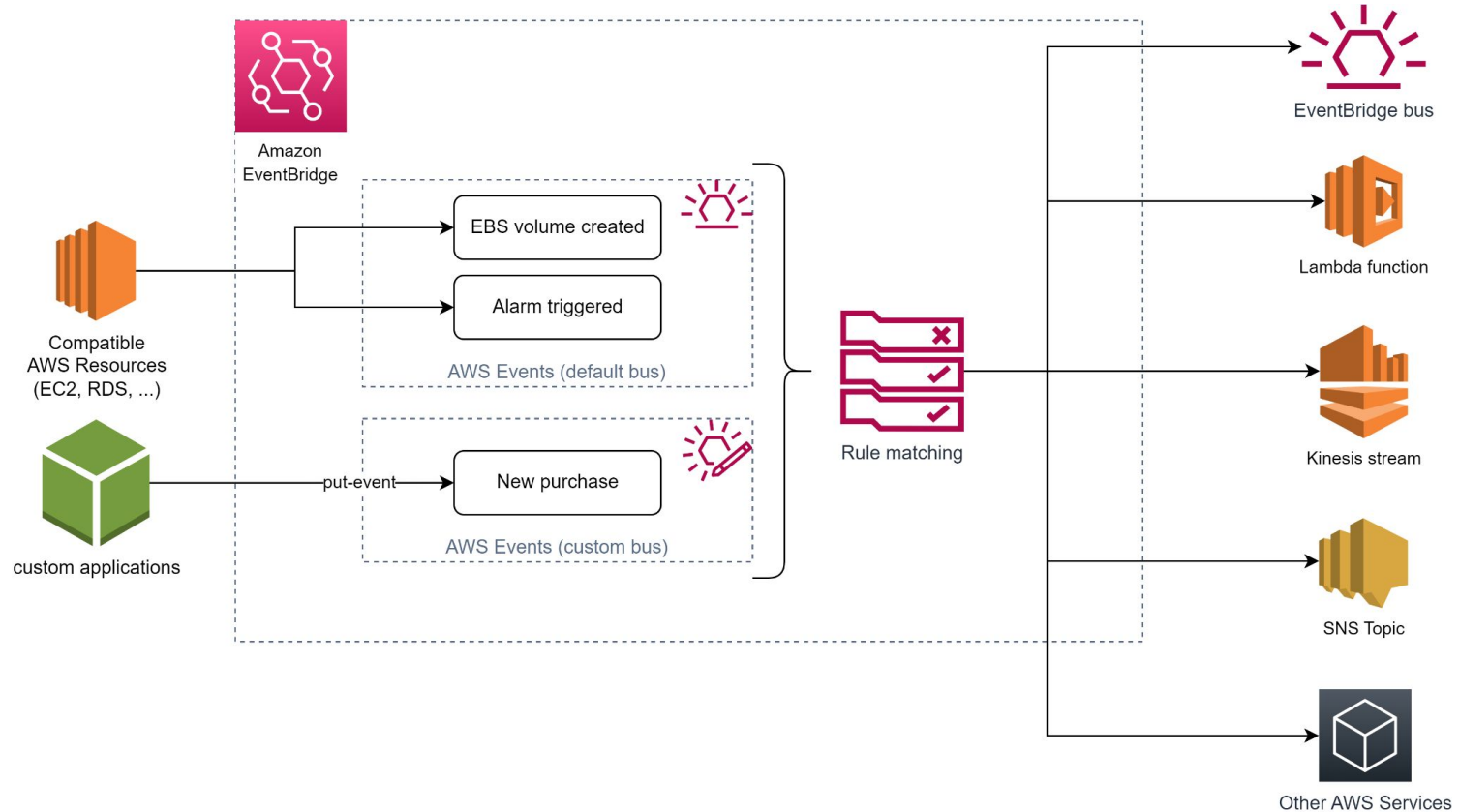
- **Evaluation period**  
(Periodo de evaluacion)
  - Ejemplo: 3
- **Datapoints to Alarm**  
(Puntos de datos para la alarma)
  - Ejemplo: 3

# Events (EventBridge)

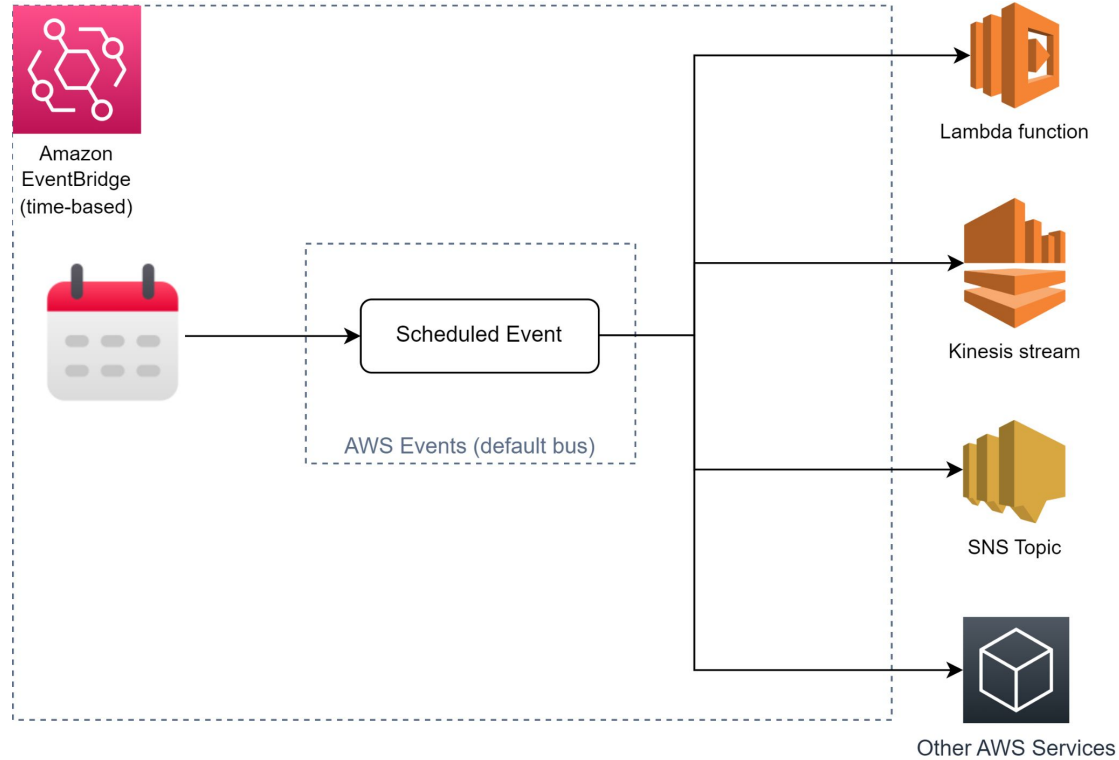


- Capturar eventos que ocurran en nuestra cuenta AWS y actuar acorde:
  - Creación de un volumen EBS
  - Activación de una alarma
- **EventBridge** es un bus de eventos sin servidor que facilita la creación de aplicaciones basadas en eventos

# Events (EventBridge) - Ejemplo



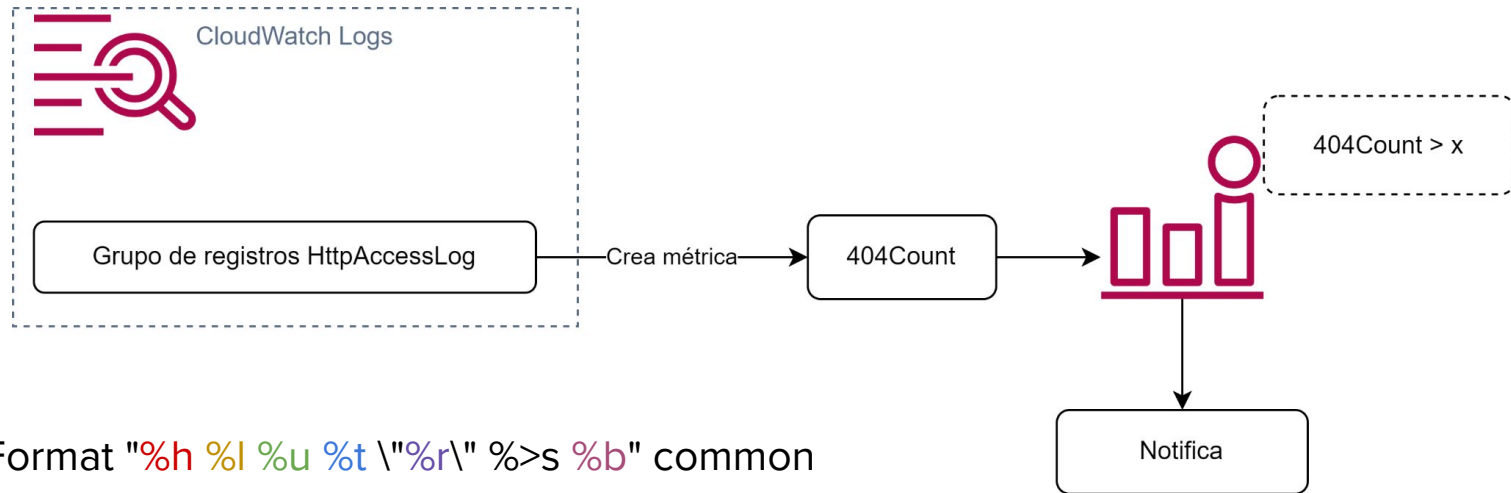
# Events (EventBridge) - Ejemplo CRON



# CloudWatch Logs

- CloudWatch logs nos permite monitorizar y almacenar archivos de registro de:
  - Instancias EC2
  - Servicios de AWS
    - AWS Route 53 / AWS Lambda / AWS CloudTrail
  - Otros
- Funcionalidades
  - Recopilación automática de registros
  - Agregación de datos en **grupos de registros**
  - Capacidad de configurar **filtros métricos**
    - Búsqueda de patrones (Errores en un Access Log de Apache)
    - Crear métricas en base a registros
  - Consulta de registros y creación de visualizaciones con **CloudWatch Logs Insights**

# CloudWatch - Logs - Creación de métricas a partir de logs



LogFormat "%h %l %u %t \"%r\" %>s %b" common

127.0.0.1 - Marc [01/Feb/2023:13:55:36 +0100] "GET /hello\_world HTTP/1.0" 200 2326

# CloudWatch - Logs - Precio

- **Capa gratuita:**
  - 5GB (recopilación, almacenamiento, análisis...)

Recopilación (captura de datos)	0,50 USD por GB
Almacenamiento (archivado)	0,03 USD por GB
Análisis (consultas de Logs Insights)	0,005 USD por GB de datos escaneados
Detección y máscara (Data Protection)	0,12 USD por GB de datos analizados

# CloudWatch - Logs - Precio

- **Ejemplo 1 - Logs HTTP:**

- Suponemos que registramos 1 GB diario durante 30 días = 30 GB al mes
- Captura de datos
  - 0 a 5GB = 0 (Capa gratuita)
  - 5 a 30 GB =  $0,5 \text{ USD} * 25 = \mathbf{12,50 \text{ USD}}$
- Almacenamiento
  - 0 a 5GB = 0 (Capa gratuita)
  - 5 a 30 GB =  $0,03 \text{ USD} * 25 = \mathbf{0,8 \text{ USD}}$
- Total
  - $\mathbf{12,50 + 0,8 = 13,3 \text{ USD}}$



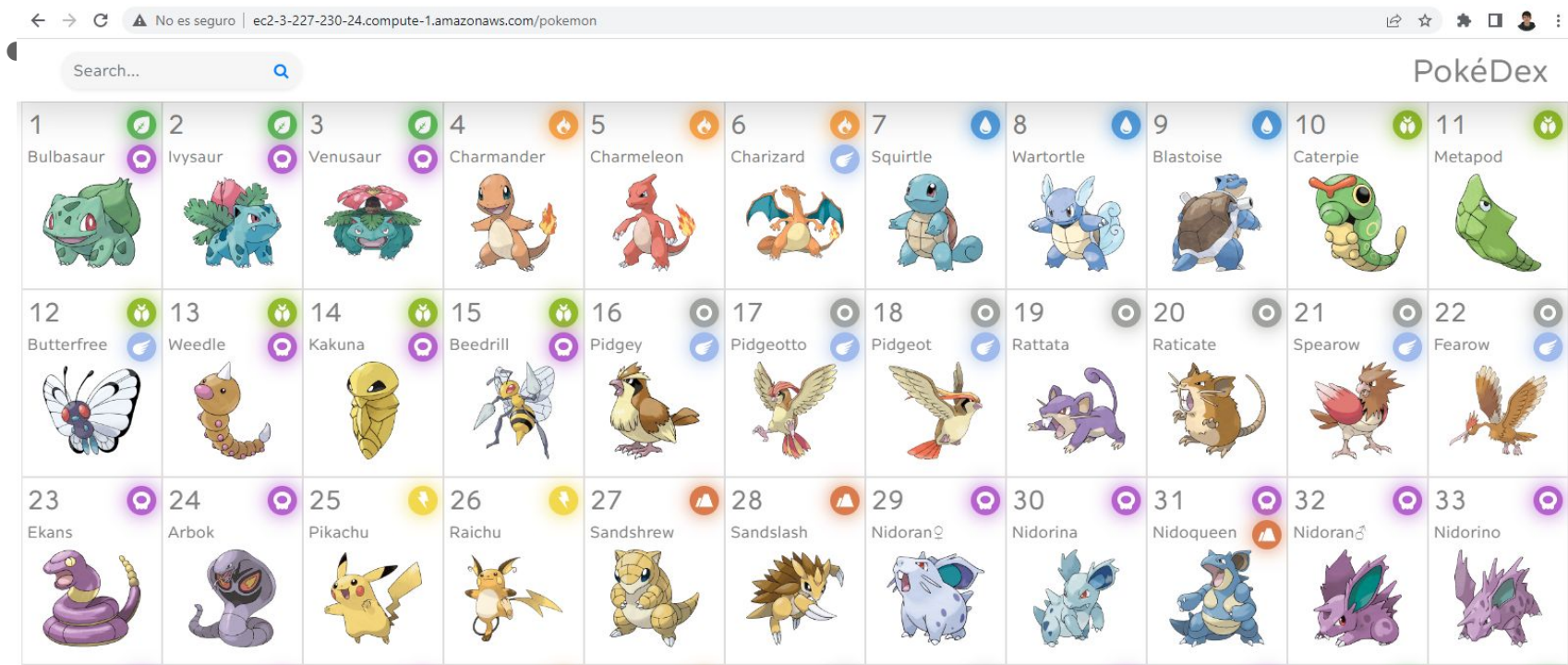
Quizz!



# Lab 1 - Pokedex

- Pokemon Go ha vuelto a hacerse viral
- El CEO ha decidido publicar una aplicación Web que ofrece el servicio de Pokedex
  - Objetivo: Conseguir aumentar la presencia de la marca en los círculos de jugadores
- La aplicación parece que está gustando aunque algunas personas se quejan de errores puntuales
- Problema! No la estamos monitorizando! ¿Errores?

# Lab 1 - Pokedex




# Lab 1 - La Pokedex

← → ↻ ⚠ No es seguro | ec2-3-227-230-24.compute-1.amazonaws.com/pokemon/94

← PokéDex

## GENGAR

Shadow Pokémon



EVOLUTION CHAIN

**ID** #94

**Height** 1.5m ( 4'11" )

**Weight** 40.5kg ( 89.3lbs. )

**Abilities** CURSED-BODY

**Type** Ghost Poison

**Forms** GENGAR MEGA GENGAR GIGANTAMAX GENGAR

**Base** **Min** **Max**

HP	60	
Attack	65	
Defence	60	
Sp. Attack		130
Sp. Defence	75	
Speed		110
Total	500	

# Lab 1 - La Pokedex

- Despliegue con **Terraform**
- Servidor NGINX
  - App Web Angular

