

# AirTrafficWatch: Observabilidad 4.0

Monitoreo activo del tráfico aéreo con Zabbix Cloud + Web

*Enriquez De La Cruz Emanuel  
Cruz Contreras Jonathan Ivan  
Martinez Muñoz Stephanie Denis  
Aguirre Rojas Mariana Isabel  
Rojas Rosales Jason Josie  
Hernandez Yonemoto Jorge Gonzalo*

**Necesidad de visibilidad unificada del tráfico aéreo y la  
infraestructura que lo soporta.**

## Temáticas abordadas

Tema 1: Flujo de Datos y Conexión en la Nube

Tema 2: App Web Clara y Sincronizada

Tema 3: Automatizaciones Visibles

Tema 4: Profundidad Técnica y Habilidades  
Demostradas

Tema 5: Observabilidad con Impacto

# Proyecto



# Flujo de Datos y Conexión en la Nube

El cerebro de la solución es Zabbix Cloud, que no solo realiza el monitoreo activo de la base y el estado de la API, sino que también actúa como nuestra capa de servicios.

La clave de la sincronización es que nuestra App Web no consulta la Base de Datos directamente; consulta la API de Zabbix utilizando métodos como item.get y host.get para garantizar que la visualización del usuario siempre refleje lo que Zabbix está viendo en tiempo real.



# Explicación Ampliada: Flujo de Datos y Conexión en la Nube

## 1. Ingesta de Datos (OpenSky API)

El proceso comienza consultando la API pública de OpenSky para obtener datos de vuelos en tiempo real sobre el área designada de la CDMX. Un script personalizado (que puede mencionar: Python/Go/etc.) es el encargado de parsear estos datos y luego persistirlos en nuestra Base de Datos.

## 2. Monitoreo Activo (Zabbix Cloud)

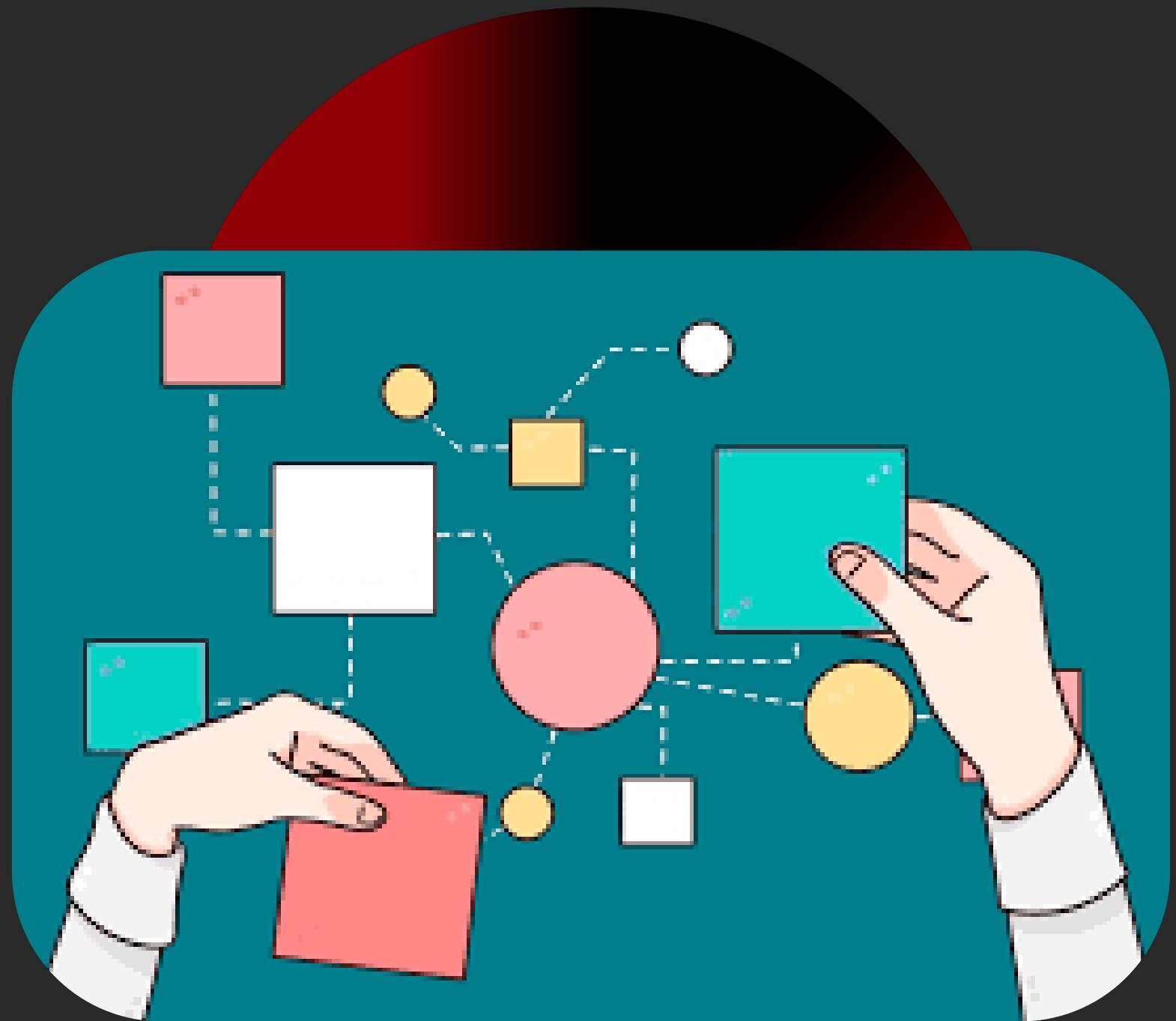
Aquí reside la observabilidad:

- Monitoreo del Servicio: Zabbix está configurado para monitorear constantemente la salud de la Base de Datos y el estado de la API de OpenSky (para saber si está caída).
- Métricas del Negocio: Creamos Items clave en Zabbix para rastrear métricas de negocio como el Conteo Total de Vuelos o los Registros de Vuelo.pdf].
- Conexión en la Nube: Al usar Zabbix Cloud, garantizamos que nuestro sistema de monitoreo sea escalable y accesible de forma remota para la App Web.

# La Web Sincronizada (¡El Foco de la Puntuación!)

Nuestra App Web cumple con los cuatro requisitos de visualización y, más importante aún, está sincronizada directamente con Zabbix.

- Vuelos Activos y Tráfico Aéreo: Utilizamos el método `host.get` para que la lista de vuelos que ven aquí sea la lista de Hosts que Zabbix está monitoreando en tiempo real.
- Estado del Sistema: El indicador superior consulta el 'último valor' de un Item clave usando `item.get`, garantizando que el estado sea un reflejo validado por el monitoreo.



# App Web Clara y Sincronizada

Visualización Unificada: La interfaz web cumple con los requisitos: Vuelos Activos, Tráfico Aéreo y Estado del Sistema.

Énfasis en el Uso de la API: El frontend consulta directamente la API de Zabbix para obtener datos frescos y ver el "Último Valor" del ítem del conteo de vuelos (`item.get`).

# Automatizaciones Visibles

Hemos configurado Triggers avanzados en Zabbix Cloud que se traducen en acciones inmediatas y visibles en nuestra web.

Por ejemplo: si la API no responde, Zabbix genera una alerta crítica, y nuestra App Web reacciona de inmediato, mostrando el indicador en ROJO VIVO. Si el tráfico excede un límite, Zabbix ejecuta una limpieza de BD y nuestra web muestra un mensaje de 'Limpieza Activa'.

Esto demuestra que transformamos la observabilidad en reacción visible para el operador."



# 25% de la Puntuación: Automatizaciones Visibles

## 1. Objetivo y Valor (Lo que tienes que decir primero)

La clave para ganar el 25% de esta sección no es solo tener triggers, sino lograr que esas acciones se reflejen en la web en tiempo real (mensajes, actualizaciones, indicadores).

- Mensaje de Impacto: "Nuestra solución transforma las alertas pasivas de monitoreo en acciones visibles y automáticas que el operador puede ver y confirmar desde nuestra interfaz web."

## 2. Mecanismos Técnicos

Explica brevemente cómo Zabbix comunica la acción:

- En Zabbix: Configuramos Triggers y Acciones que, al activarse, ejecutan un script personalizado (ej. un webhook) que actualiza un valor específico en la Base de Datos o envía un evento a un servicio que la App Web está consultando continuamente.
- En la App Web: El dashboard de Estado del Sistema está consultando constantemente la API de Zabbix (item.get) o un servicio intermedio. Cuando el valor cambia debido a una acción de Zabbix, el elemento visual se actualiza inmediatamente.

# Profundidad Técnica y Habilidades Demostradas

**Este proyecto exigió un stack completo de habilidades:**

## 1. Dominio de Zabbix Cloud:

Creamos un monitoreo funcional y proactivo (25% del peso).

## 2. Maestría en la API de Zabbix:

Utilizamos host.get para vuelos, item.get para el estado y history.get para las estadísticas, logrando la sincronización total.

## 3. Comunicación Inter-Procesos:

Conectamos las acciones de Zabbix (los Triggers) directamente a los indicadores visuales de nuestra web.

# Conclusión: Observabilidad con Impacto

El proyecto entregó una solución que es funcional, monitoreable y reactiva, cumpliendo con la sincronización y automatización requeridas.



# MUCHAS GRACIAS

Fin de la Presentacion ¡Muchas  
gracias!

Con esto concluimos la presentación de AirTrafficWatch. Queremos agradecer al jurado y a los organizadores del Hackathon. Esperamos que nuestra solución demuestre cómo el monitoreo con Zabbix puede generar valor y acción en tiempo real.

SE ACEPTAN APLAUSOS NO  
PREGUNTAS