

SISTEMA DE DETECCIÓN DE VEHÍCULOS HURTADOS

Ceiba



Planteamiento

Implementación de un software para identificar las zonas por las cuales transitan los vehículos hurtados, con el fin de generar información que ayude a recuperar estos vehículos.

Premisas / Supuestos / Conclusiones

Premisa #1) El sistema operará en múltiples países y ciudades en las cuales actualmente se cuenta con dispositivos que monitorean el tránsito en las calles, estos dispositivos son un hardware con las siguientes características:

- CPU de 2 cores a 2 Ghz
- 1 GB de memoria ram
- DD de 16GB
- Conexión a internet vía GSM
- GPS
- Cada dispositivo tiene cámaras para la captura de las imágenes de los vehículos que transitan
- Se cuenta con un software que se encarga de reconocer las matrículas de dichos vehículos, para acceder a esta información, el software provee un API.
- Los dispositivos están conectados a la red eléctrica, pero además cuentan con una batería de respaldo con una duración aproximada de 1 hora.

Supuestos de la Premisa #1)

- Las cámaras van guardando información de todos los vehículos que transitan, hurtados o no
- Las cámaras utilizan el espacio en disco para el software de reconocimiento de matrículas, las últimas imágenes grabadas y para una base de datos que guarda las matrículas reconocidas de las imágenes detectadas y analizadas por el software
- El API provisto es un API RESTful que permite recuperar en formato JSON, los datos de las matrículas de los vehículos que ya fueron procesados por el software de la cámara
- La cámara no se detiene en la captura de imágenes, si empieza a usar la batería de respaldo, esta sigue funcionando
- Si consideramos que la cámara funciona capturando video a 720p HD, que un 1 minuto de video a 720p HD ocupa aprox. 15 megas, entonces con 14 gigas de DD podemos almacenar unas 16 horas de transito. Ahora bien, en diversos puntos de Antioquia pueden pasar hasta 300 vehículos por hora, eso implica una retención de 4800 matrículas

Conclusiones sobre la Premisa #1)

- Los datos que se obtienen a través de las cámaras son del tipo histórico y/o time-stamped, por lo tanto inmutable
- Se dispone de la geolocalización de la cámara porque disponen de GPS

Premisas / Supuestos / Conclusiones

Premisa #2) La policía de cada país cuenta con una base de datos de vehículos hurtados, cada país puede tener dicha base de datos en diferente formato y tecnología, sin embargo asuma que se cuenta con la siguiente información de dichos vehículos:

- Matrícula
- Fecha de hurto
- Lugar de hurto (ciudad y país)
- DNI del propietario
- Marca del vehículo
- Clase
- Línea
- Color
- Modelo

Los anteriores datos siempre estarán disponibles pero cada país podrá proveer datos adicionales del vehículo hurtado y su propietario.

Supuestos de la Premisa #2)

- La base de datos a pesar de estar en diferentes formatos o tecnologías, es accesible de forma remota. A través de APIs o repositorios

Conclusiones sobre la Premisa #2)

- La información de la policía debe cotejarse con la información que se extraiga de las cámaras
- Los datos provistos por la policía es información semi-estructurada
- Diferentes formatos y tecnologías implica diferentes formas de leer los datos
- Los datos que se obtienen de la policía son del tipo histórico y/o time-stamped

Requerimientos

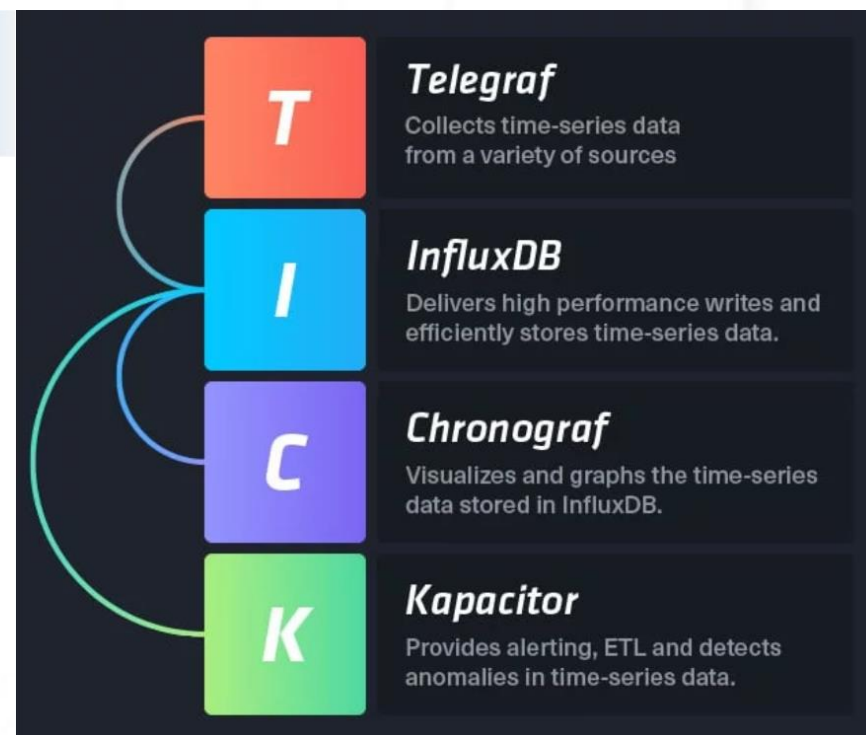
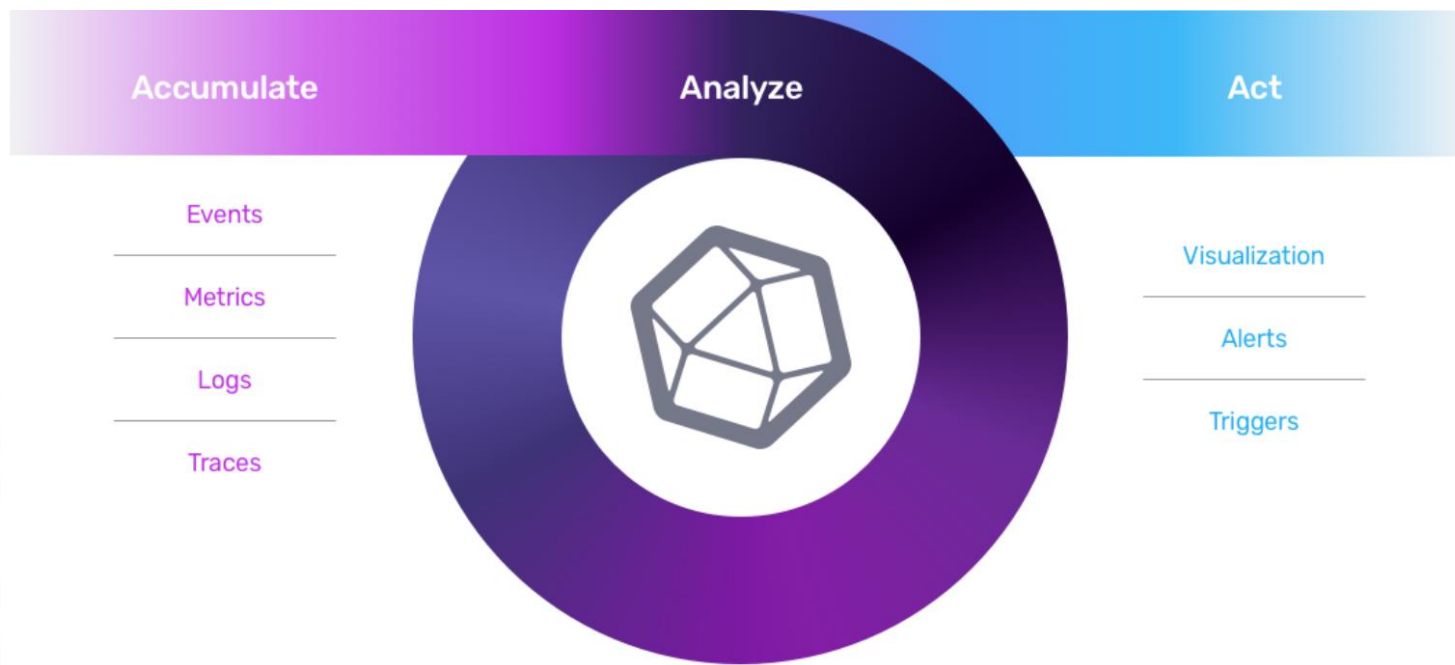
1. El software a desarrollar se pretende llevar a la nube, puede ser Amazon, Azure o Google, evitando al máximo el acoplamiento con estas nubes de manera que sea posible cambiar de proveedor, o en algún momento montar una nube privada.
2. La solución debe ser dinámicamente escalable.
3. La solución debe garantizar alta disponibilidad.
4. La comunicación entre la solución y los dispositivos debe ser segura
5. La solución debe estar en capacidad de integrarse con los sistemas de autenticación de usuarios de las instituciones policiales de los diferentes países
6. La solución debe estar en capacidad de segmentar la información por país.
7. La solución debe permitir la búsqueda de una matrícula y generar un listado de eventos en los que se ha identificado dicho vehículo incluyendo:
 - a) Fecha y hora en que fue visto el vehículo
 - b) Lugar
 - c) Prueba visual
8. Visualización de los anteriores datos en un mapa.
9. La solución debe integrarse con alguna herramienta de análisis de datos con el fin de identificar tendencias
10. La solución debe garantizar la recolección y sincronización de los datos, considerando que algunos dispositivos podrán no tener cobertura GSM continua y que la disponibilidad energética puede fallar en casos puntuales hasta por 4 horas

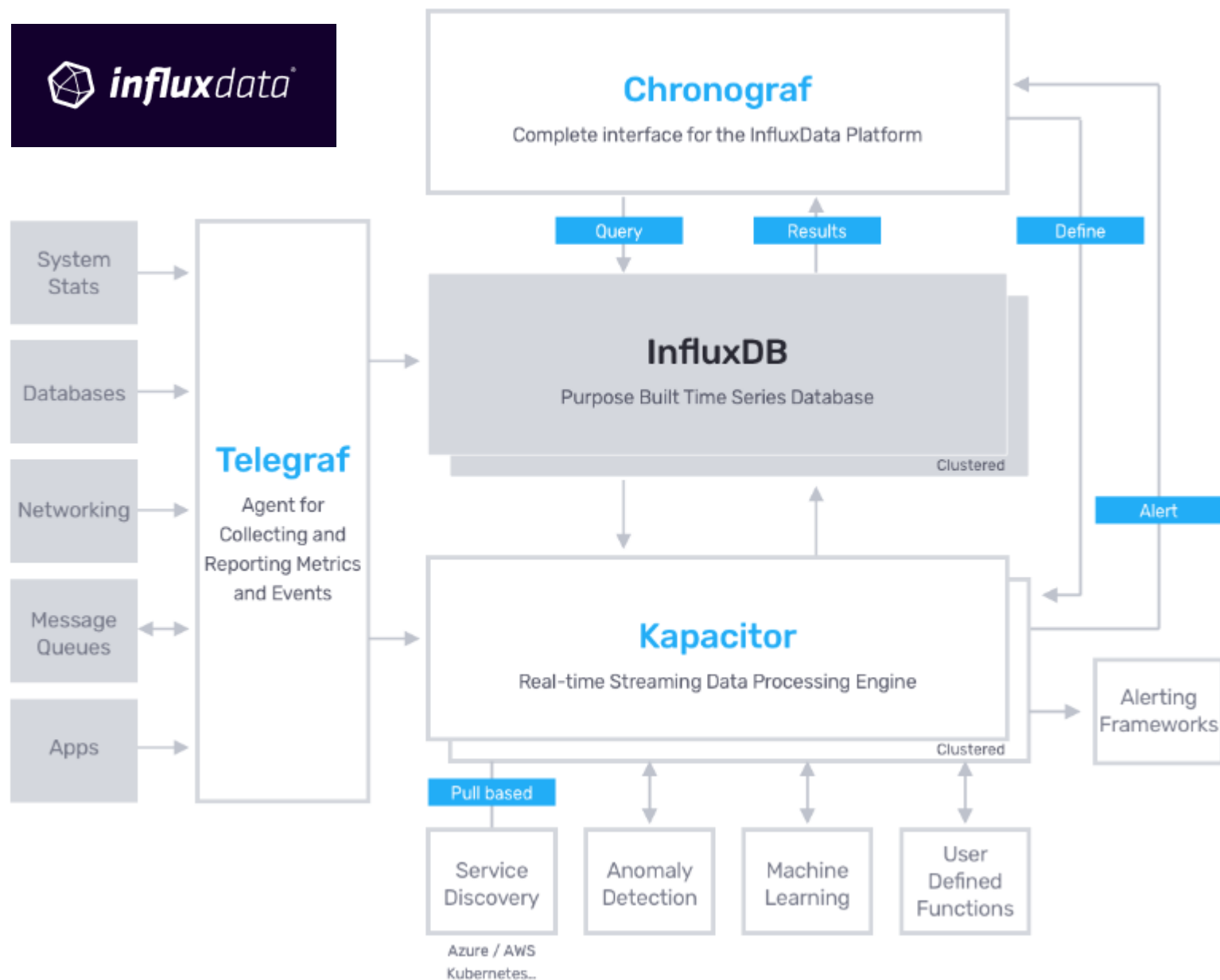
Resumen de Requerimientos

La solución es disponer de un sistema distribuido, seguro, escalable y de alta disponibilidad que en función de la lectura de eventos de tránsito organizados como data semi-estructurada en diferentes formatos, permita generar alertas, identificar tendencias, segmentar información por país y visualizarla en un mapa

La Solución

InfluxDB is used as a data store for any use case involving large amounts of time-stamped data, including DevOps monitoring, log data, application metrics, IoT sensor data, and real-time analytics.



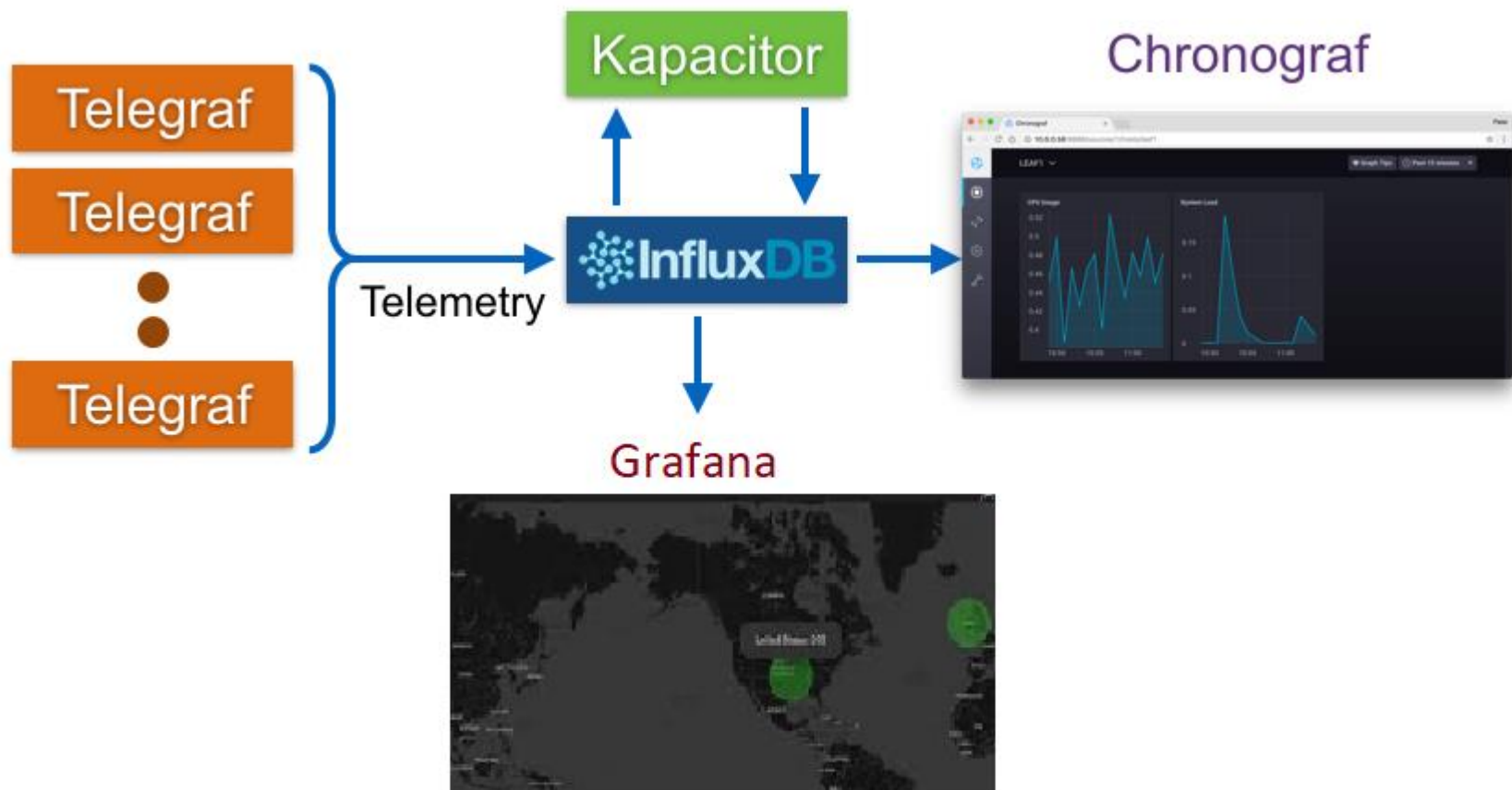


Downsampling y data retention

Downsampling: consiste en conservar los datos en bruto de alta precisión solo por tiempo limitado y almacenar la resumida, de baja precisión y/o aquella que realmente nos brinden la información que necesitamos.

Data retention: consiste en describir por cuanto tiempo se van a conservar los datos, otorgándoles un criterio para expirar.

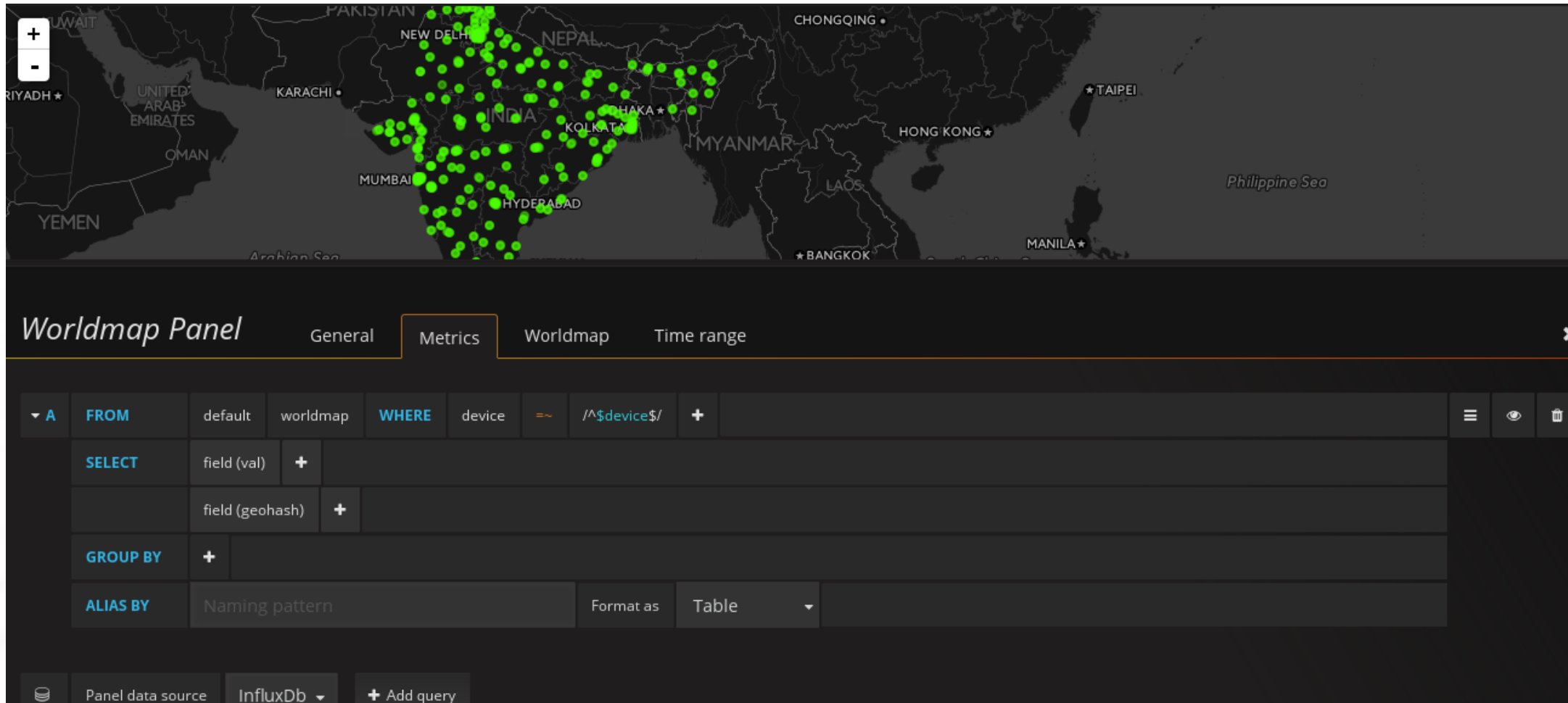
InfluxDB ofrece dos características – **Continuous Queries (CQ)** y **Retention Policies (RP)** – que automatizan el proceso de downsampling y data retention



Chronograf Dashboard



InfluxDB, Grafana y Mapas



Desplegar el TICK stack en OpenShift

SAN FRANCISCO, Oct. 17, 2017 /PRNewswire/ -- InfluxData, the modern Open Source Platform built from the ground up for metrics and events, today announced the availability of certified InfluxData container images on [Red Hat OpenShift Container Platform](#). Now developers and customers using Red Hat OpenShift Container Platform for container orchestration and management can more quickly deploy the InfluxData time series data platform in that environment. InfluxData tested and validated the interoperability of the platforms, giving open source developers and enterprise customers alike more confidence in a solution built on and certified with Red Hat's platform.

Note: *Running InfluxDB in Kubernetes in production is not recommended. See the [FAQ](#) for more info.*

OpenShift presente en proveedores de Cloud

IMPLEMENTACIÓN DE REFERENCIA

Red Hat OpenShift en AWS

Plataforma de aplicación de contenedor con organización de Kubernetes en la nube de AWS



Red Hat OpenShift on GCP

Red Hat

Container Platform for Developers

HACER UNA PRUEBA SIN CARGO [↗](#)



Microsoft Azure Red Hat OpenShift

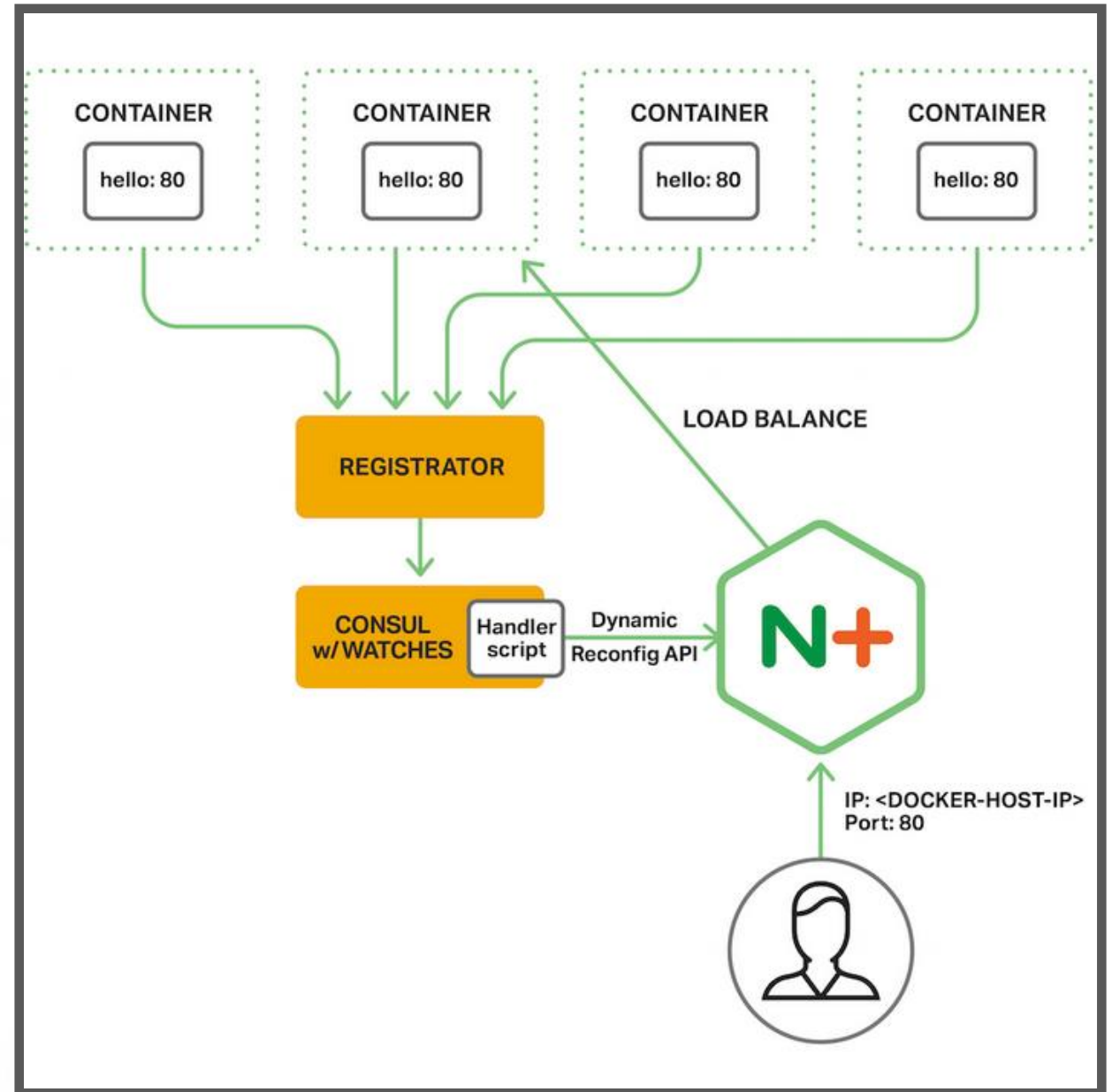
Fully managed Red Hat® OpenShift® service on Microsoft Azure

Contact Red Hat sales

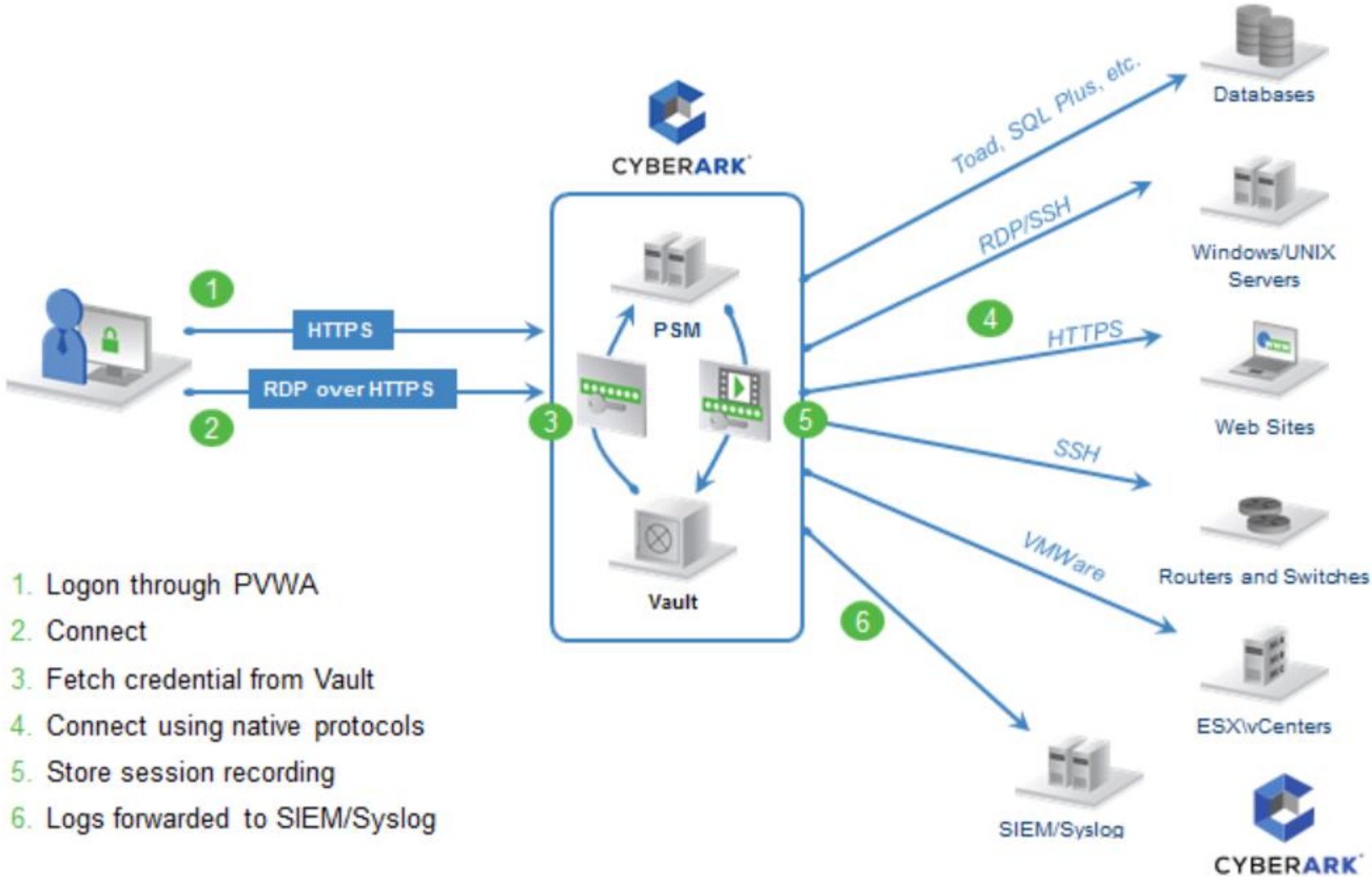
API Gateway y Service Discovery

Se tiene un contenedor separado de Docker para cada una de las siguientes aplicaciones:

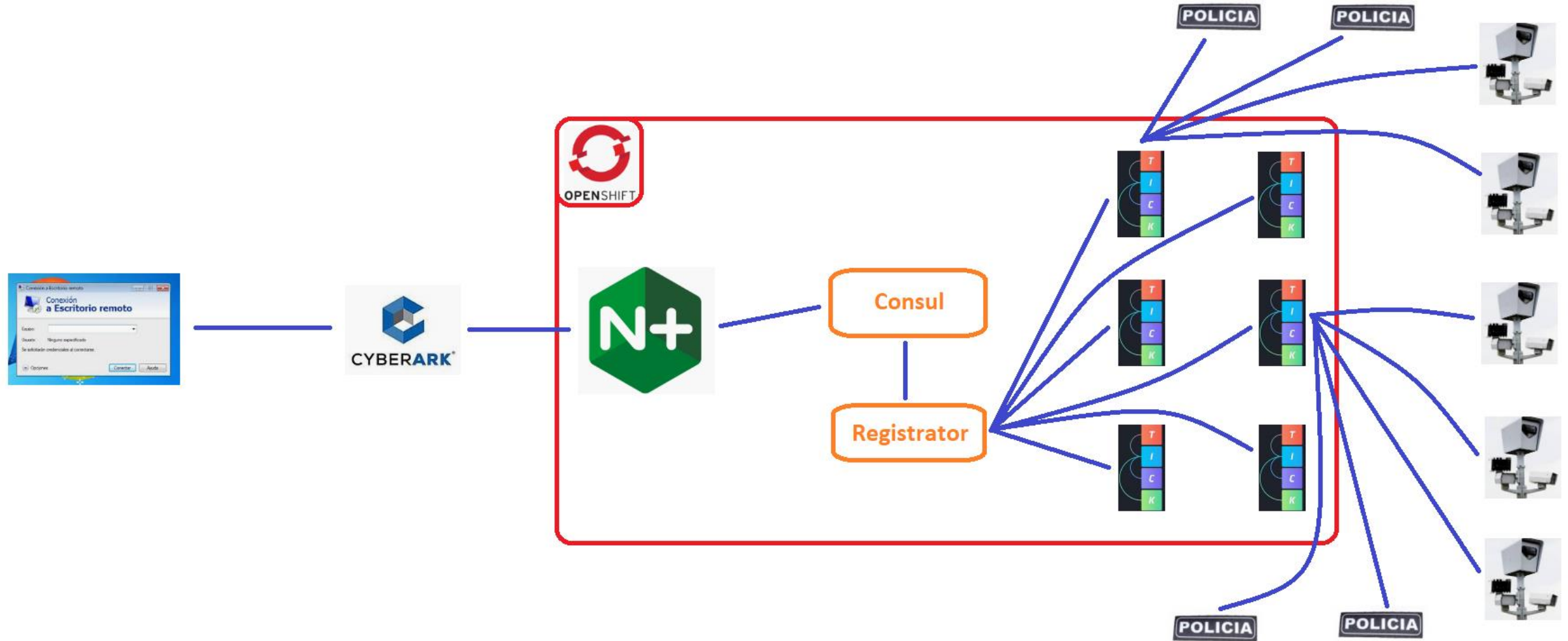
- Consul – realiza el service discovery
- Registrator – registra servicios con cónsul. Registrator monitorea los contenedores que están siendo iniciados y detenidos, y actualiza Consul cuando un contenedor cambia de estado
- NGINX Plus – API Gateway que sirve de balanceador de carga para las aplicaciones en los contenedores



Identify and Access management (IAM)



Visión Global de la Solución



Gracias !!!