

# Integrá un LLM local en Grafana y acelera tus KPIs sin exponer datos

## Introducción: una IA que te ayuda a ver más allá

Grafana es una de las herramientas más utilizadas para la visualización de datos en tiempo real. Seguramente te has cruzado con la tarea de interpretar correctamente un gráfico, especialmente si no es una tarea que hagas día a día. **Una LLM (Large Language Model) local puede hacer la diferencia al convertir información visual en explicaciones claras y útiles.**

Imaginá que tenes un dashboard con múltiples métricas y detectas una anomalía en un gráfico: una LLM **puede ayudarte a entender qué está ocurriendo y sugerir posibles causas o acciones.**

En esta guía te mostraremos **cómo integrar una LLM local en Grafana para que interprete gráficos en tiempo real** y ayude a los usuarios a tomar decisiones más informadas. Vamos a reemplazar las peticiones a la API de ChatGPT por un servicio local montado en ollama con la LLM de tu gusto favorito.

En este ejemplo vamos a usar Moondream, que con solo 1.8B es una muy básica y pequeña, pero es un placeholder que podés reemplazar según la potencia que tengas disponible.

## ¿Cómo funciona?

El proceso requiere ensamblar dos mundos: visualización y IA Generativa. Esto se basa en los siguientes pasos:

1. **Captura de datos del gráfico:** Se extraen los datos relevantes del panel de Grafana.

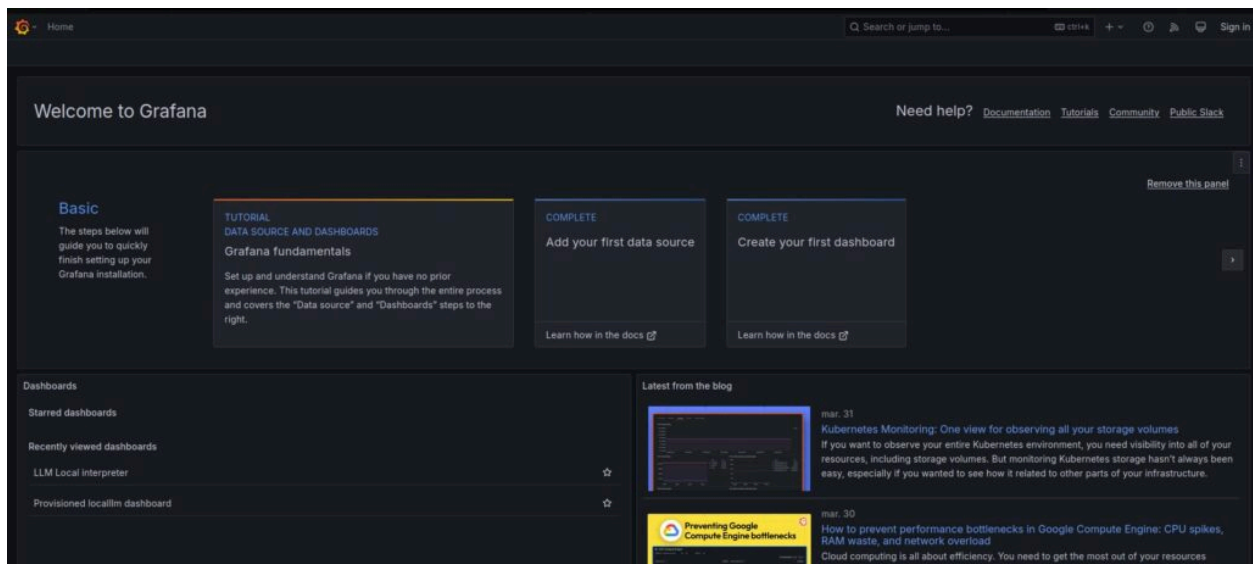
2. **Procesamiento por parte de la LLM:** La información se envía a un modelo de lenguaje local para su análisis.
3. **Generación de información:** La LLM devuelve un resumen/análisis/comparativa en lenguaje natural.
4. **Visualización en Grafana:** Los resultados generados se presentan junto al gráfico original.

## 1. Descargar el repo y la data

Primero, descargá el repo y entrá a la carpeta donde está el plugin que vamos a instalar.

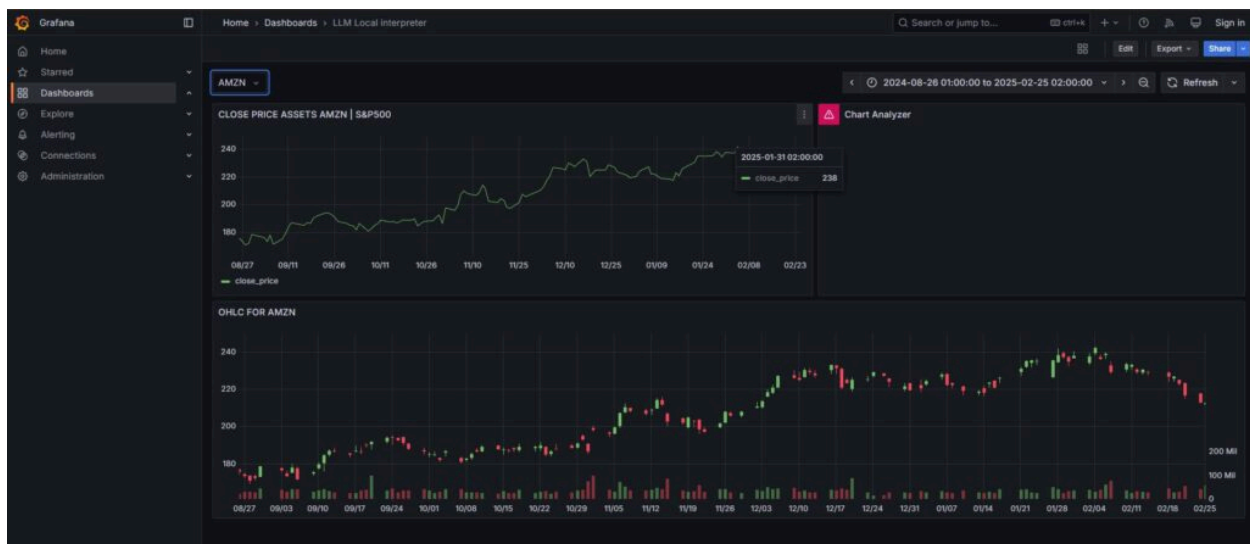
Luego, levantamos los contenedores de Grafana y de la base de datos que alberga los datos a gráficar:

Si vamos a localhost:3000, vamos a encontrar que Grafana está levantado.



Tenés dos formas distintas de traerte la data.

Con esto listo, vamos a ver que en Grafana ahora podemos acceder al dashboard que nos muestra todas las acciones del S&P500 en un rango de tiempo.



## 2. Instalar el Chart Analyzer

El Chart Analyzer todavía está vacío, porque tenemos que instalar el panel usando las dependencias del panel, y para eso necesitamos Node.js.

Node.js es un entorno que permite ejecutar código JavaScript fuera del navegador, en la computadora directamente. Aunque JavaScript nació para hacer páginas web interactivas, con Node.js ahora también se puede usar para crear programas, servidores, automatizaciones y herramientas de desarrollo. Node se ha vuelto muy popular porque es rápido, liviano y tiene una gran comunidad que comparte paquetes de código reutilizable a través de un sistema llamado NPM.

En este plugin para Grafana, usamos Node.js para manejar todo el entorno de desarrollo: instalar dependencias, ejecutar el plugin en modo prueba, revisar errores, y empaquetar el plugin listo para producción. No hace falta saber programar en profundidad para comenzar; simplemente se instalan las herramientas necesarias y se ejecutan comandos básicos

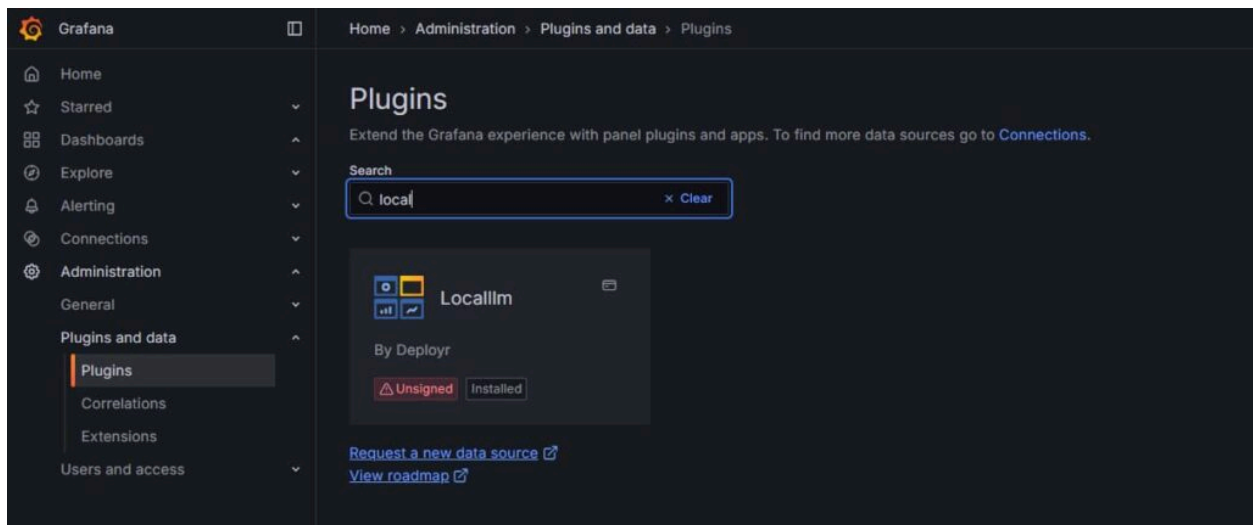
desde la terminal, como **npm install** o **npm run dev**, que hacen funcionar todo el entorno.

Hecha esta aclaración, vamos a instalar el panel.

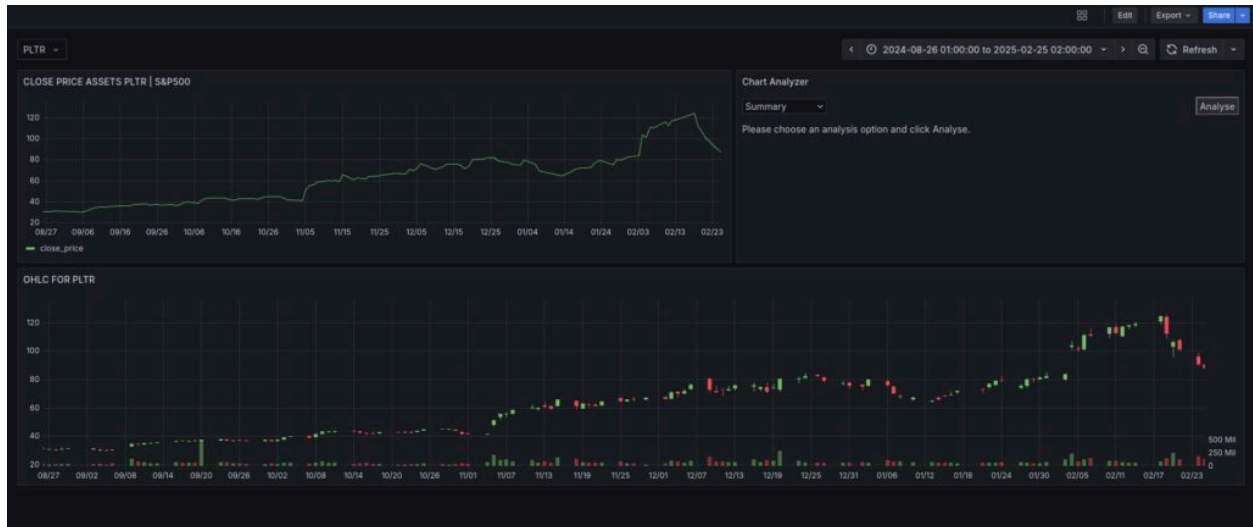
En la raíz del repo, vamos a instalar dependencias y correr el plugin.

Si este paso te da un error, es porque falta modificar los permisos de la carpeta, así que andá un paso para atrás y ejecutá esto.

Con esto listo, volvé a la carpeta e instalá todo. Ahora reiniciá los containers y vas a ver que en Grafana ya está instalado el panel.



Vas a notar que ahora el dashboard tiene habilitado el panel que acabamos de instalar, arriba a la derecha.

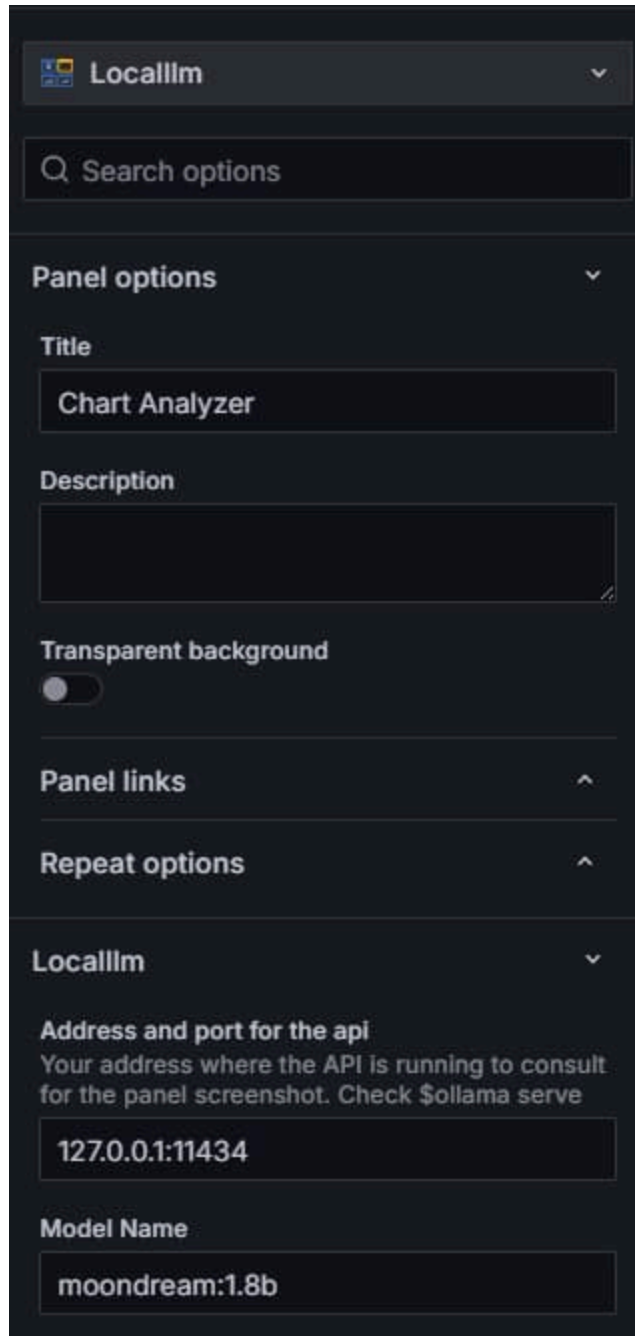


### 3. Correr el plugin

El plugin permite al usuario seleccionar distintos tipos de análisis según el gráfico mostrado. Algunas de las opciones incluyen:

- **Analizar de tendencias:** Detecta patrones crecientes o decrecientes en el tiempo.
- **Detectar anomalías:** Identifica picos o valores fuera de lo común.
- **Hacer una comparativa histórica:** Contrasta el gráfico actual con períodos previos similares.

Todas estas opciones se derivan cada una de un prompt a la LLM, que puedes editar a tu gusto en el código. Haciendo hover sobre el panel, en la parte superior derecha podés ver tres puntos. Al clickearlo, te permite editar también los parámetros del dashboard, como por ejemplo, la LLM a la que llamás.



The image shows a configuration panel for 'Localllm' in a dark-themed interface. At the top is a dropdown menu labeled 'Localllm'. Below it is a search bar with the placeholder text 'Search options'. The main section is titled 'Panel options' and contains several fields: 'Title' with the value 'Chart Analyzer', 'Description' (empty), and a 'Transparent background' toggle switch which is currently off. Below these are sections for 'Panel links' and 'Repeat options', both with upward-pointing arrows. The bottom section is also titled 'Localllm' and contains two more fields: 'Address and port for the api' with the value '127.0.0.1:11434' and 'Model Name' with the value 'moondream:1.8b'. A small note above the address field says 'Your address where the API is running to consult for the panel screenshot. Check \$ollama serve'.

Localllm

Search options

Panel options

Title

Chart Analyzer

Description

Transparent background

Panel links

Repeat options

Localllm

Address and port for the api

Your address where the API is running to consult for the panel screenshot. Check \$ollama serve

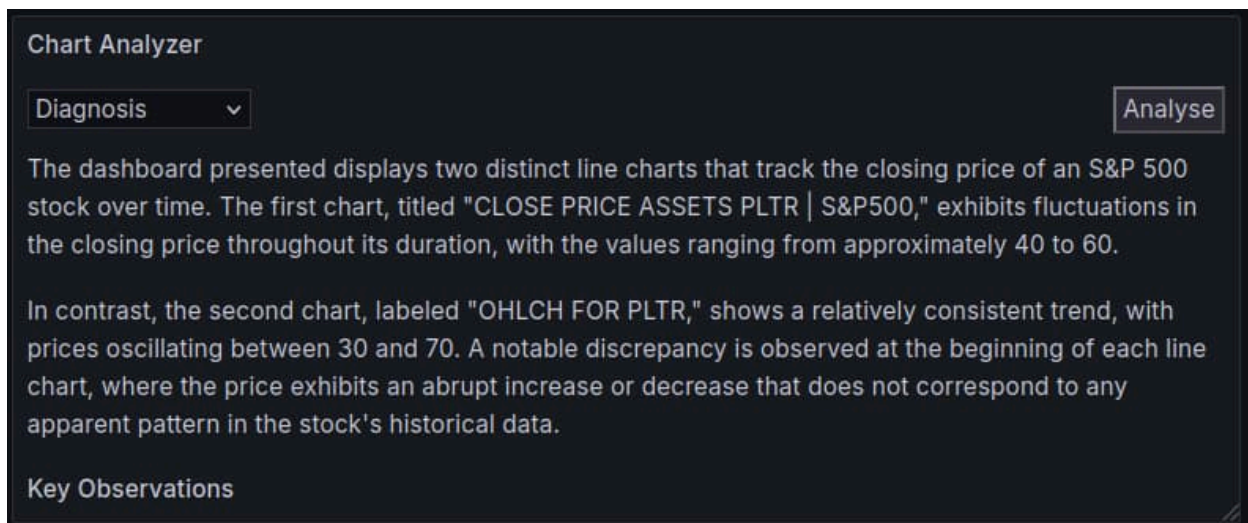
127.0.0.1:11434

Model Name

moondream:1.8b

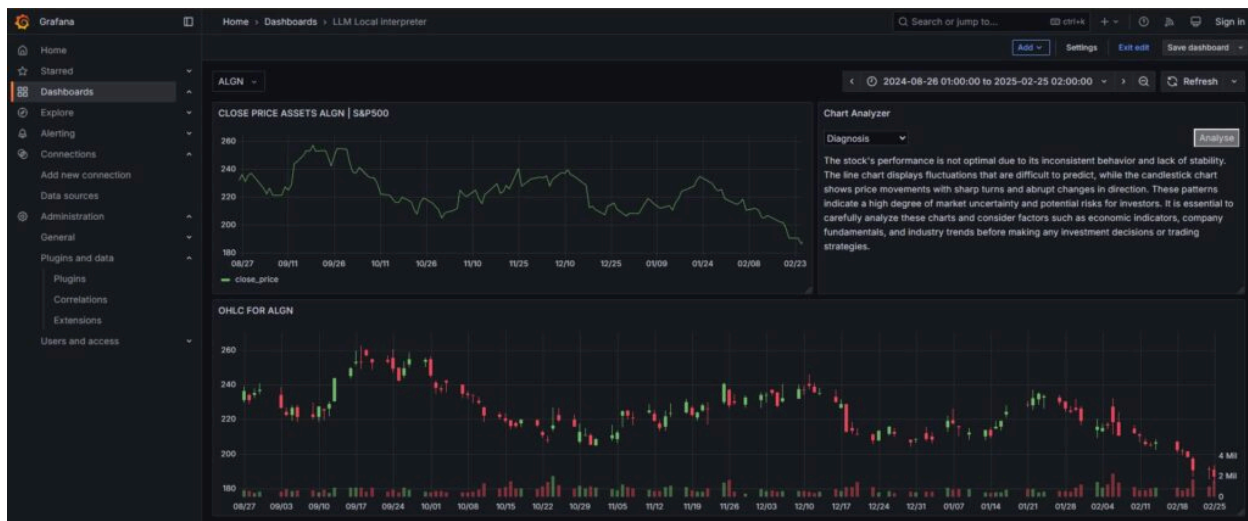
En este caso, el puerto es el que utiliza Ollama, la popular librería para deployar y servir LLMs de manera local. Si no lo tenés instalado, se baja en solo una línea de comando, y con otra más te traés el modelo que vamos a usar.

Con el modelo corriendo, estamos listos para decirle a la LLM que interprete lo que ve.



Este resultado que ves es una ejecución en la que consultamos a Llama-3.2-vision, que es bastante más potente que el buen Moondream 🤔.

Con Moondream como placeholder, la aplicación completa se debería ver así.



## Conclusión: Visualizar ya no es suficiente

Si bien Grafana ya permite visualizar datos de forma poderosa, **combinarlo con IA le da un nivel extra de inteligencia**. Implementarlo es un proceso accesible y altamente beneficioso, especialmente en entornos donde la rapidez en la toma de decisiones es clave.

Por otro lado, una implementación local da una capa extra de seguridad: te asegura que tus datos no estén por la red, y que la velocidad y la exactitud dependen del hardware que cuentes y de la LLM que quieras utilizar.

Empezá hoy mismo a mejorar tus dashboards con análisis automatizado y lleva la interpretación de tus datos al siguiente nivel. Si necesitás ayuda para incorporar estas herramientas en tu organización, no dudes en contactarnos.

¡Hasta la próxima