



# Gestión de Registros mediante Fluent-Bit con Docker

*Celia García Márquez*

*Administración de Sistemas Informáticos en Red*

*IES Gonzalo Nazareno*

# Contenido

Introducción

¿Qué es Fluent-bit?

Características de Fluent-bit

Conceptos Clave de Fluent-Bit

Canalización de Datos

Instalación de Fluent-bit

Calyptia Cloud

Escenario en Local

Escenario en Remoto



# INTRODUCCIÓN



**LA IMPORTANCIA DE GESTIONAR LOS  
REGISTROS**



# ¿Qué es Fluent-Bit?

*Recolector, procesador y reenviador de registros de código abierto.*

## Fluent-bit es capaz de:



*Recolectar registros de distintas fuentes*



*Procesar, filtrar y transformar registros*



*Enviar esos registros a diferentes destinos*



*Bajo consumo de recursos y alto rendimiento*

---

**Se propone como solución a unificar la gran cantidad de registros almacenados cada día más complejos.**

---

**Nace en el año 2014 bajo el paraguas de Fluentd desarrollado por la empresa Treasure Data.**

---

**Código Abierto  
Licencia Apache v.2.0  
Ampliamente adoptado por la industria**

# Características

- **Licencia Apache v2.0**
- **Open Source**
- **Alojado por CNCF**
- **Nivel de Producción**
- **Gran comunidad**
- **Adoptado por la industria**
- **Alto rendimiento**
- **Almacenamiento en Búfer, redundante a Fallos**

## Fluent-bit VS Fluentd

	<b>FLUENTD</b>	<b>FLUENT-BIT</b>
<b>Alcance</b>	Contenedores/ Servidores	Linux integrado/ Contenedores / Servidores
<b>Lenguaje</b>	<b>Escrito en C y Ruby</b>	<b>Escrito en C</b>
<b>Rendimiento</b>	Alto rendimiento	Alto rendimiento
<b>Dependencias</b>	<b>Al estar construido con Ruby necesita más dependencias</b>	<b>No necesita dependencias</b>
<b>Complementos</b>	<b>+1000</b>	<b>~=70</b>

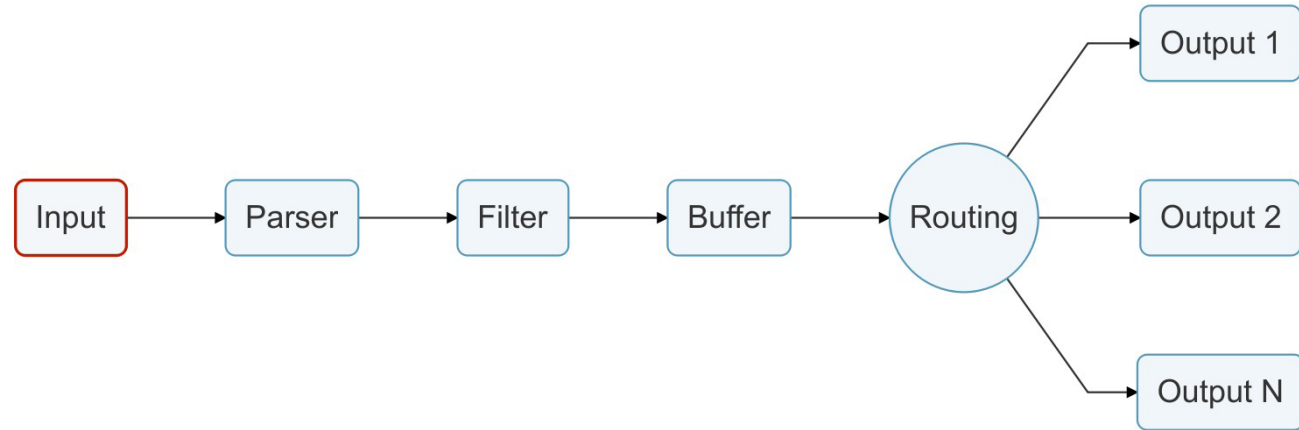
# Conceptos Clave



- **Event or Record**
- **Filtering**
- **Tag**
- **Timestamp**
- **Match**
- **Structured Message**

# Canalización de Datos

- **Input**
- **Parser**
- **Filter**
- **Buffer**
- **Router**
- **Output**



# INSTALACIÓN DE FLUENT-BIT

Si consultamos la documentación oficial de fluent-bit, podemos comprobar que hay varios caminos para ello:

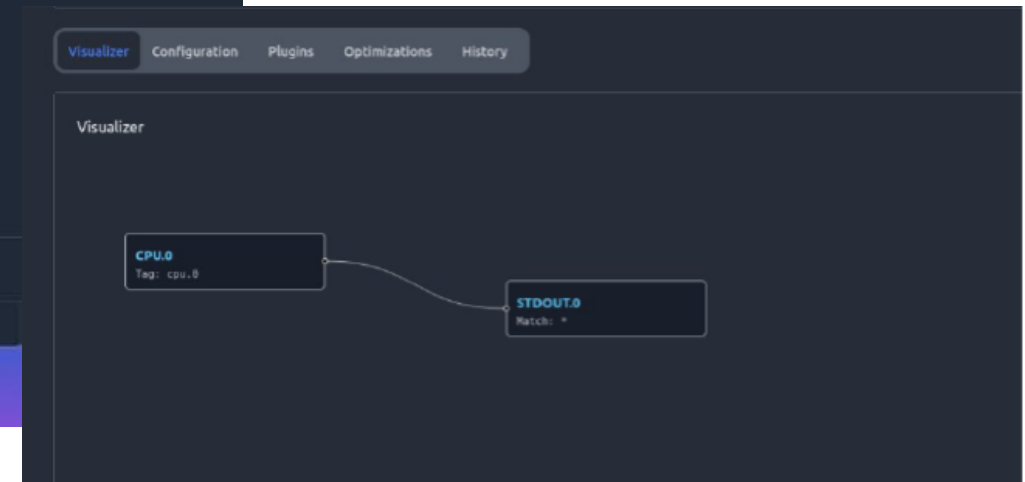
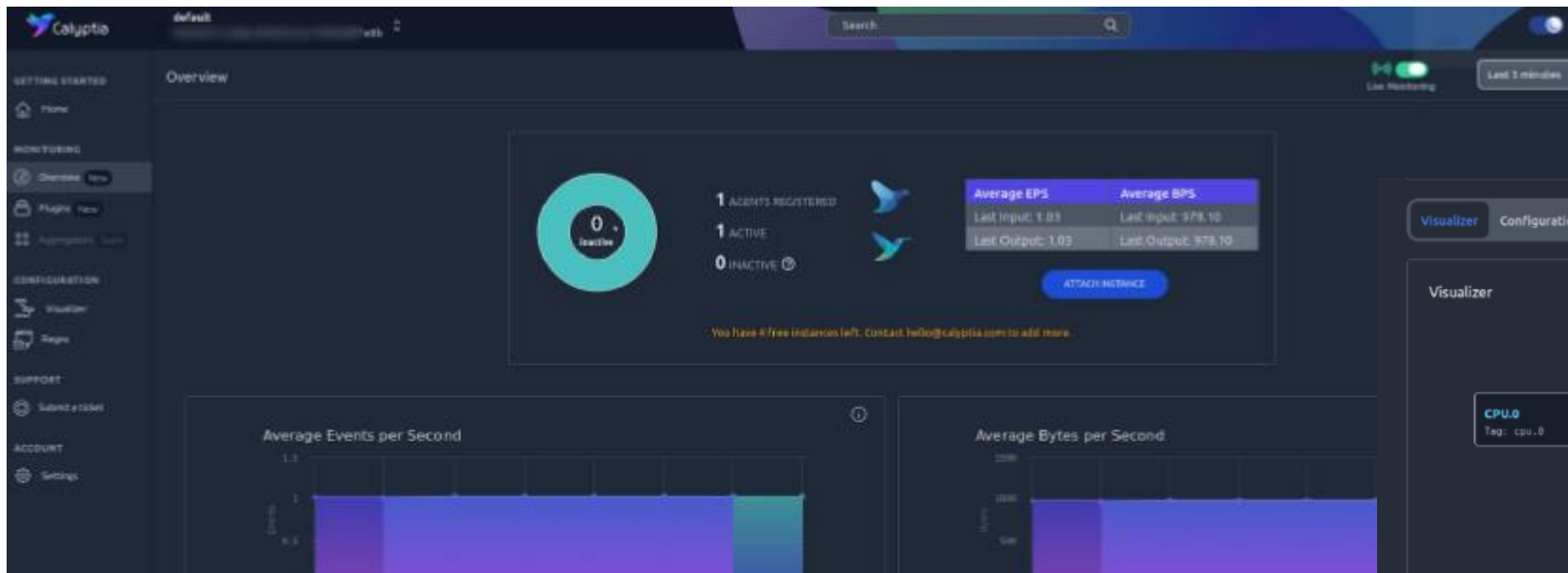
- **Desplegando contenedores** (Kubernetes, Docker y AWS)
- De forma **integrada** en Linux (Ubuntu, Debian, CentOS ...)
- Para sistemas operativos con **Windows** (Server 2019 y Win10)
- Compilando desde el **código fuente**

\* Linux -> Td-agent-bit





# CALYPTIA CLOUD



## Visualizar la canalización de datos

Calyptia Cloud permite registrar un agente en el servidor donde esté alojado Fluent-bit para analizar la canalización de datos y ver de forma gráfica la estructura.

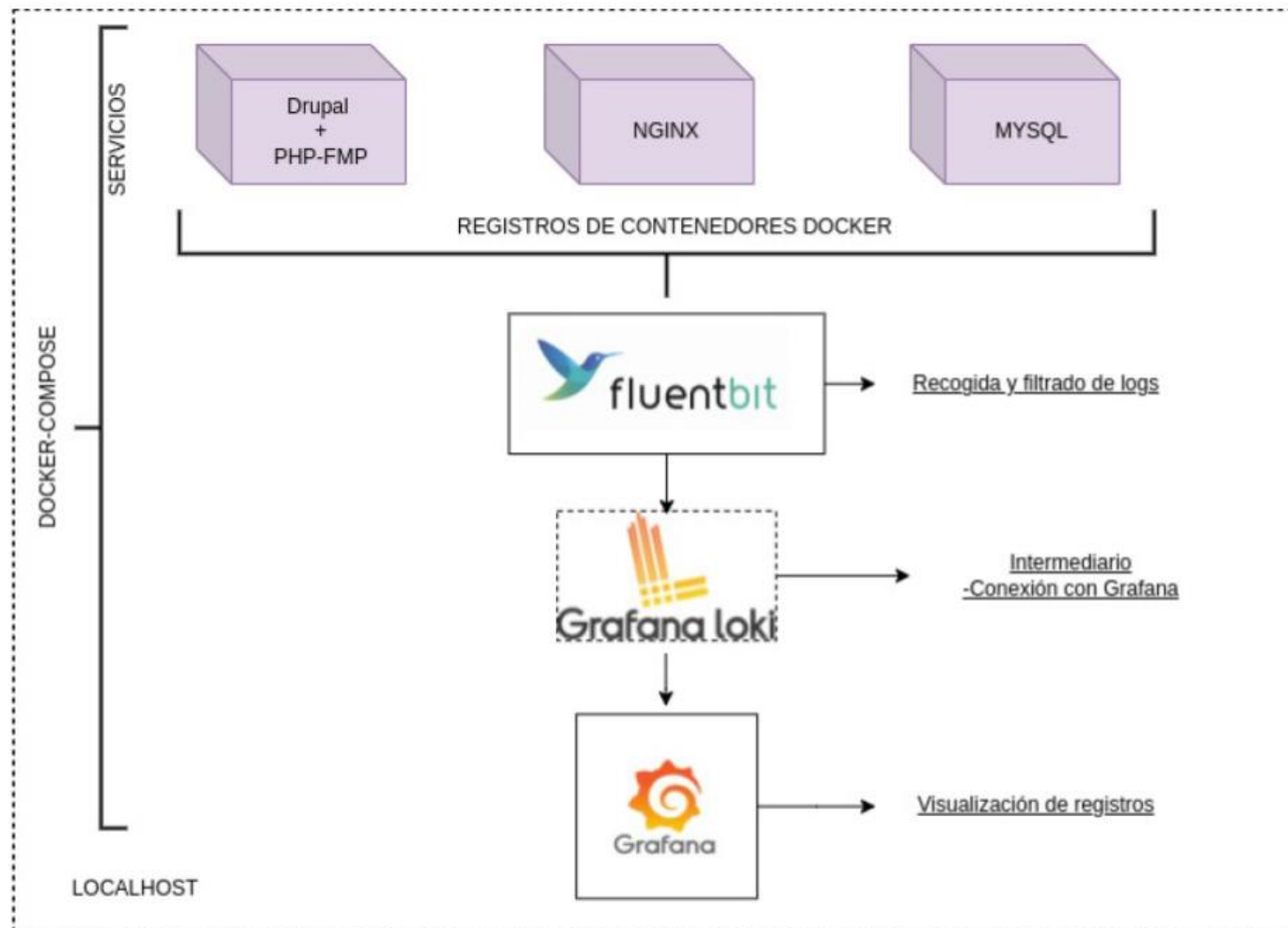
# Escenario en Local

El objetivo principal es **recolectar los registros** que generan los contenedores:

- Servidor Web
- Drupal + PHP-FPM
- Base de datos

Dotar de un filtro a esos registros y reenviarlos mediante **Fluent-Bit** a un intermediario (plugin de Grafana) llamado **Loki**, que permite la visualización de los registros en **Grafana**.

[GitHub - CeliaGMqz/fluent-bit-loki-grafana-docker-drupal](https://github.com/CeliaGMqz/fluent-bit-loki-grafana-docker-drupal)



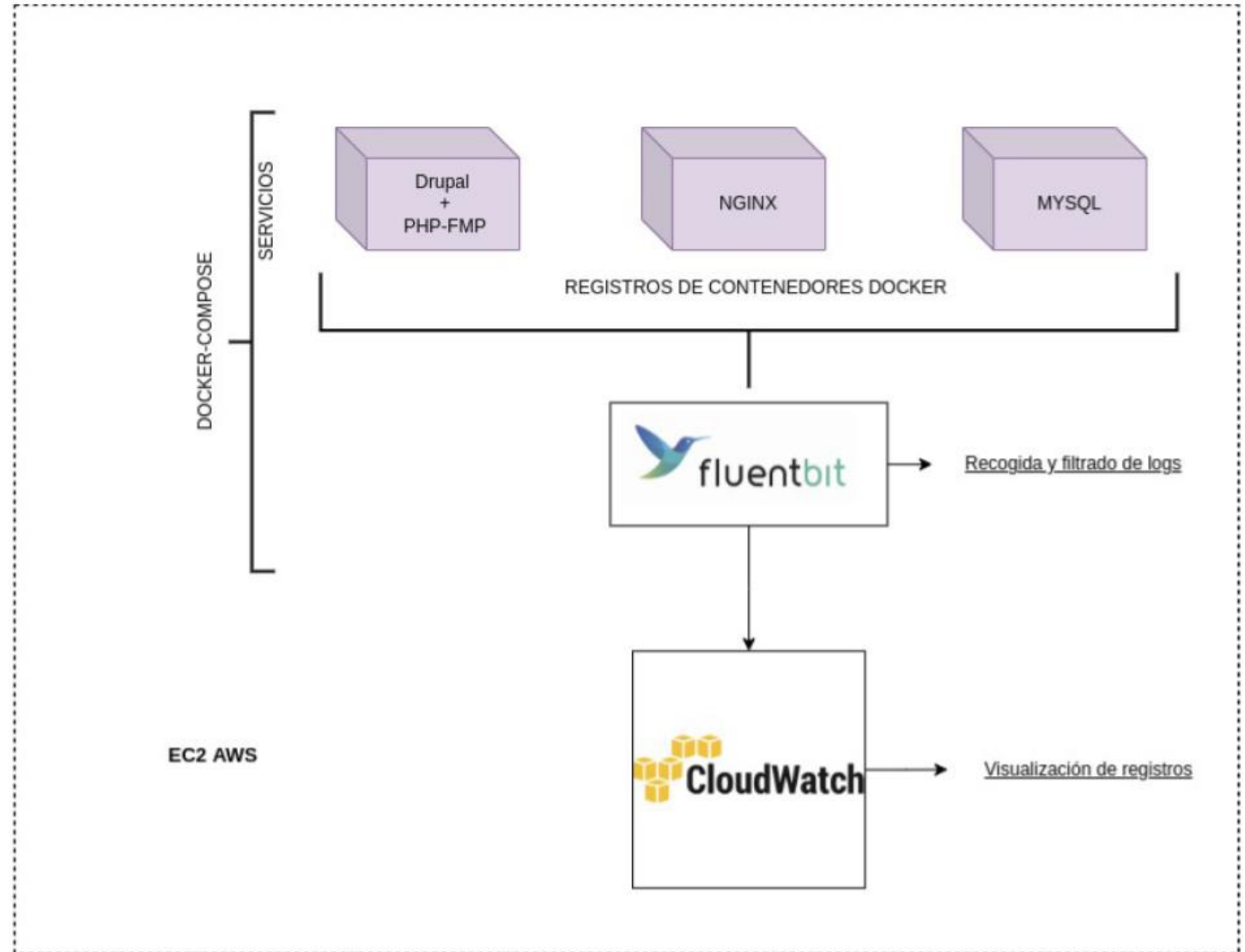
# Escenario en Remoto

El objetivo principal es **recolectar los registros** que generan los contenedores:

- Servidor Web
- Drupal + PHP-FPM
- Base de datos

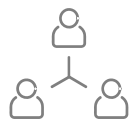
Dotar de un filtro a esos registros y reenviarlos mediante **Fluent-Bit** a una máquina **EC2** con un servicio de monitorización de Amazon llamado **CloudWatch**.

[GitHub - CeliaGMqz/ec2\\_fluent-bit\\_entorno\\_drupal](https://github.com/CeliaGMqz/ec2_fluent-bit_entorno_drupal)





fluentbit



# Gracias

Una vez explicado los dos entornos  
pasamos a la parte práctica del proyecto