

Laboratorio DevOps: Monitorización de WordPress con ELK y Grafana

Este documento describe cómo configurar un entorno de laboratorio automatizado utilizando Vagrant y Ansible para desplegar y monitorizar una aplicación WordPress en contenedores Docker mediante el stack ELK y Grafana.

Objetivo: Validar la recolección y visualización de logs de WordPress en Kibana y Grafana.

Tecnologías: Vagrant, Ansible, Docker, Docker Compose, Elasticsearch, Logstash, Kibana, Grafana, Rocky Linux 9, WordPress, MySQL.

1. Requisitos Previos

Asegúrate de tener instalado en tu máquina host:

1. **Vagrant:** [Descargar Vagrant](#)
2. **VirtualBox** (o VMWare/Hyper-V con el provider correspondiente. Yo usaría KVM pero lo que nos ha caído en suerte es esto. No es mal hipervisor pero ... es por mi ignorancia de como funcionan los sistemas Windows): [Descargar VirtualBox](#)

(OJO!!!! este documento está sistematizado por la IA de Gemini (maldita listilla ...) puede contener errores. Resumen: ficheros del repo BIEN, info de este documento DUDA).

2. Estructura del Proyecto

Crea la siguiente estructura de directorios y archivos en tu máquina local:

```
elk-grafana-wordpress-lab/
├── Vagrantfile
├── ansible/
│   ├── hosts                # Inventario Ansible
│   ├── administracion.yml    # Playbook para VM administracion
│   ├── recurso.yml           # Playbook para VM recurso
│   ├── seguimiento.yml       # Playbook para VM seguimiento
│   └── files/                # Directorio para archivos de configuración
│       ├── logstash.conf     # Configuración pipeline Logstash
│       ├── elasticsearch.yml  # Configuración Elasticsearch
│       ├── kibana.yml         # Configuración Kibana
│       ├── grafana-datasource.yml # Datasource Grafana para Elasticsearch
│       └── docker-compose.yml # Definición WordPress + MySQL
└── README.md                # (Este archivo)
```

3. Vagrantfile

Este archivo define las 3 máquinas virtuales, su red, recursos y el aprovisionamiento con Ansible Local.

Fichero en el repo

4. Aprovisionamiento Ansible

4.1. Archivo `hosts` de Ansible

Crea el archivo `ansible/hosts` con el inventario. `ansible_connection=local` es crucial para `ansible_local`. Es una mierda, en realidad, pero debe tirar (“funcionar” estimado Sergey) fichero en el repo

4.2. Playbook `administracion.yml`

Este playbook configura la VM de administración. Sintaxis antigua.

Fichero en el repo

4.3. Playbook `recurso.yml`

Este playbook instala Docker y despliega WordPress + MySQL en la VM de recursos.

Fichero en el repo

4.4. Playbook `seguimiento.yml`

Este playbook instala y configura ELK Stack y Grafana.

Fichero en el repo

5. Archivos de Configuración Adicionales

5.1. `ansible/files/docker-compose.yml`

Define los servicios de WordPress y MySQL. Configura el logging para enviar a Logstash vía syslog. Es duro instalar Wordpress pero la utilidad la conocéis todos y más simple que un chupete.

Fichero en el repo

5.2. `ansible/files/elasticsearch.yml`

Configuración mínima para Elasticsearch en un solo nodo. Se puede clusterizar (palabro) y no es complicado. Testead la clave Discovery.

Fichero en el repo

5.3. `ansible/files/logstash.conf`

Configuración del pipeline de Logstash para recibir logs de Docker (syslog) y enviarlos a Elasticsearch.

Fichero en el repo

5.4. ansible/files/kibana.yml

Configuración de Kibana para conectarse a Elasticsearch y ser accesible desde el host.

Personalmente nunca me he aclarado con Kibana. Siempre he visto capturas preciosas que nunca he obtenido en mis prototipos.

Fichero en el repo

5.5. ansible/files/grafana-datasource.yml

Aprovisiona automáticamente la fuente de datos de Elasticsearch en Grafana (maravillosa herramienta)

Fichero en el repo

6. Instrucciones Paso a Paso

1. **Clonar/Descargar:** Asegúrate de tener todos los archivos y directorios creados como se describe en la sección "Estructura del Proyecto".
2. **Abrir Terminal:** Navega hasta el directorio raíz del proyecto (elk-grafana-wordpress-lab/) en tu terminal o línea de comandos.
3. **Iniciar las VMs:** Ejecuta el comando `vagrant up`.

```
cd ruta/a/elk-grafana-wordpress-lab
vagrant up
```

Este proceso tardará varios minutos, ya que Vagrant descargará la imagen base (si no está cacheada), creará las VMs, instalará Ansible dentro de ellas y luego ejecutará los playbooks para instalar y configurar todo el software. Observa la salida en la terminal para detectar posibles errores.

4. **Esperar:** Sé paciente (me parto la "caja" ... menuda IA) . La instalación de ELK Stack, Docker y Grafana, junto con las actualizaciones del sistema, puede llevar tiempo, especialmente la primera vez.
-

7. Validación

Una vez que `vagrant up` termine sin errores:

1. **Acceder a WordPress:** Abre tu navegador web y ve a `http://localhost:8080`. Deberías ver la pantalla de instalación inicial de WordPress. Puedes completar la instalación si quieres generar más logs, o simplemente navegar por la página por defecto. Lo suyo es ir generando operaciones para que tengamos logs (registros) que puedan ser visualizados. Un par de minutos y ya tenemos métodos que puedan ser recuperados.
2. **Acceder a Kibana:** Ve a `http://localhost:5601`.
 - Es posible que Kibana te pida crear un "Data View" (anteriormente Index Pattern).
 - Ve a "Management" -> "Stack Management" -> "Data Views" (o busca "Data View" en la barra lateral).

- Haz clic en "Create data view".
- En el campo "Name", introduce un nombre como `WordPress Logs`.
- En el campo "Index pattern", escribe `wordpress-logs-*`. Debería detectar el índice creado por Logstash.
- Selecciona `@timestamp` como el campo de tiempo ("Timestamp field").
- Haz clic en "Create data view".
- Ahora, ve a la sección "Discover" (icono de brújula). Selecciona tu Data View `WordPress Logs` y deberías empezar a ver los logs de los contenedores `WordPress` que fueron enviados a Logstash. Genera actividad en `WordPress` (recarga la página, intenta acceder a `wp-admin`, etc.) para ver más logs.

3. **Acceder a Grafana:** Ve a `http://localhost:3000`.

- El usuario y contraseña por defecto son `admin / admin`. Te pedirá cambiar la contraseña en el primer login.
- Ve a "Connections" -> "Data sources" (o icono de engranaje -> Data Sources). Deberías ver la datasource "Elasticsearch-WordPress" ya configurada gracias al aprovisionamiento. Haz clic en ella y luego en "Save & test" para verificar la conexión.
- Ve a la sección "Explore" (icono de brújula).
- En el desplegable de la esquina superior izquierda, selecciona la datasource "Elasticsearch-WordPress".
- En el campo "Query", puedes simplemente dejarlo vacío o escribir una consulta Lucene básica (ej: `application:wordpress`).
- Asegúrate de que el selector de índice (generalmente justo debajo de la datasource) esté apuntando a `wordpress-logs-*`.
- Ajusta el rango de tiempo en la esquina superior derecha.
- Haz clic en "Run query". Deberías ver los logs en formato de tabla y un histograma de eventos a lo largo del tiempo.

4. **Verificar Resolución DNS Interna:** Puedes conectarte a una de las VMs usando `vagrant ssh <nombre_vm>` (e.g., `vagrant ssh administracion`) y probar a hacer ping a las otras VMs por nombre:

```
ping -c 3 recurso
ping -c 3 seguimiento
```

Deberían resolverse a sus IPs privadas (192.168.33.11 y 192.168.33.12 respectivamente).

NdA: Todo esto suponiendo que no se quede colgado VirtualBox, windows o ... vete a saber