

Conversación con Gemini

Exported: 15/6/2026 9:13:56

Jordi H.

Has dicho como puedo ver los logs de un container corriendo en podman

Gemini

Para ver los logs de un contenedor en Podman, el comando principal es prácticamente idéntico al de Docker. Aquí tienes las formas más comunes de hacerlo según lo que necesites

1. El comando básico

Para ver la salida actual del contenedor (lo que se ha impreso hasta ahora), utiliza: Bash

CODE

```
podman logs <nombre_o_id_del_contenedor>
```

2. Opciones avanzadas

Dependiendo de si estás depurando un error o simplemente monitoreando la actividad, estas banderas te serán muy útiles

- Seguimiento en tiempo real (-f o --follow): Mantiene la conexión abierta y muestra las nuevas líneas de log conforme aparecen. Bas

CODE

```
podman logs -f <nombre_del_contenedor>
```

- Ver las últimas líneas (--tail): Si el contenedor lleva mucho tiempo corriendo, los logs pueden ser enormes. Puedes limitar la salida a las últimas \$N\$ líneas. Bas

CODE

```
podman logs --tail 20 <nombre_del_contenedor>
```

- Añadir marcas de tiempo (-t o --timestamps): Útil si tu aplicación no imprime la hora por defecto. Bash

CODE

```
podman logs -t <nombre_del_contenedor>
```

- Filtrar por tiempo (--since): Para buscar eventos específicos ocurridos hace poco tiempo. Bash

CODE

```
podman logs --since 10m <nombre_del_contenedor> # Logs de los últimos 10 minutos
```

3. Diferencia clave: El Driver de Logs

A diferencia de Docker, Podman suele usar por defecto el driver de **journald** en muchas distribuciones de Linux (especialmente en RHEL, Fedora o CentOS). Si podman logs no te muestra lo que esperas, puedes consultar los logs directamente desde el sistema con journalctl:Ba

CODE

```
journalctl _COMM=podman
```

O de forma más específica para un contenedor: Bash

CODE

```
journalctl CONTAINER_ID_FULL=<id_del_contenedor>
```

Tip Pro: Como estás enfocado en **Python y Analítica de Datos**, recuerda que si corres scripts de Python en contenedores, a veces los logs no aparecen instantáneamente debido al búfer de salida. Para forzar a Python a que envíe los logs al terminal de inmediato, puedes usar la variable de entorno PYTHONUNBUFFERED=1 al lanzar tu contenedor. ¿Te gustaría saber cómo guardar esos logs directamente en un archivo .log para analizarlos luego con Pyt
