

# DIAGNÓSTICO EN TIEMPO REAL Y RETO GRUPAL

MONITOREO, REGISTROS Y USO DE `sshfs`

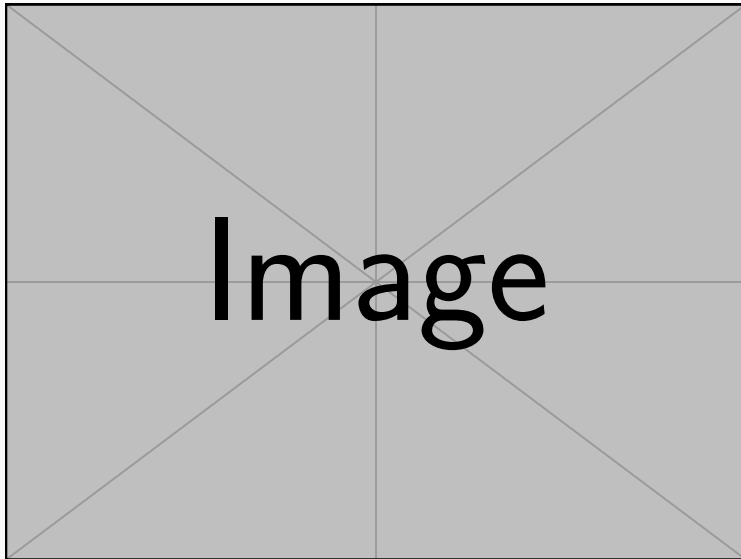
---

Equipo docente ClústerLab

8 de agosto de 2025

```
1 $ echo 'pi@head:/home/shared /home/cluster/shared fuse.sshfs defaults,_netdev 0 0' |  
  ↪ sudo tee -a /etc/fstab
```

Requiere claves SSH configuradas y el paquete **sshfs** instalado en el arranque.



```
1 $ sudo journalctl -u ssh
2 $ sudo journalctl -f
```

- Usa `-u` o `-b` para filtrar servicios o arranques.
- Con `-f` observas mensajes en tiempo real.

- Utilizar herramientas de monitoreo (`htop`, `df`, `iotop`, `watch`).
- Crear un script de salud que registre el estado del sistema cada minuto.
- Montar un directorio remoto mediante `sshfs`.
- Acceder a nodos por nombre de host usando mDNS/Avahi.

1. Monitorizar CPU, memoria, disco y sensores en tiempo real.
2. Desarrollar y ejecutar `healthcheck.sh`.
3. Instalar `sshfs` y montar `/home/shared`.
4. Configurar resolución de nombres vía Avahi (mDNS).

```
1 $ htop          # procesos y CPU en tiempo real
2 $ df -h         # uso de disco legible
3 $ iotop         # I/O por proceso (necesita root)
4 $ watch -n1 sensors # temperatura y voltajes cada segundo
```



### Nota

Ajusta el intervalo de **watch** según tus necesidades (**-n2** para cada 2 s, etc.). **iotop** requiere privilegios y puede instalarse vía **sudo apt install iotop**.

## SCRIPT healthcheck.sh

```
1  #!/usr/bin/env bash
2  LOG=/var/log/health.log
3  while true; do
4      timestamp=$(date "+%F %T")
5      load=$(uptime | awk -F'load average: ' '{print $2}')
6      temp=$(vcgencmd measure_temp | cut -d= -f2)
7      echo "$timestamp | load: $load | temp: $temp" » "$LOG"
8      sleep 60
9  done
```



### Advertencia

Asegúrate de que `healthcheck.sh` tenga permisos de ejecución (`chmod +x healthcheck.sh`). Revisa el log periódicamente con `tail -f`.

```
1 $ sudo apt install -y sshfs
2 $ mkdir -p ~/shared
3 $ sshfs pi@head:/home/shared ~/shared
4 # Trabaja como si fuera local...
5 $ fusermount -u ~/shared # para desmontar
```

### Nota

**sshfs** utiliza el mismo mecanismo de autenticación que **ssh**. Asegúrate de tener configurado tu par de claves y permisos adecuados en el directorio remoto.

Instala y habilita Avahi para resolución mDNS:

```
1 $ sudo apt install -y avahi-daemon libnss-mdns
2 $ sudo systemctl enable --now avahi-daemon
```

- Con mDNS activo, puedes conectarte usando `ssh pi@node03.local` en lugar de la IP.
- Asegúrate de que todos los nodos tengan nombres únicos y Avahi habilitado.

1. Ejecuta las herramientas de monitoreo y comparte capturas de tu Pi bajo carga.
2. Lanza `healthcheck.sh` y revisa el log después de unos minutos.
3. Consulta los últimos eventos con `journalctl -u ssh`.
4. Monta el directorio compartido con `sshfs` y crea un archivo de prueba.
5. Conéctate a otro nodo usando su nombre (`ssh pi@node02.local`) y verifica la presencia del archivo.

- Practicaste el uso de herramientas de diagnóstico en tiempo real.
- Automatizaste la recolección de métricas con un script de salud.
- Compartiste directorios mediante `sshfs` y configuraste nombres de host.

Con esto tu clúster está listo para tareas de cómputo distribuido.