🔴 DÍA 10: IMPLEMENTACIÓN DE SERVICIOS WEB Y FTP EN LINUX SERVER 🌍📡

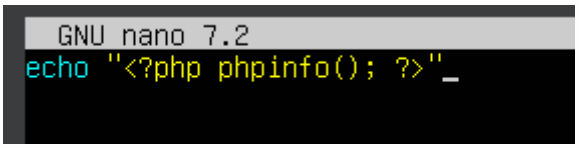
Abraham Caamaño Martínez

🚀 Fase 1: Instalación y configuración del servidor web

✅ Instalar Apache o Nginx y configurar el sitio por defecto en /var/www/html.

✅ Crear una página de prueba en HTML/PHP que muestre información del servidor (phpinfo();).

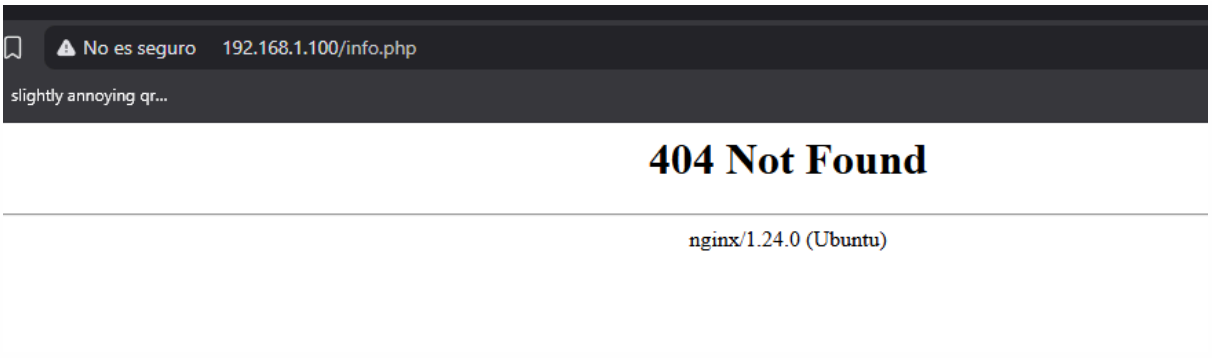
Creamos el archivo /var/www/html/info.php




```
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }

        # pass PHP scripts to FastCGI server
        #
        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
                fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #       deny all;
        #}
}


# Virtual Host configuration for example.com
```

La comprobacion:

```
vboxuser@UbuntuServer:~$
vboxuser@UbuntuServer:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
vboxuser@UbuntuServer:~$
```



✅ Configurar un vhost para alojar múltiples proyectos en el mismo servidor.

Creamos el directorio para el proyecto y asignamos permisos:

```
vboxuser@UbuntuServer:~$ sudo mkdir -p /var/www/proyecto-alpha
vboxuser@UbuntuServer:~$ sudo chown -R vboxuser:vboxuser /var/www/proyecto-alpha
vboxuser@UbuntuServer:~$
```

Creamos un archivo de configuracion:

```
  GNU nano 7.2
server {
    listen 80;
    server_name proyecto-alpha.codearts;

    root /var/www/proyecto-alpha;
    index index.html index.htm;

    location / {
        try_iles $uri $uri/ =404;
    }
}
```

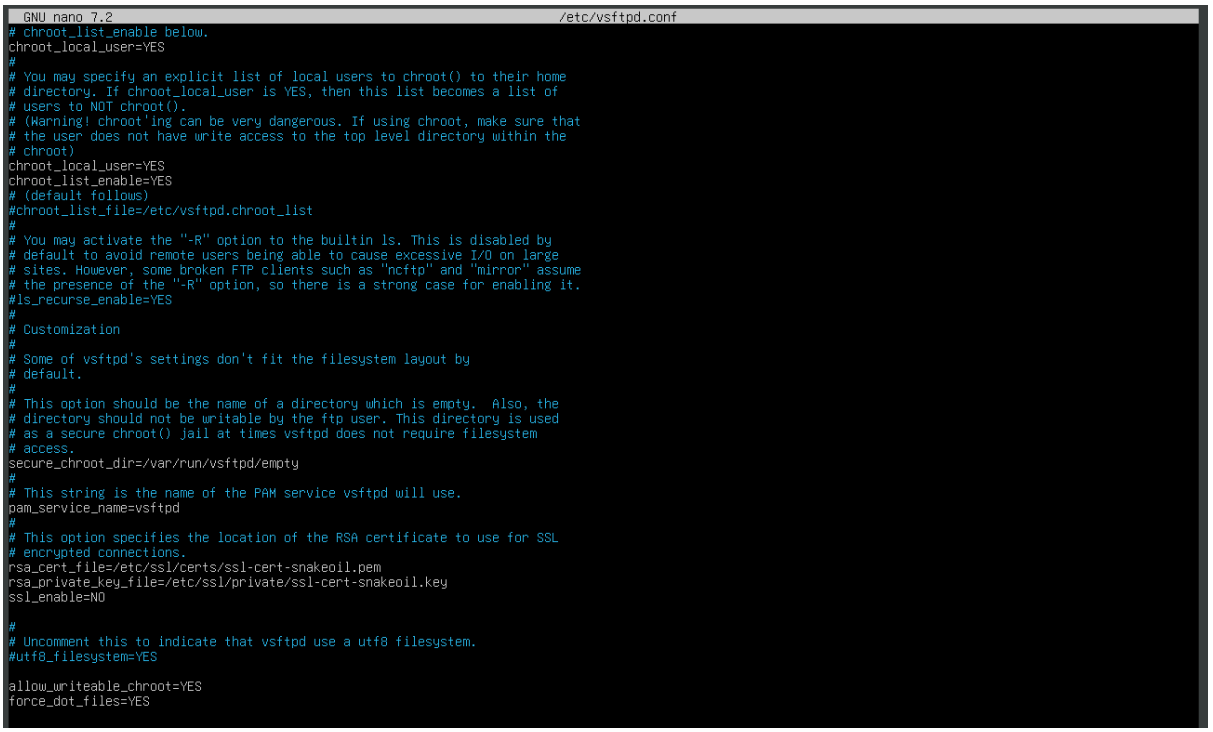
```
vboxuser@UbuntuServer:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
vboxuser@UbuntuServer:~$ sudo systemctl reload nginx
vboxuser@UbuntuServer:~$
```

🚀 Fase 2: Instalación y configuración del servidor FTP

✅ Instalar vsftpd o ProFTPD en el servidor.

He instalado vsftpd

Editamos el archivo de configuración:



✅ Crear tres usuarios FTP con acceso restringido a directorios específicos.

✅ Configurar permisos adecuados para garantizar la seguridad en las transferencias de archivos.

```
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
info: Adding new user `dev1' to supplemental / extra groups `users' ...
info: Adding user `dev1' to group `users' ...
vboxuser@UbuntuServer:~$ sudo adduser dev2 --shell /usr/sbin/nologin
info: Adding user `dev2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `dev2' (1016) ...
info: Adding new user `dev2' (1016) with group `dev2 (1016)' ...
info: Creating home directory `/home/dev2' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for dev2
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
info: Adding new user `dev2' to supplemental / extra groups `users' ...
info: Adding user `dev2' to group `users' ...
vboxuser@UbuntuServer:~$ sudo adduser dev3 --shell /usr/sbin/nologin
info: Adding user `dev3' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `dev3' (1017) ...
info: Adding new user `dev3' (1017) with group `dev3 (1017)' ...
info: Creating home directory `/home/dev3' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for dev3
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
info: Adding new user `dev3' to supplemental / extra groups `users' ...
info: Adding user `dev3' to group `users' ...
```

🚀 Fase 3: Seguridad y monitoreo de accesos

✅ Configurar firewall (ufw o iptables) para permitir solo conexiones desde la red interna.

```
vboxuser@UbuntuServer:~$ sudo ufw allow from 192.168.1.100/24 to any port 80 proto tcp
WARN: Rule changed after normalization
Skipping adding existing rule
vboxuser@UbuntuServer:~$ sudo ufw allow from 192.168.1.100/24 to any port 443 proto tcp
WARN: Rule changed after normalization
Rule added
vboxuser@UbuntuServer:~$ sudo ufw allow from 192.168.1.100/24 to any port 21 proto tcp
WARN: Rule changed after normalization
Rule added
vboxuser@UbuntuServer:~$ sudo ufw allow from 192.168.1.100/24 to any port 20 proto tcp
WARN: Rule changed after normalization
Rule added
```

```
vboxuser@UbuntuServer:~$ sudo ufw allow from 192.168.1.100/24 to any port 22 proto tcp
WARN: Rule changed after normalization
Rule added
```

```
vboxuser@UbuntuServer:~$ sudo ufw enable
Firewall is active and enabled on system startup
vboxuser@UbuntuServer:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
2222/tcp                   ALLOW IN    Anywhere
21/tcp                     ALLOW IN    192.168.1.100
40000:50000/tcp            ALLOW IN    192.168.1.0/24
22/tcp                     ALLOW IN    192.168.1.0/24
80/tcp                     ALLOW IN    192.168.1.0/24
443/tcp                    ALLOW IN    192.168.1.0/24
21/tcp                     ALLOW IN    192.168.1.0/24
20/tcp                     ALLOW IN    192.168.1.0/24
80/tcp (v6)                ALLOW IN    Anywhere (v6)
2222/tcp (v6)              ALLOW IN    Anywhere (v6)

vboxuser@UbuntuServer:~$
```

✅ Implementar logs para detectar accesos sospechosos y revisar /var/log/apache2/access.log.

```
vboxuser@UbuntuServer:~$ sudo tail -f /var/log/nginx/access.log
192.168.1.36 - - [26/Jun/2025:07:48:36 +0000] "GET / HTTP/1.1" 200 68 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
me/137.0.0.0 Safari/537.36"
192.168.1.36 - - [26/Jun/2025:07:48:36 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://192.168.1.100/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebK
/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
192.168.1.36 - - [26/Jun/2025:07:48:57 +0000] "GET /info.php HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
cko) Chrome/137.0.0.0 Safari/537.36"
^C
```

✅ Bloquear intentos de acceso no autorizados con reglas de firewall y fail2ban.

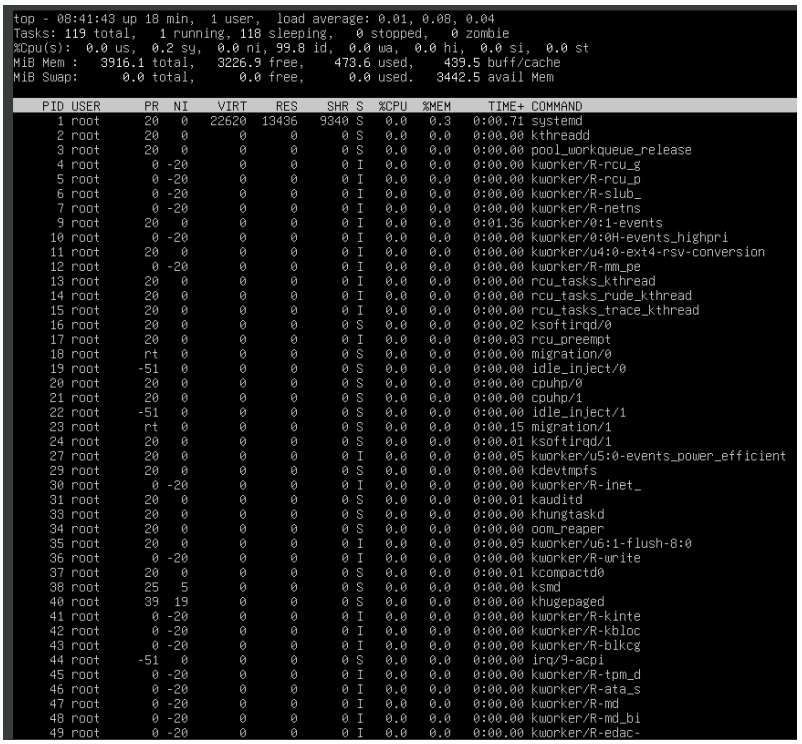
```
#
# Provide customizations in a jail.local file or a jail.
# For example to change the default bantime for all jail
# ssh-iptables jail the following (uncommented) would ap
# See man 5 jail.conf for details.
#
# [DEFAULT]
# bantime = 1h
#
# [sshd]
enabled = true

# See jail.conf(5) man page for more information



# Comments: use '#' for comment lines and ';' (following
```

```
vboxuser@UbuntuServer:~$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
vboxuser@UbuntuServer:~$
```

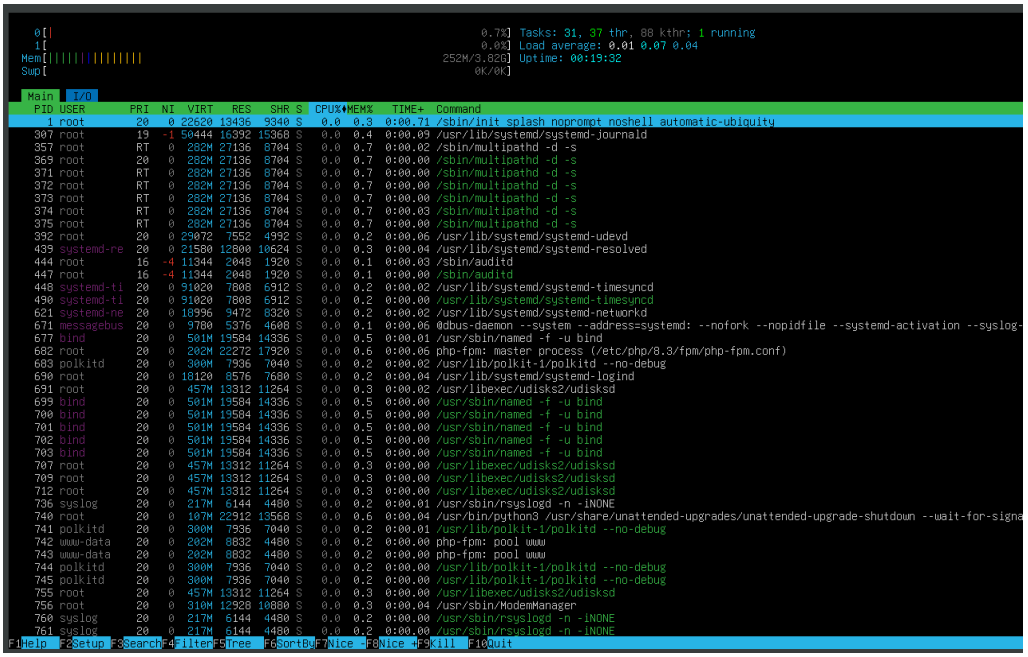
🚀 Fase 1: Análisis en tiempo real del sistema

✅ Ejecutar y analizar las siguientes herramientas:

Top:



Htop:



Uptime y Free:

```
vboxuser@UbuntuServer:~$ uptime
 08:41:29 up 18 min,  1 user,  load average: 0.02, 0.09, 0.05
vboxuser@UbuntuServer:~$ free -m
               total        used        free      shared  buff/cache   available
Mem:            3916         473        3226           3         439        3442
Swap:              0           0           0
vboxuser@UbuntuServer:~$
```

🚀 Fase 2: Gestión activa de procesos y prioridades

✅ Finalizar un proceso inactivo o que no sea esencial (kill, killall o pkill).

```
vboxuser@UbuntuServer:~$ sleep 600 &
[1] 1061
vboxuser@UbuntuServer:~$ pgrep sleep
1061
vboxuser@UbuntuServer:~$ kill 1601
-bash: kill: (1601) - No such process
vboxuser@UbuntuServer:~$ kill 1061
vboxuser@UbuntuServer:~$ _
```

✅ Cambiar la prioridad de un proceso en ejecución con renice.

```
[1]+  Terminated              sleep 600
vboxuser@UbuntuServer:~$ sleep 600 &
[1] 1072
vboxuser@UbuntuServer:~$ pgrep sleep
1072
vboxuser@UbuntuServer:~$ renice 10 1072
1072 (process ID) old priority 0, new priority 10
vboxuser@UbuntuServer:~$
```

✅ Lanzar un proceso en segundo plano (&) y enviarlo al primer plano con fg.

```
vboxuser@UbuntuServer:~$ jobs
[1]+  Running                 sleep 600 &
vboxuser@UbuntuServer:~$ fg %1
sleep 600
^Z
[1]+  Stopped                 sleep 600
vboxuser@UbuntuServer:~$ bg
[1]+ sleep 600 &
vboxuser@UbuntuServer:~$
```

✅ **Usar nice para iniciar un proceso con prioridad baja (por ejemplo, una copia pesada con cp).**

```
[1]+  Terminated              sleep 600
vboxuser@UbuntuServer:~$ sleep 600 &
[1] 1072
vboxuser@UbuntuServer:~$ pgrep sleep
1072
vboxuser@UbuntuServer:~$ renice 10 1072
1072 (process ID) old priority 0, new priority 10
vboxuser@UbuntuServer:~$ nice -n 15 cp /var/log/syslog ./backup_syslog_lento.log
cp: cannot open '/var/log/syslog' for reading: Permission denied
vboxuser@UbuntuServer:~$ sudo nice -n 15 cp /var/log/syslog ./backup_syslog_lento.log
[sudo] password for vboxuser:
vboxuser@UbuntuServer:~$
```

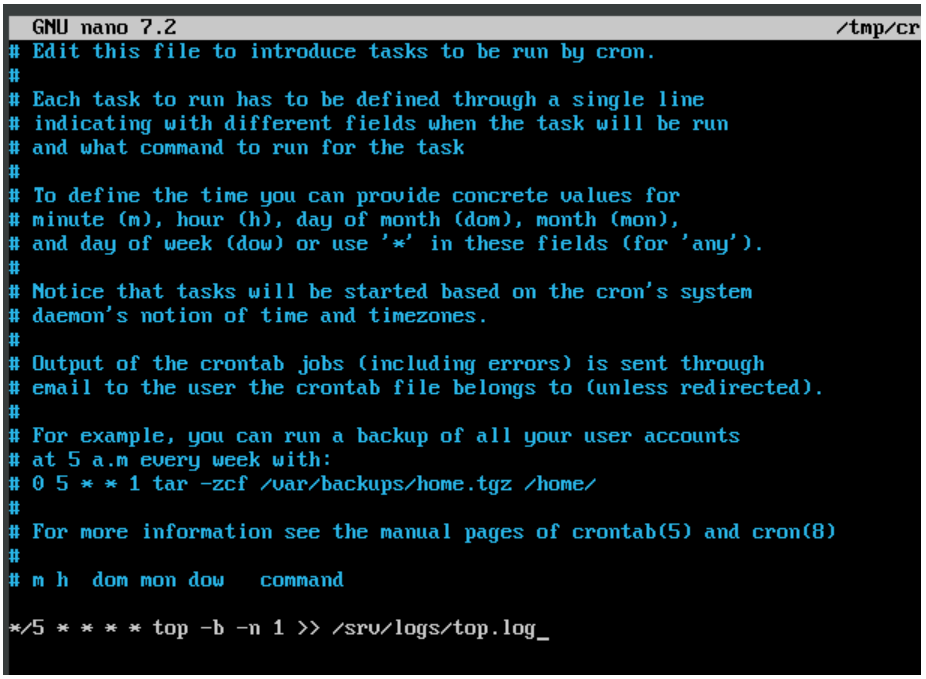
🚀 Fase 3: Monitorización y registro del uso de recursos

✅ **Usar el comando vmstat y guardar su salida en un archivo /srv/logs/vmstat.log.**

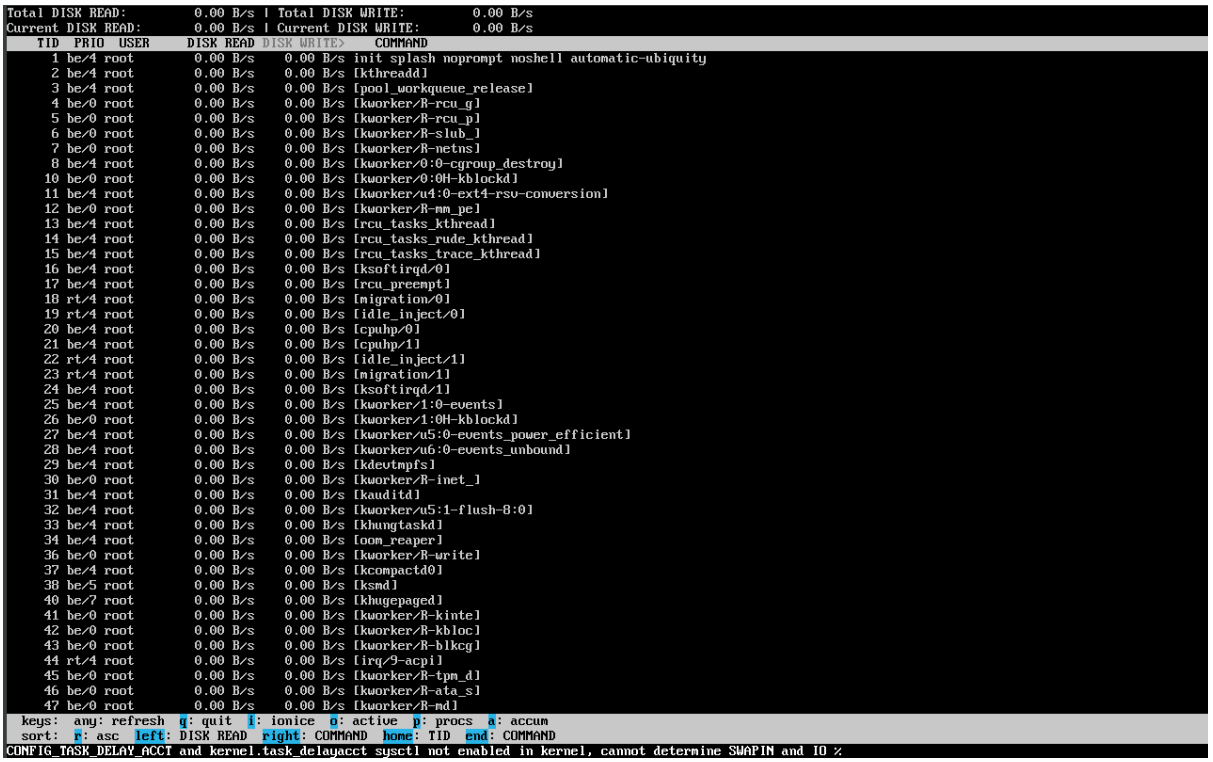
He creado la ruta /srv/logs

He ejecutado vmstat 10 > /srv/logs/vmstat.log & para que guarde datos cada 10 segundos en el archivo de log.

✅ **Configurar una tarea en crontab que guarde el uso de recursos (top -b -n 1) cada 5 minutos en /srv/logs/top.log.**



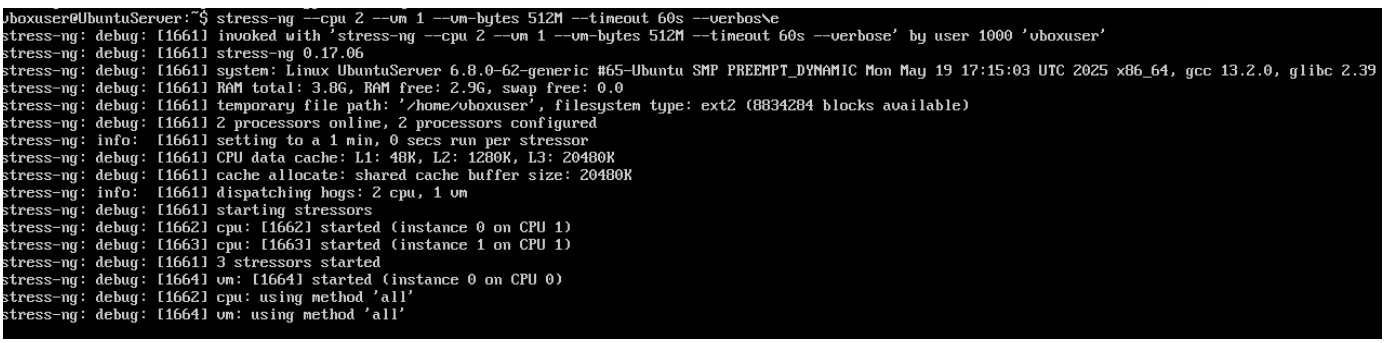
✅ Explorar iotop (si el sistema lo permite) para monitorizar I/O de disco.



🚀 Fase 4: Simulación de sobrecarga controlada

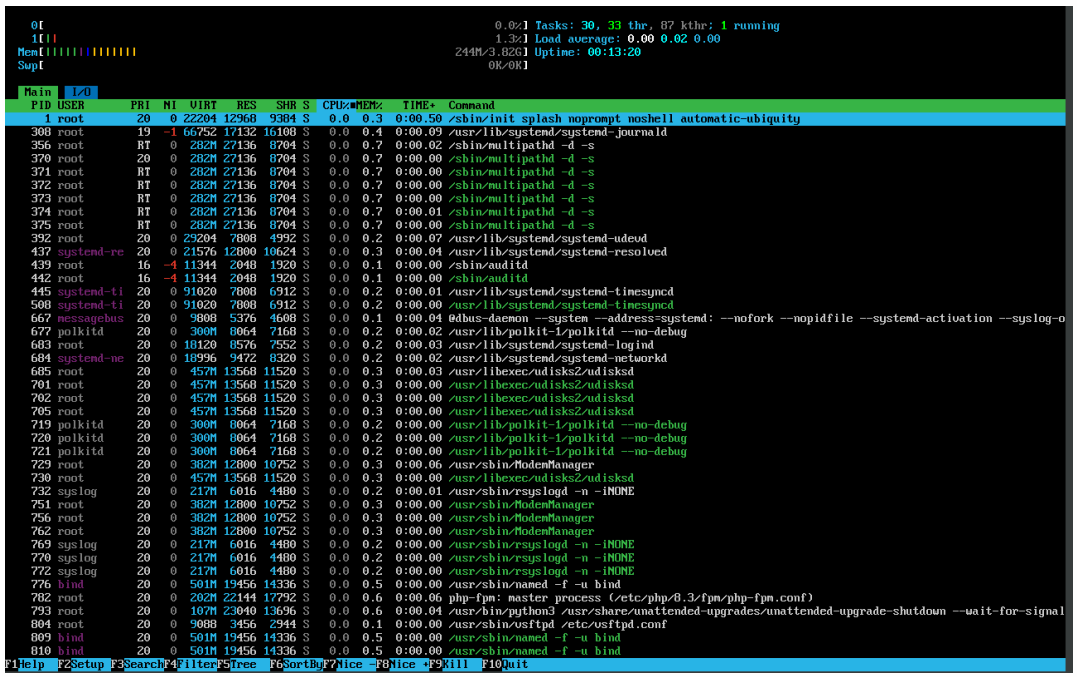
✅ Instalar el paquete stress o stress-ng.

✅ Ejecutar una prueba con carga simulada de CPU, memoria o disco durante 1 minuto.



✅ Observar el comportamiento del sistema con htop y anotar el resultado.

Antes:



Despues: