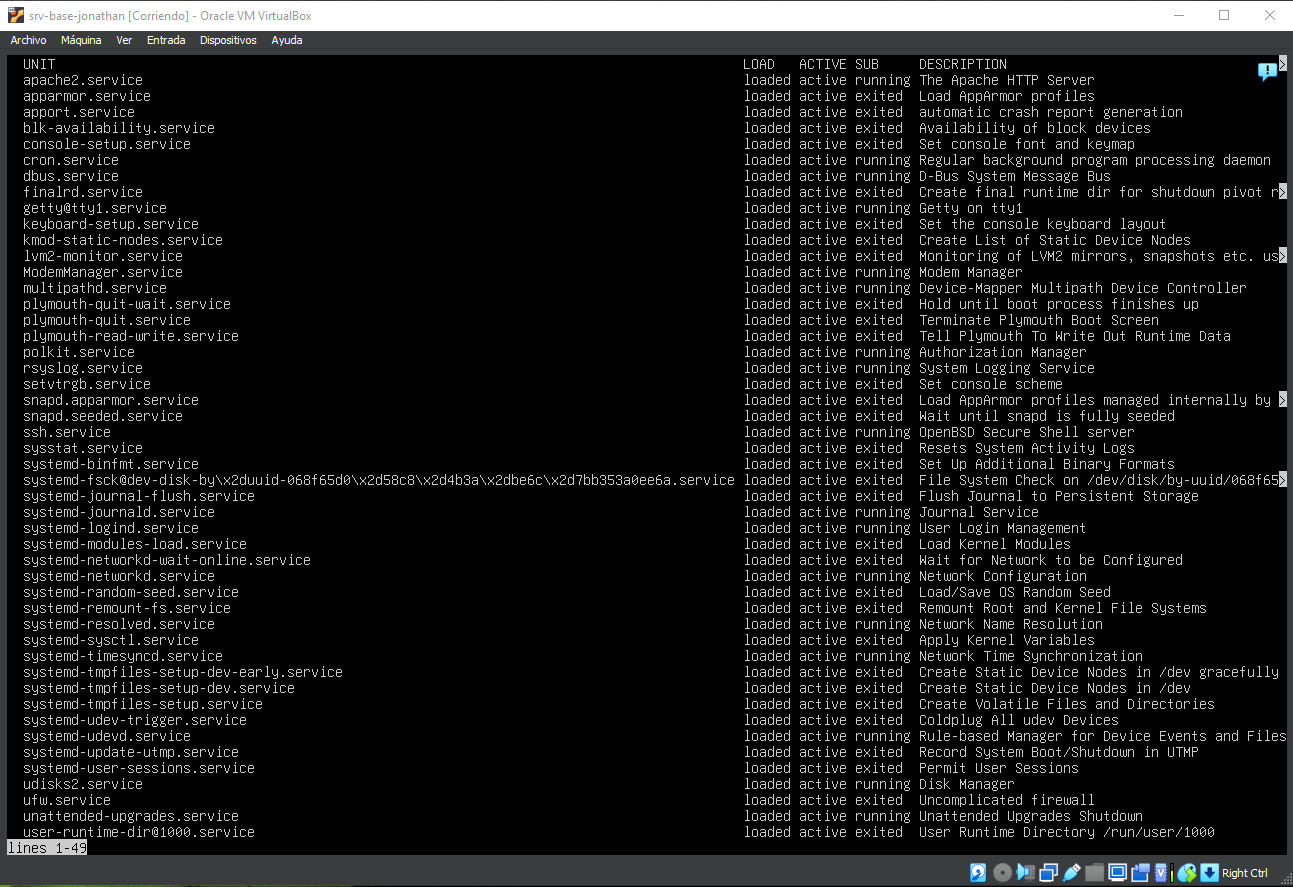
**Gestión de Servicios y Automatización con Systemd en Linux**

# Fase 1: Análisis de servicios del sistema

## **1. Listado de servicios activos**

Se ejecutó el siguiente comando para obtener una lista de todos los servicios activos del sistema:

systemctl list-units --type=service

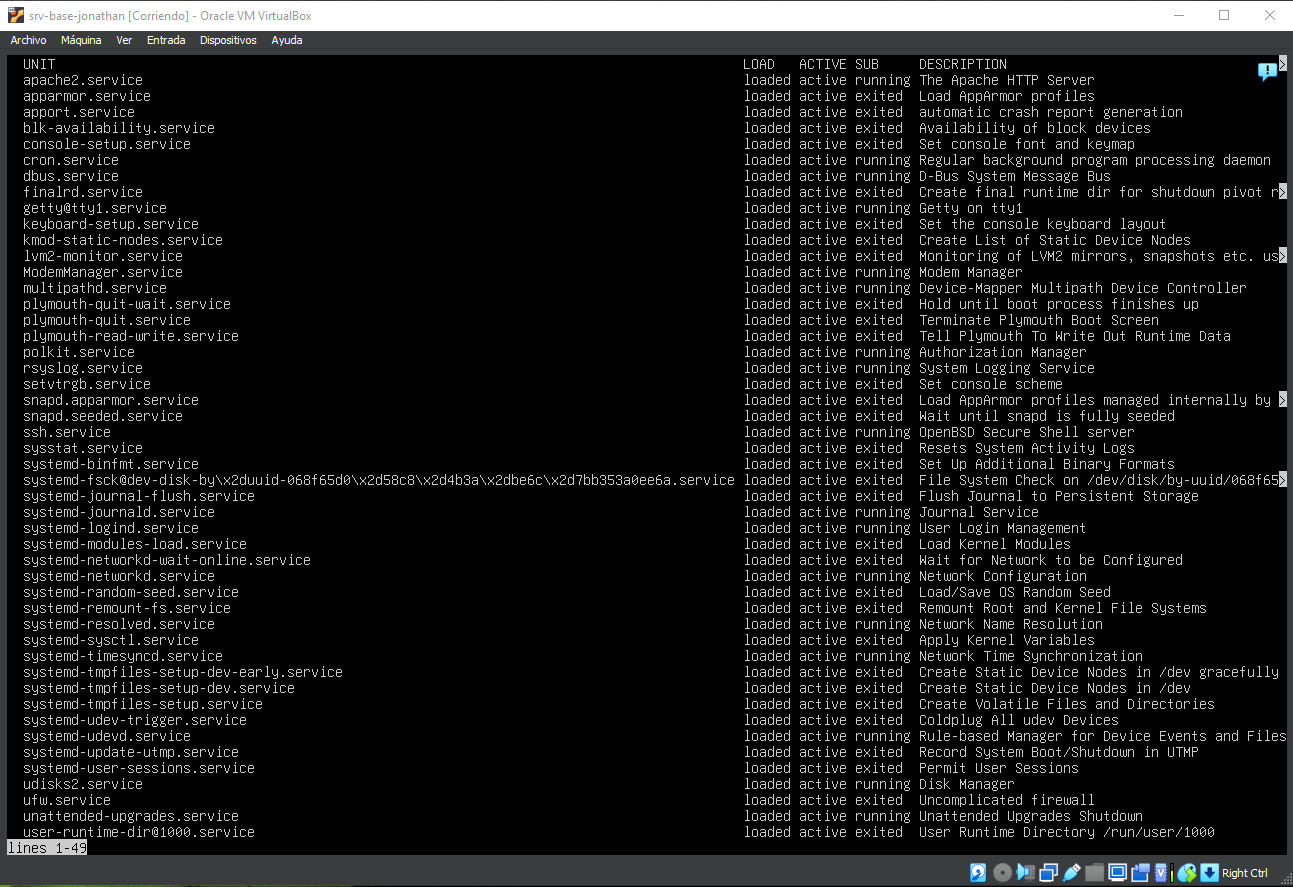


### 

## **2. Identificación de servicios fundamentales**

Se identificaron los siguientes servicios activos como fundamentales:

* **ssh.service**: Permite el acceso remoto seguro al servidor.
* **systemd-logind.service**: Gestiona las sesiones de usuario, inicio de sesión y bloqueos de pantalla.
* **ufw.service**: Controla el firewall UFW (Uncomplicated Firewall), protegiendo el sistema de accesos no autorizados.

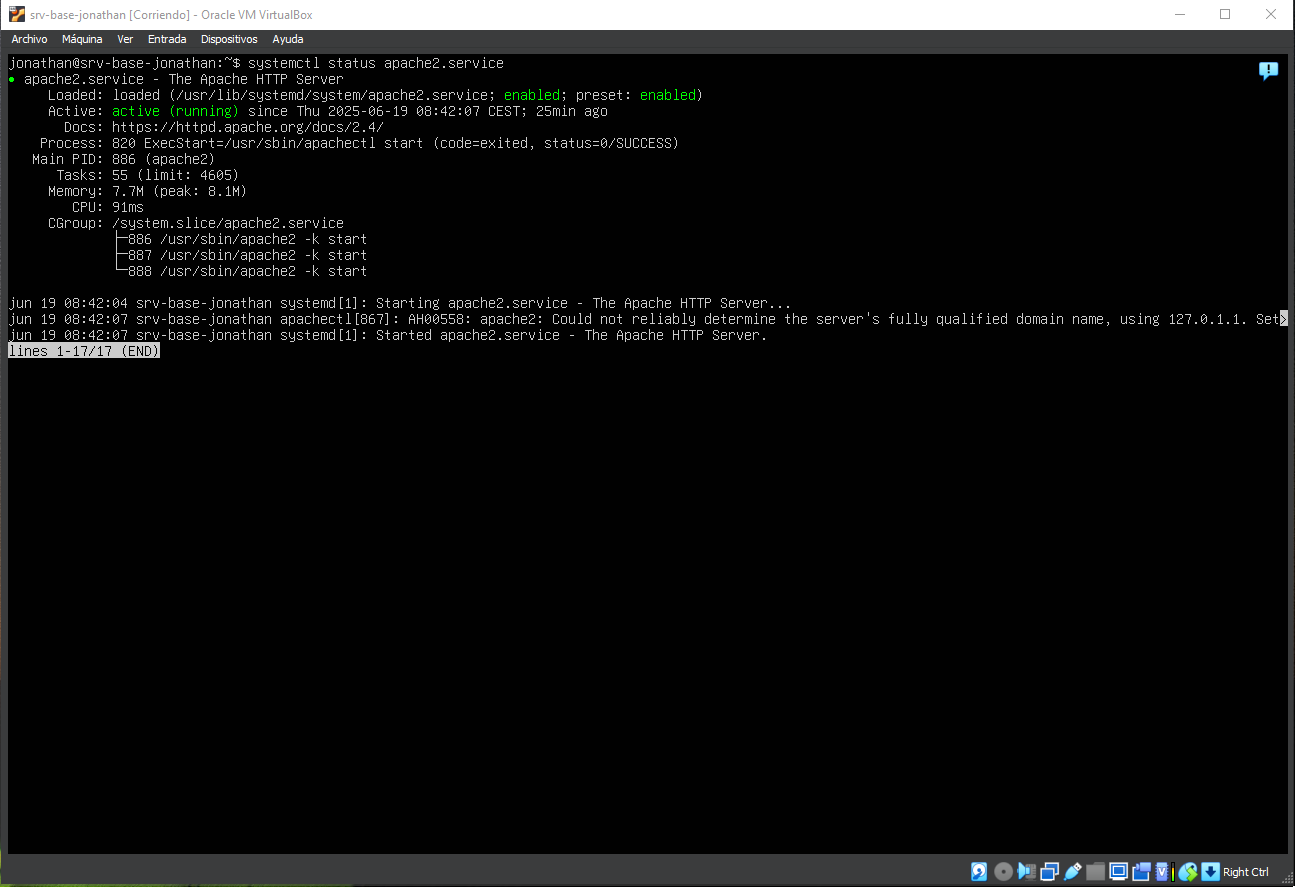


## 3. Comprobación del servidor web

Se comprueba que el servicio web (Apache2, instalado el dia anterior) esta activo y habilitado:

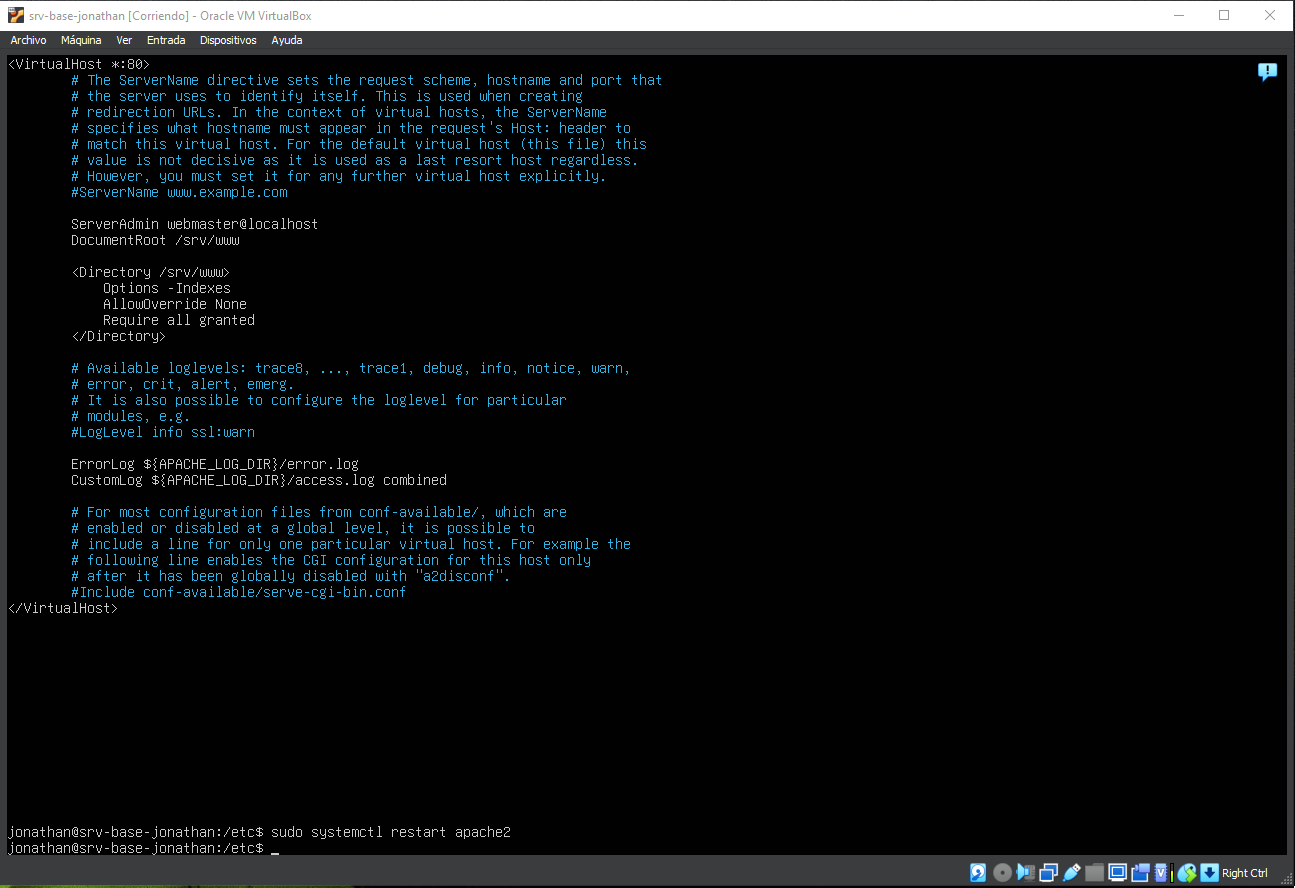
systemctl status apache2

systemctl is-enable apache2



Si no lo hicimos antes configuramos un directory para la web:

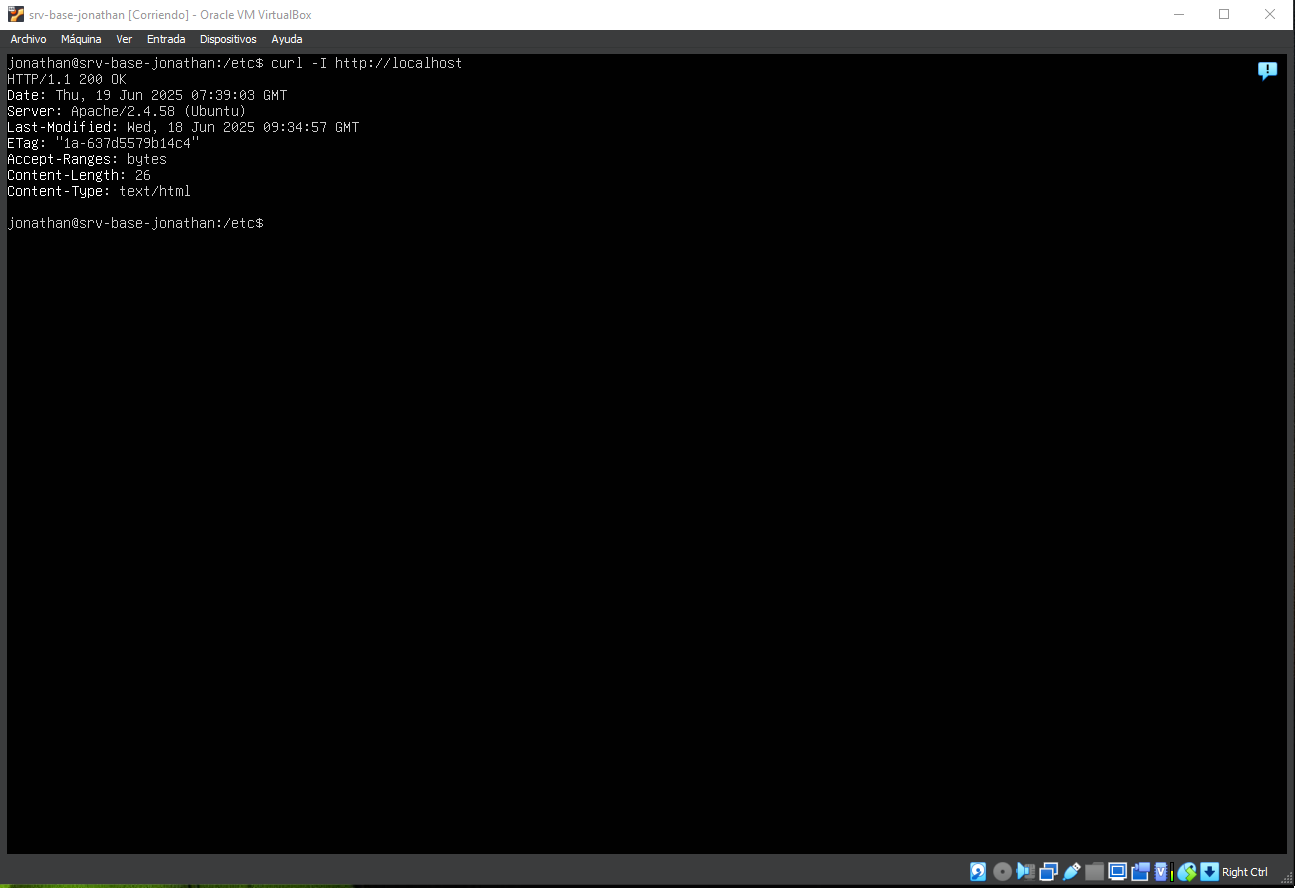
sudo nano /etc/apache2/sites-available/000-default.conf

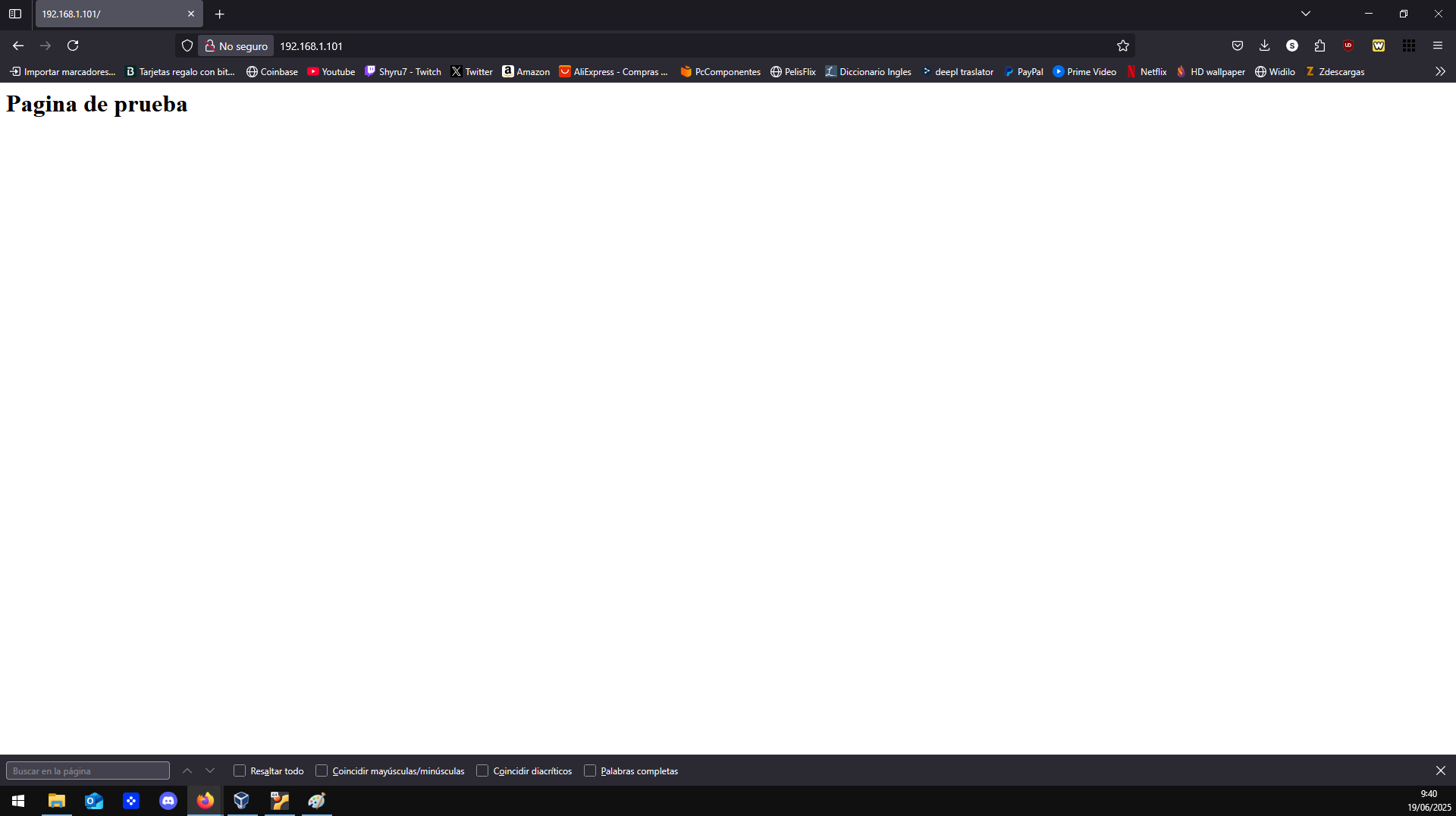


Probamos respuesta y comprobamos que podemos acceder desde un navegador:

curl -I <http://localhost>

Navegador: 192.168.1.101





# Fase 2: Gestión avanzada de servicios con Systemd

## 1. Detener, reiniciar y habilitar el servidor web

Para ello vamos a utilizar 3 comando diferentes:

sudo systemctl stop apache2

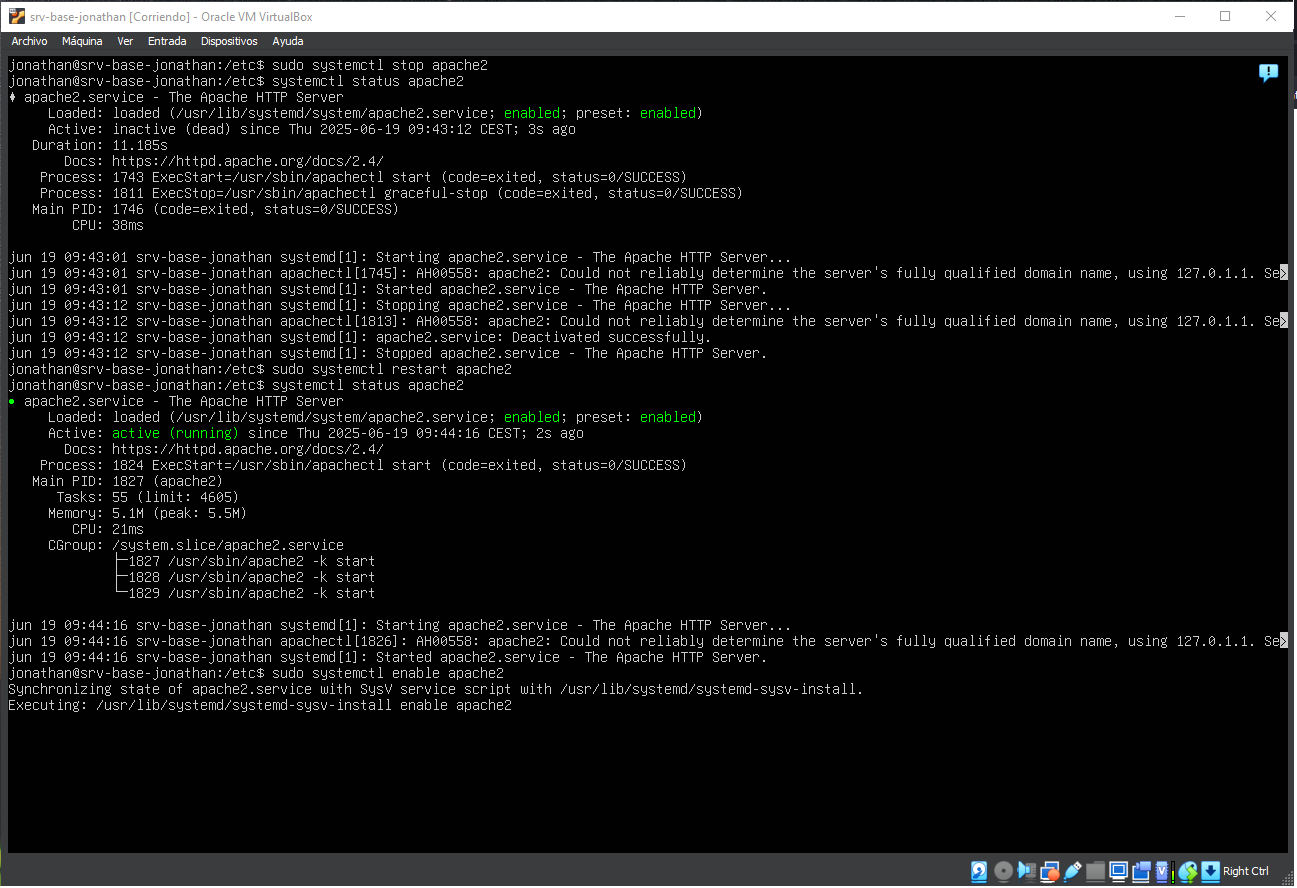
#detiene el servicio de apache2, lo que significa que el servidor web dejará de responder a peticiones.

sudo systemctl start apache2

#inicia el servicio de apache2, lo pone en funcionamiento para que pueda atender solicitudes.

sudo systemctl enable apache2

#hace que apache2 se inicie automáticamente cada vez que arranca el sistema.



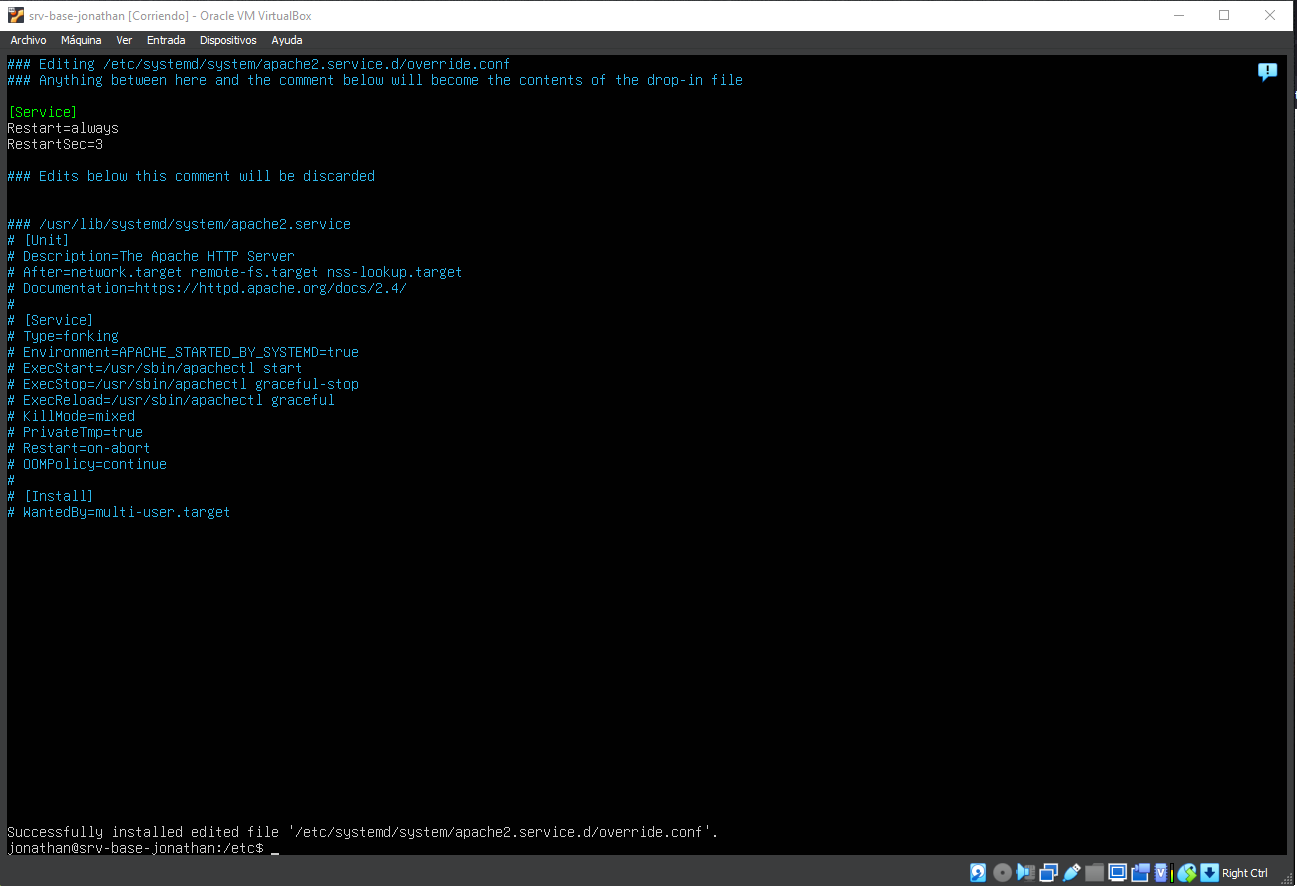
## 2. Configurar reinicio automático si falla

Editamos el archivo de unidad de apache para que reinicie si falla:

sudo systemctl edit apache2

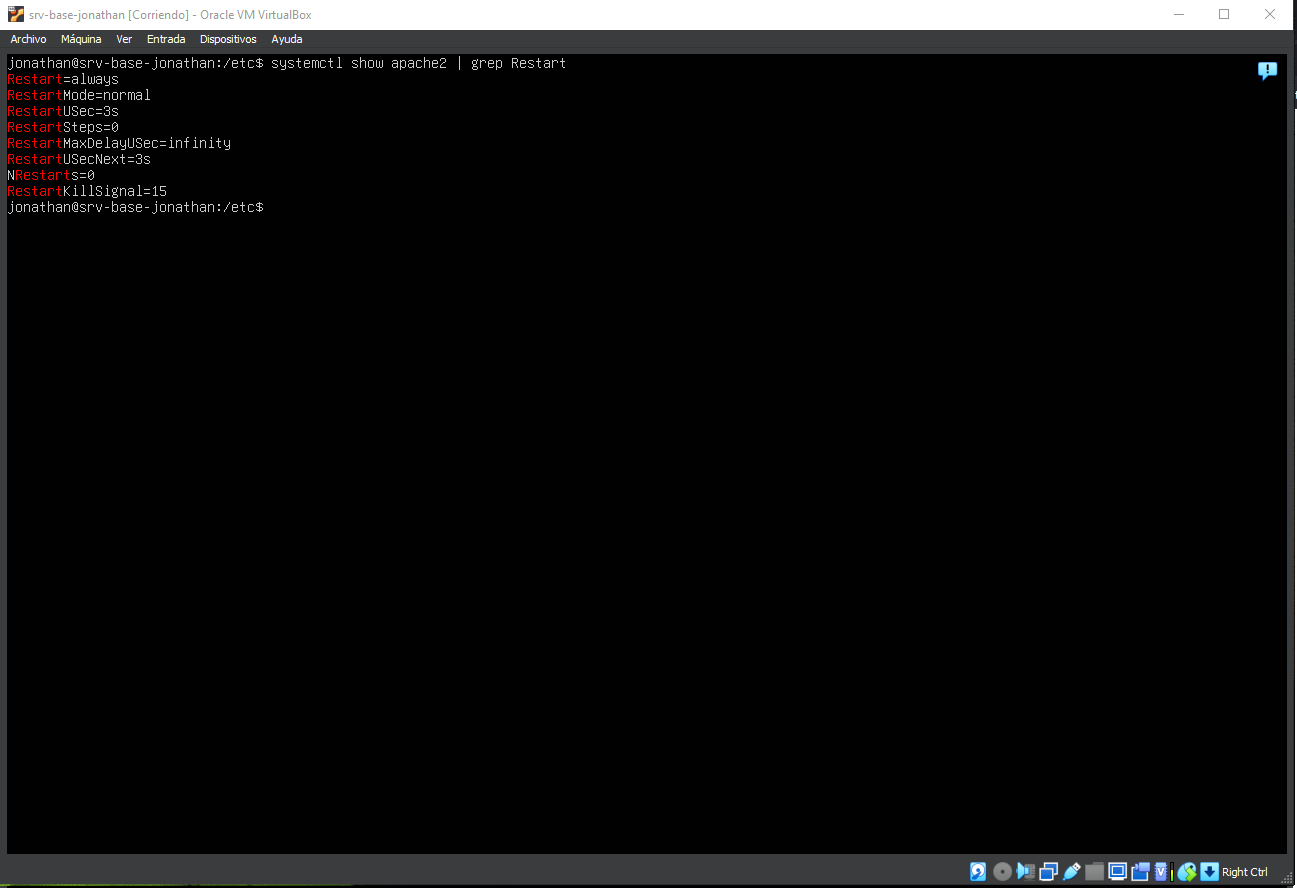
y se añade dentro de la sección, [service], Restart=always y luego recargamos el sistema con:

sudo systemctl daemon-reexec



Verificamos la configuración:

systemctl show apache2 | grep Restart



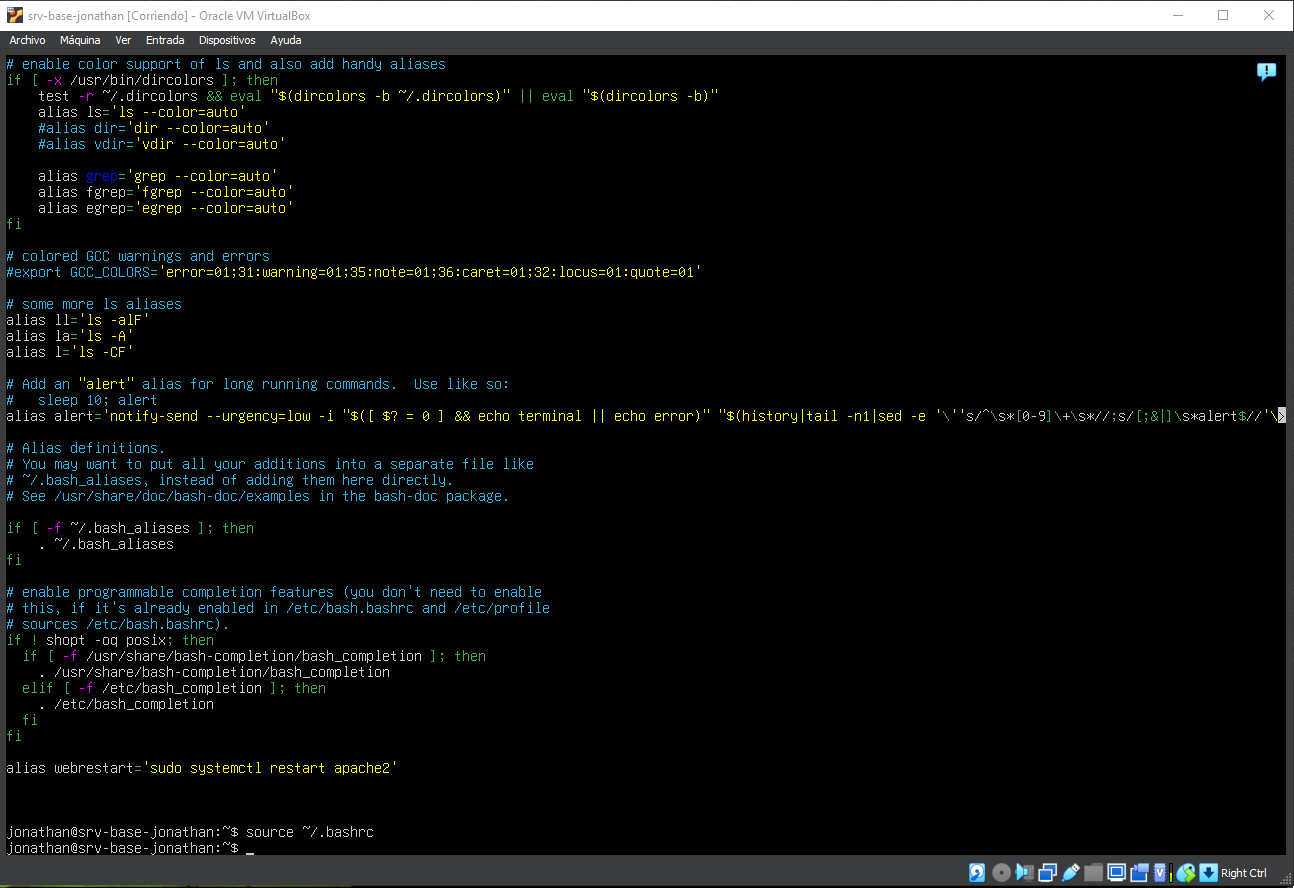
## 3. Crear un alias para reiniciar apache2

Añadiremos el siguiente alias al archivo ~/.bashrc para ello escribiremos la siguiente linea de comandos:

alias webrestart=’sudo systemctl restart apache2’

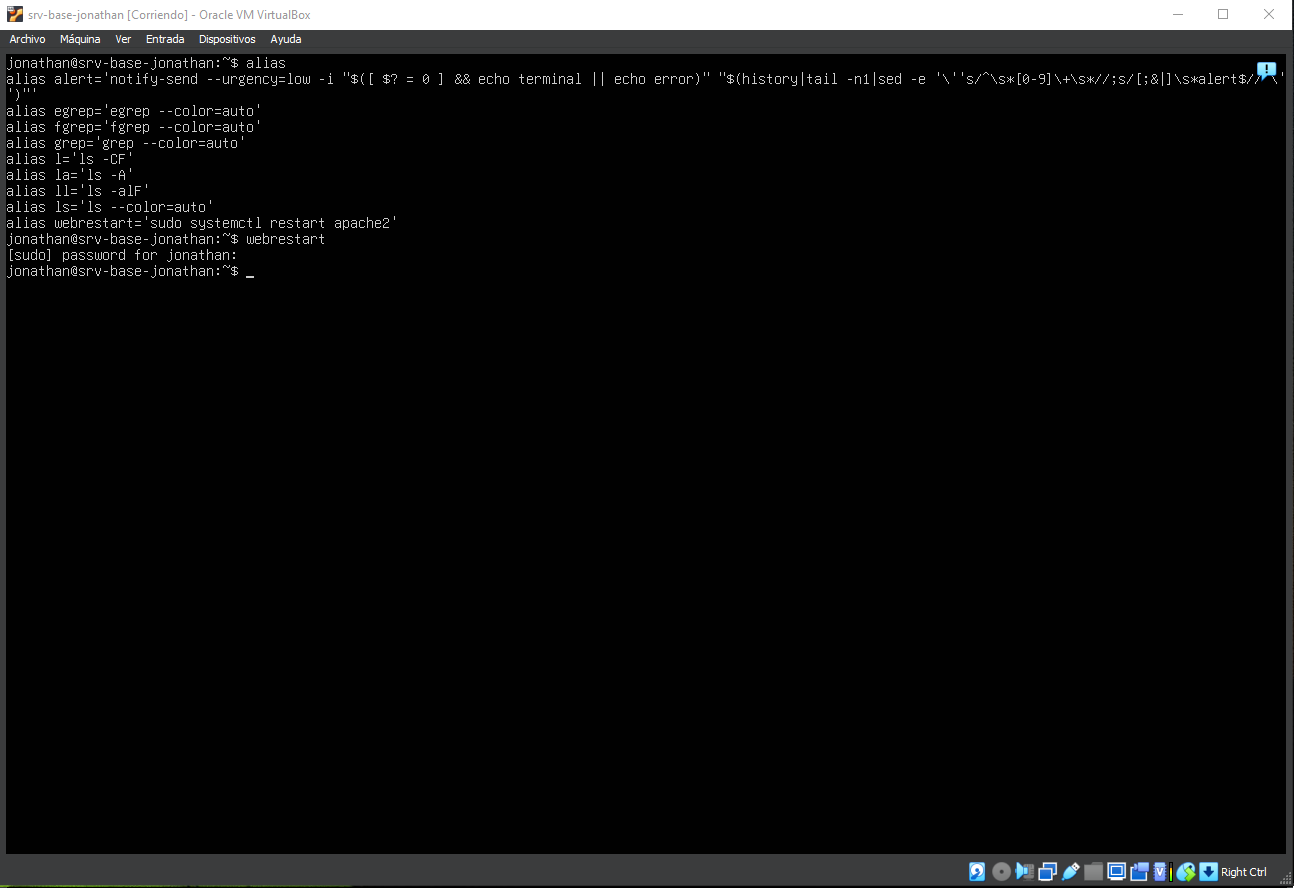
Luego lo aplicamos con:

source ~/.bashrc



y se comprueba el funcionamiento del alias:

webrestart



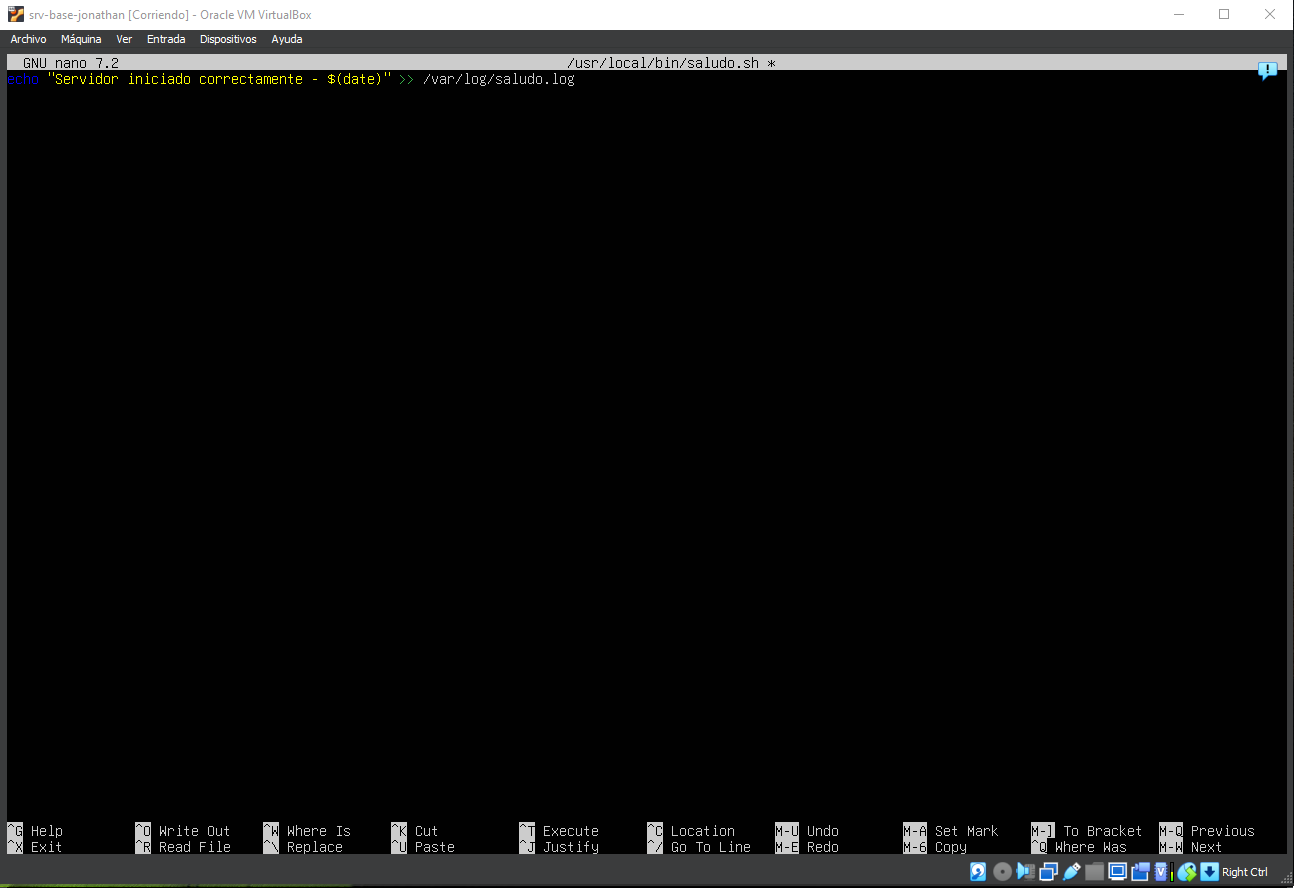
# Fase 3: Creación de un servicio personalizado

## 1. Crear el script [saludo.sh](http://saludo.sh)

Crearemos un script en /usr/local/bin/[saludo.sh](http://saludo.sh) con el siguiente comando:

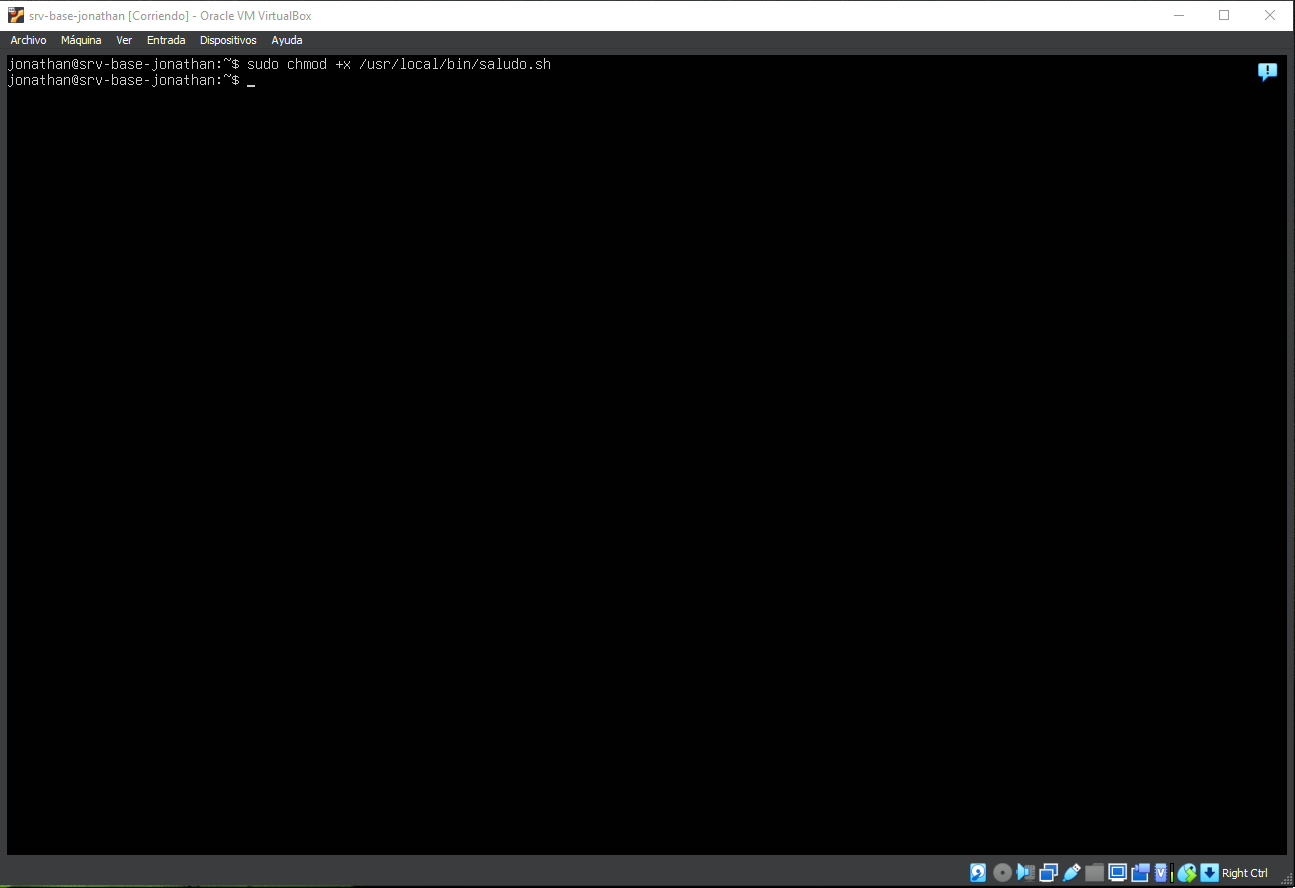
#!/bin/bash

echo "¡Servidor iniciado correctamente! - $(date)" >> /var/log/saludo.log



Dimos permisos de ejecución para esta tarea:

sudo chmod +x /usr/local/bin/saludo.sh



## 2. Crear el archivo saludo.service

Creamos el archivo /etc/systemd/system/saludo.service con el siguiente contenido:

Comando:

sudo nano /etc/systemd/system/saludo.service

[Unit]

Description=Servicio de saludo al iniciar el servidor

After=network.target

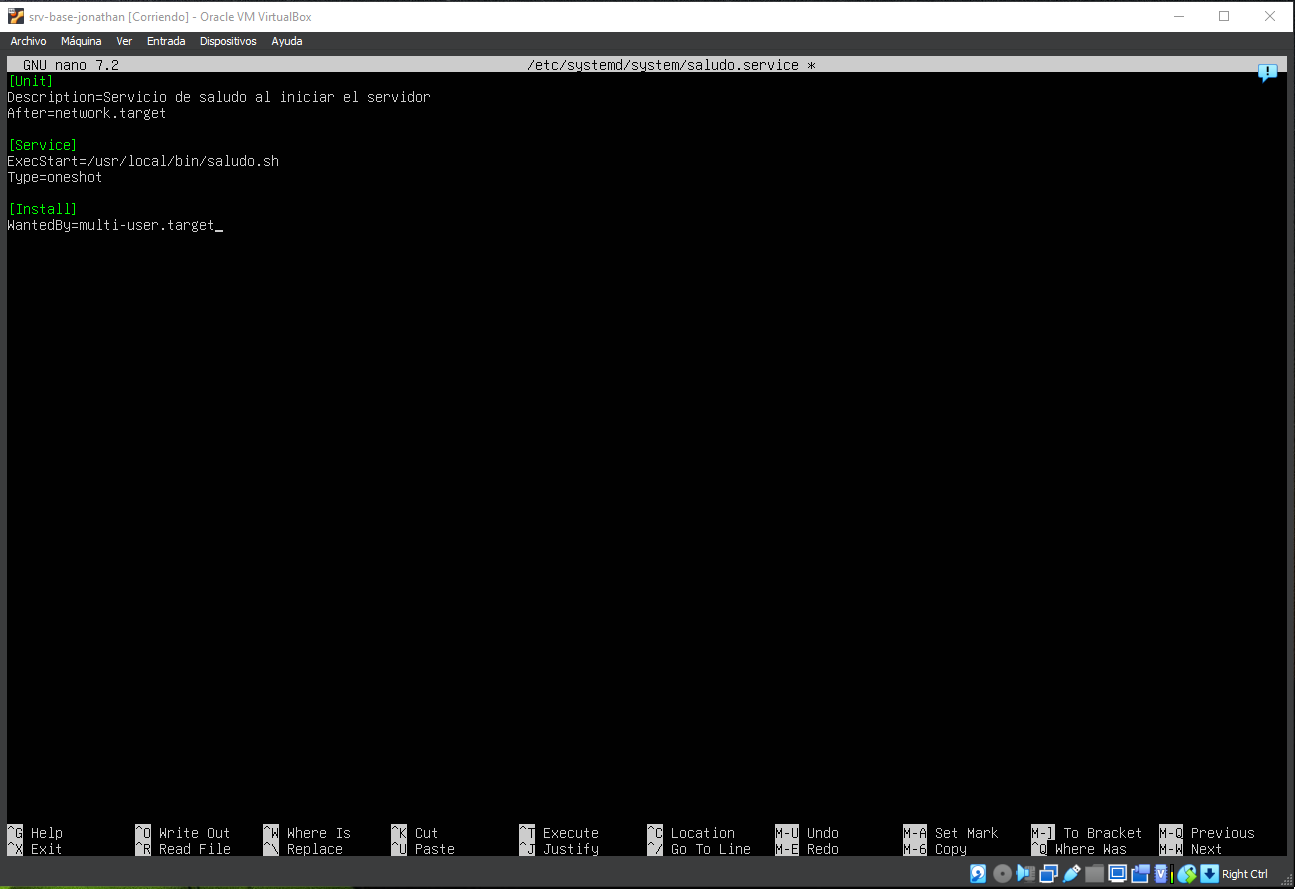
[Service]

Type=oneshot

ExecStart=/usr/local/bin/saludo.sh

[Install]

WantedBy=multi-user.target



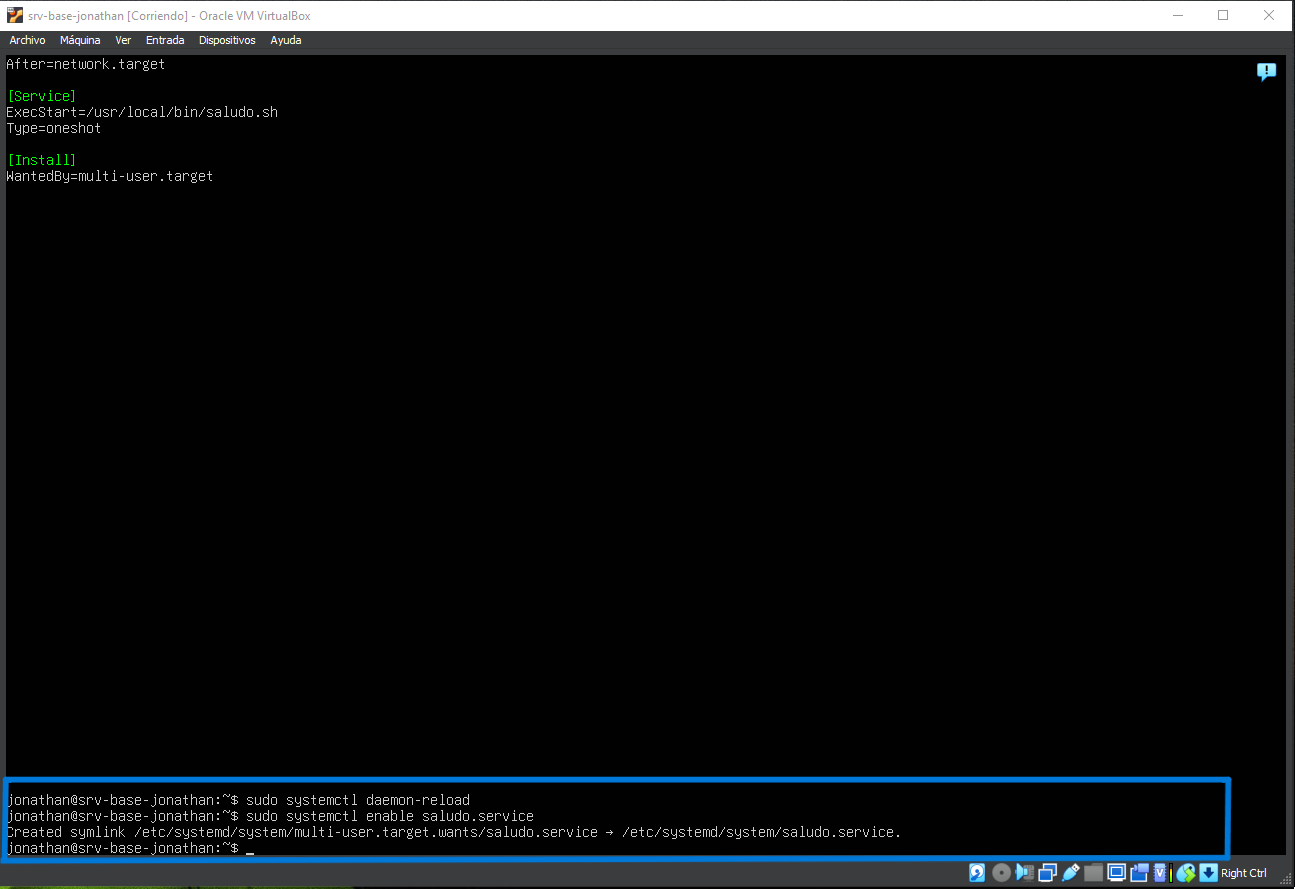
## 3. Recargar systemd y habilitar el servicio

Recargar systemd para reconocer el nuevo servicio:

sudo systemctl daemon-reload

Habilitar el servicio para que se ejecute al iniciar:

sudo systemctl enable saludo.service

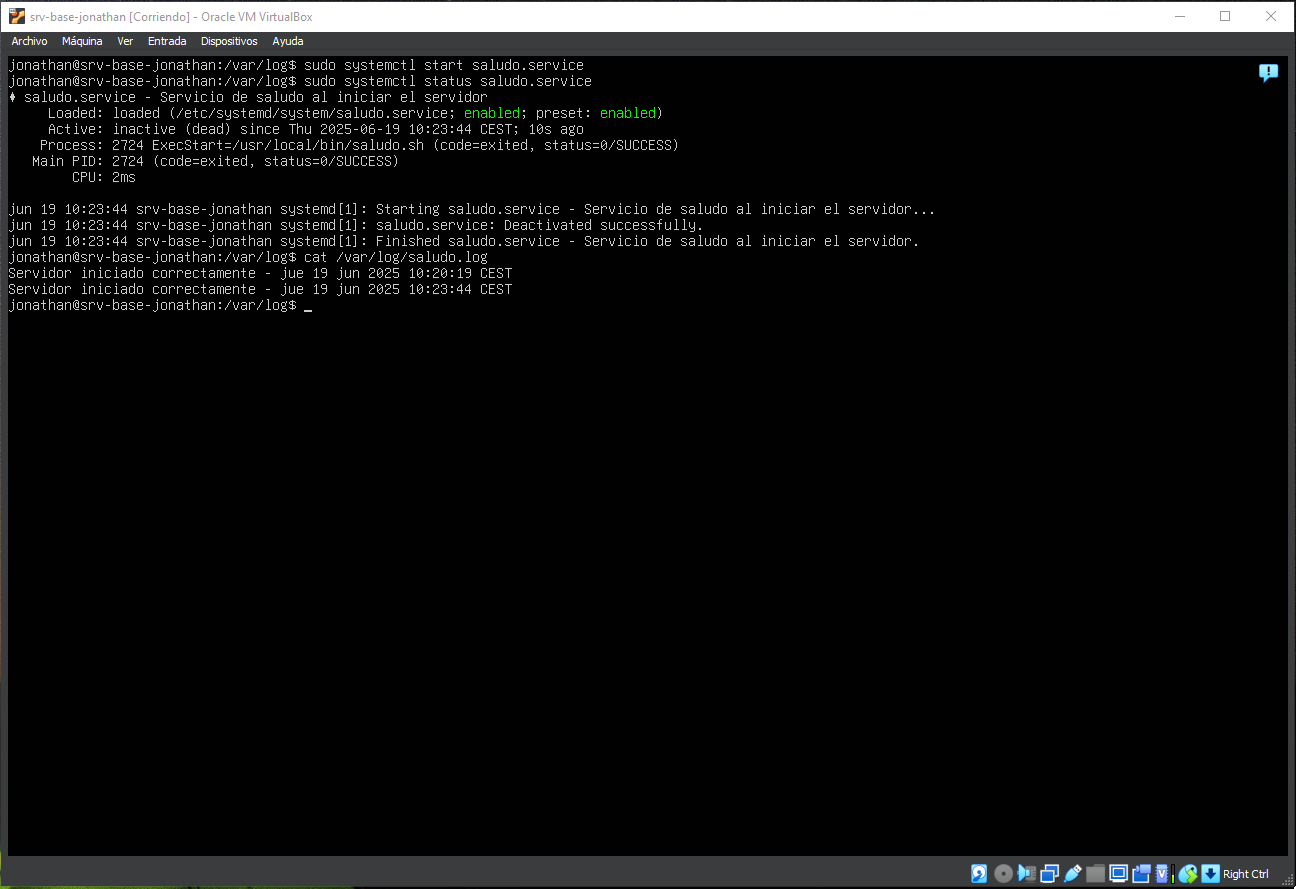


## 4. Probar y verificar el servicio

Ejecutamos manualmente para comprobar y verificar el estado del servicio:

sudo systemctl start saludo.service

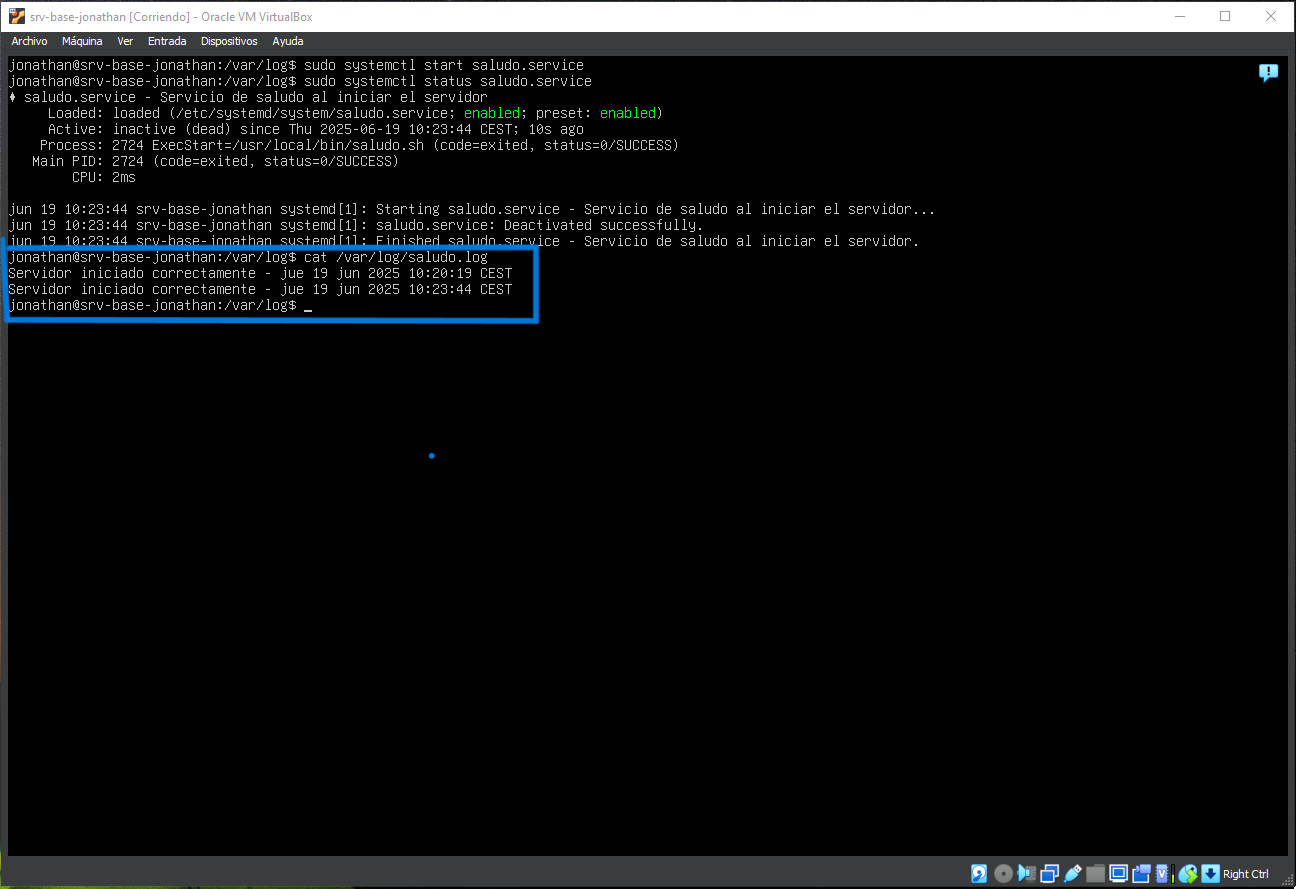
systemctl status saludo.service



## 5. Verificar archivo de log

Usamos el comando:

cat /var/log/saludo.log



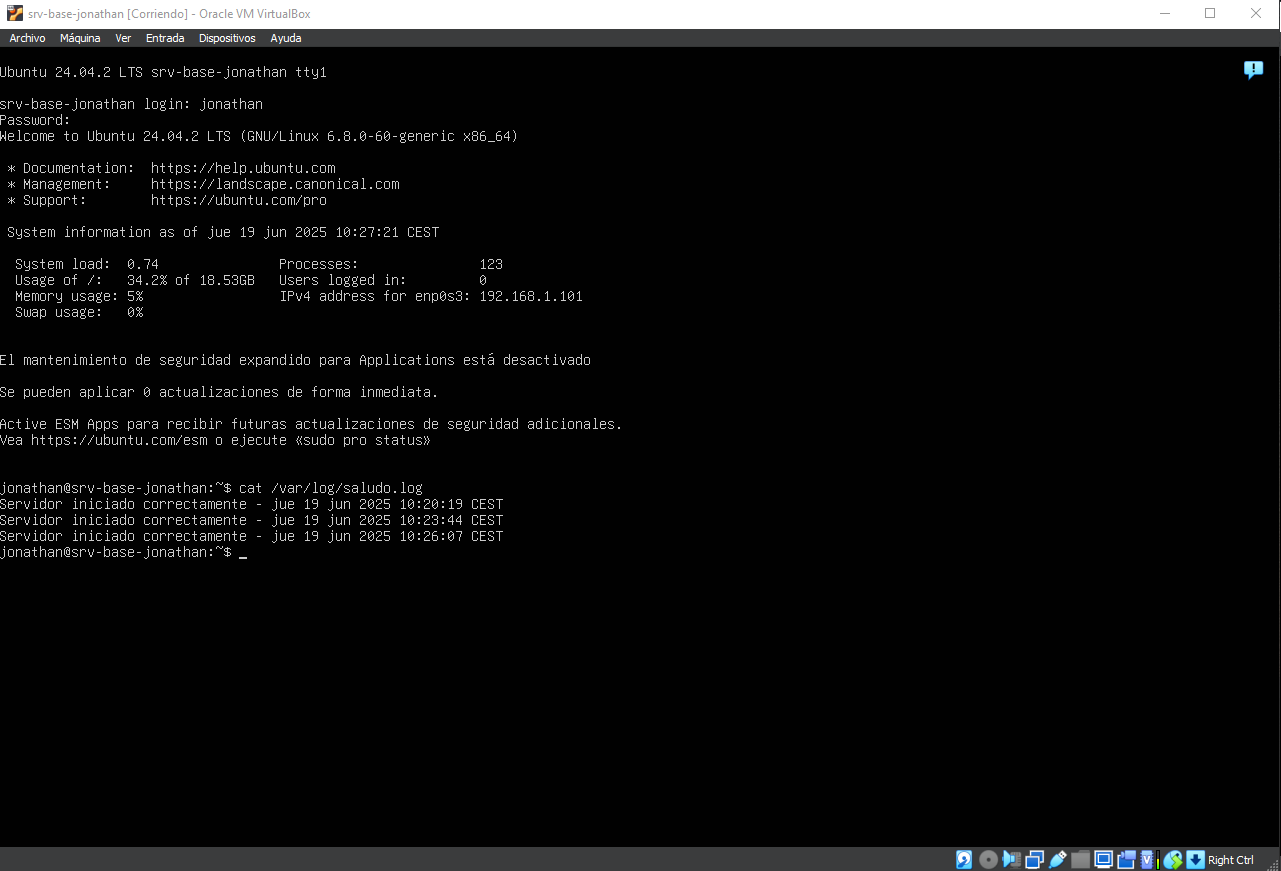
## 6. Confirmar que se ejecuta al arrancar

Vamos a confirmar que funciona reiniciando la maquina, usamos el comando:

sudo reboot

Se nos reiniciara la maquina, y despues de reiniciarla volveremos a comprobar el archivo de log:

car /var/log/saludo.log

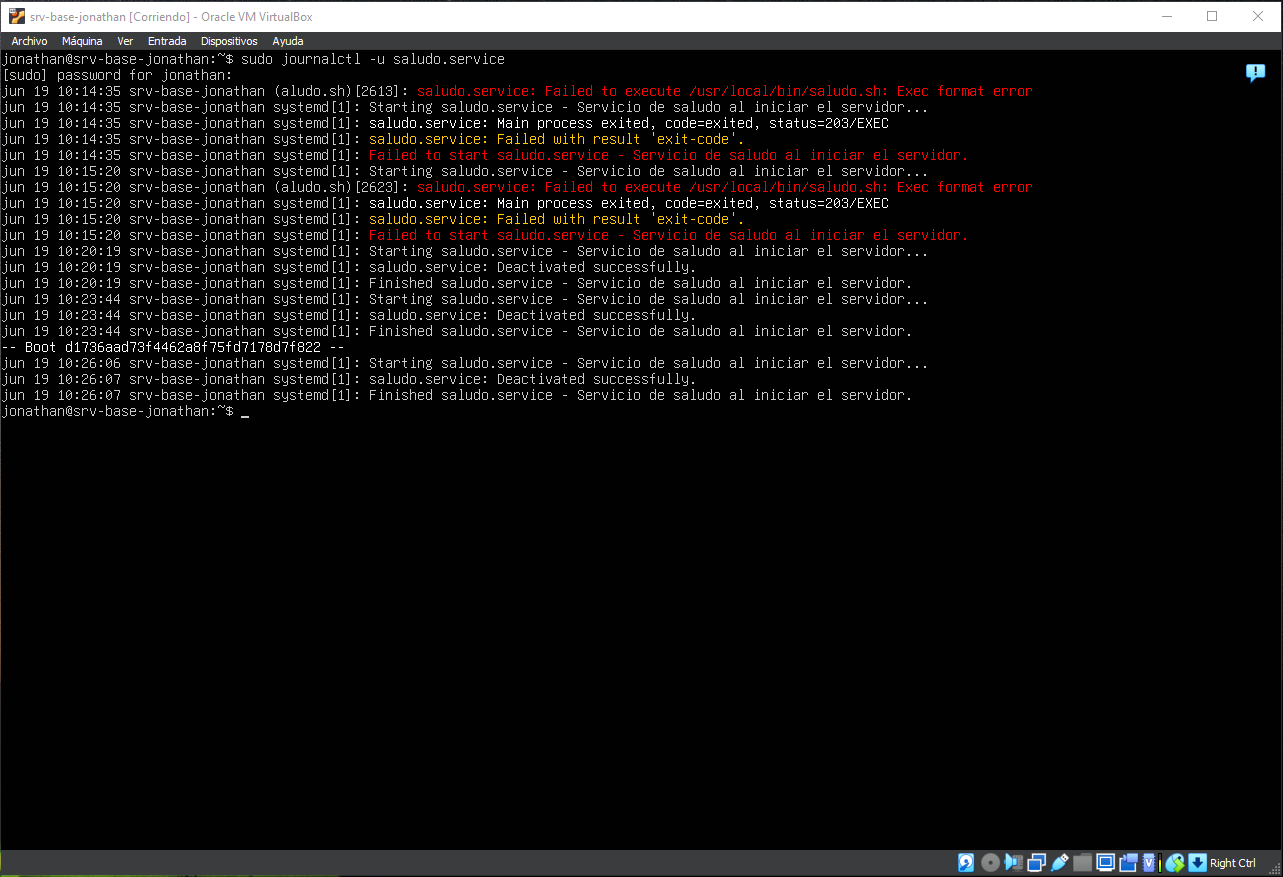


# Fase 4: Monitorización y logs

## 1. Ver logs con journalctl

Los primeros logs que vamos a ver seran los del servicio saludo:

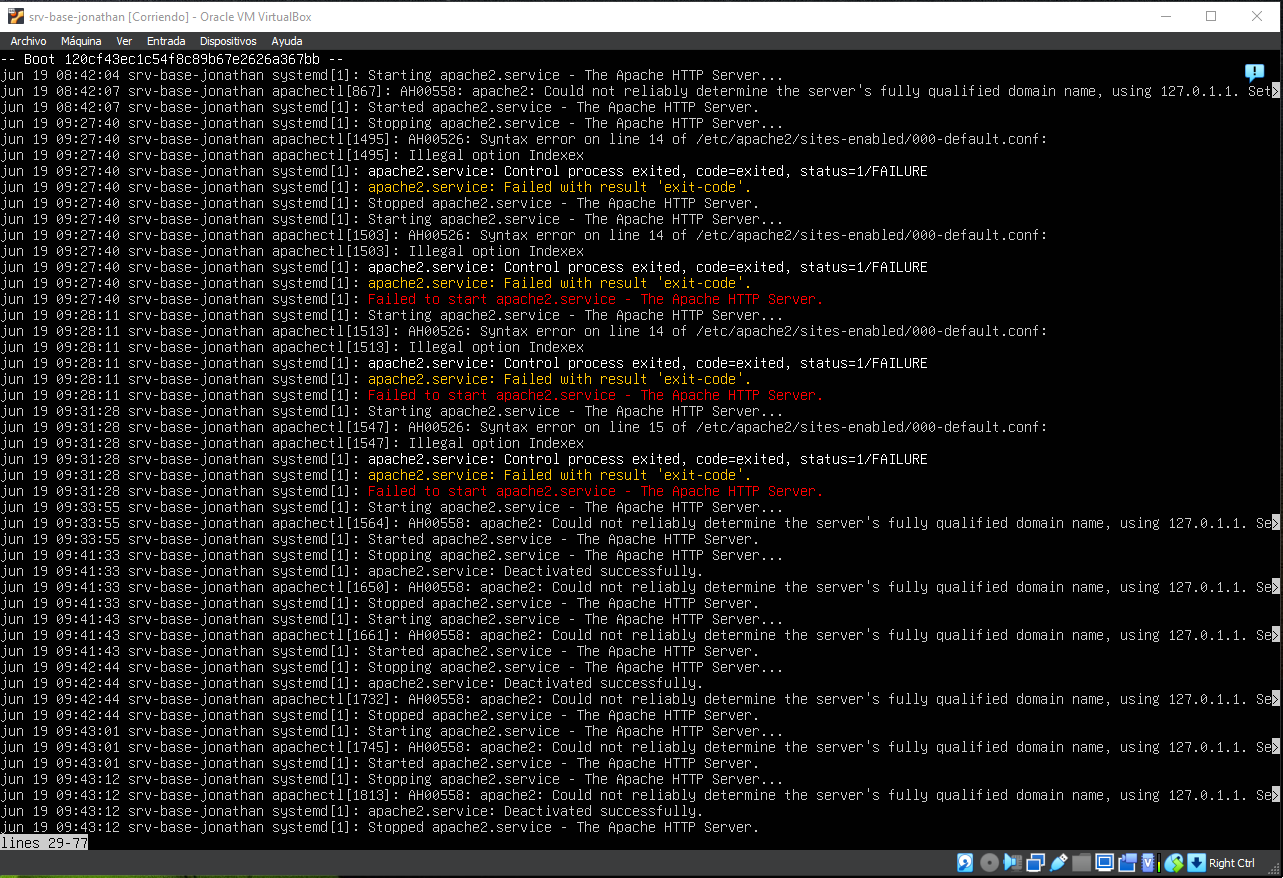
journalctl -u saludo.service



## 2. Filtrar mensajes de error o advertencia

Vamos a usar el siguiente comando para ver errores del arranque actual:

journalctl -p 3 -xb

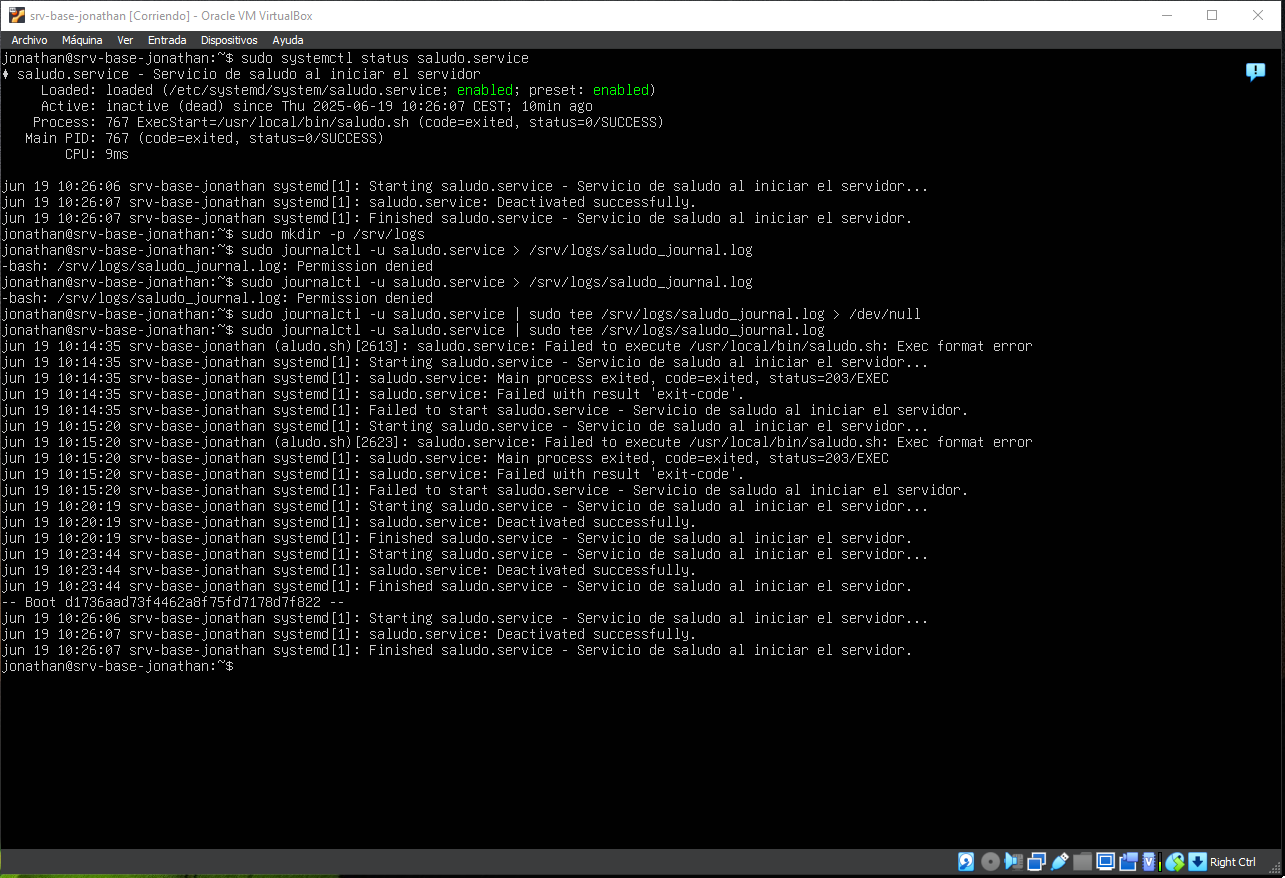


## 3. Guardar logs en un archivo externo

Vamos a crear el directorio /srv/logs y se guardaran los logs del servicio saludo:

sudo mkdir -p /srv/logs

sudo journalctl -u saludo.service | sudo tee /srv/logs/saludo\_journal.log



## Contenido del Script y el archivo de unidad service.

### **Script: saludo.sh**

Ubicación: /usr/local/bin/[saludo.sh](http://saludo.sh)

bash

#!/bin/bash

echo "¡Servidor iniciado correctamente! - $(date)" >> /var/log/saludo.log

### **Servicio: saludo.service**

Ubicación: /etc/systemd/system/saludo.service

[Unit]

Description=Servicio de saludo al iniciar el servidor

After=network.target

[Service]

Type=oneshot

ExecStart=/usr/local/bin/[saludo.sh](http://saludo.sh)

[Install]

WantedBy=multi-user.target

**FIN**