



## APLICACIONES DISTRIBUIDAS

### TEMA: ALTA DISPONIBILIDAD

### TALLER N°4

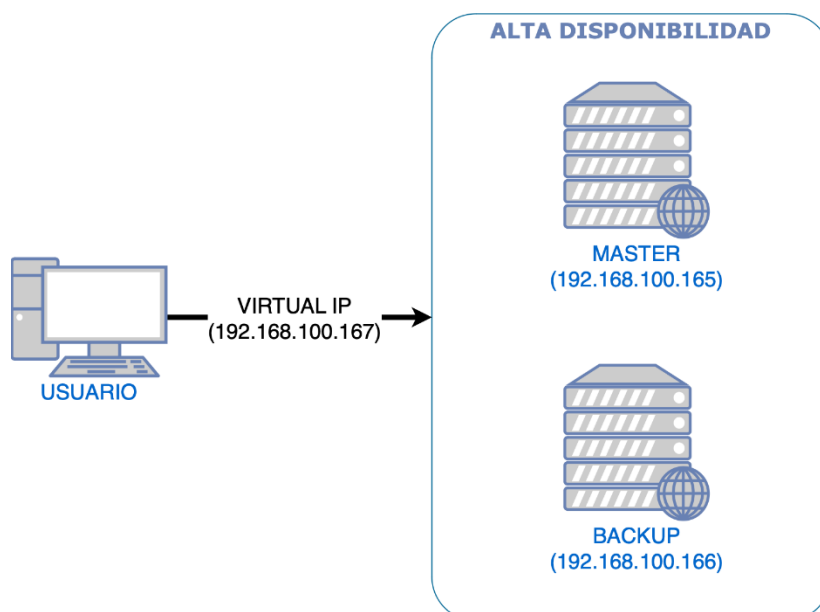
**Objetivo:** Comprender la arquitectura clúster mediante la configuración de la alta disponibilidad para establecer un servicio permanente en situaciones de fallos de hardware, software o red, además de entender cómo se integran en los modelos arquitectónicos de los sistemas distribuidos.

#### INDICACIONES GENERALES:

- El trabajo se desarrollará de forma individual.
- Desarrollar las actividades mencionadas en el siguiente punto y realizar un informe con la explicación de cada uno de los pasos que siguió para la elaboración de la tarea.
- Subir el documento en formato PDF al aula virtual en el espacio habilitado para la práctica.
- La práctica tiene un puntaje de 10 puntos

#### ACTIVIDAD EN CLASE:

1. Desarrollar un de alta disponibilidad en base a la siguiente configuración de red utilizando el servidor web NGINX y la herramienta Keepalived:



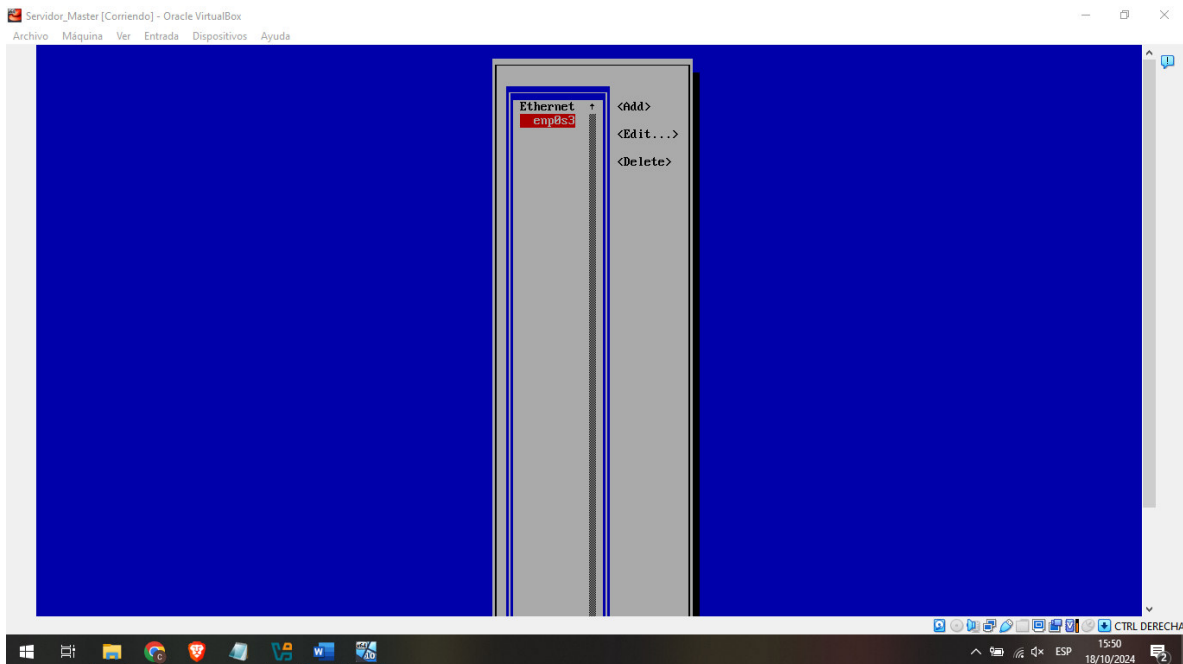
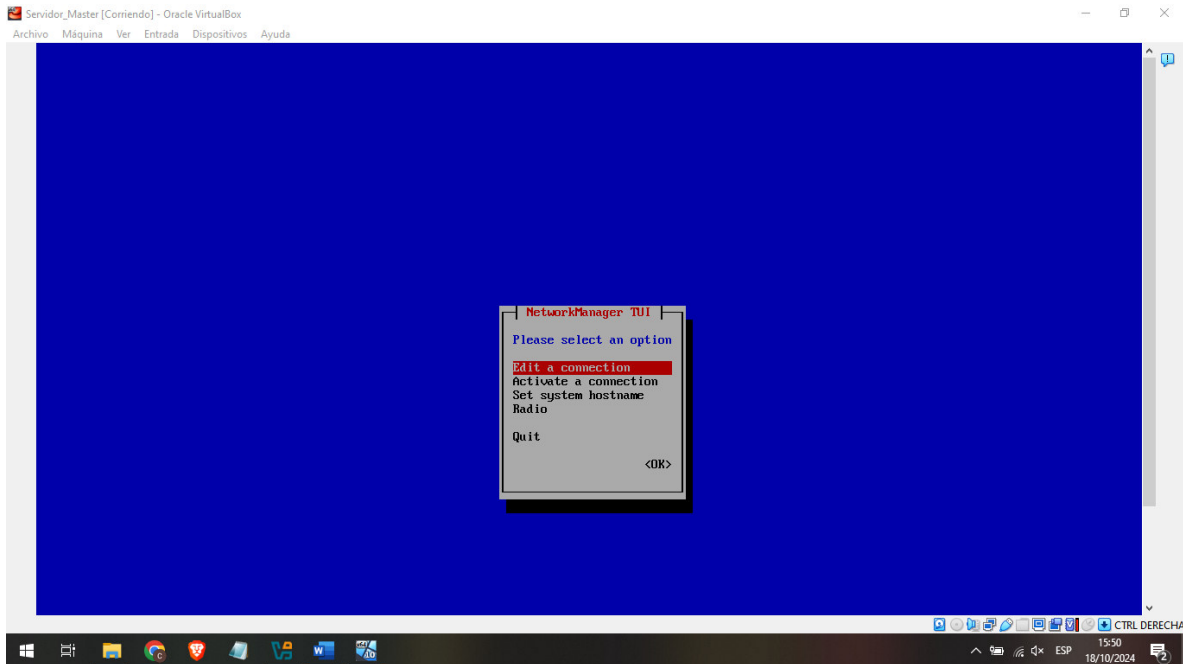


- Las máquinas MASTER y BACKUP deben tener instalado el sistema operativo Alma Linux.
- Configurar una máquina con el Sistema Operativo Windows para el acceso al servidor de alta disponibilidad.
- Crear una página web básica con estilos, la misma que tiene que estar configurada en cada nodo.
- Al esquema anterior agregar un nodo adicional para fortalecer la alta disponibilidad en el laboratorio. Asignar al nuevo servidor la IP que crea conveniente.
- Al final evidenciar la alta disponibilidad del laboratorio, dando de baja algunos nodos.
- Agregue una conclusión y recomendación sobre el tema



- **Configuraciones de las máquinas virtuales**

Lo primero será ocupar el comando nmtui y cambiar la IP. Esto en las dos máquinas.

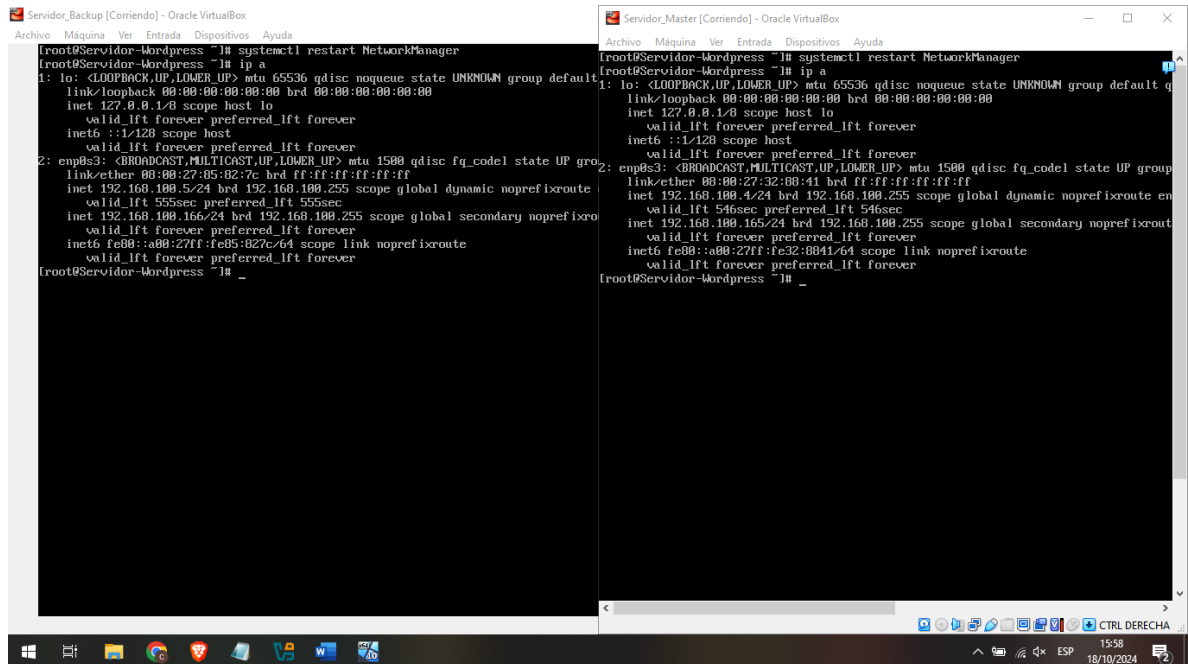
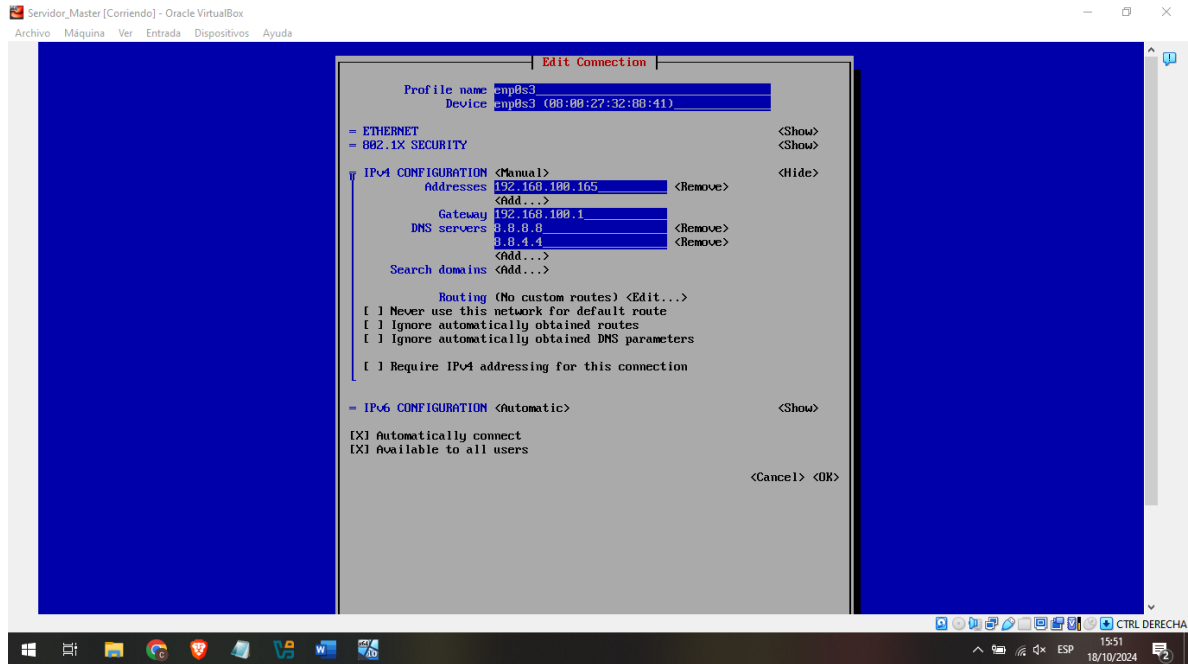




# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

PERÍODO ACADÉMICO: SEPTIEMBRE 2024 - FEBRERO 2025



De esta forma se cambian las direcciones IP en las máquinas virtuales.

The image shows two terminal windows side-by-side, both running AlmaLinux. The left window is titled 'root@Master:-' and the right window is titled 'root@Backup:-'.

**Left Window (root@Master:-):**

```
almaLinux-logos-httpd
noarch 90.5.1-1.1.el9 appstream 18 k
nginx-core x86_64 1:1.20.1-16.el9_4.1 appstream 565 k
nginx-filesystem noarch 1:1.20.1-16.el9_4.1 appstream 7.9 k

Transaction Summary
-----
Install 4 Packages

Total download size: 626 k
Installed size: 1.8 M
Downloading Packages:
(1/4): almaLinux-logos-httpd 50 kB/s | 18 kB 00:00
(2/4): nginx-1.20.1-16.el9_4 92 kB/s | 35 kB 00:00
(3/4): nginx-filesystem-1.20 310 kB/s | 7.9 kB 00:00
(4/4): nginx-core-1.20.1-16 1.0 MB/s | 565 kB 00:00
-----
Total 548 kB/s | 626 kB 00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Running scriptlet: nginx-filesystem-1:1.20.1-16.el9_4.1 1/4
Installing : nginx-filesystem-1:1.20.1-16.el9_4.1 1/4
Installing : nginx-core-1:1.20.1-16.el9_4.1.x86_64 2/4
Installing : almaLinux-logos-httpd-90.5.1-1.1.el9_4.1 3/4
Installing : nginx-1:1.20.1-16.el9_4.1.x86_64 4/4
Running scriptlet: nginx-1:1.20.1-16.el9_4.1.x86_64 4/4
Verifying : almaLinux-logos-httpd-90.5.1-1.1.el9_4.1 1/4
Verifying : nginx-1:1.20.1-16.el9_4.1.x86_64 2/4
Verifying : nginx-core-1:1.20.1-16.el9_4.1.x86_64 3/4
Verifying : nginx-filesystem-1:1.20.1-16.el9_4.1 4/4

Installed:
almaLinux-logos-httpd-90.5.1-1.1.el9.noarch
nginx-1:1.20.1-16.el9_4.1.x86_64
nginx-core-1:1.20.1-16.el9_4.1.x86_64
```

**Right Window (root@Backup:-):**

```
No match for argument: nginx
Error: Unable to find a match: nginx
[root@Backup ~]# yum install nginx -y
Last metadata expiration check: 0:01:28 ago on Fri Oct 18 16:08:02 2024.
Dependencies resolved.
=====
Package Arch Version Repo Size
=====
Installing:
nginx x86_64 1:1.20.1-16.el9_4.1 appstream 35 k
Installing dependencies:
almaLinux-logos-httpd
nginx-core x86_64 1:1.20.1-16.el9_4.1 appstream 565 k
nginx-filesystem noarch 1:1.20.1-16.el9_4.1 appstream 7.9 k

Transaction Summary
=====
Install 4 Packages

Total download size: 626 k
Installed size: 1.8 M
Downloading Packages:
(1/4): almaLinux-logos-httpd 43 kB/s | 18 kB 00:00
(2/4): nginx-1.20.1-16.el9_4 83 kB/s | 35 kB 00:00
(3/4): nginx-filesystem-1.20 295 kB/s | 7.9 kB 00:00
(4/4): nginx-core-1.20.1-16 1.0 MB/s | 565 kB 00:00
-----
Total 547 kB/s | 626 kB 00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Running scriptlet: nginx-filesystem-1:1.20.1-16.el9_4.1 1/4
Installing : nginx-filesystem-1:1.20.1-16.el9_4.1 1/4
Installing : nginx-core-1:1.20.1-16.el9_4.1.x86_64 2/4
```

Hay que ocupar el mismo comando en los dos servidores: `Yum install nginx -y`

The image shows two terminal windows side-by-side. The left window, titled 'root@Master:-', displays the command '[root@Master ~]# yum install keepalived -y'. The right window, titled 'root@Backup:-', displays a long list of installed packages, including mariadb, net-snmp, perl, and various perl modules, all installed from the 'noarch' repository.

Hay que instalar keepalived en ambos servidores.



```
WINDOWS [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@Master:~# service nginx start
Redirecting to /bin/systemctl start nginx.service
root@Master:~# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service - /usr/lib/systemd/system/nginx.service.
root@Master:~#

root@Backup:~# service nginx start
Redirecting to /bin/systemctl start nginx.service
root@Backup:~# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service - /usr/lib/systemd/system/nginx.service.
root@Backup:~#
```

Iniciar y activar el servicio de nginx

Service nginx start

Systemctl enable nginx

```
WINDOWS [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@Master:~# firewall-cmd --permanent --add-service=http
success
root@Master:~# firewall-cmd --reload
success
root@Master:~#

root@Backup:~# firewall-cmd --permanent --add-service=http
Warning: ALREADY ENABLED: http
success
root@Backup:~# firewall-cmd --reload
success
root@Backup:~#
```

Dar permisos en el firewall: firewall-cmd --permanent -add-service=http

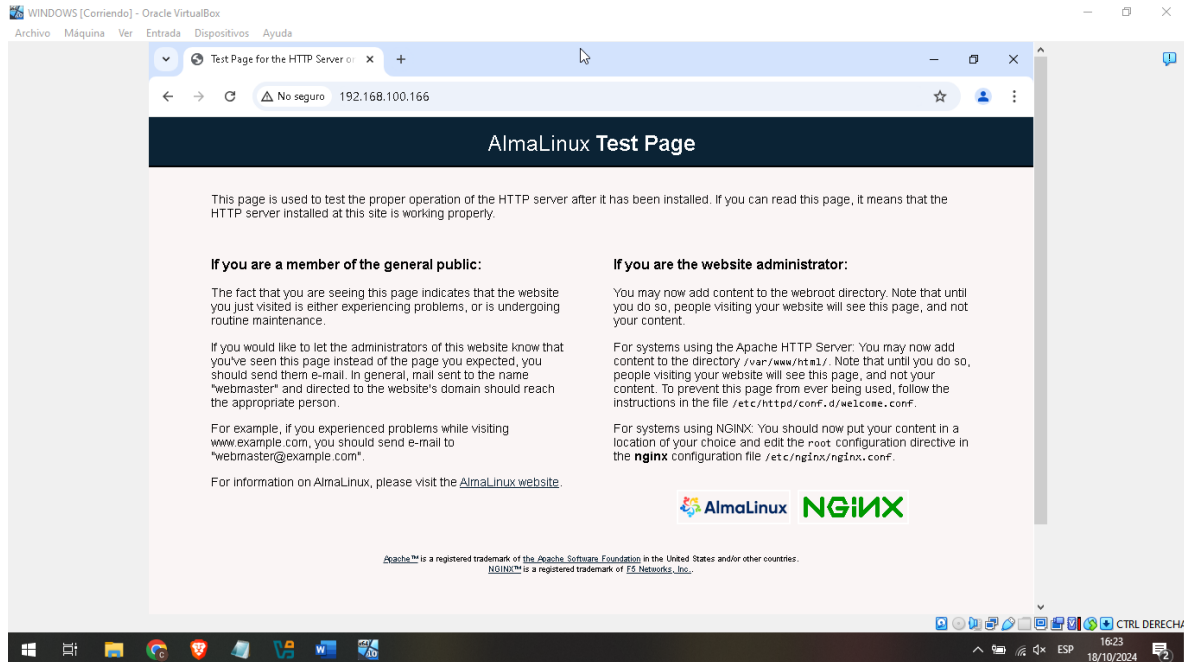
firewall-cmd --reload



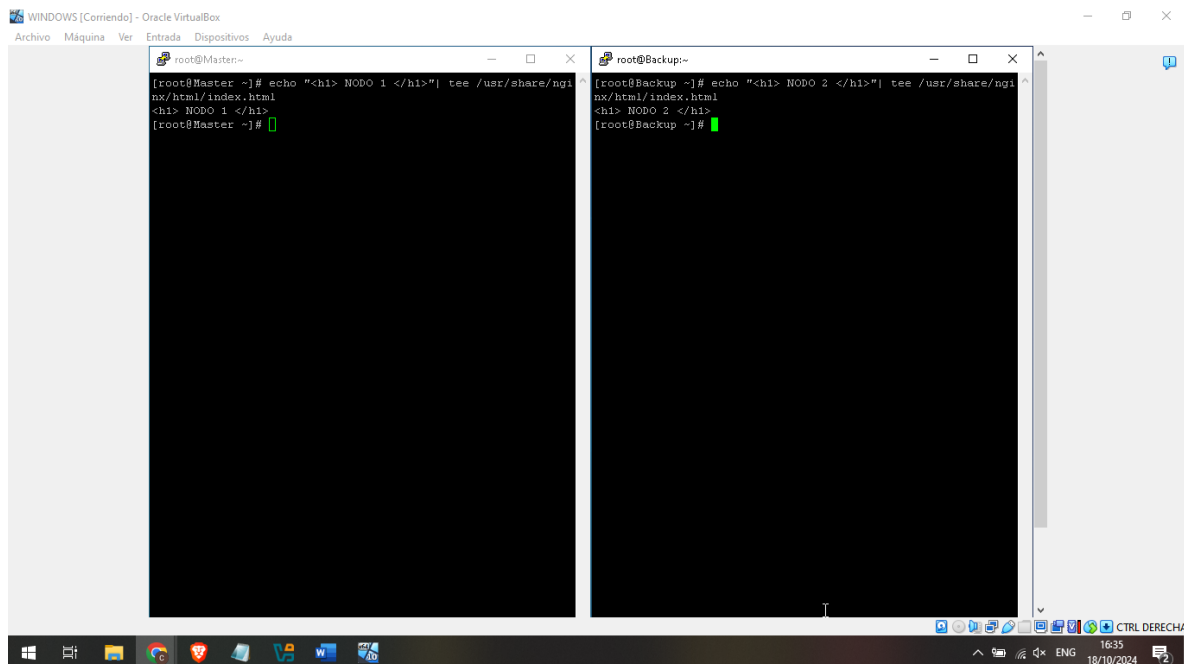
# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

PERÍODO ACADÉMICO: SEPTIEMBRE 2024 - FEBRERO 2025



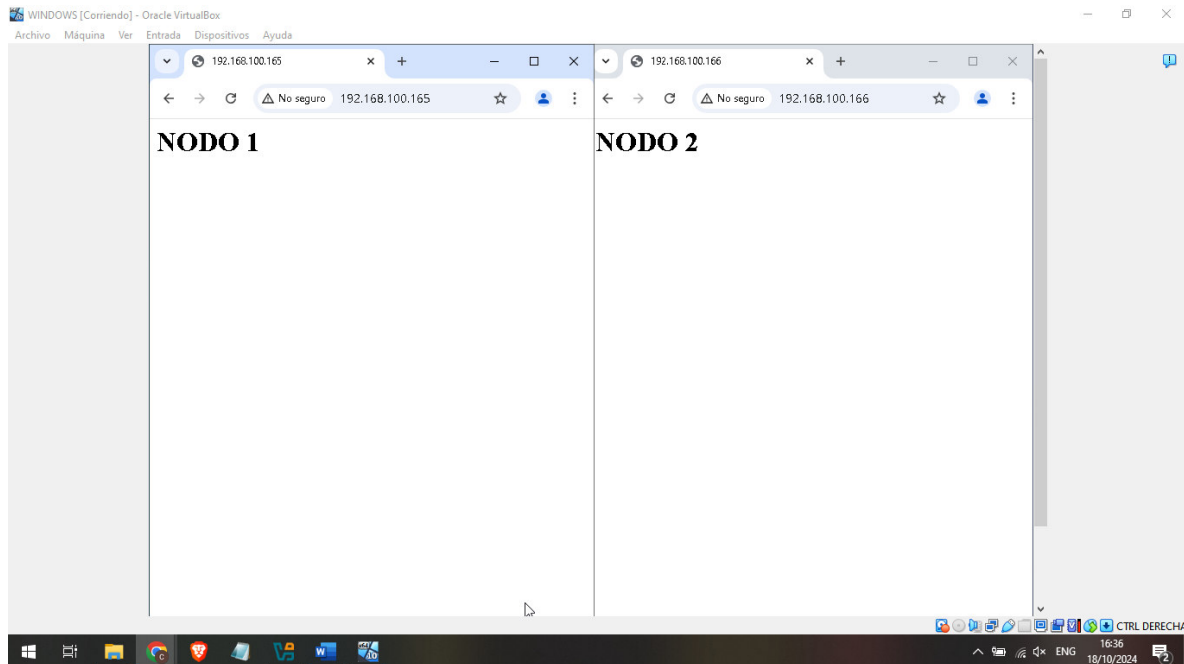
Como se puede apreciar, al ingresar desde el navegador colocando la IP del servidor aparece el logo de NGINX y el servidor funcionando.



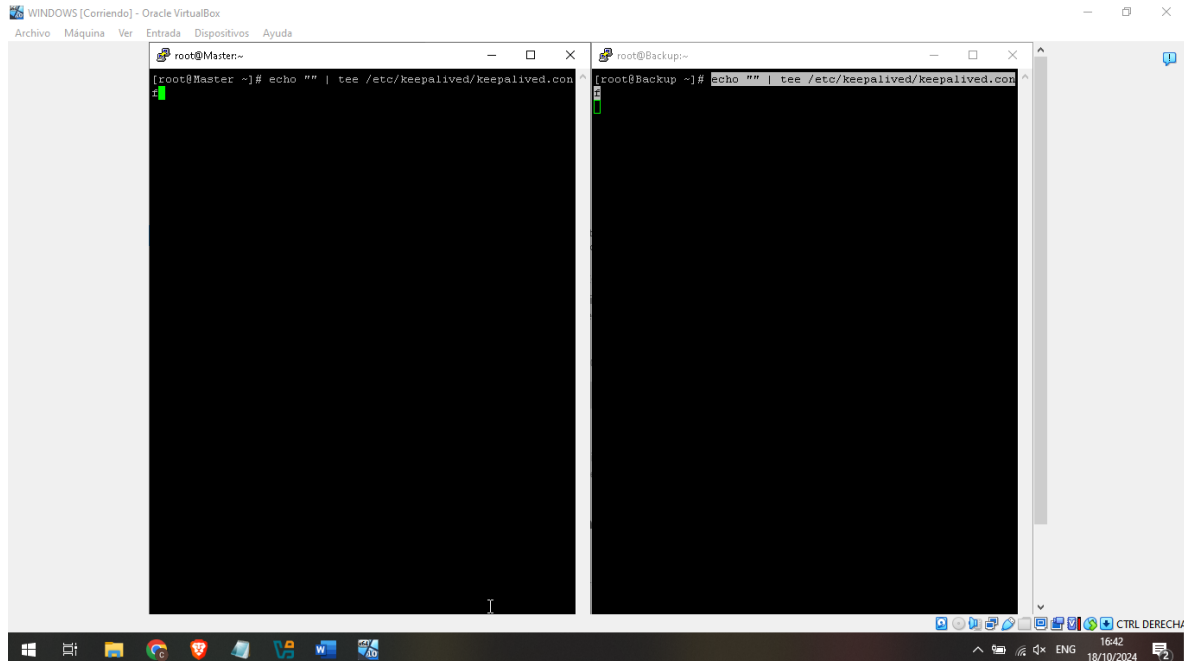
Crear un html simple para diferenciar el servidor

```
echo "<h1> NODO 1</h1>" | tee /usr/share/nginx/html/index.html
```

Ing. Mg. Pablo I. Morales P.



Como se puede apreciar al ingresar desde el navegador aparece NODO 1 y 2 respectivamente.



Se debe limpiar el archivo de configuración:

```
echo "" | tee /etc/keepalived/keepalived.conf
```





# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

PERÍODO ACADÉMICO: SEPTIEMBRE 2024 - FEBRERO 2025



```
root@Master~# GNU nano 5.6.1 /etc/keepalived/keepalived.conf
##### ALTA DISPONIBILIDAD #####
global_defs {
    # Keepalived process identifier
    router_id nginx # Se puede cambiar dependiendo del servicio
}

# Script to check whether Nginx is running or not
vrrp_script check_nginx {
    script "/bin/check_nginx.sh" # Verifica que el servicio funcione
    interval 2
    weight 50
}

# Virtual interface - The priority specifies the order in
# which the assigned interface to take over in a failover
vrrp_instance VI_01 {
    state MASTER # Dependiendo el nodo se configura MASTER o BACKUP
    interface enp0s3 # Agregar la interfaz de red eth0, enp0s3
    virtual_router_id 50 # De 0-255, el mismo en todos los nodos
    priority 110 # 110 para MASTER o 100 para BACKUP

    # The virtual ip address shared between the two NGINX
    # Web Server which will float
    virtual_ipaddress {
        192.168.100.167/24 # Esta IP debe ir en todos los nodos.
    }
    track_script {
        check_nginx
    }
    authentication {
        auth_type AH
        auth_pass secret
    }
}

root@Backup~# GNU nano 5.6.1 /etc/keepalived/keepalived.conf Modified
##### ALTA DISPONIBILIDAD #####
global_defs {
    # Keepalived process identifier
    router_id nginx # Se puede cambiar dependiendo del servicio
}

# Script to check whether Nginx is running or not
vrrp_script check_nginx {
    script "/bin/check_nginx.sh" # Verifica que el servicio funcione
    interval 2
    weight 50
}

# Virtual interface - The priority specifies the order in
# which the assigned interface to take over in a failover
vrrp_instance VI_01 {
    state BACKUP # Dependiendo el nodo se configura MASTER o BACKUP
    interface enp0s3 # Agregar la interfaz de red eth0, enp0s3
    virtual_router_id 50 # De 0-255, el mismo en todos los nodos
    priority 100 # 110 para MASTER o 100 para BACKUP

    # The virtual ip address shared between the two NGINX
    # Web Server which will float
    virtual_ipaddress {
        192.168.100.167/24 # Esta IP debe ir en todos los nodos.
    }
    track_script {
        check_nginx
    }
    authentication {
        auth_type AH
        auth_pass secret
    }
}
```

Se debe configurar en MASTER o en BACKUP respectivamente.

Con el comando nano /etc/keepalived/keepalived.conf entramos al archivo de configuración y se debe colocar el código respectivo en cada servidor.

```
root@Master~# GNU nano 5.6.1 /bin/check_nginx.sh Modified
#!/bin/sh
if [ -z "$(pidof nginx)" ]; then
    exit 1
fi

root@Backup~# GNU nano 5.6.1 /bin/check_nginx.sh Modified
#!/bin/sh
if [ -z "$(pidof nginx)" ]; then
    exit 1
fi
```

En el archivo nano /bin/check\_nginx.sh se debe configurar un pequeño código.



# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

PERÍODO ACADÉMICO: SEPTIEMBRE 2024 - FEBRERO 2025



Se debe otorgar permisos con cmod.

cmod 755 /bin/check\_nginx.sh

```
root@Master:~# service keepalived start
Redirecting to /bin/systemctl start keepalived.service
root@Master:~# service keepalived status
Redirecting to /bin/systemctl status keepalived.service
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; vendor preset: enabled)
   Active: active (running) since Fri 2024-10-18 17:04:41; 1min 1s ago
     Main PID: 12291 (keepalived)
       Tasks: 2 (limit: 10806)
      Memory: 7.0M
         CPU: 91ms
    CGroup: /system.slice/keepalived.service
            └─12291 /usr/sbin/keepalived --dont-fork -D
              └─12292 /usr/sbin/keepalived --dont-fork -D

Oct 18 17:04:44 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:44 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:44 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:49 Master Keepalived_vrrp[12292]: (VI_01) Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:49 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:49 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:49 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:49 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:49 Master Keepalived_vrrp[12292]: Sending gratuitous ARP for 10.10.10.10

root@Master:~# systemctl enable keepalived
Created symlink /etc/systemd/system/multi-user.target.wants/keepalived.service → /usr/lib/systemd/system/keepalived.service.
root@Master:~#
```

```
root@Backup:~# service keepalived start
Redirecting to /bin/systemctl start keepalived.service
root@Backup:~# service keepalived status
Redirecting to /bin/systemctl status keepalived.service
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; vendor preset: enabled)
   Active: active (running) since Fri 2024-10-18 17:04:39; 1min 1s ago
     Main PID: 12279 (keepalived)
       Tasks: 2 (limit: 10806)
      Memory: 7.0M
         CPU: 74ms
    CGroup: /system.slice/keepalived.service
            └─12279 /usr/sbin/keepalived --dont-fork -D
              └─12280 /usr/sbin/keepalived --dont-fork -D

Oct 18 17:04:43 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:43 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:43 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:48 Backup Keepalived_vrrp[12280]: (VI_01) Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:48 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:48 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:48 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:48 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10
Oct 18 17:04:48 Backup Keepalived_vrrp[12280]: Sending gratuitous ARP for 10.10.10.10

lines 1-21/21 (END)
systemctl enable keepalived
Created symlink /etc/systemd/system/multi-user.target.wants/keepalived.service → /usr/lib/systemd/system/keepalived.service.
root@Backup:~#
```

Iniciar y activar el servicio de keepalived:

service keepalived start

service keepalived status

systemctl enable keepalived

```
root@Master:~# firewall-cmd --add-rich-rule='rule protocol value="vrrp" accept' --permanent
success
root@Master:~# firewall-cmd --reload
success
root@Master:~#
```

```
root@Backup:~# firewall-cmd --add-rich-rule='rule protocol value="vrrp" accept' --permanent
success
root@Backup:~# firewall-cmd --reload
success
root@Backup:~#
```

Ing. Mg. Pablo I. Morales P.

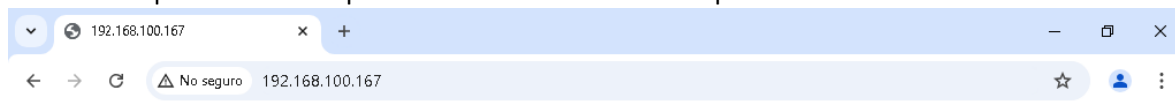


```
root@Master~# firewall-cmd --add-rich-rule='rule protocol value="vrrp" accept' --permanent
success
[root@Master ~]# firewall-cmd --reload
success
[root@Master ~]# service keepalived restart
Redirecting to /bin/systemctl restart keepalived.service
[root@Master ~]#

root@Backup~# firewall-cmd --add-rich-rule='rule protocol value="vrrp" accept' --permanent
success
[root@Backup ~]# firewall-cmd --reload
success
[root@Backup ~]#
[root@Backup ~]#
[root@Backup ~]# service keepalived restart
Redirecting to /bin/systemctl restart keepalived.service
[root@Backup ~]#
```

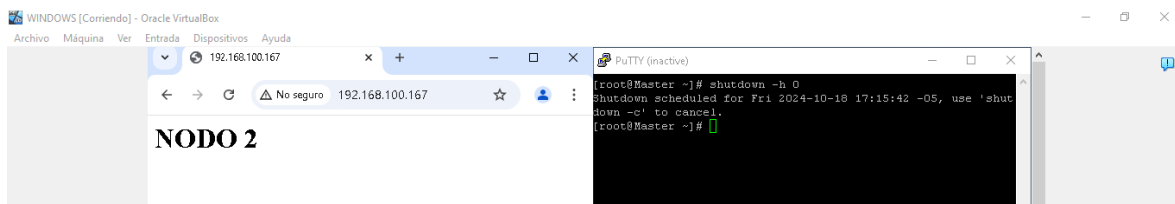
Dar permisos del firewall.

“service keepalived restart” para reiniciar el servicio de keepalived.



## NODO 1

Como se puede apreciar entrando a la dirección 192.168.100.167 aparece el nodo 1 que es de MASTER.



Al apagar el servidor MASTER