



Universidad Autónoma de Sinaloa

ADMINISTRACIÓN DE SISTEMAS

Nombre de la práctica:

Tarea 1: Entorno de Virtualización e Infraestructura Base

Grupo:

3-02

Alumno:

Montes Vázquez

Adrián Tadeo

Profesor:

Herman Geovany Ayala Zuñiga

Repositorio de GitHub:

<https://github.com/ADNTD1/Administracion-de-Sistemas-T1>

2. Introducción y Arquitectura

Objetivo:

El objetivo de esta práctica es implementar un entorno de red virtualizado compuesto por tres nodos funcionales:

- Linux (NixOs Minimal) como servidor
- Linux (NixOS) como cliente
- Windos Server 2022 (Minimal) como servidor

Asegurando la conectividad bidireccional mediante una red interna aislada y acceso a internet a través de NAT.

Asimismo, se busca preparar los sistemas operativos para tareas posteriores de automatización mediante el uso de scripts de diagnóstico inicial.

Arquitectura del Sistema

La arquitectura del laboratorio se basa en un hipervisor de virtualización (VMware Workstation), donde cada máquina virtual cuenta con dos adaptadores de red:

- **Adaptador NAT:** proporciona acceso a internet para la instalación y actualización de paquetes.
- **Adaptador de Red Interna (red_sistemas):** permite la comunicación directa y aislada entre los nodos a través de un comando ping para verificar la conectividad.

Los tres nodos se comunican exclusivamente por la red interna utilizando direccionamiento IP estático, lo que garantiza estabilidad y control del entorno.

Diagrama de topología de red:

En el diagrama se representa la conectividad de las máquinas virtuales agrupadas por la red interna y la que da salida a internet.

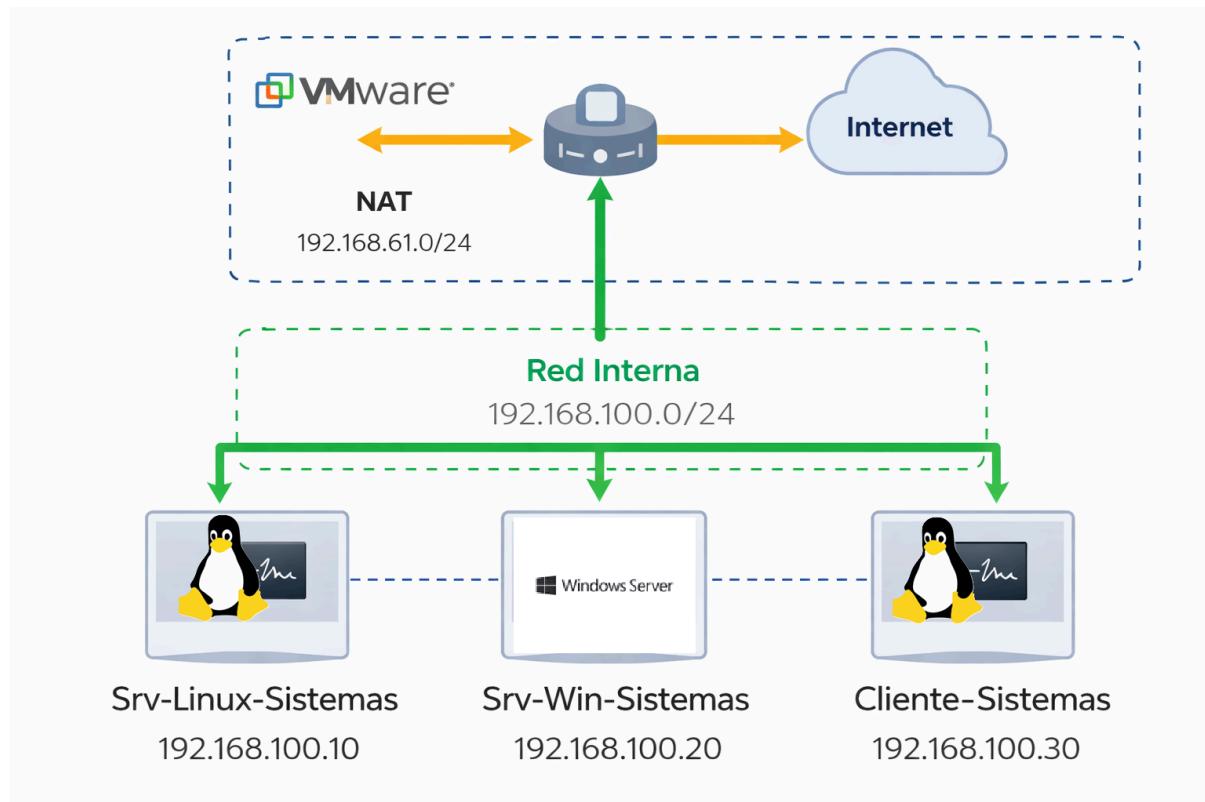


TABLA DE DIRECCIONAMIENTO DE IP:

Nodo	Sistema Operativo	Dirección IP	Máscara de Red
Srv-Linux-Sistemas	NixOS Server	192.168.100.30	255.255.255.0
Srv-Win-Sistemas	Windows Server 2022	192.168.100.20	255.255.255.0
Cliente-Sistemas	NixOs	192.168.100.10	255.255.255.0

3. Guía de Uso de los Scripts (Manual de Usuario)

Requisitos previos a la ejecución de los Scripts:

En NixOS (Minimal y con interfaz):

- Sistema operativo funcional con particiones de disco hechas.
- Haber configurado usuario y contraseña para acceder al modo Root/Super Usuario.
- Acceso a la terminal
- Tener instalado Nano en cualquier versión para la edición/creación de los archivos Bash.

En Windows Server 2022:

- Tener habilitado PowerShell
- Tener acceso a la terminal en Administrador.
- Habilitar la ejecución de scripts con una ExecutionPolicy.

Instrucciones de ejecución de los Scripts:

LINUX:

- Entrar a la carpeta donde estén los scripts ya creados:

```
[adrian@Srv-Linux-Sisremas:~/Scripts]$ cd ./Scripts/_
```

- Dar permisos de ejecución a el bash que se va a ejecutar:

```
[adrian@Srv-Linux-Sisremas:~/Scripts]$ chmod +x check_status.sh
```

- Ejecutarlo con ./ y el nombre del script:

```
[adrian@Srv-Linux-Sisremas:~/Scripts]$ ./check_status.sh
```

- Poner la contraseña del superusuario para poder ejecutarlo:

```
Srv-Linux-Sisremas login: adrian  
Password:
```

WINDOWS SERVER:

-Dar permisos para la ejecución de Scripts:

```
m>Set-ExecutionPolicy -Scope Process -ExecutionPolicy BypassS
```

-Ejecutarlo igual que en linux (dentro de el mismo directorio):

```
m>.\check_status.ps1S
```

Flujo de Interacción

Los scripts no solicitan datos al usuario.

Al ejecutarse, obtienen automáticamente la información del sistema y muestran en consola:

- Nombre del equipo.
- Dirección IP activa.
- Espacio disponible en disco.

Ejemplo:

```
PS C:\Users\Administrator\Desktop\Scripts> notepad check_status.ps1
PS C:\Users\Administrator\Desktop\Scripts> ./check_status.ps1
Maquina:
WIN-9M8V3M39GMK

Direccion IP:
169.254.183.107
Espacio del disco:

Name      Used (GB)    Free (GB) Provider      Root          CurrentLocation
----      -----      -----      -----      -----
C           6.78        52.59 FileSystem      C:\          Users\Administrator\Desktop\Scripts
D           4.70        0.00 FileSystem      D:\

PS C:\Users\Administrator\Desktop\Scripts> S_
```

BITACORA DEL DESARROLLO Y CONFIGURACION

Explicación del Script

Script en Linux (check_status.sh)

El script obtiene el nombre del equipo utilizando el comando hostname.

La dirección IP se extrae filtrando las interfaces IPv4 activas mediante ip addr.

El espacio en disco se consulta con df -h, mostrando únicamente la partición raíz para simplificar la salida.

Script en Windows (check_status.ps1)

El script utiliza la variable de entorno \$env:COMPUTERNAME para identificar el nombre del equipo.

La dirección IP se obtiene mediante Get-NetIPAddress, filtrando direcciones IPv4 activas.

El espacio en disco se consulta usando Get-CimInstance Win32_LogicalDisk, calculando el espacio libre y total en la unidad principal.

EVIDENCIAS DE CONFIGURACIÓN:

Adaptadores de red:

 Hard Disk (NVMe)	00 GB
 CD/DVD (SATA)	Using file C:\Users\wdcnm\Do...
 Network Adapter	Custom (VMnet0)
 Network Adapter 2	Host-only
 USB Controller	Present

Ejecución de los Scripts:

NixOS Server Minimal:

```
[adrian@Srv-Linux-Sisremas:~/Scripts]$ chmod +x check_status.sh
chmod: cannot access 'check_status.sh': No such file or directory
[adrian@Srv-Linux-Sisremas:~/Scripts]$ chmod +x check_status.sh

[adrian@Srv-Linux-Sisremas:~/Scripts]$ ./check_status.sh
-bash: ./check_status.sh: No such file or directory

[adrian@Srv-Linux-Sisremas:~/Scripts]$ ./check_status.sh
Maquina:
Srv-Linux-Sisremas
Direccion IP:
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    altname enp2s1
    altname enx000c294b5616
    inet 192.168.61.129/24 brd 192.168.61.255 scope global dynamic noprefixroute ens33
        valid_lft 1294sec preferred_lft 1294sec
Espacio del disco:
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        20G   2.1G   16G  12% /
[adrian@Srv-Linux-Sisremas:~/Scripts]$
```

NixOs:

```
[adrian@nixos:~/Escritorio/Scripts]$ chmod +x Check_status.sh
[adrian@nixos:~/Escritorio/Scripts]$ ./Check_status.sh
Maquina:
nixos
Direccion ip:
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    altname enp2s1
    altname enx000c297a0c60
    inet 192.168.61.128/24 brd 192.168.61.255 scope global dynamic noprefixroute ens33
        valid_lft 1302sec preferred_lft 1302sec
Espacio del disco:
S.ficheros      Tamaño Usados  Disp Uso% Montado en
/dev/sda1        20G     9.8G   8.8G  53% /
[adrian@nixos:~/Escritorio/Scripts]$
```

Windows Server 2022:

```
PS C:\Users\Administrator\Desktop\Scripts> notepad check_status.ps1
PS C:\Users\Administrator\Desktop\Scripts> ./check_status.ps1
Maquina:
WIN-9M8V3M39GMK

Direccion IP:
169.254.183.107
Espacio del disco:

Name      Used (GB)    Free (GB) Provider      Root          CurrentLocation
----      -----      -----      -----      -----
C           6.78       52.59 FileSystem    C:\           Users\Administrator\Desktop\Scripts
D           4.70       0.00 FileSystem    D:\          

PS C:\Users\Administrator\Desktop\Scripts> S-
```

Pruebas de funcionamiento (Ping):

Ping entre NixOs Cliente y Windows Server 2022:

```
vatty_ttt forever preferred_ttt forever

[adrian@nixos:~/Escritorio/Scripts]$ ping 192.168.100.20
PING 192.168.100.20 (192.168.100.20) 56(84) bytes de datos.
64 bytes desde 192.168.100.20: icmp_seq=1 ttl=128 tiempo=1.29 ms
64 bytes desde 192.168.100.20: icmp_seq=2 ttl=128 tiempo=0.920 ms
64 bytes desde 192.168.100.20: icmp_seq=3 ttl=128 tiempo=1.32 ms
64 bytes desde 192.168.100.20: icmp_seq=4 ttl=128 tiempo=1.29 ms
^C
--- 192.168.100.20 estadísticas ping ---
4 paquetes transmitidos, 4 recibidos, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 0.920/1.204/1.319/0.164 ms

[adrian@nixos:~/Escritorio/Scripts]$
```

Ping entre Windows Server 2022 y NixOs Cliente:

```
Default Gateway . . . . . :
PS C:\Users\Administrator\Desktop\Scripts> ping 192.168.100.10

Pinging 192.168.100.10 with 32 bytes of data:
Reply from 192.168.100.10: bytes=32 time=1ms TTL=64
Reply from 192.168.100.10: bytes=32 time<1ms TTL=64
Reply from 192.168.100.10: bytes=32 time<1ms TTL=64
Reply from 192.168.100.10: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.100.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\Administrator\Desktop\Scripts>
```

Ping entre NixOs server y NixOs cliente:

```
[adrian@Srv-Linux-Sisremas:~/Scripts]$ ping -c 4 192.168.85.128
PING 192.168.85.128 (192.168.85.128) 56(84) bytes of data.
64 bytes from 192.168.85.128: icmp_seq=1 ttl=64 time=4.09 ms
64 bytes from 192.168.85.128: icmp_seq=2 ttl=64 time=7.55 ms
64 bytes from 192.168.85.128: icmp_seq=3 ttl=64 time=8.10 ms
64 bytes from 192.168.85.128: icmp_seq=4 ttl=64 time=1.06 ms

--- 192.168.85.128 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 1.061/5.199/8.098/2.841 ms

[adrian@Srv-Linux-Sisremas:~/Scripts]$
```

Ping entre NixOs cliente y NixOs server:

```

rtt min/avg/max/mdev = 0.920/1.204/1.319/0.164 ms

[adrian@nixos:~/Escritorio/Scripts]$ ping 192.168.100.30
PING 192.168.100.30 (192.168.100.30) 56(84) bytes de datos.
64 bytes desde 192.168.100.30: icmp_seq=1 ttl=64 tiempo=0.966 ms
64 bytes desde 192.168.100.30: icmp_seq=2 ttl=64 tiempo=1.20 ms
64 bytes desde 192.168.100.30: icmp_seq=3 ttl=64 tiempo=0.652 ms
64 bytes desde 192.168.100.30: icmp_seq=4 ttl=64 tiempo=0.886 ms
^C
--- 192.168.100.30 estadísticas ping ---
4 paquetes transmitidos, 4 recibidos, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 0.652/0.926/1.201/0.196 ms

[adrian@nixos:~/Escritorio/Scripts]$ 

```

Ping entre WindowsServer y NixOS server:

```

Default Gateway . . . . .
PS C:\Users\Administrator\Desktop\Scripts> ping 192.168.100.30

Pinging 192.168.100.30 with 32 bytes of data:
Reply from 192.168.100.30: bytes=32 time=1ms TTL=64
Reply from 192.168.100.30: bytes=32 time<1ms TTL=64
Reply from 192.168.100.30: bytes=32 time=1ms TTL=64
Reply from 192.168.100.30: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.100.30:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\Administrator\Desktop\Scripts>

```

Ping entre NixOs Server y Windows Server 2022:

```

[adrian@Srv-Linux-Sisremas:~/Scripts]$ ping -c 4 192.168.100.20
PING 192.168.100.20 (192.168.100.20) 56(84) bytes of data.
64 bytes from 192.168.100.20: icmp_seq=1 ttl=128 time=1.49 ms
64 bytes from 192.168.100.20: icmp_seq=2 ttl=128 time=1.47 ms
64 bytes from 192.168.100.20: icmp_seq=3 ttl=128 time=1.08 ms
64 bytes from 192.168.100.20: icmp_seq=4 ttl=128 time=0.967 ms

--- 192.168.100.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 0.967/1.252/1.493/0.234 ms

[adrian@Srv-Linux-Sisremas:~/Scripts]$ 

```

CONCLUSION:

Como conclusion quiero recalcar que la configuracion del Linux Server sin interfaz estuvo muy tardado (4-5 Horas aproximadamente), lo demas estuvo relativamente sencillo, solo era desactivar Firewall de windows para que no rechace la solicitudes del ping y lo demas estuvo mas sencillo. Con esta practica ademas se reforzo el uso de PowerShell y de Bash

LECCIONES APRENDIDAS:

Durante la práctica se identificaron y resolvieron problemas relacionados con el direccionamiento IP, configuración del firewall de Windows y la diferencia entre configuraciones temporales y persistentes en NixOS.

Se reforzó el uso de PowerShell y Bash como herramientas fundamentales para la administración de sistemas y la automatización inicial.

REFERENCIAS:

Documentación oficial de NixOS: <https://nixos.org/manual>

Microsoft Learn – Windows Server: <https://learn.microsoft.com>

NixOs:

<https://www.youtube.com/watch?v=T1NH2LuGx5w>

<https://www.youtube.com/watch?v=lUB2rwDUm5A>

IAs Utilizadas:

ChatGPT (OpenAI) para aclaración de conceptos y apoyo en redacción técnica.

Gemini para resolver errores con los scripts.

Copilot para ayuda en la instalacion de los sistemas operativos.