



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERÍA MOCHIS

Ingeniería de Software

Alumno:

Cervantes Gil Ángel Alfredo

Profesor:

Dr. Herman Geovany Ayala Zuñiga

Grupo: 3-02

Materia:

Administración de sistemas

Practica:

P1. Entorno de Virtualización e Infraestructura Base

Historial de cambios

Version	Fecha	Descripcion del cambio	Commit
V1.0	03-Feb-2025	Creacion del repositorio, Cambio a README.md	cambio aplicado a README.md, contiene informacion acerca de la practica realizada. Breve descripcion, topologia, direccionamiento de IP y el status de algunas tareas
V1.1	03-Feb-2025	Se agrego el script de .sh al repositorio	Se agrego a scripts el script check_status.sh, se comprobo que funciona de una manera correcta y limpia
V1.2	04-Feb-2025	Se agrego el script de .ps1 al repositorio	add check_status.ps1 script powershell
V1.3	04-Feb-2025	Correccion de ortografia en README.md	correccion de ortografia en README.md
V1.4	04-Feb-2025	Se añadió el doc 1 de 2 al repositorio	add doc 1/2 to files

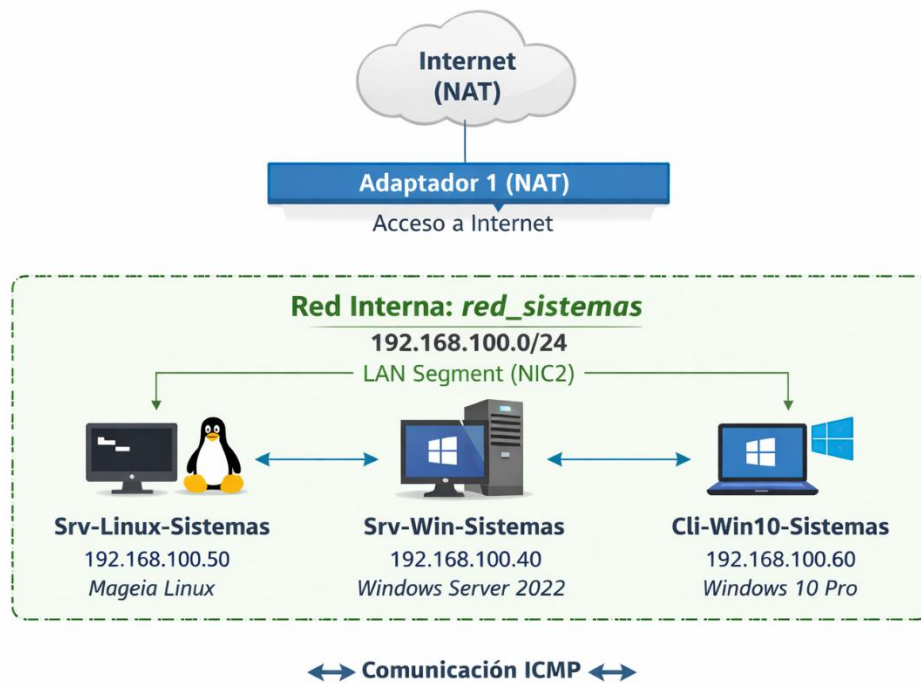
Introducción

En esta práctica se implementa un entorno de red virtualizada compuesto por tres nodos (Linux Server, Windows Server y Windows Cliente), con el objetivo de garantizar conectividad interna aislada y preparar los sistemas para tareas de automatización y diagnóstico. La arquitectura emplea doble adaptador de red para separar el tráfico interno del acceso a Internet.

Objetivo

Configurar y validar un entorno virtualizado con comunicación bidireccional mediante ICMP, manteniendo acceso a Internet vía NAT, y desarrollar scripts de diagnóstico que muestren el estado básico de cada nodo.

Topología – RED INTERNA VIRTUAL



Guia de uso de los scripts.

Esta sección describe los requisitos y el procedimiento necesario para ejecutar los scripts de diagnóstico inicial incluidos en el proyecto. Su propósito es permitir que cualquier administrador pueda verificar rápidamente el estado básico de cada nodo sin necesidad de revisar manualmente la configuración del sistema.

Requisitos previos

En Linux (Mageia):

- Sistema operativo Linux Mageia instalado.
 - Conectividad de red funcional.
 - Acceso a una terminal.
- Permisos de superusuario (root o sudo).
- Herramientas básicas del sistema (ip, df, hostname) instaladas por defecto.
 - Descargar paquetes esenciales como nano.

En Windows (Server 2022 / Windows 10 Pro)

- PowerShell ejecutado con permisos de Administrador.
- Política de ejecución configurada para permitir scripts: Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
 - Conectividad de red funcional.

Instrucciones de ejecución.

Mageia

1. Acceder al directorio donde se encuentra el archivo con terminación sh.. Ej -> cd scripts/check_status.sh.
2. Asignar permisos de ejecución(solo la primera vez) -> chmod +x check_status.sh.
3. Ejecutar el script -> ./check_status.sh.

Windows 10 Pro / Windows Server 2022

Aplica para ambos

1. Ejecutar PowerShell como administrador.
2. Acceder al Directorio del script -> cd scripts/check_status.ps1
3. Ejecutar el script .\check_status.ps1.

Flujo de interaccion

Los scripts no requieren interacción del usuario ni solicitan datos de entrada.

Al ejecutarse, realizan automáticamente las siguientes acciones:

1. Obtienen el nombre del equipo (hostname).
2. Identifican las direcciones IP configuradas en el sistema.
3. Muestran el espacio disponible en disco.

Toda la información se presenta directamente en la consola, permitiendo un diagnóstico rápido del estado del sistema.

Bitácora de Desarrollo y Configuración

Esta sección describe la lógica de funcionamiento de los scripts y las decisiones técnicas tomadas durante su desarrollo.

Explicación del Script en Linux (check_status.sh)

El script de Linux fue desarrollado en Bash y sigue una estructura simple y modular:

- Se utiliza el comando hostname para identificar el nombre del equipo.
- El comando ip permite obtener las direcciones IPv4 activas del sistema, mostrando la interfaz asociada.
- El comando df -h se emplea para consultar el uso de espacio en disco de la partición raíz (/) de forma legible.

El objetivo principal del script es ofrecer una vista rápida del estado del servidor, sin modificar la configuración del sistema ni requerir dependencias externas.

Explicación del Script en Windows (check_status.ps1)

El script de Windows fue desarrollado en PowerShell utilizando cmdlets nativos del sistema:

- La variable de entorno \$env:COMPUTERNAME se usa para obtener el nombre del equipo.
- El cmdlet Get-NetIPAddress permite listar las direcciones IPv4 configuradas en las interfaces de red.
- El cmdlet Get-PSDrive se emplea para mostrar el espacio utilizado y disponible en las unidades de almacenamiento.

El script está diseñado para ser seguro, informativo y no interactivo, facilitando su uso en entornos administrativos y educativos.

N	Prueba	Entrada	Salida Esperada	Salida Obtenida
1	Ping Linux → Windows Server	ping 192.168.100.40 ejecutado desde Mageia	Respuesta ICMP exitosa sin pérdida de paquetes	Respuesta ICMP exitosa
2	Ping Linux → Windows 10	ping 192.168.100.60 ejecutado desde Mageia	Respuesta ICMP exitosa sin pérdida de paquetes	Respuesta ICMP exitosa
3	Ping Windows Server → Linux	ping 192.168.100.50 ejecutado desde Windows Server	Respuesta ICMP exitosa sin pérdida de paquetes	Respuesta ICMP exitosa
4	Ping Windows 10 → Linux	ping 192.168.100.50 ejecutado desde Windows 10	Respuesta ICMP exitosa sin pérdida de paquetes	Respuesta ICMP exitosa
5	Ejecución de script en Linux	<code>./check_status.sh</code>	Muestra nombre del equipo, IP actual y espacio en disco	Información mostrada correctamente
6	Ejecución de script en Windows	<code>.\check_status.ps1</code>	Muestra nombre del equipo, IP actual y espacio en disco	Información mostrada correctamente

Tabla de "Entrada - Salida Esperada - Salida Obtenida".

Prueba N1

```
Inicio X Windows 10 x64 X Windows Server 2022 (2) X Srv-Linux-... X
[root@Srv-Linux-Sistemas scripts (develop)]# ping -c 4 192.168.100.40
PING 192.168.100.40 (192.168.100.40) 56(84) bytes of data.
64 bytes from 192.168.100.40: icmp_seq=1 ttl=128 time=11.1 ms
64 bytes from 192.168.100.40: icmp_seq=2 ttl=128 time=2.51 ms
64 bytes from 192.168.100.40: icmp_seq=3 ttl=128 time=3.95 ms
64 bytes from 192.168.100.40: icmp_seq=4 ttl=128 time=3.71 ms

--- 192.168.100.40 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 2.511/5.320/11.118/3.390 ms
[root@Srv-Linux-Sistemas scripts (develop)]#
```

Prueba N2

```
Inicio X Windows 10 x64 X Windows Server 2022 (2) X Srv-Linux-... X
[root@Srv-Linux-Sistemas scripts (develop)]# ping -c 4 192.168.100.60
PING 192.168.100.60 (192.168.100.60) 56(84) bytes of data.
64 bytes from 192.168.100.60: icmp_seq=1 ttl=128 time=3.41 ms
64 bytes from 192.168.100.60: icmp_seq=2 ttl=128 time=1.10 ms
64 bytes from 192.168.100.60: icmp_seq=3 ttl=128 time=2.17 ms
64 bytes from 192.168.100.60: icmp_seq=4 ttl=128 time=1.15 ms

--- 192.168.100.60 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 1.096/1.958/3.412/0.941 ms
[root@Srv-Linux-Sistemas scripts (develop)]# _
```

Prueba N3

```
PS C:\Users\Administrator> ping 192.168.100.50

Pinging 192.168.100.50 with 32 bytes of data:
Reply from 192.168.100.50: bytes=32 time=2ms TTL=64
Reply from 192.168.100.50: bytes=32 time=10ms TTL=64
Reply from 192.168.100.50: bytes=32 time=2ms TTL=64
Reply from 192.168.100.50: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.100.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 10ms, Average = 3ms
PS C:\Users\Administrator> _
```


Prueba 4

```
PS C:\Windows\system32> ping 192.168.100.50

Pinging 192.168.100.50 with 32 bytes of data:
Reply from 192.168.100.50: bytes=32 time=1ms TTL=64
Reply from 192.168.100.50: bytes=32 time=9ms TTL=64
Reply from 192.168.100.50: bytes=32 time=1ms TTL=64
Reply from 192.168.100.50: bytes=32 time=9ms TTL=64

Ping statistics for 192.168.100.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 9ms, Average = 5ms
PS C:\Windows\system32>
```

Prueba 5

```
[root@Srv-Linux-Sistemas scripts (develop)]# ./check_status.sh

-----
                Angel Distro Mageia MSG Bienvenida - LINUX
-----

Nombre del equipo:
Srv-Linux-Sistemas

IP actual:
- 127.0.0.1/8 (lo)
- 192.168.222.129/24 (ens33)
- 192.168.100.50/24 (ens34)

Espacio en disco:
S.ficheros      Tamaño Usados  Disp Uso% Montado en
/dev/sda1        17G   4.0G   12G  25% /

-----

[root@Srv-Linux-Sistemas scripts (develop)]#
```

Prueba 6

```
PS C:\Users\Win Cliente SRV\Desktop\redes-sistemas-virtualizadas\scripts> ./check_status.ps1

-----
                Script de bienvenida - Windows
-----

Nombre del equipo:
CLI-WIN10-SISTE

IP actual:

InterfaceAlias  IPAddress          PrefixLength
-----
Ethernet1       192.168.100.60     24
Ethernet0       192.168.222.131    24

Espacio en disco:

Name Used (GB) Free (GB) Total (GB)
-----
C:    28        31.42    59.41
D:    4.56       0        4.56
```

Conclusión

El aprendizaje en la realización de esta practica fue basto ya que comencé a utilizar un nuevo sistema operativo, me familiaricé con los comandos tanto en Windows como en Linux (Mageia) después de tantos errores y correcciones, aprendí como funcionan las redes, como interactúan por ejemplo que la NAT me daba conexión directamente a internet sin afectar al adaptador de red2, que era donde están las maquinas conectadas entre si.

Aprendi que hacer y que no, y como manejar el repositorio de github entre cada sistema en la VM y como me facilitaba la compartición de archivos mas la edición.

Fuentes consultadas

<https://learn.microsoft.com/powershell/>

<https://learn.microsoft.com/powershell/module/microsoft.powershell.management/get-psdrive>

<https://learn.microsoft.com/powershell/module/nettcpip/get-netipaddress>

Linux Foundation. (s. f.). *Linux Manual Pages*

<https://man7.org/linux/man-pages/>

Mageia Project. (s. f.). *Mageia Documentation*.

<https://wiki.mageia.org/>

ChatGPT (openAI)

Utilizado como apoyo para:

- Resolución de errores en scripts Bash y PowerShell
 - Explicación de comandos Git y SSH
 - Corrección de sintaxis y buenas prácticas