Universidad Autónoma de Sinaloa

ADMINISTRACIÓN DE SISTEMAS

Nombre de la práctica:
Tarea 2: Automatización y Gestión del Servidor DHCP

Grupo:

3-02

Alumno:

Montes Vázquez Adrián Tadeo

Profesor:

Herman Geovany Ayala Zuñiga

Repositorio de GitHub:

https://github.com/ADNTD1/Administracion-de-Sistemas-T2

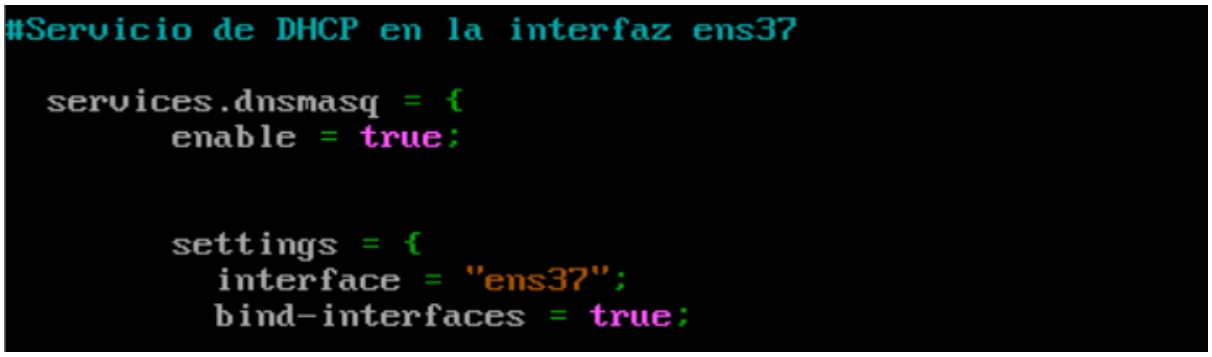
ENTREGABLES:

1. Implementación de la Lógica de Instalación (Idempotencia)

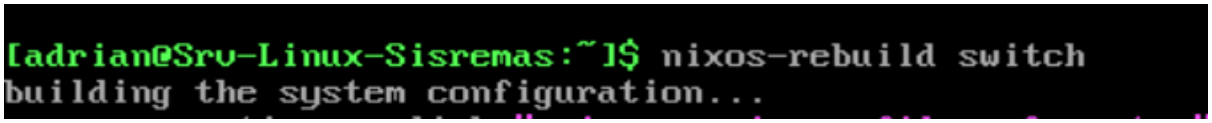
Se desarrolló un script de automatización que implementa una lógica idempotente para la instalación y habilitación del servicio DHCP. El script verifica de forma autónoma la presencia del servicio antes de realizar cualquier acción, evitando instalaciones duplicadas o configuraciones inconsistentes.

Linux (NixOS):

En distribuciones Linux basadas en NixOS, el demonio isc-dhcp-server fue retirado por fin de vida (EOL). Por este motivo, se implementó el servicio dnsmasq, el cual cumple la misma función de asignación dinámica de direcciones IP y es compatible con la arquitectura declarativa de NixOS**.**



Esta captura es del archivo “Configuration.nix” donde se crea soporte para el servicio y se le asigna la interfaz de red en la cual va a trabajar.



Después de aplicar la configuración y guardarla se tiene que rebuldear por que se cambió la configuración, con eso el servicio DHCP ya está habilitado y listo para poder configurarse.

2. Orquestación de Configuración Dinámica:

Linux:

La automatización implementada no depende de valores fijos. El script solicita de manera interactiva al administrador los parámetros necesarios para la configuración del servicio DHCP, validando que las direcciones IP ingresadas cumplan con el formato IPv4.

Los parámetros solicitados son:

-Scope

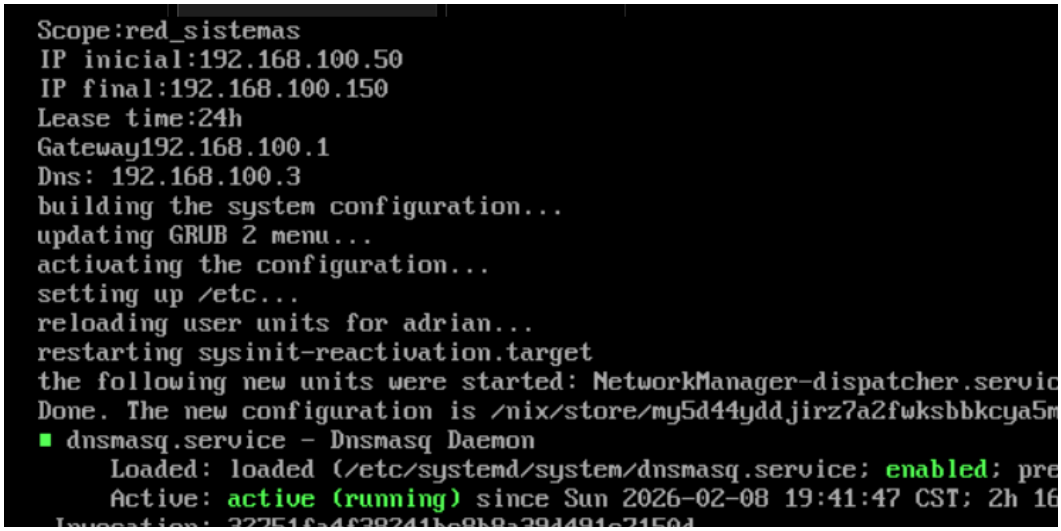
-IP inicial y final (rango)

-Lease time

-Gateway

-DNS

Ejecución del script donde se piden estos parámetros:



Para la validacion de los parametros se uso un script aparte donde se contenian las funciones que se usaban para validar llamado net_utils.sh:

```
set  -euo pipefail

die() {
  echo "error: $*" >&2
  exit 1
}



valid_ipv4() {
  local ip="$1"
  local a b c d

  IFS='.' read -r a b c d <<< "$ip"

 [[  -n "${a:-}" && -n "${b:-}" && -n "${c:-}" && -n "${d:-}" ]] || return 1
 [[ "$a" =~ ^[0-9]+$ && "$b" =~ ^[0-9]+$ && "$c" =~ ^[0-9]+$ && "$d" =~ ^[0-9]+$ ]] || return 1



 (( a >= 0 && a <= 255)) || return 1
 (( b >= 0 && b <= 255)) || return 1
 (( c >= 0 && c <= 255)) || return 1
 (( d >= 0 && d <= 255)) || return 1

 return 0;

}



ip_to_int() {
   local ip="$1"
   IFS='.' read -r a b c d <<< "$ip"
   echo $(( (a << 24) + (b << 16) + (c<<8) +d))
}


require_root() {
   [[ "${EUID:-$(id -u)}" -eq 0 ]] || die "Ejecuta como root (sudo)"
}
```

El script ejecutado para inciar el servicio dinamicamente se llama dhcp_linux_nixos.sh

```
set -euo pipefail
source "./net_utils.sh"
require_root

IFACE="ens37"
NETMASK="255.255.255.0"
NIX_DHCP_FILE="/etc/nixos/dhcp-dnsmasq.nix"
NIX_MAIN_FILE="/etc/nixos/configuration.nix"

read -r -p "Scope:" SCOPE
read -r -p "IP inicial:" IP_START

valid_ipv4 "$IP_START" || die "IP inicial Invalida"

read -r -p "IP final:" IP_END
valid_ipv4 "$IP_END" || die "IP final invalida"

read -r -p "Lease time:"  LEASE
read -r -p "Gateway" GATEWAY
valid_ipv4 "$GATEWAY" || die "Gateway invalido"

read -r -p "Dns: " DNS
valid_ipv4 "$DNS" || die "DNS invalido"

cat > /tmp/dhcp-dnsmasq.nix << EOF
{ config, pkgs, ...  }:
{
    services.dnsmasq = {
        enable = true;
        settings = {
          interface = "${IFACE}";
          bind-interfaces = true;
          dhcp-range = "${IP_START},${IP_END},${NETMASK},${LEASE}";
          dhcp-option = [ "3,$GATEWAY" "6,${DNS}" ];
          port = 0;
        };
      };
    networking.firewall.allowedUDPPorts = [67];
}
EOF

mv /tmp/dhcp-dnsmasq.nix "$NIX_DHCP_FILE"

grep -q "dhcp-dnsmasq.nix" "$NIX_MAIN_FILE" || \
```

Dividiendo el script en 2 partes era mas facil saber en que me habia equivocado al hacerlos y ejecutarlos.

3. Módulo de Monitoreo y Validación de Estado:

Para monitorear el servidor y las conexiones del mismo se uso un script llamado dhcp_monitor con el cual se lista las direcciones mac y el nombre del host el cual solicitó la ip.

Antes de usar este script se usa un comando que le pregunta al systemd si el servicio dnsmasq esta activo:

```
[adrian@Srv-Linux-Sisremas:~/Scripts/DHCP]$ systemctl is-active dnsmaq
```

con esto se puede saber si el servicio esta activo en tiempo real.

Concesiones Activas:

Una concesión es un registro de que el cliente o la maquina cliente pidio una ip, el servidor se la asigno y esa misma ip sigue reservada por el tiempo especificado (24h)

Listear los Leases te permite:

-saber que equipos estan conectados

-verificar si el servidor esta asignado ips

-comprobar que los parámetros se estan entregando

-saber el host el cual esta conectado.

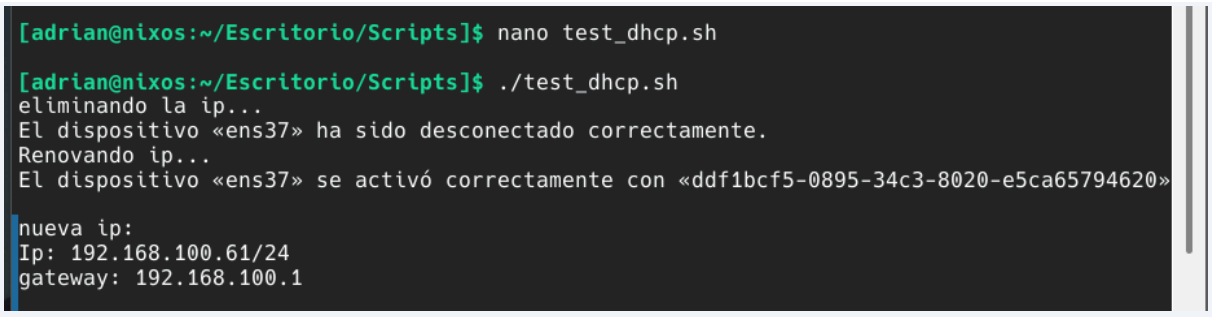
Para identificar equipos conectados y validar concesiones, se implementó un script de monitoreo (dhcp_monitor.sh) que analiza los registros del servicio dnsmasq en el journal del sistema. El script filtra eventos DHCPACK, que representan la confirmación de asignación de IP, mostrando la dirección IP otorgada y el hostname del cliente

```
[adrian@Srv-Linux-Sisremas:~/Scripts/DHCP]$ ./dhcp_monitor.sh
estado del servicio:
active
concesiones activas:
IP: 192.168.100.61 Host: nixos

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

Prueba del cliente:

En la prueba se ejecuta un script el desconecta y conecta la interfaz de red para resetear la ip que se le asigno anteriormente.



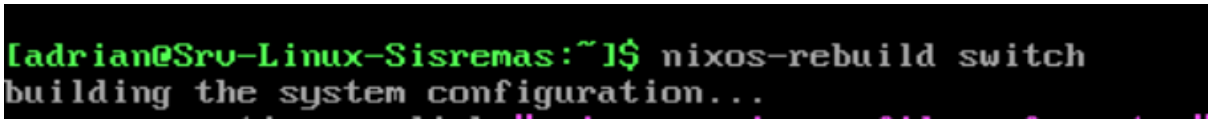
Como se puede observar se le asigna la ip que se observa arriba en el monitor de red.

Comandos Básicos para a practica:

CONFIGURACION:

Archivos de configuracion:
nano /etc/configuration.nix
Con este comando se accede a la configuracion para habilitar el servicio DHCP.

Rebuild de el sistema:



Se usa para aplicar los cambios hechos en el archivo de configuracion.

ESTADO DEL SERVICIO:

Verificar el estado del servicio:
Systemctl Status Dnsmasq

LOGS Y CONCESIONES:

Journalctl -u dnsmasq

Para saber que concesiones se establecieron con el servicio DHCP.

MONITOREO MEDIANTE EL SCRIPT:

Salida:

```
[adrian@Srv-Linux-Sisremas:~/Scripts/DHCP]$ ./dhcp_monitor.sh
estado del servicio:
active
concesiones activas:
IP: 192.168.100.61 Host: nixos

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

Script:

```
set -euo pipefail
SERVICE="dnsmasq"


echo "estado del servicio:"
systemctl is-active "$SERVICE" || echo "Servicio apagado"
echo "concesiones activas: "

journalctl -u "$SERVICE" --no-pager  | \
grep DHCPACK | \
awk '{print "IP:", $8, "Host:",  $NF}' | \
sort -u || echo "no hay conexiones"
```

Windows Server:

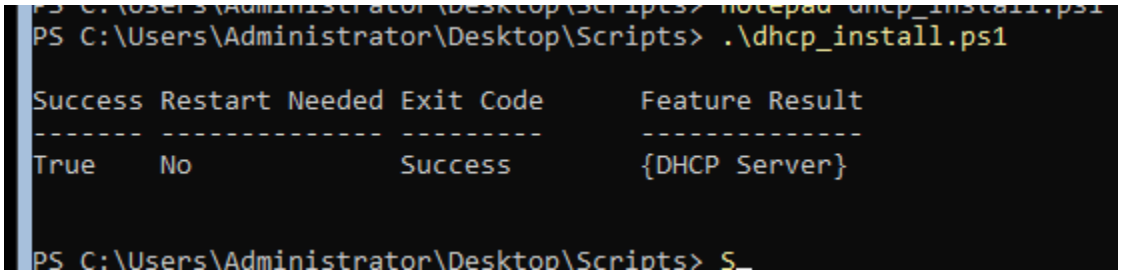
Para el entorno Windows se desarrollo un script en PowerShell que implementa una lógica idempotente para la instalación del rol DHCP Server. La idempotencia garantiza que el script pueda ejecutarse múltiples veces sin provocar instalaciones duplicadas o configuraciones inconsistentes.

Para instalar el paquete necesario para la creacion del servidor DHCP se uso este script;

```
File  Edit  Format  View  Help
$feature = Get-WindowsFeature -Name DHCP

if (-not $feature.Installed) {
        Install-WindowsFeature -Name DHCP -IncludeManagementTools
} else {
    Write-Host "dhcp ya instalado"
}
```

Al ejecutarse el script dio de salida:



donde dice que la feature se instaló correctamente.

2. Orquestación de Configuración Dinámica

La automatización implementada no se limita a valores estáticos. Durante la ejecución del script, el administrador introduce de forma interactiva los parámetros necesarios para la configuración del servicio DHCP, lo que permite adaptar la solución a distintos entornos de red sin modificar el código.

Los parámetros solicitados son:

-Nombre descriptivo del ámbito (Scope).

-Dirección IP inicial del rango DHCP.

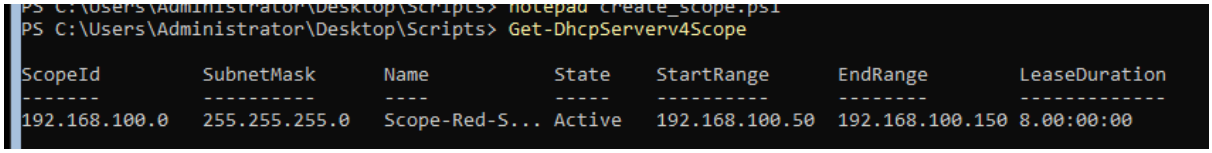
-Dirección IP final del rango DHCP.

-Puerta de enlace predeterminada (Gateway).

-Servidor DNS

-Tiempo de concesión (Lease Time) expresado en días.
Con estos datos, el script crea dinámicamente un ámbito IPv4 correspondiente al segmento 192.168.100.0/24, asegurando que las direcciones entregadas se encuentren dentro del rango definido.
El tiempo de concesión se establece utilizando un objeto time span lo que permite un control preciso del lease.

Con este comando se puede saber los parametros del scope;



para la creacion del servicio se uso este script:

```
$feature = Get-WindowsFeature DHCP
if(-not $feature.Installed) {
        Install-WondowsFeature DHCP -IncludeManagementTools
}

Add-DhcpServerSecurityGroup | Out-Null
Restart-Service DHCPServer

$scopeName = Read-Host "Nombre del scope: "
$startIP = Read-Host "IP Inicial: "
$endIP = Read-Host "IP final: "
$gateway = Read-Host "Gateway: "
$dns = Read-Host "dns: "
$leaseDays = [int](Read-Host "Lease: ")

$ScopeID = "192.168.100.0"

if(-not(Get-DhcpServerv4Scope -ErrorAction SilentlyContinue | Where-Object {$_.ScopeId -eq $scopeId})) {
        Add-DhcpServerv4OScope -Name $scopeName -StartRange $startIP -EndRange $endIP -SubnetMask 255.255.255.0 -LeaseDuratio
}
Set-DhcpServerv4OptionValue -ScopeId $scopeId -OptionId 3 -Value $gateway

try {
    Set-DhcpServerv4OptionValue -ScopeId $scopeId -OptionId 6 -Value $dns
} catch {
        Write-Host "Dns no aplicado"
}

Set-DhcpServerv4OptionValue -ScopeId $scopeId -State Active
Get-Service DHCPServer
Get-DhcpServerv4Lease -ScopeId $scopeId
```

en el se especifica los parámetros y con ello se inicia el servicio con los mismos parámetros proporcionados a traves de la consola.

3. Activación y Preparación del Servicio DHCP

Una vez creado el ámbito, el script activa explícitamente el scope para permitir la entrega de direcciones IP a los clientes.
Adicionalmente, se ejecutan tareas de post-instalación necesarias, como la creación de grupos de seguridad del servicio DHCP y el reinicio del servicio, garantizando que el servidor quede operativo.

4. Módulo de Monitoreo y Validación de Estado

Para cumplir con el módulo de monitoreo, se desarrolló un script independiente de diagnóstico que permite al administrador verificar el estado del servicio DHCP y las concesiones activas en tiempo real.

Este módulo realiza las siguientes funciones:

-Consulta del estado del servicio DHCP mediante PowerShell.

-Listado de concesiones DHCP activas utilizando el cmdlet get-dhcpservervLease

-Visualización de información relevante de los clientes conectados, como dirección IP asignada, identificador del cliente (MAC), nombre del host y tiempo de expiración de la concesión.

El módulo de monitoreo no solicita parámetros al usuario, ya que opera sobre la configuración previamente aplicada, permitiendo una validación rápida y no intrusiva del estado del servidor.

Para el monitoreo se uso un script llamado Dhcp_monitor.ps1

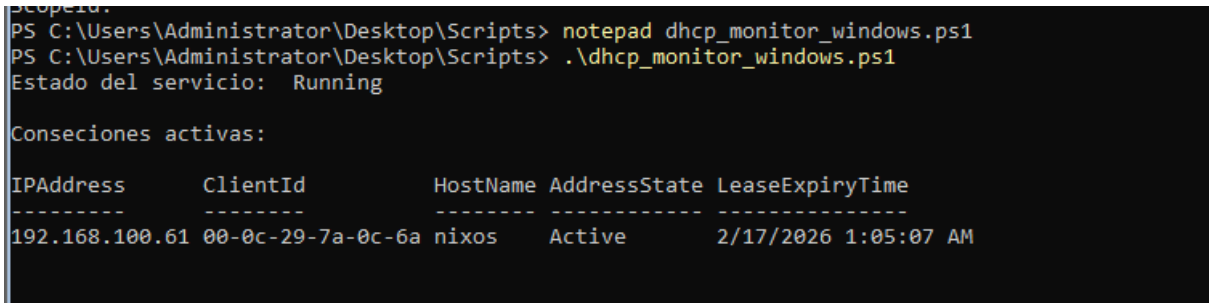
```
$ScopeID = "192.168.100.0"

$svc = Get-Service DHCPServer
Write-Host "Estado del servicio: " $svc.Status
Write-Host ""

Write-Host "Conseciones activas: "
Get-DhcpServerv4Lease -ScopeId $ScopeId |
Select-Object IPAddress, ClientId, HostName, AddressState, LeaseExpiryTime |
Format-Table -AutoSize
```

ahi se especifica el scopeID y los parametros que vamos a necesitar para monitorear.

La salida de datos del monitor con un cliente conectado es esta:

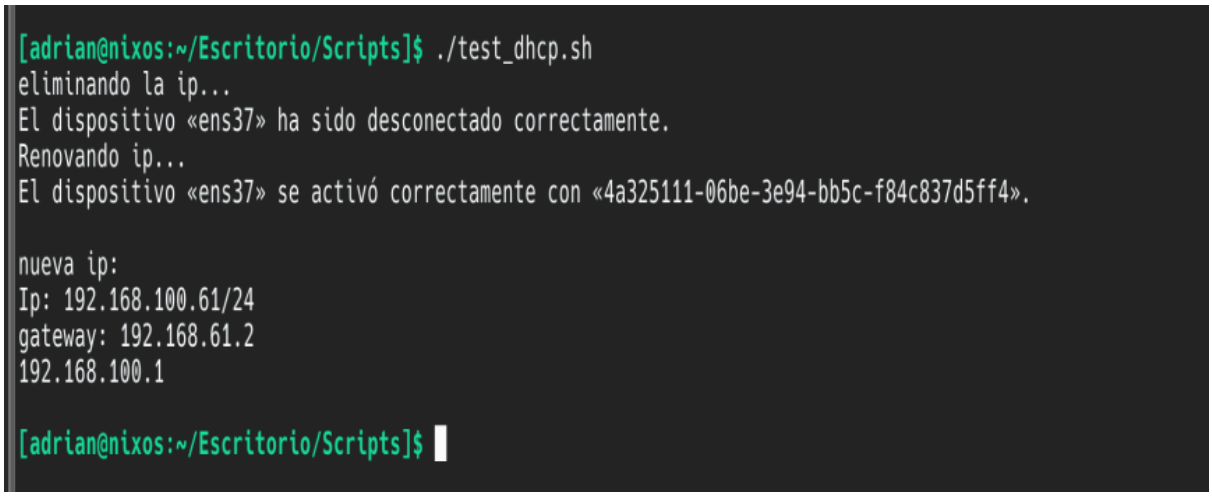


5. Prueba de Cliente (Release / Renew)

Como parte de la validación del funcionamiento del servidor DHCP, se realizó una prueba de cliente mediante la renovación forzada de la configuración de red.

Desde el equipo cliente se ejecutó un proceso de renovar la ip, provocando que el cliente solicitara nuevamente parámetros de red al servidor DHCP.
El resultado fue la asignación exitosa de una dirección IP dentro del rango configurado, junto con la entrega de la puerta de enlace predeterminada.

en este caso se uso un script que automariza este proceso en mi linux:



COMANDOS BASICOS PARA LA PRACTICA:

Get-WindowsFeature -Name DHCP : con este se verifica que el servicio esta instalado.

Get-Service DHCPServer: Comando para saber el estado del servicio

Set-DhcpServerv4OptionValue -ScopeId 192.168.100.0 -OptionId 3 -Value 192.168.100.1 : Para agregar la puerta de enlace manualmente