

Infraestructura Gaming con IPv6 + NAT64/DNS64

Proyecto completo de Ansible para crear una infraestructura gaming automatizada con Ubuntu Desktop, usando IPv6 puro con traducción NAT64/DNS64 a internet IPv4.

Arquitectura

```

internet (ipv4/Vm network)
↓
router (fisico)
↓
switch (fisico)
↓
switch_virtual (----no hecho----)
↓
Servidor Ubuntu (UBPC)
├─ ens33: Internet IPv4 (DHCP/VM network)
└─ ens34: Red interna IPv6 (2025:db8:10::/64)
    ├─ DHCP IPv6 (asigna IPs automáticamente)
    ├─ DNS + DNS64 (traduce nombres a IPs IPv6)
    ├─ Tayga NAT64 (traduce paquetes IPv6→IPv4)
    ├─ Squid Proxy (HTTP/HTTPS sobre IPv6)
    └─ radvd (Router Advertisements)
        ↓
        switch virtual (M_vm's)
        ↓
        VMs Ubuntu Desktop (solo IPv6)
        ├─ IP automática por DHCP: 2025:db8:10::100-200
        ├─ DNS64 traduce google.com → 64:ff9b::xxx
        ├─ NAT64 traduce paquetes a IPv4
        └─ Acceso completo a internet
  
```

Red: 2025:db8:10::/64

Dominio: gamecenter.local

Servidor: 2025:db8:10::2

VMs (DHCP): 2025:db8:10::10-200 (debe ser literalmente la ip 2025:db8:10::10 y así, sin cosas en medio)

Estructura del Proyecto

```

ansible/
├─ playbooks/           # Playbooks principales
│   ├─ setup-complete-infrastructure.yml # Configura TODO el servidor
│   ├─ create-vm-ubuntu-desktop.yml     # Crea VMs en ESXi
│   ├─ configure-ubuntu-desktop.yml     # Configura usuarios en VMs
│   └─ setup-gaming-desktop.yml         # Instala software gaming
  
```

```
├── roles/                                # Roles de Ansible
│   ├── network/                         # Configuración de red, NAT64, Squid
│   ├── dhcpv6/                          # Servidor DHCP IPv6
│   ├── dns_bind/                        # DNS con BIND9 + DNS64
│   ├── firewall/                       # Firewall con UFW
│   ├── ubuntu_gaming/                  # Software y optimizaciones gaming
│   └── storage/                         # NFS y almacenamiento
├── scripts/                             # Scripts auxiliares
│   ├── install-nat64-tayga.sh           # Instala Tayga manualmente
│   ├── install-squid-proxy.sh          # Instala Squid manualmente
│   ├── fix-nat64-routes.sh             # Corrige rutas NAT64
│   └── check-nat64-status.sh           # Verifica estado NAT64
├── inventory/                           # Inventarios de hosts
│   └── hosts.ini                       # Definición de servidores y VMs
└── group_vars/                          # Variables globales
    ├── all.yml                        # Variables comunes
    └── all.vault.yml                  # Contraseñas encriptadas
```

Guía de Uso Rápida

1 Configurar Servidor Completo

```
# Activa el entorno virtual
source .ansible-venv/bin/activate

# Configura TODO: Red, DHCP, DNS, NAT64, Squid, Firewall
ansible-playbook -i inventory/hosts.ini setup-complete-infrastructure.yml -K
```

Esto configura:

- ☒ Red IPv6 en ens34
- ☒ DHCP IPv6 (rango 2025:db8:10::10d-200)
- ☒ DNS con BIND9 + DNS64
- ☒ Tayga NAT64 (traduce IPv6→IPv4)
- ☒ Squid Proxy (HTTP/HTTPS)
- ☒ radvd (Router Advertisements)
- ☒ Firewall configurado

2 Crear VM Ubuntu Desktop

```
# Crea una VM en ESXi con Ubuntu Desktop
ansible-playbook -i inventory/hosts.ini create-vm-ubuntu-desktop.yml
```

Especificaciones de la VM:

- 8GB RAM
- 4 CPUs
- 40GB disco
- Conectada a red M_vm's (ens34 del servidor)

Después:

1. Instala Ubuntu Desktop manualmente
2. Crea usuario inicial: **administrador** / **123456**
3. Configura red IPv6 (ver sección "Configurar Red en VM")

3 Configurar Usuarios en la VM

```
# Crea 3 usuarios: admin, auditor, gamer01
ansible-playbook -i inventory/hosts.ini configure-ubuntu-desktop.yml
```

Usuarios creados:

- **admin**: Administrador con sudo (contraseña: 123456)
- **auditor**: Solo lectura (contraseña: 123456)
- **gamer01**: Usuario gaming (contraseña: 123456)

4 Instalar Software Gaming

```
# Instala y optimiza para gaming
ansible-playbook -i inventory/hosts.ini setup-gaming-desktop.yml
```

Software instalado:

- Steam, Lutris, Heroic Games Launcher
- Discord, OBS Studio
- GameMode, MangoHud, ProtonUp-Qt
- Bottles, emuladores (RetroArch, PCSX2, Dolphin)

Optimizaciones:

- Kernel XanMod gaming
- CPU governor en performance
- Swap optimizado (swappiness=10)
- Audio de baja latencia

Personalización:

- Tema Sweet Dark
- Iconos Papirus

- Wallpapers gaming
- Conky para monitoreo

Playbooks Disponibles

Playbooks Principales

Playbook	Descripción	Uso
<code>setup-complete-infrastructure.yml</code>	Configura TODO el servidor desde cero	<code>ansible-playbook -i inventory/hosts.ini setup-complete-infrastructure.yml -K</code>
<code>create-vm-ubuntu-desktop.yml</code>	Crea VM en ESXi	<code>ansible-playbook -i inventory/hosts.ini create-vm-ubuntu-desktop.yml</code>
<code>configure-ubuntu-desktop.yml</code>	Configura usuarios en VM	<code>ansible-playbook -i inventory/hosts.ini configure-ubuntu-desktop.yml</code>
<code>setup-gaming-desktop.yml</code>	Instala software gaming	<code>ansible-playbook -i inventory/hosts.ini setup-gaming-desktop.yml</code>

Playbooks por Componente

Playbook	Descripción
<code>playbook-network.yml</code>	Solo configuración de red
<code>playbook-dhcp.yml</code>	Solo DHCP IPv6
<code>playbook-dns.yml</code>	Solo DNS + DNS64
<code>playbook-firewall.yml</code>	Solo firewall

Scripts de Ejecución Rápida

Script	Descripción
<code>run-network.sh</code>	Ejecuta playbook de red
<code>run-dhcp.sh</code>	Ejecuta playbook de DHCP
<code>run-dns.sh</code>	Ejecuta playbook de DNS
<code>run-firewall.sh</code>	Ejecuta playbook de firewall

Scripts Auxiliares

Scripts de NAT64

Script	Descripción	Uso
--------	-------------	-----

Script	Descripción	Uso
<code>install-nat64-tayga.sh</code>	Instala Tayga NAT64 manualmente	<code>sudo bash install-nat64-tayga.sh</code>
<code>install-squid-proxy.sh</code>	Instala Squid Proxy manualmente	<code>sudo bash install-squid-proxy.sh</code>
<code>fix-nat64-routes.sh</code>	Corrige rutas y reglas de NAT64	<code>sudo bash fix-nat64-routes.sh</code>
<code>check-nat64-status.sh</code>	Verifica estado completo de NAT64	<code>sudo bash check-nat64-status.sh</code>
<code>fix-dhcp-quick.sh</code>	Corrige servicio DHCP rápidamente	<code>sudo bash fix-dhcp-quick.sh</code>

Configurar Red en VM

Después de instalar Ubuntu Desktop en la VM, configura la red IPv6:

```
# Editar netplan
sudo nano /etc/netplan/01-netcfg.yaml
```

Contenido:

```
network:
  version: 2
  ethernets:
    ens34:
      dhcp4: no
      dhcp6: yes
      accept-ra: yes
      nameservers:
        addresses:
          - 2025:db8:10::2
      search:
        - gamecenter.local
```

Aplicar:

```
sudo netplan apply

# Verificar IP obtenida
ip -6 addr show ens34

# Probar internet
ping6 google.com
```

Verificación y Diagnóstico

En el Servidor

```
# Ver servicios activos
sudo systemctl status isc-dhcp-server6
sudo systemctl status bind9
sudo systemctl status radvd
sudo systemctl status squid

# Ver NAT64
ps aux | grep tayga
ip addr show nat64
ip -6 route | grep 64:ff9b

# Ver reglas de firewall
sudo iptables -L -v -n
sudo ip6tables -L -v -n

# Verificar estado completo
sudo bash check-nat64-status.sh
```

En la VM

```
# Ver IP obtenida
ip -6 addr show ens34

# Ver rutas
ip -6 route show

# Probar DNS64
dig @2025:db8:10::2 google.com AAAA

# Probar internet
ping6 google.com
curl http://google.com
```

Roles Detallados

network

Configura red IPv6, NAT64, Squid Proxy y radvd.

Tareas:

- Configura interfaces ens33 (WAN) y ens34 (LAN)

- Habilita IP forwarding
- Instala y configura radvd
- Instala y configura Tayga NAT64
- Instala y configura Squid Proxy
- Configura iptables para NAT

dhcpv6

Configura servidor DHCP IPv6.

Tareas:

- Instala isc-dhcp-server
- Configura rango 2025:db8:10::100-200
- Configura permisos y AppArmor
- Crea directorio PID correcto

dns_bind

Configura DNS con BIND9 + DNS64.

Tareas:

- Instala BIND9
- Configura zona gamecenter.local
- Configura DNS64 (prefijo 64:ff9b::/96)
- Configura forwarders a 8.8.8.8

firewall

Configura firewall con UFW.

Tareas:

- Instala UFW
- Abre puertos: SSH (22), DNS (53), DHCP (546/547)
- Configura rate limiting para SSH

ubuntu_gaming

Instala software gaming y optimizaciones.

Tareas:

- Instala Steam, Lutris, Heroic, Discord, OBS
- Instala kernel XanMod gaming
- Optimiza CPU, swap, audio
- Instala tema Sweet Dark
- Configura Conky para monitoreo

Contraseñas

Las contraseñas están en `group_vars/all.vault.yml` (encriptado con Ansible Vault).

Contraseñas por defecto:

- Usuarios VM: `123456`
- Usuario servidor: (tu contraseña actual)

Encriptar/Desencriptar

```
# Encriptar archivo
ansible-vault encrypt group_vars/all.vault.yml

# Desencriptar
ansible-vault decrypt group_vars/all.vault.yml

# Editar
ansible-vault edit group_vars/all.vault.yml
```

Solución de Problemas

DHCP no asigna IPs

```
# Verificar servicio
sudo systemctl status isc-dhcp-server6

# Ver logs
sudo journalctl -u isc-dhcp-server6 -n 50

# Corregir permisos
sudo bash fix-dhcp-quick.sh
```

NAT64 no funciona

```
# Verificar Tayga
ps aux | grep tayga
ip addr show nat64

# Corregir rutas
sudo bash fix-nat64-routes.sh

# Verificar estado completo
sudo bash check-nat64-status.sh
```


HTTP/HTTPS no funciona

```
# Usar Squid Proxy en la VM
echo 'Acquire::http::Proxy "http://[2025:db8:10::2]:3128";' | sudo tee
/etc/apt/apt.conf.d/proxy.conf

# Verificar Squid en servidor
sudo systemctl status squid
```

DNS no resuelve

```
# Verificar BIND
sudo systemctl status bind9

# Probar DNS64
dig @2025:db8:10::2 google.com AAAA

# Ver logs
sudo journalctl -u bind9 -n 50
```

Documentación Adicional

- [GUIA-RAPIDA.md](#): Guía rápida de uso
- [TOPOLOGIA-RED.md](#): Diagrama de red detallado
- [USUARIOS-Y-CONTRASEÑAS.md](#): Lista de usuarios y contraseñas
- [DONDE-EJECUTAR-PLAYBOOKS.md](#): Dónde ejecutar cada playbook
- [SCRIPTS-Y-PLAYBOOKS.md](#): Descripción de scripts

Comandos Útiles Gaming

En la VM

```
# Optimizar para jugar
sudo gaming-mode.sh

# Restaurar configuración normal
sudo normal-mode.sh

# Ver FPS y stats
mangohud <juego>

# Monitoreo del sistema
btop
```

Contribuir

Este proyecto es para uso educativo y gaming. Siéntete libre de adaptarlo a tus necesidades.

Licencia

MIT License - Ver [LICENSE.txt](#)

✦ Características Principales

- ☒ IPv6 puro en VMs (sin IPv4)
 - ☒ NAT64/DNS64 funcional
 - ☒ DHCP IPv6 automático
 - ☒ Software gaming completo
 - ☒ Optimizaciones de rendimiento
 - ☒ Personalización visual gaming
 - ☒ Todo automatizado con Ansible
 - ☒ Fácil de replicar y mantener
-

¡Disfruta tu infraestructura gaming!  