

Despliegue del servidor DNS (VM controller)

```
# En el controller (192.168.200.200)
sudo apt update
sudo apt install dnsmasq -y

sudo systemctl stop dnsmasq

# Hacer backup de la configuración original
sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.backup

# Limpiar configuraciones existentes
sudo rm -f /etc/dnsmasq.d/*

sudo bash -c 'cat > /etc/dnsmasq.d/dhcp-sdn.conf << EOF
# =====
# Configuración DHCP para red SDN
# Fecha: 2025-11-01 13:34:37 UTC
# Usuario: Cjfs2005
# Lease Time: 4 horas
# =====

# Interface conectada a los switches OVS
interface=ens4
bind-interfaces

# No leer archivos del sistema
no-resolv
no-hosts

# Rango DHCP con lease de 4 horas
dhcp-range=192.168.200.10,192.168.200.254,255.255.255.0,4h

# ===== Opciones DHCP (RFC 2132) =====

# Option 3: Router/Gateway
dhcp-option=3,192.168.200.200

# Option 6: DNS Servers
dhcp-option=6,8.8.8.8,8.8.4.4

# Option 15: Domain Name
dhcp-option=15,sdn.local

# Option 28: Broadcast Address
dhcp-option=28,192.168.200.255

# Option 51: IP Address Lease Time (4 horas = 14400 segundos)
```

```
dhcp-option=51,14400
```

```
# Option 58: Renewal Time (50% del lease = 2 horas = 7200 segundos)
```

```
dhcp-option=58,7200
```

```
# Option 59: Rebinding Time (87.5% del lease = 3.5 horas = 12600 segundos)
```

```
dhcp-option=59,12600
```

```
# ===== Logging =====
```

```
# Habilitar logs detallados
```

```
log-dhcp
```

```
log-queries
```

```
# ===== Configuración de leases =====
```

```
# Archivo de leases
```

```
dhcp-leasefile=/var/lib/misc/dnsmasq.leases
```

```
# Modo autoritativo (responde NACK a solicitudes incorrectas)
```

```
dhcp-authoritative
```

```
# =====  
EOF'
```

```
# Ver contenido del archivo
```

```
cat /etc/dnsmasq.d/dhcp-sdn.conf
```

```
# Verificar que no haya errores de sintaxis
```

```
sudo dnsmasq --test
```

```
# Crear directorio
```

```
sudo mkdir -p /var/lib/misc/
```

```
# Crear archivo de leases vacío
```

```
sudo touch /var/lib/misc/dnsmasq.leases
```

```
# Establecer permisos correctos
```

```
sudo chown dnsmasq:nogroup /var/lib/misc/dnsmasq.leases
```

```
sudo chmod 644 /var/lib/misc/dnsmasq.leases
```

```
# systemd-resolved puede interferir con dnsmasq
```

```
sudo systemctl stop systemd-resolved 2>/dev/null || true
```

```
sudo systemctl disable systemd-resolved 2>/dev/null || true
```

```
sudo systemctl start dnsmasq
```

```
sudo systemctl status dnsmasq
```

Reglas para tráfico DHCP, ARP y HTTP hacia el módulo de autenticación (VM swX)

sw=\$(hostname)

```
sudo ovs-ofctl add-flow $sw "priority=300,udp,tp_src=68,tp_dst=67,dl_vlan=0xffff,actions=mod_vlan_vid:100,NORMAL,C  
ONTROLLER:65535"
```

```
sudo ovs-ofctl add-flow $sw "priority=300,udp,tp_src=68,tp_dst=67,dl_vlan=100,actions=NORMAL"  
sudo ovs-ofctl add-flow $sw "priority=300,udp,tp_src=67,tp_dst=68,actions=NORMAL"
```

```
sudo ovs-ofctl add-flow $sw "priority=300,arp,actions=NORMAL"
```

```
sudo ovs-ofctl add-flow $sw "priority=300,tcp,nw_dst=192.168.200.200,tp_dst=5000,actions=NORMAL"
```

```
sudo ovs-ofctl add-flow $sw "priority=300,tcp,nw_src=192.168.200.200,tp_src=5000,actions=NORMAL"
```

```
sudo ovs-ofctl add-flow $sw "priority=0,actions=drop"
```

Únicamente ejecutar en el switch al que está conectado el servidor DHCP (VM controller)

```
sudo ovs-ofctl add-flow $sw "priority=300,udp,tp_src=68,tp_dst=67,dl_vlan=100,actions=strip_vlan, NORMAL"
```

Despliegue del servidor FLASK (VM controller)

En el controller (192.168.200.200)

```
sudo apt update  
sudo apt install -y python3 python3-pip
```

```
# Verificar versión de Python  
python3 --version
```

```
# Instalar Flask (solo si tu proyecto no tiene virtualenv)  
sudo -H pip3 install flask
```

```
# Instalar Gunicorn para Python3 (MUY IMPORTANTE)  
sudo -H pip3 install --upgrade gunicorn
```

```
# Verificar que gunicorn correcto está en /usr/local/bin  
which gunicorn
```

Si la salida es /usr/bin/gunicorn → NO USAR

El correcto debe estar en /usr/local/bin/gunicorn

```
# Ir al directorio del proyecto  
cd /home/ubuntu/Documents/Proyecto
```

```
# Lanzar la aplicación Flask con Gunicorn (modo prueba)
# auth_portal.py debe contener "app = Flask(__name__)"
/usr/local/bin/gunicorn --bind 0.0.0.0:5000 auth_portal:app
```

Opcional: Para ejecutar en segundo plano

```
nohup /usr/local/bin/gunicorn --bind 0.0.0.0:5000 auth_portal:app &
```