

SOFTWARE-DEFINED NETWORK



Proyecto Integrado | IES Gonzalo Nazareno Curso 2024/2025

Miguel Figueroa Escribano

Índice

1.	OBJ	JETIVOS	3
2.	ESC	CENARIO	4
2	.1	TABLAS RESUMEN DEL ESCENARIO	7
2	.2	VPN PARA ACCESO EXTERIOR	8
3.	FUN	NDAMENTOS TEÓRICOS Y CONCEPTOS	11
3	.1	SDN	11
3	.2	SDN EN PROXMOX	12
4.	DES	SCRIPCIÓN	14
5.	CON	NCLUSIONES Y PROPUESTAS	24
6.	DIF.	ICULTADES	25
7.	BIB	LIOGRAFÍA	26



1. OBJETIVOS

El principal objetivo de este proyecto es implementar y desarrollar una solución de redes definidas por software (SDN) en un entorno basado en Proxmox, lo que permitirá dotar a la infraestructura de una serie de funcionalidades avanzadas que van más allá de la simple conectividad entre máquinas virtuales. Al introducir esta característica, se busca conseguir, entre otras cosas, la creación de redes privadas completamente aisladas por cada nodo, la posibilidad de establecer redes superpuestas que abarquen varios clústeres y la habilitación de escenarios multiusuario o multitenant, donde distintos grupos de usuarios puedan trabajar de forma segura y segmentada dentro de la misma plataforma.

Durante el desarrollo del proyecto, se abordarán diferentes casos prácticos que evidenciarán el valor añadido de la solución propuesta. Por ejemplo, se diseñarán y desplegarán redes virtuales aisladas específicamente para distintos grupos de máquinas virtuales, lo que permitirá garantizar la segmentación y la seguridad de los entornos. Además, se realizarán pruebas exhaustivas de conectividad y de aislamiento entre las diferentes redes, asegurando que solo las máquinas virtuales autorizadas puedan comunicarse entre sí y que no existan fugas de información entre redes independientes.

Otro aspecto importante será la integración de un sistema de asignación automática de direcciones IP mediante un servicio de DHCP, lo que facilitará la gestión y el escalado de las redes virtuales sin intervención manual. Finalmente, se evaluará la escalabilidad del entorno SDN, añadiendo nuevas redes o nodos al sistema y verificando que estos elementos se integran correctamente y que el rendimiento y la funcionalidad se mantienen estables a medida que crece la infraestructura. En definitiva, este proyecto pretende no solo demostrar la viabilidad técnica de la solución, sino también resaltar las ventajas prácticas que aporta en términos de flexibilidad, seguridad y gestión avanzada de redes en entornos virtualizados.



2. ESCENARIO

Para la realización del proyecto hemos montado el escenario en un servidor local, concretamente en un mini PC de las siguientes características:

- MinisForum UM773 Lite
- AMD Ryzen 7 7735HS
- 32GB RAM
- 1TB SSD

Este servidor está ubicado en la red de mi vivienda, que tiene configurada una IP fija con la compañía Avatel (la siguiente dirección apunta hacia esta red → jaen.miguelfigueroa.es)

Sobre él hemos instalado un Proxmox VE 8.4, al que le hemos asignado una IP estática dentro de nuestra red local que es la 192.168.1.200/24. Esta máquina será la que nos servirá de base (la llamaremos la máquina base) para montar el resto del escenario.

🛑 😑 📄 📶 Google Calendar - junio de 2000	× A Proyecto Integrad	- FCT - PI × X base - Proxmox Virt	ual Enviro × +		Ŭ
← → C ⊗ No es seguro https://19	2.168.1.200:8006/#v1:0	:=node%2Fbase:4:5:::::			☆ 👂 🗅 🤹 🗄
먦 🛛 🗯 iCloud 🗅 Google 🗅 Finanzas	🔞 Calendar 🛛 🦛 Facebo	ok 💶 Youtube 🚳 NubeBMM 🛃 G	lissandoo 🧕 Amazon 🧕 Amazon Photos	🕌 Netflix 🕟 Prime Video 🗀 Software	
	3.4.1 Search			Documentation	ate VM 🜍 Create CT 💄 root@pam 🗸
Server View 🗸 🔅	Node 'base'			් Reboot 🖒 Shutdown >_ S	hell 🗸 🚦 Bulk Actions 🗸 😧 Help
✓ ■ Datacenter ✓ ■ base	Q Search				Hour (average)
Iocalnetwork (base)	Summary				
€ ☐ local-lvm (base)	D Notes >_ Shell ס? System ▼	() CPU usage ■ Load average	0.33% of 16 CPU(s) 1.70,0.75,0.28	O IO delay	0.00%
	≓ Network	🚥 RAM usage	6.21% (1.81 GiB of 29.12 GiB)	KSM sharing	
	Certificates	A / HD space	4.79% (4.50 GiB of 93.93 GiB)	C SWAP usage	0.00% (0 B of 8.00 GiB)
	 ONS Hosts Options Time ≣ System Log Updates ▼ 	CPU(s) Kernel Version Boot Mode Manager Version Repository Status		16 x AMD Ryzen 7 7738 Linux pv O Proxmox VE updates 🕦 Non pro	S with Radeon Graphics (1 Socket) 6.8.12-11-pve (2025-05-22109-392) EFI e-manager/8.4.1/2a5/fa54a850396d duction-ready repository enabled! >
	C Repositories				😑 CPU usage 🍵 IO delay 😑
	♥ Firewall ▶				

Una vez tenemos ya el equipo base funcionando, crearemos un contenedor que nos permitirá acceder desde el exterior a la interfaz web de Proxmox de la máquina.



Para ello, con la ayuda de este proyecto de la comunidad <u>https://community-scripts.github.io/ProxmoxVE/</u>, desplegaremos un contenedor Caddy, que es un servidor web y que nos servirá de proxy inverso. Este contenedor LXC tiene asignada la IP estática 192.168.1.201/24. En él, configuramos el caddyfile con el siguiente contenido:

```
base.miguelfigueroa.es {
    reverse_proxy <https://192.168.1.200:8006> {
        transport http {
            tls_insecure_skip_verify
        }
    }
}
```

Configuramos el reenvío de los puertos 80 y 443 de nuestro router hacia el contenedor caddy y ya tenemos acceso a nuestra interfaz de Proxmox desde el exterior a través de la URL <u>https://base.miguelfigueroa.es</u>



Ahora, dentro de base, crearemos dos nuevas máquinas virtuales que serán dos nodos de Proxmox. Una vez instalados ambos nodos haremos una copia de seguridad de uno de ellos, por si estropeamos algo del escenario o para futuras pruebas que haremos.





De la misma forma que hicimos antes, añadiremos a nuestro dominio los accesos a la interfaz web de los nodos srv1 y srv2, añadiendo primero los registros CNAME en nuestro dominio y posteriormente añadiendo las direcciones públicas de los nodos al archivo caddyfile de nuestro proxy inverso:

Тіро	CNAME
Nombre de host	srv1
Apunta a	jaen.miguelfigueroa.es
Π	1 hora v
Vista previa	srv1.miguelfigueroa.es 3600 IN CNAME jaen.miguelfigueroa.es
Cancelar	Guardar



2.1 TABLAS RESUMEN DEL ESCENARIO

Tabla resumen de IPs del escenario					
192.168.1.200 Nodo PVE - base					
192.168.1.201	CT – Caddy				
192.168.1.202	Nodo PVE (VM) – srv1				
192.168.1.203	Nodo PVE (VM) – srv2				
192.168.1.204	Nodo PVE (VM) — srv3 No funcionando, preparado para exposición				

Tabla resumen registros DNS del dominio miguelfigueroa.es						
А	jaen	45.8.49.181				
CNAME	base	jaen.miguelfigueroa.es				
CNAME	srv1	jaen.miguelfigueroa.es				
CNAME	srv2	jaen.miguelfigueroa.es				
CNAME	srv3	jaen.miguelfigueroa.es				



2.2 VPN PARA ACCESO EXTERIOR

Para acceder desde fuera de la red local donde está montado el escenario usaremos una VPN, en este caso **Tailscale**.

Tailscale es una solución de VPN de última generación que permite crear redes privadas virtuales de forma sencilla y segura entre dispositivos distribuidos en diferentes ubicaciones y redes. A diferencia de los VPN tradicionales, Tailscale utiliza el protocolo WireGuard para establecer conexiones punto a punto cifradas, formando una red privada (tailnet) donde solo los dispositivos autorizados pueden comunicarse entre sí.

Su principal ventaja es la facilidad de uso: basta con instalar el cliente en cada dispositivo y autenticarse para que automáticamente se configuren las conexiones, sin necesidad de cambios complejos en routers o firewalls. Tailscale gestiona el descubrimiento de dispositivos, el cruce de NAT y la asignación de IPs privadas, permitiendo que los dispositivos se conecten directamente siempre que sea posible, o a través de relay si no lo es.

Además, Tailscale ofrece control de acceso granular basado en identidades de usuario, integración con proveedores de identidad (como Google o Microsoft Entra ID), y políticas de acceso centralizadas, facilitando la adopción de arquitecturas Zero Trust y la segmentación de la red. Su enfoque mesh evita cuellos de botella y puntos únicos de fallo, mejorando el rendimiento y la fiabilidad respecto a VPNs centralizadas.

Para la instalación, deberemos añadir nuestro servidor base a nuestra tailnet, y ya desde él saltaremos a los distintos nodos, máquinas virtuales y contenedores. Para ello accedemos al panel web y añadimos un nuevo dispositivo Linux, donde nos da el comando de instalación para ejecutar directamente en nuestro nodo de Proxmox.



8

💀 miguelfigueroaescribano@g~	acil com				Download	Support Docs 🛐
	Add device			×		
🔚 Machines 👶 Apps 🎅 Ser	To add a new device t	o your tailnet, simply i	nstall Tailscale on that	5		☆ Get started
Machines	device and log in as m	niguelfigueroaescriba	no@gmail.com.	_		
Manage the devices connected to vo	E Linux	Windows	🛂 macOS			Add device 🗸
	🗯 iPhone & iPad	📥 Android	Synology			
Q Search by name, owner, tag, ve		_				Ł
5 machines	Email link 🔗 Co	py command 🔲 Rea	ad guide	- 1		
	<pre>\$ curl -fsSL https</pre>	://tailscale.com/in	stall.sh sh	G		
MACHINE				- 1	LAST SEEN	

Ejecutamos la shell desde la interfaz web de Proxmox y lanzamos el comando obtenido anteriormente. Una vez instalado Tailscale, lanzamos el siguiente comando para levantar el servicio:

tailscale up



Nos proporciona ese link al que accederemos, nos logueamos con nuestra cuenta y confirmamos que queremos agregar este dispositivo a nuestra tailnet:

:: tailscale	miguelfigueroaescribano@gmail.com 🚯				
	Connect device				
You are abo	ut to connect the device base to the				
miguelfigu	eroaescribano@gmail.com tailnet.				
Connect					
	Device details				
Public key	nodekey:f2ad52a098c3b943c290aa85b				
Hostname	base				
Hostname Operating system	base linux (6.8.12-11-pve)				



Una vez agregado ya podemos comprobar desde el panel de control de Tailscale los dispositivos que están unidos a nuestra tailnet, que son lo mismos desde los que podemos acceder a ellos:

Machines				
Manage the devices connected to your tailnet. Le	arn more a		l	Add device 🗸
Q Search by name, owner, tag, version		√ Filters ✓		¥
6 machines				
MACHINE	ADDRESSES ()	VERSION	LAST SEEN	
base miguelfigueroaescribano@gmail.com	100.87.43.57 ∨	1.84.0 Linux 6.8.12-11-pve	Connected	***

Ahora nos conectamos desde otro dispositivo ya fuera de la red local y que tenemos agregado a nuestra tailnet, introducimos la IP proporcionada por la VPN y comprobamos que podemos conectarnos:

Imiguel@Mac-mini-de-Miguel ~ % ssh root@100.87.43.57 The authenticity of host '100.87.43.57 (100.87.43.57)' can't be established. ED25519 key fingerprint is SHA256:EpTt7UZTzfA5a6fztW5hydy4Wpo3h98+cq9Lj8DkgDo. This key is not known by any other names. Are you sure you want to continue connecting (yes/no/[fingerprint])? yes Warning: Permanently added '100.87.43.57' (ED25519) to the list of known hosts. Iroot@100.87.43.57's password: Linux base 6.8.12-11-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-11 (2025-05-22T09:39Z) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sun Jun 15 19:50:58 2025



root@base:~#

3. FUNDAMENTOS TEÓRICOS Y CONCEPTOS

A partir de la versión 8, Proxmox VE incluyó la característica de las SDN (software-defined networks). Esta funcionalidad permite la creación, gestión y control centralizado de redes virtuales de manera dinámica y programable.

Entre sus características principales, permite crear zonas virtuales redes virtuales (VNets) directamente desde la interfaz web, permite la segmentación de red para mejorar la seguridad y simplificar la gestión de topologías complejas, centraliza el control de la red, permite la asignación dinámica de recursos de red ajustando el ancho de banda y garantizando que las aplicaciones reciban los recursos necesarios según la demanda, y permite también la integración de servicios como DHCP e IPAM para la gestión automática de direcciones IP.

3.1 SDN

Las Software-Defined Networking (SDN) es un concepto arquitectónico que separa el plano de control (gestión de políticas y rutas) del plano de datos (reenvío de tráfico), permitiendo redes programables y centralizadas. En Proxmox, SDN se integra directamente con el hipervisor para gestionar redes virtualizadas de forma ágil.

Sus características claves son:

- Abstracción de la red física: Las redes se definen mediante software, independientemente del hardware subyacente.
- Automatización: Configuración dinámica de VLANs, subredes y políticas de seguridad.
- Multitenant: Aislamiento lógico de redes para distintos usuarios o cargas de trabajo.



3.2 SDN EN PROXMOX

Zonas

- Una zona es un área virtual de red separada dentro de Proxmox SDN.
 Define el tipo de red (por ejemplo, simple, VLAN, VXLAN, EVPN, etc.) y determina cómo se comportan las redes virtuales dentro de ella.
- Las zonas permiten aislar redes y controlar cómo se conectan y enrutan las máquinas virtuales (MV) y contenedores dentro del clúster.
- Es como una especie de "LAN virtual" dentro de la infraestructura.

VNets

- Una VNet (Virtual Network) es una red virtual que se crea dentro de una zona. Funciona como un switch virtual (puente) al que se conectan las MV y contenedores.
- Cada VNet puede tener uno o varios subredes asociadas, y se despliega localmente en cada nodo del clúster.
- Las VNets permiten segmentar el tráfico y definir dominios de broadcast independientes dentro de una zona.

Options

- Se refiere a las opciones de configuración disponibles tanto para zonas como para VNets. Estas opciones determinan el comportamiento de la red virtual, como la activación de DHCP, la configuración de gateways, la selección del backend de IPAM, el tipo de aislamiento, etc.
- Por ejemplo, puedes habilitar DHCP automático para una zona simple o definir rangos de direcciones IP para una subred dentro de un VNet.

IPAM (IP Address Management)

 IPAM es la gestión de direcciones IP dentro del entorno SDN. Proxmox utiliza IPAM para asignar, liberar y rastrear direcciones IP de las máquinas virtuales y contenedores de forma automática. El plugin IPAM integrado permite ver y administrar las asignaciones de IP desde la interfaz, y existen integraciones con soluciones externas como NetBox o phpIPAM.

 Por ejemplo, cuando una VM se conecta a un VNet con DHCP habilitado, IPAM asigna automáticamente una IP libre del rango configurado y mantiene el registro de esa asignación.

VNet Firewall

- El VNet Firewall es la integración del firewall de Proxmox con las redes definidas por SDN. El sistema genera automáticamente IPSets (listas de IPs o rangos) para cada VNet y sus subredes, que pueden ser usados en las reglas del firewall para controlar el tráfico de red. Por ejemplo, se crean conjuntos como vnet-all (todas las subredes del VNet), vnetgateway (IP de gateways), vnet-no-gateway (subredes excluyendo gateways) y vnet-dhcp (rango DHCP).
- Esto permite crear reglas de firewall específicas para cada VNet o subconjunto de direcciones, facilitando la administración segura y dinámica del tráfico entre redes virtuales.



4. DESCRIPCIÓN

Una vez montado el escenario, accedemos a nuestro nodo srv1. En él deberemos realizar algunas configuraciones para que las SDN funcionen sin problema. Para ello, en primer lugar, deberemos instalar el paquete dnsmasq.

```
apt install dnsmasq
```

Tras esto, deberemos editar el archivo /etc/network/interfaces y añadir la siguiente línea al final del mismo si no está:

```
source /etc/network/interfaces.d/*
```

Con esto verificado ya podemos acceder al apartado SDN de la interfaz web de Proxmox:



Una vez aquí crearemos una nueva zona de prueba. Será de tipo "Simple" y la llamaremos zona2. La configuraremos con el MTU automático y disponible para todos los nodos del cluster, aunque ahora mismo sólo tengamos 1.



También deberemos activar la opción "automatic DHCP", dentro del menú "Advanced" y que configuraremos más adelante en la subnet:

Edit: Simple		5 S
ID:	zona2	
MTU:	auto	$\hat{}$
Nodes:	All (No restrictions)	\sim
IPAM:	pve	~
DNS Server:		\sim
Reverse DNS Server:		\sim
DNS Zone:		
automatic DHCP:		
Help	Advanced	ОК

Con la zona ya creada, pasaremos a crear una red y posteriormente una subred. Para ello, nos dirgimos al apartado "VNets" y crearemos una nueva "VNet":

Edit: VNet		S C
Name: Alias: Zone:	vnet2 zona2	~
Isolate Ports: VLAN Aware:		
P Help		Advanced 🗹 OK

Con esta red creada la seleccionamos y justo en la ventana lateral de "Subnets" crearemos una nueva, con el direccionamiento 192.168.100.0/24:



Edit: Subnet				
General DHC				
Subnet:	192.168.200.0/24			
Gateway:	192.168.200.1			
SNAT:				
DNS Zone Prefix:				
		ОК		

Habilitaremos la opción de SNAT para que todas las máquinas que coloquemos dentro de esta subred puedan salir al exterior y además en la pestaña de "DHCP Ranges" le asignaremos un rango de direcciones para que el servidor DHCP de esta subred pueda asignar IPs a las máquinas que coloquemos en esta subred:



Creamos la subred y ahora comprobamos que tanto la VNet como la subnet están pendiente de cambios. Para aplicar estos cambios debemos dirigirnos al apartado de "SDN" y clickar en el botón de aplicar. Confirmamos:

Confirm	\otimes
Applying pending SDN changes will also apply any pending local node network changes. Proceed?	
Yes No	



Una vez confirmamos se recargan todas las configuraciones de red de los nodos y ya observamos como aparece la zona2 como disponible:

Status		
Apply		
SDN	Node	Status
localnet	srv1	ok
zona1	srv1	available
zona2	srv1	available

Para hacer una primera comprobación de esta zona, su red y su subred está funcionando correctamente lanzaremos la ejecucción de un contenedor de debian12. Avanzamos en la creación del contenedor en Proxmox como solemos hacer y al llegar a la pestaña "Network" observamos que podemos seleccionar como bridge la vnet2 que creamos anteriormente. La seleccionamos y además activaremos que obtenga una dirección IPv4 por DHCP.

Name:	eth0			IPv4: 🔵 Static (DHCP		
MAC address:	BC:24:11:6D:7A:	64		IPv4/CIDR:			
Bridge:	vnet2	`	~	Gateway (IPv4):			
VLAN Tag:	Bridge 个	Active	С	omment		AC	
Firewall:	vmbr0	Yes					
	vnet1	Yes					
	vnet2	Yes					
Disconnect:				Rate limit (MB/s):	unlimited		\bigcirc
MTU:	Same as bridge		0				



Terminamos de configurar el contenedor, lanzamos su creación y accedemos a él:

	1.4.0 Search
Server View 🗸 🗘	Container 200 (debian 12-test1) on node 'srv1' No Tags
✓ ■ Datacenter ✓ ₩ srv1	ℬ Summary root@debian12-test1:-# ip a 1: lo: <loopback,up,lower_up> mtu 65536 qdisc noqueue state UNKNOWN group default glen 1000</loopback,up,lower_up>
200 (debian12-test1) Eccalestwork (srv1) zona1 (srv1) zona2 (srv1) Ccal (srv1) Ccal (srv1) Ccal-ivm (srv1)	 Console Link/Loopback 00:00:00:00:00:00:00:00:00:00:00:00:00:
	O Snapshot 64 bytes from 8.8.8.8 iomp_seq=1 t12.48 time=10.2 ms O Forewall °C - 8.8.8.8 iomp_seq=2 t12.48 time=10.8 ms °C °C - 8.8.8.8 iomp_seq=2 t12.48 time=10.8 ms °C °C • Permissions °C • Permissions °C • O To the state transmitted, 2 received, 0% packet loss, time 1002ms • rott@eblanl2-test1:-#

Comprobamos que ha obtenido la IP 192.168.200.100/24, que está dentro del rango DHCP que le hemos asignado a la subnet y además tiene conectividad con el exterior. Si nos dirigimos a la pestaña "IPAM" observamos de forma gráfica todo los componentes de las SDN, así como sus clientes e IPs asignadas. También podemos hacer modificaciones o mapeos de MACs con IPs de forma manual:

Reload				
Name / VMID 个	IP Address 个	MAC	Gateway	Actions
zona1				
금 Kan vnet1				0
192.168.100.0/24				
Gateway	192.168.100.1		1	
100	192.168.100.100	BC:24:11:C7:15:4F		ar 🕮
zona2				
- 器 vnet2				8
- 192.168.200.0/24				
Gateway	192.168.200.1		1	
200	192.168.200.100	BC:24:11:6D:7A:64		ar 🖻

Además, ahora mismo el CT que hemos creado tiene alcance al resto de dispositivos de la red local, por ejemplo a nuestro equipo que tiene la IP 192.168.1.110:



root@debian12-test1:~# ping 192.168.1.110					
PING 192.168.1.110 (192.168.1.110) 56(84) bytes of data.					
64 bytes from 192.168.1.110: icmp_seq=1 ttl=63 time=0.612 ms					
64 bytes from 192.168.1.110: icmp_seq=2 ttl=63 time=0.613 ms					
64 bytes from 192.168.1.110: icmp_seq=3 ttl=63 time=0.577 ms					
64 bytes from 192.168.1.110: icmp_seq=4 ttl=63 time=0.568 ms					
^C					
192.168.1.110 ping statistics					
4 packets transmitted, 4 received, 0% packet loss, time 3072ms					
rtt $min/avg/max/mdev = 0.568/0.592/0.613/0.020$ ms					
root@debian12-test1:~# traceroute 192.168.1.110					
traceroute to 192.168.1.110 (192.168.1.110), 30 hops max, 60 byte packets					
1 192.168.200.1 (192.168.200.1) 0.037 ms 0.009 ms 0.007 ms					
2 192.168.1.110 (192.168.1.110) 0.361 ms 0.326 ms *					
r = r + 0 $d = h + r = 10$ $d = r = 10$ $d = 10$					

Para concluir con esta primera prueba lanzaremos un nuevo CT que estará en la zona3, y que tendrá direccionamiento 10.0.0/24 y comprobamos que tiene conectividad con el CT que está en la vnet2:



Ahora uniremos el nodo srv2 al clúster para comprobar cómo se expande las zonas definidas anteriormente al resto de nodos que vayamos agregando. Para ello en primer lugar debemos crear un clúster desde srv1. Lo hacemos desde el menú Datacenter > Cluster > Create cluster:

srv1 - Proxmox Virtual Enviror	× +				•
\leftrightarrow \rightarrow C \sim srv1.miguelfigueroa.es	/#v1:0:18:4:::::=cluster		@ \$	🔊 🖸 🕹 🔕) :
🔡 🧯 iCloud 🗅 Google 🗅 Finanzas	🔢 Calendar 🛛 (7) Facebook	🖸 Youtube 🛞 NubeBMM 🛃 Glissandoo 🧕 Amazon	a. Amazon Photos 🔋 Netflix	>> 🗀 Todos los marca	adores
	3.4.0 Search		Documentation Create VM	🕞 Create CT 💄 root@pi	iam v
Server View 🗸 🌣	Datacenter			0	Help
Datacenter Srv1 100 (debian12-vnet3)	Q Search	Cluster Information			
 200 (debian12-test1) 201 (debian12-local) 	 Summary Notes 	Standalone node - no cluster defined			
<pre>iocalnetwork (srv1) zona1 (srv1)</pre>	E Cluster	Cluster Nodes			
zona2 (srv1)	Ceph Continue	Nodename		ID↑ Votes	
E conas (srv1) E cal (srv1) E cal-lvm (srv1)	Storage				



	Create Cluster	\otimes
	Cluster Name:	escenarioASIR
÷	Cluster Network:	Link: 0 🗘 192.168.1.202 🗸 💼
DI 888		Add Multiple links are used as failover, lower numbers have higher priority.
4	Help	Create

Y ya tenemos el clúster creado:

Cluster Information					
Create Cluster Join Information Join Cluster					
Cluster Name: escenarioASIR Config Version: 1 Number of Nodes: 1				1	
Cluster Nodes					
Nodename		ID 个	Votes	Link 0	
srv1		1	1	192.168.1.202	

Ahora nos dirigimos a srv2 para agregarlo al clúster creado. Para ello nos vamos al menú Datacenter > Cluster > Join cluster. Aquí deberemos pegar la información copiada anteriormente desde el botón "Join information" de srv1.

Cluster Join	0008			\otimes		
Assisted join: Paste encoded cluster join information and enter password.						
Information:	RToxMjo3MDoxNSIsInBIZXJMaW5rcyl6eyIwljoi LjIwMiJdLCJ0b3RlbSI6eyJjbHVzdGVyX25hbW a251bWJIcil6IjAifX0sInZlcnNpb24iOilyIiwiaXBf 2ZXJzaW9uIjoiMSIsImxpbmtfbW9kZSI6InBhc3	MTkyLjE2OC4xLjIw ′UiOiJlc2NIbmFyaW dmVyc2lvbil6Imlwdj NpdmUifX0=	MiJ9LCJyaW5nX2FkZHliOlsiMTkyLjE2OC4x 9BU0lSliwiaW50ZXJmYWNIIjp7ljAiOnsibGlu (QtNiIsInNIY2F1dGgiOiJvbiIsImNvbmZpZ19			
Peer Address:	192.168.1.202	Password:	Peer's root password			
Fingerprint:	E5:A3:1A:69:CA:17:25:94:06:0B:3F:73:C4:19:9	1:95:62:77:83:0B:B	8:F4:38:26:56:EC:15:5F:AE:12:70:15			
Cluster Network: Link: 0 IP resolved by node's hostname v peer's link address: 192.168.1.202						
Help			Join 'escenarioA	SIR'		

Deberemos introducir la contraseña del nodo y pulsar finalmente sobre el botón "Join". El proceso se realizará automáticamente y cuando finalice ya observaremos como desde el nodo srv2 ya vemos también el resto de nodos que están dentro del cluster:



\leftrightarrow \rightarrow \mathbb{C} $\stackrel{\bullet\bullet}{\to}$ srv2.miguelfigueroa.es	/#v1:0:18:4:::::=cluster				© ☆ 🛞 🗅	⊻ 🚯 :
🔡 🛛 🗯 iCloud 🗅 Google 🗀 Finanzas	🔢 Calendar 🛛 存 Facebook	🕑 Youtube 🛛 🚯 NubeBMM 🛛 🛃 Glissandoo	a Amazon a	Amazon Photos	Netflix >> 🗀 Tod	os los marcadores
	3.4.0 Search		8	Documentation	Create VM 🝞 Create CT	💄 root@pam 🗸
Server View 🗸 🌣	Datacenter					Ø Help
Datacenter (escenarioASIR) Vertical Structure (escenarioASIR)	Q Search	Cluster Information				
100 (debian12-vnet3)	Summary	Create Cluster Join Information Join Cluster	er			
200 (debian12-test1)	Notes					
201 (debian12-local)	Cluster	Cluster Name: escenarioASIR Con	ig Version: 2		Number of Nodes: 2	
localnetwork (srv1)	Ceph	Cluster Nodes				
zona2 (srv1)	Options	Nodename	ID 个	Votes	Link 0	
zona3 (srv1)	S Storage	srv1	1	1	192 168 1 202	
local (srv1)	e storage	50/2	2	1	102.168.1.202	
Iocal-lvm (srv1)	🖺 Backup	5172	2	1	192.100.1.203	
√ 🛃 srv2	Replication					
localnetwork (srv2)	Permissions					
zona1 (srv2)	•					
zona2 (srv2)	 Users 					
zona3 (srv2)	API Tokens					
local (srv2)	A Two Factor					
Cocal-lvm (srv2)	📽 Groups					

Ya podemos ver la distintas zonas que teníamos creadas previamente, pero aparecen con un símbolo de advertencia y es porque están pendientes de aplicar (cómo ya nos pasaba cuando creábamos una zona y no la aplicábamos):

	.4.0 Search			
Server View 🗸 🌣	Datacenter			
✓ ■ Datacenter (escenarioASIR) ✓ ₽ srv1	Q Search	Status		
 100 (debian12-vnet3) 200 (debian12-test1) 201 (debian12-local) 	SummaryNotes	Apply SDN	Node	Status
 Los (costanz local) Iocalnetwork (srv1) zona1 (srv1) 	Cluster	localnet	srv2 srv1	ok ok
zona2 (srv1) zona3 (srv1) □ local (srv1)	OptionsStorage	zona1 zona2	srv1 srv1	available available
© local (siv1) © local-lvm (srv1) ∨ ௵ srv2	BackupReplication	zona3 zona1	srv1 srv2	available pending
Iocalnetwork (srv2) zona1 (srv2)	Permissions	zona2 zona3	srv2 srv2	pending pending
Zona2 (srv2) Zona3 (srv2) Gocal (srv2) Gocal (srv2)	API Tokens			
U local-ivm (srv2)	GroupsPools			
	🛉 Roles <table-of-contents> Realms</table-of-contents>			
	SDN			

Pulsamos el botón "Apply" y esto hará que recargue toda la configuración de las redes definidas por SDN en todos los nodos. Una vez termina ya comprobamos que están todas disponibles:



	.4.0 Search	
Server View 🗸	Datacenter	
 Datacenter (escenarioASIR) srv1 100 (debian12-vnet3) 200 (debian12-test1) 	Q Search	Status Apply
 201 (debian12-local) localnetwork (srv1) zona1 (crv1) 	Cluster	SDN Node Status localnet srv2 ok
zona2 (srv1) zona3 (srv1)	 Options Storage 	localnetsrv1okzona1srv1availablezona2srv1available
€∐ local (srv1) €∏ local-lvm (srv1) ∨ ∰ srv2	Backup Replication	zona3 srv1 available zona1 srv2 available
 localnetwork (srv2) zona1 (srv2) zona2 (srv2) 	 Permissions Users 	zona2srv2availablezona3srv2available
zona3 (srv2) local (srv2) local-lvm (srv2)	 API Tokens Two Factor Groups 	

La extensión automática de zonas y VNets frente a la configuración manual de subnets responde a diferencias fundamentales en sus funciones dentro de la arquitectura de red:

- Las VNets operan como switches virtuales distribuidos dentro de una zona, proporcionando conectividad de capa 2 entre nodos. Su propagación automática permite migraciones transparentes de VMs y balanceo de carga.
- Las subnets son entidades de capa 3 vinculadas a VNets específicas, encargadas de gestión de direcciones IP (DHCP), configuración de gateways y reglas NAT/SNAT.

Algunos motivos de la no extensión automática de las subnets son:

• Especificidad de configuración: Cada subnet requiere ajustes particulares (rangos IP, reglas de firewall, integración con DNS) que varían según el nodo y su rol en la red.



- Prevención de conflictos: La replicación automática podría generar duplicados de rangos IP en diferentes nodos, causando colisiones.
- Arquitectura descentralizada: Las subnets se gestionan localmente en cada nodo a través de dnsmasq, requiriendo sincronización explícita.



5. CONCLUSIONES Y PROPUESTAS

La implementación de SDN en Proxmox ha supuesto un avance significativo en la gestión y flexibilidad de la infraestructura de red virtualizada. Gracias a la integración de tecnologías como Open vSwitch y la centralización del control de red, el entorno ahora permite crear redes privadas aisladas, redes superpuestas entre múltiples clústeres y escenarios multiusuario, lo que facilita la segmentación lógica y la seguridad de los recursos virtuales. La virtualización de red posibilita una administración dinámica y programable, permitiendo adaptar la topología a las necesidades cambiantes del proyecto y optimizar el uso de recursos de red.

Durante el desarrollo del proyecto se han validado casos prácticos como la creación de redes virtuales aisladas para diferentes grupos de máquinas virtuales, pruebas de conectividad y aislamiento, la asignación automática de IPs mediante DHCP integrado y la escalabilidad al añadir nuevas redes o nodos al entorno SDN. Estas funcionalidades han demostrado ser eficaces tanto en escenarios de pequeña escala como en despliegues más complejos, manteniendo un rendimiento elevado y una baja latencia.

Sin embargo, también se han identificado áreas de mejora. Por ejemplo, la gestión de subnets requiere una configuración manual y cuidadosa para evitar conflictos de direccionamiento y garantizar la correcta integración de nuevos nodos, lo que añade un nivel de complejidad operativa. Además, la extensión de ciertas configuraciones de red, como las subnets, no es automática, lo que obliga a definir procesos claros para su despliegue y sincronización en todos los nodos del clúster.



6. DIFICULTADES

Durante el desarrollo e implementación del proyecto de redes definidas por software (SDN) en Proxmox, se han identificado diversas dificultades que han condicionado tanto el avance como la calidad de los resultados obtenidos.

Una de las principales barreras ha sido la escasez de documentación oficial y de calidad sobre SDN en Proxmox. Aunque existen manuales básicos y algunos apartados en la documentación oficial, la información suele ser limitada, en inglés y, en muchos casos, incompleta o en desarrollo. Esto nos obliga a recurrir a foros, vídeos de Youtube, hilos de discusión y experiencias de la comunidad para resolver dudas o problemas específicos, lo que puede resultar ineficiente y poco fiable. Además, la mayoría de los ejemplos prácticos disponibles se centran en escenarios muy básicos, sin abordar casos más complejos o situaciones reales de despliegue en entornos empresariales o multiusuario.

Otra dificultad recurrente es la falta de casos prácticos detallados. La documentación y los recursos existentes suelen limitarse a la explicación de conceptos teóricos o a la configuración inicial, sin profundizar en ejemplos de integración con servicios avanzados, resolución de incidencias, migraciones entre nodos o automatización de tareas. Esta ausencia de guías paso a paso y de soluciones a problemas habituales dificulta la curva de aprendizaje y aumenta la probabilidad de cometer errores de configuración, especialmente en escenarios donde se requiere alta disponibilidad o integración con otras tecnologías.



7. BIBLIOGRAFÍA

- <u>https://base.miguelfigueroa.es/pve-docs/index.html</u>
- <u>https://doc-proxmox.datosporlasnubes.com/books/redes-en-proxmox-</u> <u>ii/page/41-que-son-las-sdn</u>
- <u>https://tecnocratica.net/wikicratica/books/proxmox-ve</u>
- https://www.youtube.com/watch?v=vRniNRKxhWE
- <u>https://mattglass-it.com/software-defined-network-proxmox/</u>

