IMPLEMENTACIÓN DE PROXMOX EN UN ENTORNO DE LABORATORIO

Miguel E. Pérez F.

1. Objetivo general

Implementar un entorno de pruebas que permita evaluar el rendimiento, la disponibilidad y la resiliencia de los servicios virtualizados, así como el comportamiento de la infraestructura bajo condiciones controladas de prueba.

2. Objetivos específicos del Proyecto

- Instalación de Proxmox en cada uno de los tres servidores, el cual actúa como virtualizador y permite la instalación de máquinas virtuales sin necesidad de software adicional.
- Creación de un clúster que permita la distribución de carga de los distintos nodos.
- Implementación de máquinas virtuales distribuidas entre los nodos que brinden diferentes servicios a los potenciales clientes.
- Configuración de almacenamiento compartido CEPH para lograr una mayor resiliencia y poder brindar mayor disponibilidad de los servidores.
- Documentación de procesos y creación de manuales para permitir el fácil aprendizaje de futuros profesionales en el área o una guía para futuras configuraciones que sean requeridas.

3. Recursos Utilizados

- **Hardware:** 3 servidores HP, 1 switch de red, 6 discos HDD de 146 GB y 3 discos SSD de 1,92 TB.
- **Software:** Rufus, Proxmox VE, máquinas virtuales (Kali Linux, Windows Server 2022, Metasploitable2, VulnHub, Ubuntu Server), CEPH para almacenamiento compartido.

4. Implementación

La implementación se realizó en 5 fases:

1. Instalación de Proxmox en los tres servidores

1.1 Preparación del medio de instalación
La instalación de Proxmox VE se realizó mediante un dispositivo USB booteable, creado con la herramienta Rufus, lo que permitió iniciar el proceso desde el menú de arranque de cada servidor.

1.2 Configuración del controlador RAID
Los servidores utilizados cuentan con un controlador RAID por hardware, el cual no es compatible directamente con Proxmox VE.
Por este motivo, fue necesario acceder a la configuración del controlador y establecer cada disco duro como RAID 0 individual, con el fin de que el instalador de Proxmox reconociera correctamente todos los discos disponibles.

1.3 Configuración del almacenamiento durante la instalación
Durante la instalación del sistema, se configuró un RAID 1 (Mirror) entre los dos primeros discos (146 GB cada uno), destinado al alojamiento del sistema operativo anfitrión (Proxmox VE).
Adicionalmente, se estableció un ashift=13, lo que permite crear bloques de mayor tamaño dentro del sistema de archivos, optimizando las operaciones de lectura y escritura en discos duros mecánicos (HDD).

1.4 Asignación de red y credenciales
Finalizada la configuración de almacenamiento, se procedió a asignar una dirección IP estática a cada nodo del clúster, junto con un usuario y contraseña para su posterior administración remota.

1.5 Primer inicio y actualización de repositorios
Una vez completada la instalación y reiniciado cada nodo, se accedió al sistema con las credenciales definidas.
Posteriormente, se realizaron los siguientes pasos para modernizar los repositorios del sistema:

1.5.1 Eliminar el contenido del directorio **/etc/apt/sources.list.d/.**

1.5.2 Ejecutar el siguiente comando para actualizar la estructura de repositorios:

apt modernize-sources

1.5.3 Crear los archivos **debian.sources** y **proxmox.sources** dentro del directorio **/etc/apt/sources.list.d/,** con el siguiente contenido:

Archivo: **debian.sources**

Types: deb deb-src

URIs: http://deb.debian.org/debian/

Suites: trixie trixie-updates

Components: main non-free-firmware

Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg

Types: deb deb-src

URIs: http://security.debian.org/debian-security/

Suites: trixie-security

Components: main non-free-firmware

Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg

Archivo: **proxmox.sources**

Types: deb

URIs: http://download.proxmox.com/debian/pve

Suites: trixie

Components: pve-no-subscription

Signed-By: /usr/share/keyrings/proxmox-archive-keyring.gpg

1.6 Actualización del sistema

Para finalizar, se ejecutaron los siguientes comandos con el propósito de actualizar completamente el sistema operativo y los paquetes del entorno:

apt update

apt dist-upgrade

1.7 Repetición del procedimiento en los tres nodos
Todo el proceso descrito anteriormente fue repetido en cada uno de los tres nodos, garantizando la homogeneidad de la configuración del clúster.

2. Creación del clúster.

2.1 Acceso a la interfaz de administración
Una vez instalados los distintos nodos y confirmado el acceso a la interfaz web de administración mediante la dirección:

https://<Dirección IP>:8006

2.2 Creación del clúster desde el nodo principal
Desde la interfaz de administración del nodo 1, se ingresó a la siguiente ruta: **Datacenter > Cluster > Create Cluster.**
En esta sección se inició el proceso de creación del clúster, tras lo cual el sistema generó la información de unión necesaria para incorporar los demás nodos. Esta información incluye el Cluster Name, Fingerprint, y el Join Information, que deberán utilizarse posteriormente en los otros nodos.

2.3 Unión del nodo 2 al clúster
Con el clúster ya creado, se accedió al nodo 2 a través de su interfaz web y se siguió la ruta:
Datacenter > Cluster > Join Cluster.
A continuación, se copió la información de unión previamente generada en el nodo 1 y se pegó en el formulario correspondiente.
Al confirmar la operación con OK, el nodo se integró exitosamente al clúster.

2.4 Unión del nodo 3 al clúster
Se repitió el mismo procedimiento con el nodo 3, utilizando nuevamente la información del nodo principal para completar la unión al clúster.

2.5 Verificación de la unión de los nodos
Una vez finalizado el proceso, se verificó que todos los nodos fueran visibles y administrables desde la interfaz del nodo 1, confirmando así la correcta creación y sincronización del clúster Proxmox VE.

3. Configuración de almacenamiento compartido CEPH.

3.1 Instalación del módulo Ceph en los nodos:

Para habilitar el almacenamiento distribuido Ceph, es necesario instalar el módulo correspondiente en cada uno de los nodos del clúster.
Esta instalación puede realizarse de dos maneras:

3.1.1 Instalación mediante terminal

Desde la interfaz web de Proxmox, acceder a cada nodo individualmente, abrir la consola (Shell) y ejecutar el siguiente comando:

apt install ceph -y

3.1.2 Instalación mediante interfaz gráfica (UI)

Alternativamente, se puede realizar la instalación desde el navegador ingresando a: **Datacenter > pve01 > Ceph > Install Ceph**, lo que iniciará el proceso de instalación de manera automática.

3.2 Selección y configuración de discos OSD

Una vez instalado Ceph, se deben seleccionar los discos duros que funcionarán como OSD (Object Storage Daemons), los cuales almacenarán los datos del clúster de manera distribuida.
Para optimizar el rendimiento, se seleccionó un disco SSD en cada nodo, destinado al almacenamiento de las imágenes de disco de las máquinas virtuales (VMs).

3.3 Creación del pool de almacenamiento compartido

Con los discos OSD configurados, se procedió a crear un pool de almacenamiento denominado ssdpool, compuesto por los tres discos SSD de los nodos.
Este pool se utilizó como almacenamiento compartido de alta velocidad para las máquinas virtuales del entorno.

3.4 Verificación del almacenamiento compartido

Tras finalizar la configuración, el pool ssdpool quedó visible y accesible desde cada uno de los nodos del clúster, confirmando su correcta integración como almacenamiento Ceph compartido. (Anexo 1)

4. Instalación de máquinas virtuales:

4.1 Carga de imágenes ISO al servidor

Para realizar la instalación de máquinas virtuales en los distintos nodos, primero es necesario **cargar las imágenes ISO** al almacenamiento del servidor.
Desde un equipo con acceso remoto a los servidores, se debe descargar la imagen ISO requerida, por ejemplo, la ISO de *Kali Linux*.

Una vez descargada, se accede a la interfaz web de Proxmox a través del navegador y se sigue la siguiente ruta:
Datacenter > pve01 > local (pve01) > Imágenes ISO > Cargar. (Anexo 2)

4.2 Subida de la imagen ISO al servidor

En este apartado se selecciona el archivo ISO que se desea subir (por ejemplo, *kali-linux.iso*) y se hace clic en **Cargar**.
Una vez completada la transferencia, la imagen quedará disponible en el almacenamiento local para su uso en la creación de máquinas virtuales. (Anexo 3).

4.3 Creación de la máquina virtual (VM)

Con la imagen ISO disponible, se procede a crear una nueva máquina virtual. Para ello, desde la interfaz web se selecciona la opción **Crear VM**, ubicada en la parte superior derecha de la interfaz de usuario.

Se elige el **nodo** donde se instalará la máquina virtual.

Se asigna un ID de máquina virtual (VM ID) —el cual debe ser mayor a 100— y un nombre descriptivo. (Anexo 4)
En este caso:

- Nodo: pve01
- VM ID: 105
- Nombre: Prueba1

4.4 Selección de la imagen ISO y tipo de sistema operativo

En la siguiente ventana del asistente, se selecciona la imagen ISO previamente cargada y se define el tipo de sistema operativo, por ejemplo, *Linux*. (Anexo 5)

4.5 Configuración del sistema

En la sección Sistema, se recomienda seleccionar el chipset Q35, ya que es un modelo moderno compatible con tecnologías como PCIe y permite migraciones entre nodos.
Además, se debe activar el QEMU Agent, el cual permite al hipervisor comunicarse directamente con el sistema operativo invitado para realizar tareas de administración, control y monitoreo. (Anexo 6)

4.6 Configuración del almacenamiento

En el apartado de Disco, se selecciona como destino de almacenamiento la CEPH pool creada previamente (ssdpool) y se asigna el tamaño del disco virtual, expresado en GB. (Anexo 7).

4.7 Configuración de CPU:

A continuación, se define la cantidad de sockets y núcleos que utilizará la máquina virtual.
Respecto al tipo de procesador, se dispone de las siguientes opciones:

- qemu64: mayor compatibilidad, menor rendimiento.
- kvm64: mejor rendimiento, menor compatibilidad.
- x86-64-v2-AES: soporte de cifrado por hardware.
  En este caso se seleccionó qemu64 por su estabilidad y compatibilidad general. (Anexo 8).

4.8 Configuración de memoria (RAM)

Se asigna la cantidad de memoria RAM que la máquina podrá utilizar.
En este caso, se estableció 4092 MB y se habilitó la función Ballooning, que permite ajustar dinámicamente la memoria asignada según la carga de trabajo de la VM. (Anexo 9).

4.9 Configuración de red

En el apartado de red, se selecciona la interfaz o puente (bridge) que utilizará la máquina virtual para conectarse.
En este entorno se empleó el modelo Realtek RTL8139 y el puente vmbr0, el cual está vinculado a la interfaz física enp3s0f0 debido a que se realiza en un entorno de laboratorio.
En entornos de producción se recomienda el uso del adaptador VirtIO, que ofrece mayor rendimiento y menor latencia aunque en máquinas virtuales Windows necesita la instalación de drivers. (Anexo 10)

4.10 Finalización de la creación de la máquina virtual

Una vez configurados todos los parámetros, se presiona Finalizar.
La máquina virtual aparecerá inmediatamente bajo el nodo correspondiente en la interfaz de Proxmox (Anexo 11).

4.11 Distribución de máquinas virtuales por nodo

El proceso anterior debe repetirse cada vez que se desee crear una nueva máquina virtual.
En este caso, se implementó la siguiente distribución:

- Nodo 1 (10.10.100.93): *Kali Linux*
- Nodo 2 (10.10.100.94): *Windows Server 2022*
- Nodo 3 (10.10.100.95): *Ubuntu Server*

5. Instalación de VMs desde archivos no .iso

5.1 Introducción

Existen dos métodos principales para crear máquinas virtuales en Proxmox a partir de archivos previamente exportados: a partir de un **archivo .ova** o a partir de un **archivo .vmdk**. A continuación se describen ambos procesos paso a paso.

5.2 Creación de VM a partir de un archivo .ova

5.2.1 Copiar el **archivo .ova** al servidor

Método A. Desde un equipo con acceso remoto (por ejemplo PowerShell en Windows), navegar a la carpeta que contiene el **.ova** y ejecutar:

scp .\archivo.ova root@10.10.100.93:/tmp/

Esto copia archivo.ova al directorio **/tmp/** del nodo (en este ejemplo, 10.10.100.93).

Método B. Copiar el **archivo.ova** a un disco externo o USB, conectarlo al servidor, crear un punto de montaje del disco y copiarlo al directorio **/tmp/.** Finalizado esto, puedes eliminar el punto de montaje para evitar corrupción en el disco.

5.2.2 Extraer el contenido del .ova en el nodo

En la consola del nodo (o mediante la interfaz web > Consola), ejecutar:

cd /tmp
tar -xvf archivo.ova

El comando extraerá archivos como .vmdk, .ovf, etc.

5.2.3 **Crear la VM y preparar su configuración**

Crear la VM con parámetros básicos (ID, nombre, memoria, cores, red):

qm create 103 --name vulnhub --memory 2048 --cores 2 --net0 virtio,bridge=vmbr0

5.2.4 Importar el disco VMDK al almacenamiento Ceph (ssdpool)

Importar el **archivo .vmdk** extraído al pool ssdpool:

qm importdisk 103 archivo.vmdk ssdpool

5.2.5 **Asignar el disco importado a la VM y configurar arranque**

Configurar la VM para usar el disco importado vía SCSI y marcarlo como disco de arranque:

qm set 103 --scsihw virtio-scsi-pci --scsi0 ssdpool:vm-103-disk-0

qm set 103 --boot c --bootdisk scsi0

5.2.6 Iniciar la VM

Para arrancar la máquina virtual colocar el siguiente comando:

qm start 103

Al completar estos pasos, la VM habrá sido creada y tendrá asignada la imagen de disco proveniente del .ova.

5.3 Creación de VM a partir de un archivo .vmdk

5.3.1 Subida del archivo .vmdk al servidor

Se realiza el mismo proceso que en el punto 5.2.1 pero con el archivo .vmdk, ambos métodos son válidos.

5.3.2 Crear una VM vacía en la UI

En la interfaz web crear una VM nueva; en el paso de **Disco** seleccionar **No usar ningún medio** (dejando BIOS y SCSI controller por defecto). Asignar sockets, núcleos, memoria y la interfaz de red, y finalizar la creación. Anotar el VM ID asignado (por ejemplo, 104).

5.3.3 Importar el disco VMDK al almacenamiento deseado

Desde la consola del nodo, ejecutar:

qm importdisk 104 /tmp/Metasploitable2-Linux.vmdk ssdpool

Esto transfiere/importa el disco al pool ssdpool.

5.3.4 Adjuntar el disco importado a la VM y configurar arranque

Asignar el disco importado a la VM y configurarlo como disco de arranque:

qm set 104 --scsihw virtio-scsi-pci --scsi0 ssdpool:vm-104-disk-0

qm set 104 --boot c --bootdisk scsi0

Tras estos pasos, la VM 104 dispondrá del contenido del .vmdk y estará lista para iniciar.

5. Resultados Obtenidos

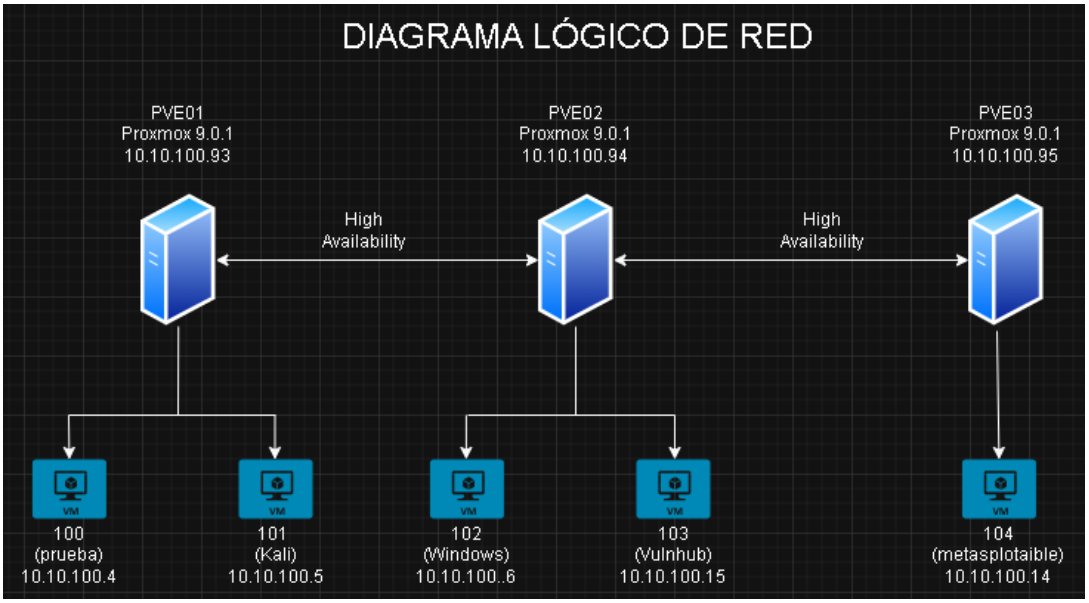
- Configuración de clúster de Proxmox para mayor resiliencia.
- Instalación exitosa de 5 máquinas virtuales distribuidas entre los nodos
- Creación y configuración de almacenamiento compartido CEPH.
- Adaptación de instalación para VMs como VulnHub y Metasploitable2.
- Migración de máquinas virtuales entre nodos, aun estando en funcionamiento.

Identificación de nodos:

Nombre del nodo	Dirección IP
pve01	10.10.100.93
pve02	10.10.100.94
pve03	10.10.100.95

Identificación de máquinas virtuales:

ID	Nodo	Nombre	Dirección IP	Sistema Operativo
100	pve01	prueba	10.10.100.4	Ubuntu Server
101	pve01	Kali	10.10.100.5	Kali Linux
102	pve02	Windows	10.10.100.6	Windows
103	pve02	Vulnhub	10.10.100.15	Windows
104	pve03	metasplotaible	10.10.100.14	Linux



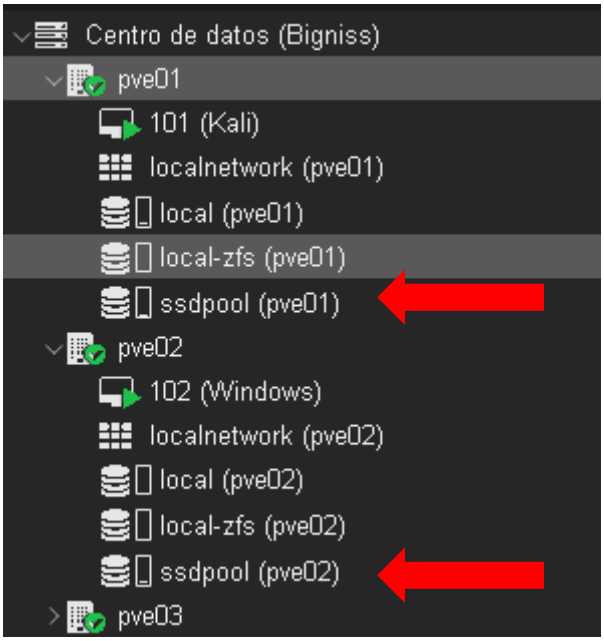
6. Desafíos y Soluciones

- **Desafío:** Proxmox solo permite instalación de VMs desde archivos .iso.
    - **Solución:** Se realizaron configuraciones específicas para permitir la instalación de VulnHub y Metasploitable2.
- **Desafío:** Proxmox es incompatible con controladores de RAID de hardware
    - **Solución:** Se realizó una configuración a través del controlador del RAID que permite a Proxmox usar los discos en la distribución que desee o sea necesitado.
- **Desafío:** Se encontró un servidor incompatible con Proxmox debido a su longevidad
    - **Solución:** Se llevó a cabo un cambio de servidor para poder instalar correctamente el sistema operativo.

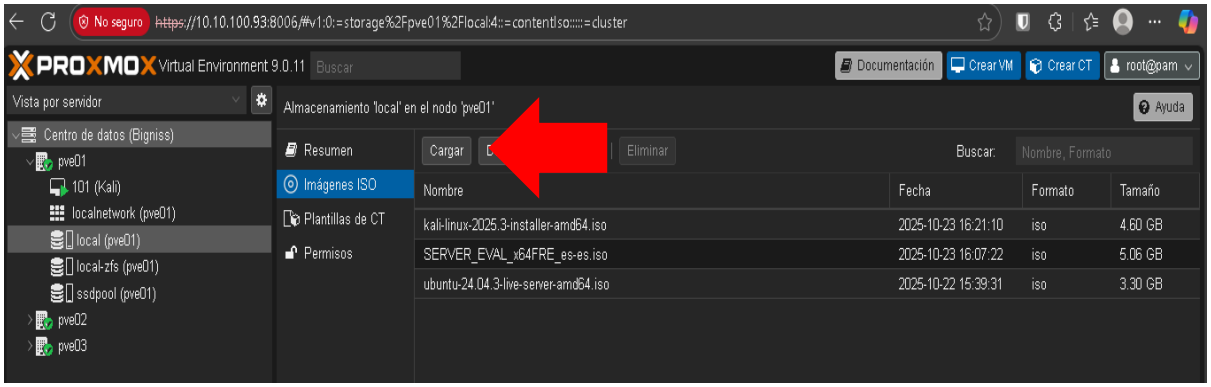
7. Próximos Pasos

- Instalación de servidor de respaldo para VMs, nodos y logs para comprobar la factibilidad de un servidor de respaldo y comprobar el proceso de respaldo que se llevará a cabo.
- Implementación de Wazuh Agent en nodos y VMs a modo de facilitar la detección de intrusiones en los servidores.

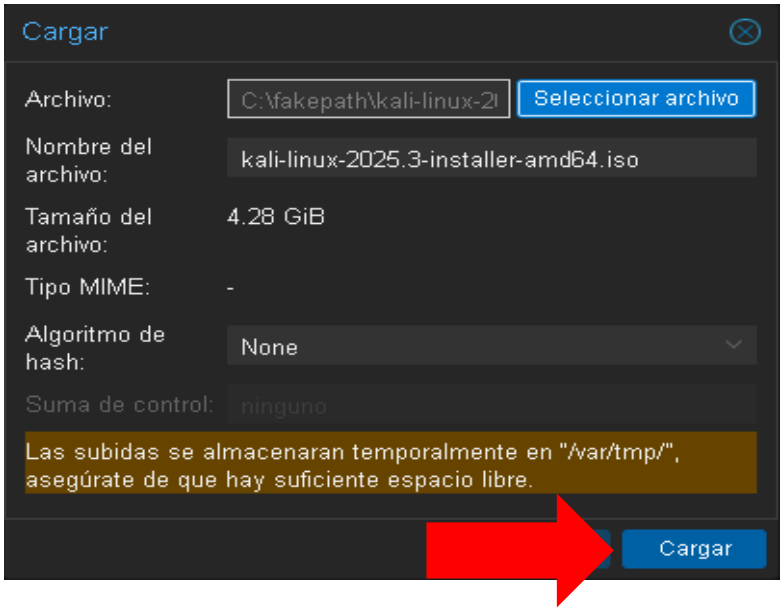
8. Anexos



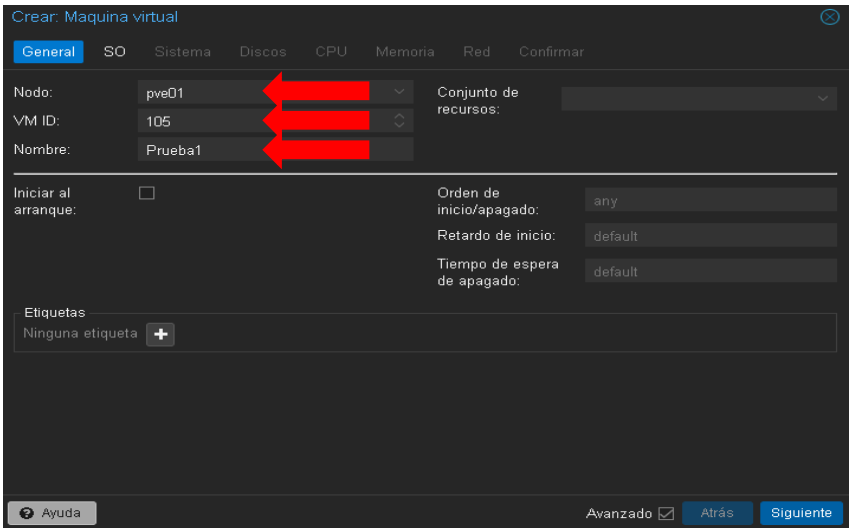
Anexo 1. Muestra la creación del pool “ssdpool” en los distintos nodos



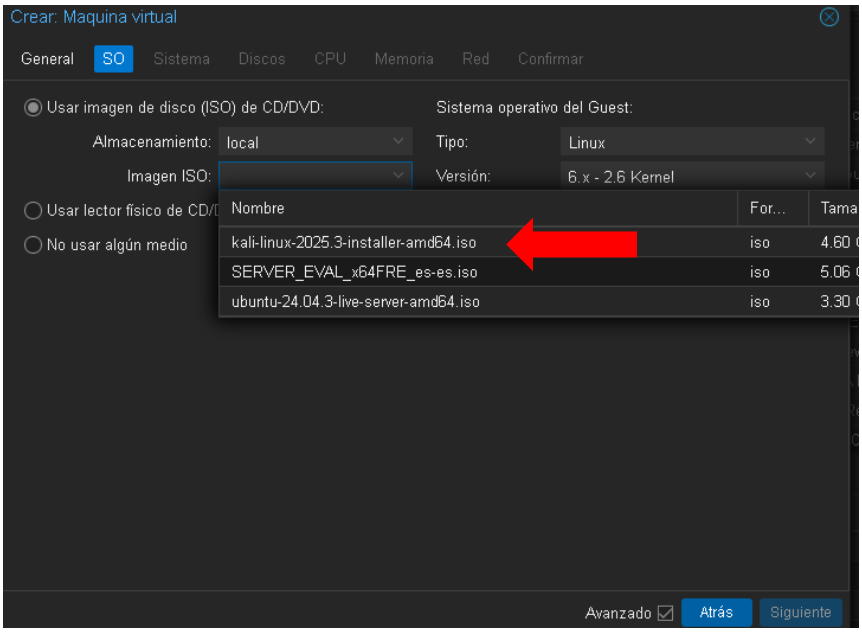
Anexo 2. Muestra donde se deben cargar los archivos .iso



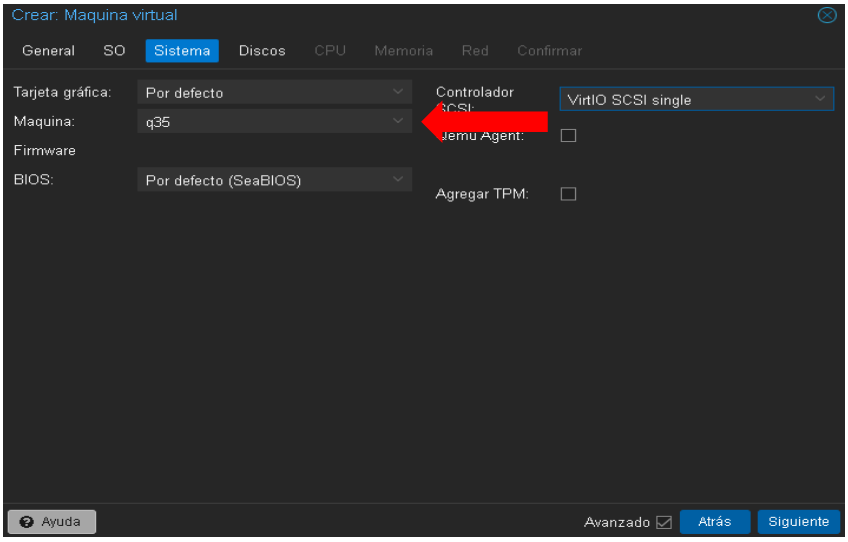
Anexo 3. Ejemplo de transferencia de imagen .iso al servidor



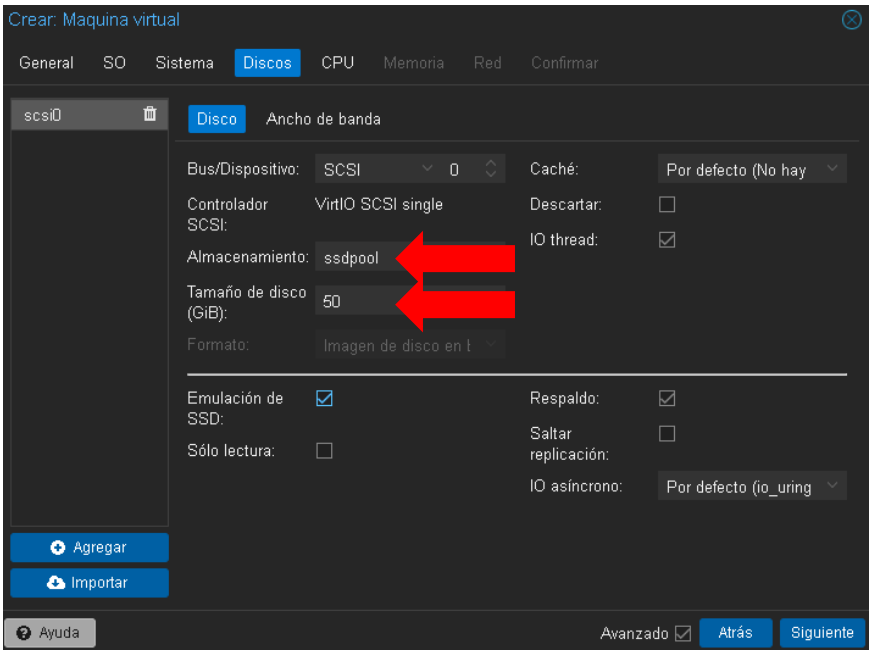
Anexo 4. Muestra la configuración inicial para la identificación de la máquina.



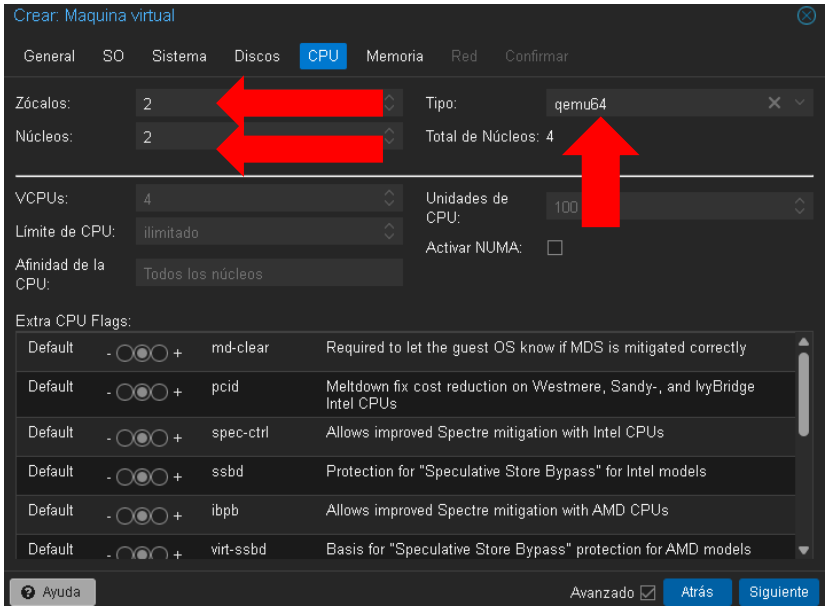
Anexo 5. Muestra la selección del archivo .iso y el tipo de SO que se desea instalar en la VM.



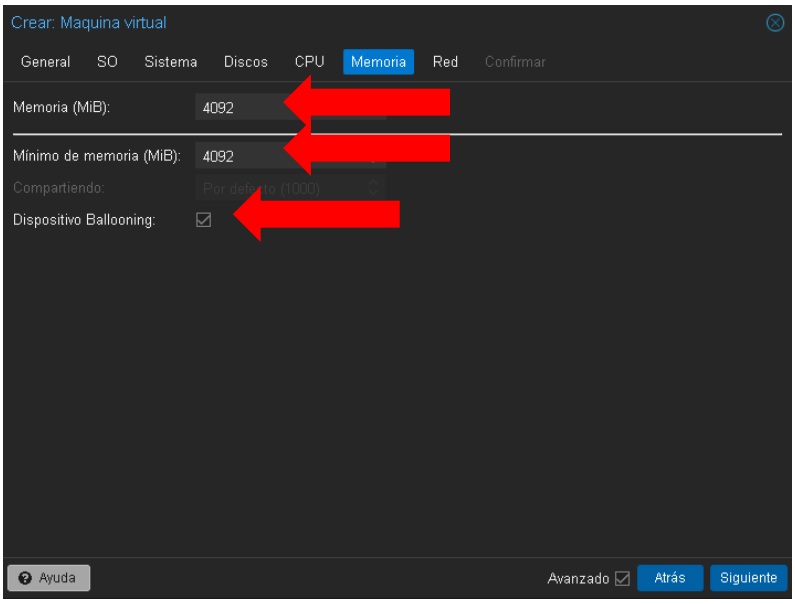
Anexo 6. Configuración del sistema de la VM donde se selecciona el chipset y se activa el Qemu Agent.



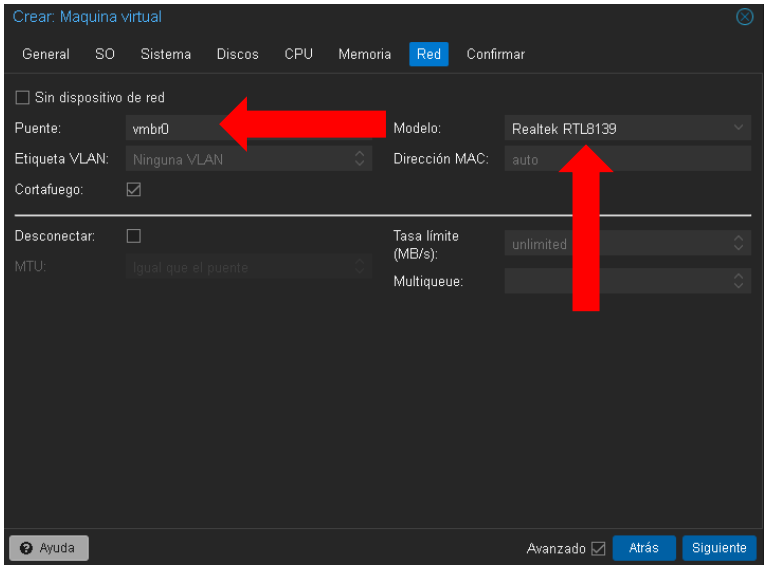
Anexo 7. Muestra donde se guardará la imagen del disco y el tamaño que tendrá dicho disco.



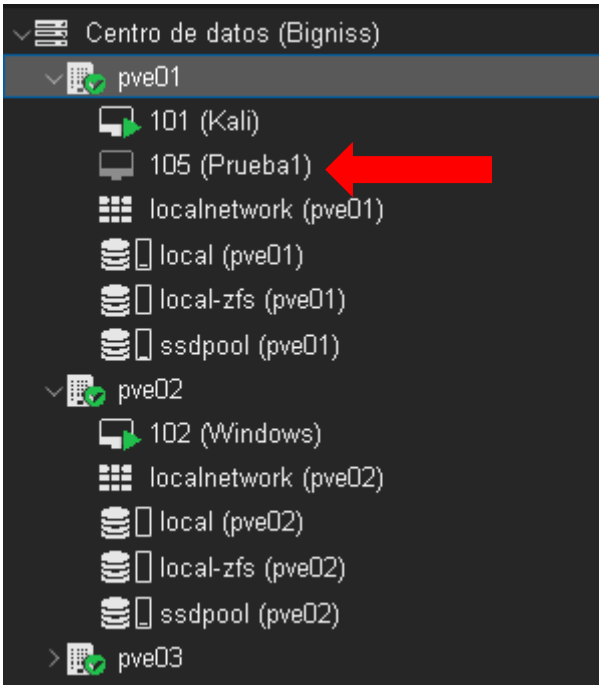
Anexo 8. Configuración de núcleos y tipo de procesador que tendrá la VM.



Anexo 9. Muestra la configuración de memoria RAM en la VM junto con la activación del ballooning.



Anexo 10. Muestra la configuración de la tarjeta de red junto con el puente al que estará conectado.



Anexo 11. Muestra la creación de la máquina virtual finalizada en el nodo correspondiente.