

COMANDOS EN PROXMOX

GESTIONANDO MÁQUINAS VIRTUALES CON EL COMANDO "qm"

- **Listar todas las MV del host:** `qm list`
- **Creación de una MV:**
 - **Máquina virtual:** `qm create <vm id> --name <nombre_vm> --memory <memoria_MB> --cores <núcleos> --net0 <configuración_red> --socket <> --ostype <tipo_sistema> --kvm 0`

Descripción de los parámetros básicos:

<vmid>: ID único de la máquina virtual.

--name <nombre_vm>: Nombre de la máquina virtual.

--memory <memoria_MB>: Cantidad de memoria RAM en MB.

--cores <núcleos>: Número de núcleos de CPU asignados.

--socket<zócalos>: Número de CPUs.

--net0 virtio,bridge=vmbr0: configura una interfaz de red en modo "VirtIO" y usa el bridge vmbr0.

--ostype <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> : Tipo de Sistema Operativo

--kvm 0: Desactiva KVM, ejecutando la VM en un modo de emulación completo (sin aprovechar la aceleración de hardware), útil en casos específicos, como en máquinas virtuales que necesitan una emulación pura.

Otros parámetros:

ide2 local:<Ruta_iso>,media=cdrom: monta una ISO como unidad de CD-ROM

boot c: establece el disco duro como dispositivo de arranque.

scsi0 local-lvm:32: crea un disco SCSI de 32 GB en el almacenamiento local-lvm

--bootdisk scsi0: Define que el disco principal de arranque será el disco asignado a `scsi0`. En este caso, `scsi0` es el identificador de un disco virtual conectado a la VM usando el controlador **SCSI**.

- **Creación del disco:** `qm set <vm id> --scsihw virtio-scsi-single - -scsi0 <nombre-almacenamiento>:<Tamaño en GB>`
Crea un disco de tipo scsi en el almacenamiento que le indiquemos
- **Agregar la iso en CDROM:** `qm set <vm id> --ide0 <Ruta_iso>,media=cdrom`
- **Eliminar una MV:** `qm destroy <vm id>`
- **Resetear una MV:** `qm reset <vm id>`
- **Apagar una MV:** `qm shutdown <vm id>`
- **Apagar todas las MVs:** `qm shutdown all`
- **Encender/Arrancar una MV:** `qm start <vm id>`
- **Parar una MV de inmediato (forzar apagado):** `qm stop <vm id>`
- **Suspender la MV seleccionada:** `qm suspend <vm id>`
- **Desbloquear la MV seleccionada:** `qm unlock <vm id>`
- **Incrementar el tamaño del disco de la MV:** `qm resize <vm id><disco><tamaño en GB>`

Ejemplo: Incrementar el disco virtio en 50 GB en la MV 100: `qm resize 100 virtio +50G`

- **Ayuda sobre el comando qm:** `qm help`
- **Muestra el estado de la MV seleccionada:** `qm status <vm id>`
- **Muestra la configuración de una determinada MV:** `qm config <vm id>`
- Saber la RAM de una MV determinada: `grep config vmid | grep ^memory`
- **Acceder a la consola (Terminal) monitor de una MV indicando su ID:** `qm monitor <vm id>`
Una vez dentro de la consola podemos obtener información de la MV con el comando `info`. **`info status` o `info name`**
- **Mostrar información del build de una MV (sacar la MAC por ejemplo de la tarjeta de red virtual, el nombre de la máquina virtual, unidades, tarjeta gráfica):** `qm showcmd <vm id>`
- **Cambiar la memoria RAM de una máquina virtual:** `qm set <vm id> --memory <tamaño en mb>`
- **Cambiar el idioma del teclado de una MV:** `qm set <vm id> --keyboard "es"`
- **Proteger una MV para que no se pueda eliminar:** `qm set <vm id> --protection 1`
Es booleano. 1 para habilitar y 0 para deshabilitar la protección. Por defecto estará a 0.
- **Activar el arranque automático de una MV:** `qm set <vm id> --onboot 1`

Nota: Las configuraciones de las máquinas virtuales las encontraréis en la ruta: `/etc/pve/qemu-server/`

GESTIONANDO CONTENEDORES CON EL COMANDO “pct”

- **Consulta del comando pct:** `pct --help` o `man pct`
- **Crear un contenedor:** `pct create <ct id> <plantilla> --arch <nombre-arquitectura> --ostype <nombre_sistema> --hostname <nombre_contenedor> --memory <memoria_MB> --cores <núcleos> --net0 <configuración_red> --rootfs <almacenamiento>:<tamaño_GB> --password <mypassword>`

Descripción de los parámetros básicos:

<ct id>: ID único del contenedor.

<plantilla>: Ruta de la plantilla (template) del contenedor, que suele estar en formato .tar.gz en `/var/lib/vz/template/cache`.

--ostype<nombre_sistema>: `debian, ubuntu, centos, fedora, alpine` o `archlinux`

--arch<nombre_arquitectura>: `amd64:` Arquitectura de 64 bits (x86_64), que es la más común en sistemas modernos. `i386:` Arquitectura de 32 bits (x86).

--hostname <nombre_contenedor>: Nombre del contenedor.

--memory <memoria_MB>: Cantidad de memoria RAM en MB.

--cores <núcleos>: Número de núcleos de CPU asignados.

--net0 <configuración_red>: Configuración de red. Por ejemplo, `name=eth0,bridge=vmbbr0`

--rootfs <almacenamiento>:<tamaño_GB>: Almacenamiento y tamaño del sistema de archivos raíz del contenedor. Por ejemplo, `--rootfs local-lvm:8:` Asigna 8 GB de almacenamiento en `local-lvm`.

--password mypassword: Establece `mypassword` como contraseña para el usuario `root`.

- **Listado de contenedores del nodo:** `pct list`
- **Muestra la configuración de un determinado contenedor:** `pct config <ct id>`
- **Arrancar un contenedor:** `pct start <ctd id>`
- **Apagar un contenedor de forma ordenada:** `pct shutdown <ct id>`
- **Detener un contenedor de inmediato:** `pct stop <ct id>`
- **Reiniciar un contenedor:** `pct reboot <ct id>`
- **Eliminar un contenedor:** `pct destroy <ct id>`
- **Acceder al terminal del contenedor:** `pct enter <ct id>`
- **Cambiar la memoria de un contenedor:** `pct set <ct id> --memory <size in mb>`
Ejemplo: reducir la RAM de un contenedor: `pct set <ct id> - -memory 1024`
- **Ver la configuración de un contenedor:** `pct config <ct id>`

PARA LAS PLANTILLAS

- **Listado de todas las plantillas en el repositorio (para descargar):** `pveam available`
- **Listado de plantillas en el sistema:** `pveam available - -section system`
- **Descargar una plantilla:** `pveam download local <nombre plantilla.tar.gz>`
Ejemplo: descargar una plantilla: `pveam download local debian-10.0-standard_10.0-1_amd64.tar.gz`

PARA CLONAR

- **Clonar una MV indicando su vmid actual vmid asignado:** `qm clone <vm id> <vm id-clone>`
- **Ponerle nombre a una máquina virtual:** `qm set <vm id-clone> -name "MV-clonada"`

PARA SNAPSHOT (INSTÁNTANEAS)

- **Realizar Snapshot de una MV determinada incluyendo una descripción:** `qm snapshot <vm id> "prueba-snapshot_20210603" -description "snapshot antes de actualizar wordpress"`
- **Eliminar el Snapshot de una MV:** `qm delsnapshot <vm id> "snapshot-name"`
- **Listar Snapshot de un MV determinada:** `qm listsnapshot <vm id>`
- **Restaurar la instantánea (Snapshot) de una MV:** `qm rollback <vm id> "snapshot-name"`

CREACIÓN DE USUARIOS CON EL COMANDO pveum

pveum es un comando específico para administración de usuarios y permisos. Más sencillo y directo para tareas de gestión de usuarios.

- **Creación de un usuario:** `pveum useradd <user id>@<dominio>`
- **Asignar una contraseña:** `pveum passwd <user id>@<dominio>`

CREACIÓN DE COPIAS DE SEGURIDAD

- **Para crear backups de una MV:** `vzdump ID`
- **Restaurar un backups de una MV:** `vzrestore ID`
- **Otra forma de restaurar un backup de una MV:** `qmrestore ID`

GESTIONANDO EL CLUSTER CON COMANDOS

- **Administración del Sistema**
 - **Obtener la versión de Proxmox:** `pvesh get /versión`
 - **Información del estado del clúster:** `pvesh get /cluster/status`
 - **Obtener información de los logs:** `pvesh get /cluster/log`
 - **Mostrar todos los recursos del clúster:** `pvesh get /cluster/resources`
 - **Mostrar información de los nodos:** `pvesh get /nodes`
 - **Creación de un nuevo clúster:** `pvecmd create <nombre cluster>`
 - **Añadir un nodo al clúster:** `pvecmd add <IP NODE o DNS>`
 - **Ver estado de los nodos:** `pvecmd nodes`
 - **Realizar un test del host físico:** `pveperf`
- **Administración de MV**
 - **Mostrar los recursos del clúster:** `pvesh get /cluster/resources`
 - **Obtener un listado de todas las máquinas virtuales de un nodo:** `pvesh get /nodes/<node id>/qemu`
 - **Mostrar las MVs del clúster:** `pvesh get /cluster/resources --type vm`
 - **Arrancar una MV:** `pve create /nodes/<node id>/qemu/<vm id>/status/start`
Ejemplo: arrancar la VM con id 100: `pvesh create /nodes/virtual1/qemu/100/status/start`
 - **Obtener la configuración de una MV:** `pvesh create /nodes/<node id>/qemu/<vm id>/config`
Ejemplo: obtener toda la configuración de la VM con id 100: `pvesh get /nodes/virtual1/qemu/100/config`
 - **Borrar una MV:** `pvesh delete /nodes/<node id>/qemu/<vm id>`
Ejemplo: Borrar la máquina virtual con id 100: `pvesh delete /nodes/virtual1/qemu/100`
- **Administración de Usuarios con el comando pvesh**

pvesh es el comando general para interactuar con la API de Proxmox. Útil para configuraciones avanzadas y para automatizar tareas en todo el sistema, incluyendo pero no limitado a la gestión de usuarios.

 - **Obtener información de los usuarios:** `pvesh get /access/users`
 - **Crear un usuario con contraseña:** `pvesh create /access/users --userid <user id>@pve --password "<password>"`
 - **Borrar un usuario:** `pvesh delete /access/users/<user id>@pve`
 - **Asignar contraseña a un usuario:** `pvesh set /access/password - - userid "<user id>@pve" - - password "password.123"`
 - **Creación de un pool y asignarlo a un usuario:** `pvesh create pools -poolid "usuario"`

- **Dar permiso al usuario sobre el pool y asignar el rol PVEVMAdmin para que pueda administrar las MV de su pool:** `pvesh set /Access/acl -path /pool/"usuario" -roles PVEVMAdmin -users "<user id>@pve"`
- **Clonar cada plantilla y asociarlas al pool:** `pvesh create /nodes/pve/qemu/501/clone - -pool="usuario" - -newid $(pvesh get /cluster/nextid) - -name "usuario-maquina01" - -full`
Donde 501 es el id de la plantilla que vamos a clonar.
Usuario-maquina01 es el nombre de la MV resultante.
full es clonación completa.
- `pvesh get /cluster/nextid` nos devuelve el siguiente identificador disponible en Proxmox VE, con el fin de asignárselo a la máquina que estamos creando.
- **Administración de grupos**
 - **Obtener información de los grupos:** `pvesh get /access/groups`
 - **Crear un grupo:** `pvesh create /access/groups --groupid <group name>`
 - **Eliminar un grupo:** `pvesh delete /access/groups/<group name>`
 - **Añadir un usuario a un grupo:** `pvesh set /access/users/<user id>@<dominio> --groups <group name>`
 - **Quitar un usuario de un grupo:** `pveum usermod <usuario> -deletgroup <grupo>`
- **Administración de Pools y almacenamiento**
 - **Mostrar los pools existentes:** `pvesh get pools`
 - **Creación de un pool:** `pvesh create /pools -poolid "<pool id>"`
 - **Borrar un pool (Solo si está vacío):** `pvesh delete /pools/<pool id>`
 - **Asignación almacenamiento a un pool:** `pvesh set /pools/nombre-pool -storage local,local-lvm`
 - **Nos muestra los almacenamientos disponibles:** `pvesh get /storage`
 -
- **Gestión de Roles**
 - **Visualizar los roles existentes:** `pvesh get /access/roles/`
 - **Crear un rol y asignar permisos:** `pvesh create /access/roles --roleid<role id> --privs nombre-privilegio`
Ejemplo: crear un rol y asignar permisos (por ejemplo el privilegio VM.Audit):
`pvesh create /access/roles --roleid nombre-rol --privs VM.Audit`
Nota: los privilegios irán separados por comas
 - **Asignar roles a un pool para que puedan acceder los usuarios:** `pvesh set /access/acl -path /pool/"Proyecto.$usuario" -roles rol_proyecto -users "$usuario@pve"`
 - **Asignar a un pool un rol para que pueda acceder un grupo:** `pvesh set /access/acl -path /pool/nombre-pool -roles nombre-rol -groups nombre-grupo`
 - **Eliminar un rol:** `pvesh delete /access/roles/<rol id>`