

API Proxmox

Diciembre 11, 2023

En el presente documento, se busca documentar de manera exhaustiva la interacción con la API de Proxmox, con el propósito de automatizar el proceso de gestión de contenedores y usuarios

El objetivo principal al revisar este documento es comprender el progreso alcanzado con la API de Proxmox. Asimismo, se espera que, como colaborador, puedas construir sobre esta base y continuar trabajando hacia los mismos objetivos. Para lograr una comprensión integral, se considera esencial haber revisado previamente los documentos adjuntos en el repositorio. Este conocimiento previo proporcionará una perspectiva adecuada para aprovechar plenamente la información presentada aquí. Este enfoque está diseñado para satisfacer las necesidades del Departamento de Electrónica e Informática, siendo de utilidad tanto para los estudiantes que buscan familiarizarse con el entorno como para los docentes que desean aprovechar las capacidades avanzadas de la plataforma.

1 Paso a Paso para Crear Usuarios mediante la API con .sh:

1. Creación de Usuario Global y Configuración de Permisos:

Inicialmente, se crea un usuario global en la GUI de Proxmox con el propósito de ser utilizado por la API. Este usuario se configura con los permisos y roles pertinentes para garantizar una gestión de acceso segura al cluster. Es fundamental almacenar el token asignado durante este proceso.

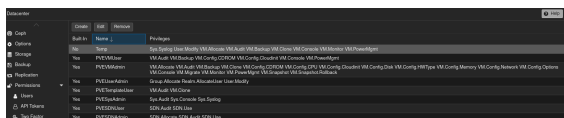


Figure 1: Configuración de Permisos

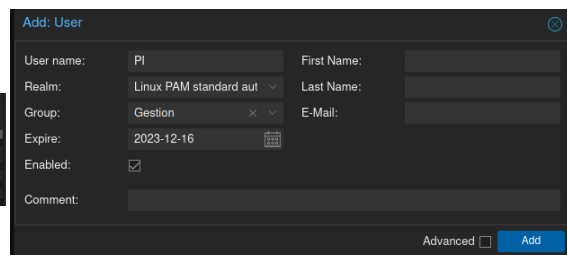


Figure 2: Usuario Global para la interacción con la API

2. Modificación y Ejecución del Script:

Accedemos a nuestra terminal y procedemos a modificar el script llamado **"usuarios.sh"**, el cual se encuentra en el repositorio. Asegúrese de cargar el archivo en su máquina antes de continuar. Abra el archivo con su editor de texto preferido, como vim, y adapte los parámetros según las configuraciones de su nodo de Proxmox. Y finalmente ejecute

```
root@pve:/home/PROXMOX-api-bash# ./alumno.sh
Usuario creado correctamente:
Username: syspve@pam
Password: /BeVB64PGo+mMlXPbvQ=
root@pve:/home/PROXMOX-api-bash#
```

Figure 3: Ejecución de código

```
#!/bin/bash

PROXMOX_HOST="ip"
PROXMOX_USER="user_cred"
PROXMOX_PASSWORD="pass_cred"
REALM="pam"

NEW_USERNAME="sys$(hostname)@pam"
NEW_PASSWORD=$(openssl rand -base64 14)
NEW_ROLE="audi"
API_URL="https://${PROXMOX_HOST}/api2/json/access/users"

create_user() {
    local response
    response=$(curl -s -k -b "PVEAuthCookie=${PROXMOX_TICKET}" -d "roleid=${NEW_ROLE}" -d "realms=${REALM}" "${API_URL}")
}
```

Figure 4: Código de referencia para realizar las modificación y adaptarlo a nuestro nodo

3. Revisión en la GUI o Archivo JSON:

Luego de ejecutar el script, se recomienda revisar la interfaz gráfica (GUI) de Proxmox o el archivo JSON correspondiente para verificar la correcta creación de los usuarios. Este paso garantiza que los usuarios se hayan generado de manera efectiva y estén listos para su utilización.



Figure 5: Verificación en archivo JSON

2 Paso a Paso para la Creación Automática de LXC con .sh:

1. Modificación del Archivo para Crear Containers:

Para iniciar la creación automática de un contenedor, se procederá a modificar el archivo con el nombre **"vm.sh"**. Ajuste los parámetros según sus necesidades específicas. Luego, ejecute el archivo para dar inicio al proceso de creación de las máquinas virtuales. Este script ofrece

diversas opciones, incluyendo create, start, stop y delete.



```

#!/bin/bash

PROXMOX_NODE_IP="IP"
PROXMOX_NODE_NAME=what
PROXMOX_STORAGE="local-lvm"

API_USER="@username_cred "
API_USER_PASSWORD=1 "@password_cred"

CRED="username=$API_USER@pve&password=$API_USER_PASSWORD"

curl --silent --insecure --data $CRED https://$PROXMOX_NODE_IP:8006/
e
curl --silent --insecure --data $CRED https://$PROXMOX_NODE_IP:8006/
ionToken://> token

# container config

CPU=1
CPUUNITS=512
MEMORY=512
DISK=4G
SWAP=0
OS_TEMPLATE="local:vztmpl/debian-12-standard_12.2-1_amd64.tar.zst"

```

```

root@pve:/home/PROXMOX-api-bash# ./proxmox_api.sh start 101
{"data":{"UPID:pve:0000073F:0000AF1A:65842CA8:vzstart:101:api@pve:"}} done.
root@pve:/home/PROXMOX-api-bash# ./proxmox_api.sh start 100
{"data":{"UPID:pve:0000078D:0000BCB4:65842CCB:vzstart:100:api@pve:"}} done.

```

Figure 6: Ejecución de código

Figure 7: Código de referencia para realizar las modificación y adaptarlo a nuestro nodo

2. Verificación en el Nodo Asignado:

Después de ejecutar el script, es recomendable revisar el nodo asignado para confirmar la creación efectiva del contenedor. Además, se sugiere examinar los registros ubicados en la ruta `/var/logs/` para obtener referencias más detalladas en caso de posibles errores durante el proceso.



```

JSON  Datos en bruto  Cabeceras
Guardar Copiar Contraer todo Expandir todo Filtrar
▼ data:
  ▼ 0:
    name: "ff"
    status: "running"
    netin: 0
    disk: 637513728
    diskwrite: 74973184
    mem: 17293312
    netout: 0
    pid: 2051
    maxmem: 536870912
    type: "lxc"
    swap: 0
    maxswap: 536870912
    uptime: 2587
    maxdisk: 8350298112
    cpus: 1
    vmid: "100"
    diskread: 398286848
    cpu: 0.000107357868126587

```

Figure 8: Archivo JSON con la nueva VM

- Con el fin de abordar ambas tareas, se ha adaptado un código en formato `.py` que se encuentra dentro del repositorio para brindar flexibilidad en su uso. Es esencial que este archivo `.py` cuente con las configuraciones especificadas en los pasos 1 y 2 mencionados previamente, asegurando así una ejecución sin complicaciones.

References

https://pve.proxmox.com/wiki/Proxmox_VE_API_Test_auth_credentials
<https://github.com/baseblack/Proxmoxia>
<https://github.com/sitrox/proxmox-utils>
<https://github.com/Corsinvest/cv4pve-api-dotnet>
<https://github.com/Corsinvest/cv4pve-cli>
<https://pve.proxmox.com/pve-docs/pve-admin-guide.html>
<https://pve.proxmox.com/pve-docs/api-viewer/#nodes/node/qemu/vmid/clone>
`curl -k -d 'username=root@pam' -data-urlencode 'password=xxxxxxxx'`
<https://10.0.0.1:8006/api2/json/access/ticket>
`curl -k -d 'username=root@pam' -data-urlencode "password@$HOME/.pve-pass-file"`
<https://10.0.0.1:8006/api2/json/access/ticket>

