ChatGPT

Tài liệu API REST Proxmox VE 8.4.1 - Nhóm Access (Quản lý truy cập)

---

1. Đồng bộ người dùng/group từ LDAP

Endpoint: `POST /access/domains/{realm}/sync`

Chức năng: Đồng bộ người dùng và nhóm từ domain LDAP về Proxmox (dữ liệu local)

CLI tương đương:

```
pvesh create /access/domains/{realm}/sync
```

Tham số:

- `realm` (string) - Tên domain LDAP (bắt buộc)
- `dry-run` (bool) - Chỉ test, không ghi dữ liệu (default: 0)
- `enable-new` (bool) - Kích hoạt user mới ngay (default: 1)
- `full` (bool) - **Deprecated** – thay bằng `remove-vanished`
- `purge` (bool) - **Deprecated** – thay bằng `remove-vanished`
- `remove-vanished` (string) – Loại bỏ user/group không còn trong LDAP, có thể là:
- `entry` – xoá luôn user không có trong LDAP
- `properties` – xoá các thuộc tính local đã chỉnh tay
- `acl` – gỡ bỏ quyền ACL nếu không còn user
- `none` – không làm gì cả
- `scope` (enum) – Lựa chọn `users`, `groups` hoặc `both`

---

2. 🏷 Xem thông tin group

Endpoint: `GET /access/groups/{groupid}`

Chức năng: Lấy cấu hình chi tiết của 1 group theo ID

CLI: `pvesh get /access/groups/{groupid}`

Tham số:

- `groupid` (string) – Tên nhóm

Kết quả trả về:

```
{
  "members": ["user1@pam", "user2@ldap"],
  "comment": "Group for DevOps team"
}
```

---

3. Tạo URL OpenID

Endpoint: `POST /access/openid/auth-url`

Chức năng: Lấy URL để xác thực OpenID ở 1 realm

Tham số:

- `realm` (string): Tên realm
- `redirect-url` (string): URL redirect của client (thường là `window.location.origin`)

Trả về: Chuỗi URL để chuyển hướng trình duyệt xác thực OpenID

---

4. Xác thực OpenID (login)

Endpoint: `POST /access/openid/login`

Chức năng: Dùng mã `code` nhận từ OpenID để tạo ticket đăng nhập vào Proxmox

Tham số:

- `code` (string) – mã OpenID authorization
- `redirect-url` (string)
- `state` (string)

Trả về:

```
{
  "ticket": "PVE:...",
  "username": "user@realm",
  "cap": {...},
  "CSRFPreventionToken": "..."
}
```

---

5. 🛡️ Lấy thông tin Role

Endpoint: `GET /access/roles/{roleid}`

Chức năng: Trả về danh sách các quyền mà Role đang sở hữu

Tham số:

- `roleid` (string): Tên Role (VD: `PVEAdmin`)

Trả về: object chứa các quyền dạng boolean như:

- `Datastore.Allocate`, `Group.Allocate`, `Mapping.Modify`, ...

---

6. Xem danh sách TFA của user

Endpoint: `GET /access/users/{userid}/tfa`

Chức năng: Trả về danh sách các loại xác thực 2 bước của user

Tham số:

- `userid` (string): định danh user (VD: `admin@pam`)
- `multiple` (boolean, default 0): nếu bật sẽ trả về danh sách nhiều thiết lập

Trả về:

```
{
  "types": ["totp", "recovery"],
  "user": "totp",
  "realm": "pam"
}
```

---

7. 🔓 Mở khoá TFA cho user

Endpoint: `PUT /access/users/{userid}/unlock-tfa`

Chức năng: Mở khoá xác thực 2 bước nếu người dùng bị lock

Tham số:

- `userid` (string) – định danh user (VD: `admin@pam`)

Trả về: boolean (`true` nếu mở khoá thành công)

Yêu cầu quyền: `User.Modify`

---

8. ⛓ Xem danh sách ACL

Endpoint: `GET /access/acl`

Chức năng: Trả về danh sách các dòng phân quyền (ACL – Access Control List)

Tham số bắt buộc:

- `path` – đường dẫn (VD: `/vms`, `/`)
- `roleid` – tên vai trò
- `ugid` – ID user hoặc group
- `type` – kiểu: `user`, `group`, hoặc `token`

Tuỳ chọn:

- `propagate` – cho phép kế thừa quyền xuống cấp con (mặc định: true)

---

9. 🎫 Lấy thông tin API Token

Endpoint: `GET /access/users/{userid}/token/{tokenid}`

Chức năng: Lấy chi tiết API token đã tạo cho user

Tham số:

- `userid` (string): định danh user (VD: `admin@pam`)
- `tokenid` (string): định danh token (VD: `apitoken`)

Trả về:

```
{
  "comment": "Token truy cap API",
  "expire": 0,
  "privsep": true
}
```

Yêu cầu quyền: `User.Modify` hoặc truy cập chính token của mình

Cần giúp viết thêm tài liệu cho nhóm cluster, nodes hay storage không? Để mình làm luôn nhé!