

✓ پیشنهاد عملی

بخش اول: استفاده از netbox-agent ²

این ابزار از nmap در پشت صحنه استفاده می‌کند. پس باید nmap نصب باشد.

روی سروری که:

- دسترسی به کل رنج IP مورد نظر دارد
- به اینترنت یا حداقل به NetBox API دسترسی دارد

1. نصب کن:

```
pip install netbox-agent
```

```
pip install --upgrade netbox-agent
```

2. و بعد اجرا کن مثلاً:

```
netbox-agent discover --range 192.168.0.0/24 --netbox-url http://192.168.0.10 --token YOUR_TOKEN
```

3. و در صورت نیاز می‌تونی با cron یا systemd timer این کار رو زمان‌بندی کنی.

4.

خودش به صورت خودکار IP های کشف شده رو در NetBox ثبت می‌کنه) در (IPAM → IP Addresses).

```
netbox-agent discover \
```

```
--range 192.168.0.0/24 \
```

```
--netbox-url http://192.168.0.10 \
```

```
--token YOUR_NETBOX_TOKEN
```

بخش دوم: استفاده از اسکریپت برای ثبت ip و hostname به صورت اتوماتیک: ²

اینجا یک اسکریپت پایتون برات آماده می‌کنم که:

1. شبکه رو اسکن می‌کنه) از رنج IP مشخص شده).
2. اطلاعات IP ها رو می‌گیره) مثل MAC Address و (Hostname).
3. به صورت اتوماتیک اطلاعات IP ها رو داخل NetBox ثبت می‌کنه.
4. در صورت نیاز، Device و Interface جدید می‌سازه.

مراحل کار اسکریپت:

1. اسکن شبکه با nmap برای پیدا کردن IP ها و اطلاعات مرتبط.

2. ارسال اطلاعات به NetBox از طریق API برای ثبت IP Address ها.

3. اگر دستگاه‌های جدید پیدا بشه، Device و Interface جدید ایجاد می‌کنه.

پیش‌نیازها:

- نصب کتابخانه‌های پایتون requests, nmap :

```
pip install python-nmap requests
```

اسکرپت پایتون :

```
import nmap
```

```
import requests
```

```
import socket
```

```
#تنظیمات NetBox
```

```
NETBOX_URL = "http://your-netbox.local" # آدرس NetBox خود
```

```
API_TOKEN = "your_api_token" # توکن API که قبلاً ساخته‌ای
```

```
HEADERS = {
```

```
"Authorization": f"Token {API_TOKEN}",
```

```
"Content-Type": "application/json"
```

```
}
```

```
def scan_network(cidr):
```

```
    """ شبکه رو اسکن می‌کنه و IP ها رو پیدا می‌کنه """
```

```
    scanner = nmap.PortScanner()
```

```
    scanner.scan(hosts=cidr, arguments='-sn') # فقط پینگ (برای شناسایی فعال بودن)
```

```
    return scanner.all_hosts()
```

```
def get_host_info(ip):
```

```
    """ اطلاعات Host مثل MAC و Hostname رو می‌گیره """
```

```
    try:
```

```
host_info = {}
```

```
    گرفتن Hostname از IP    host_info['hostname'] = socket.gethostbyaddr(ip)[0] #
```

```
except (socket.herror, socket.gaierror):
```

```
    در صورتی که نتونه Hostname پیدا کنه    host_info['hostname'] = None #
```

```
return host_info
```

```
def create_ip_address(ip):
```

```
    """ IP جدید رو در NetBox ثبت می‌کنه """
```

```
    payload = {
```

```
        "address": f"{ip}/32",
```

```
        "status": "active",
```

```
    }
```

```
response = requests.post(f"{NETBOX_URL}/api/ipam/ip-addresses/", headers=HEADERS,  
                          json=payload)
```

```
if response.status_code == 201:
```

```
print(f"IP {ip} successfully added to NetBox.")
```

```
else:
```

```
print(f"Failed to add IP {ip} to NetBox: {response.text}")
```

```
def create_device(ip, hostname):
```

```
    """ اگر دستگاه جدید پیدا شد، Device جدید در NetBox می‌سازد """
```

```
    payload = {
```

```
        "name": hostname if hostname else ip,
```

```
        "device_type": 1, # باید نوع Device رو بر اساس پیکربندی NetBox انتخاب کنی
```

```
        "device_role": 1, # باید رول دستگاه رو انتخاب کنی
```

```
        "site": 1, # باید سایت رو انتخاب کنی
```

```
    }
```

```
response = requests.post(f"{NETBOX_URL}/api/dcim/devices/", headers=HEADERS, json=payload)
```

```
if response.status_code == 201:
```

```
    print(f"Device {hostname if hostname else ip} successfully created.")
```

else:

```
print(f"Failed to create device {hostname if hostname else ip}: {response.text}")
```

```
def create_interface(device_id, ip):
```

```
    """ Interface جدید رو برای Device می‌سازه """
```

```
        payload = {
```

```
            "name": "eth0", # اسم Interface رو مشخص کن
```

```
            "device": device_id,
```

```
            "ip_addresses": [{"address": f"{ip}/32"}]
```

```
        }
```

```
response = requests.post(f"{NETBOX_URL}/api/dcim/interfaces/", headers=HEADERS, json=payload)
```

```
if response.status_code == 201:
```

```
    print(f"Interface for device {device_id} successfully created.")
```

else:

```
print(f"Failed to create interface for device {device_id}: {response.text}")
```

```
def main():
```

```
    cidr = "192.168.1.0/24" # رنج شبکه‌ای که می‌خواای اسکن کنی
```

```
    hosts = scan_network(cidr)
```

```
    for host in hosts:
```

```
        host_info = get_host_info(host)
```

```
        create_ip_address(host) # ثبت IP در NetBox
```

```
        if host_info['hostname']: # اگر Hostname پیدا شد
```

```
            create_device(host, host_info['hostname']) # Device جدید می‌سازد
```

```
    else:
```

```
        create_device(host, None) # اگر Hostname نبود، Device با IP می‌سازد
```

```
if __name__ == "__main__":
```

```
    main()
```


توضیحات:

- اسکن شبکه: با nmap به رنج IP مشخص شده پینگ می‌زنیم تا دستگاه‌های فعال رو پیدا کنیم.
- گرفتن اطلاعات Hostname: برای هر IP از (`socket.gethostbyaddr()`) استفاده می‌کنیم تا اگر hostname در دسترس بود، اون رو ثبت کنیم.
- ثبت IP در NetBox: با استفاده از API ، IP جدید رو داخل بخش ipam NetBox ثبت می‌کنیم.
- ساخت Device و Interface: در صورتی که دستگاه جدید پیدا بشه، Device جدید در NetBox ساخته میشه.

نکات:

1. نوع Device و Site: در API باید `device_type`, `device_role` و `site` مشخص بشه. این‌ها رو باید بر اساس پیکربندی NetBox خودت تنظیم کنی.
2. توکن API: مطمئن شو که توکن API دسترسی کافی به بخش‌های IPAM و DCIM رو داشته باشه.
3. بسته به نیازت: می‌تونی اسکریپت رو برای نیازهای خاص خودت تغییر بدی (مثلاً گرفتن اطلاعات بیشتر از دستگاه‌ها یا تنظیمات پیشرفته‌تر).