

Лабораторная работа № 3.

Развертывание Netbox, сеть связи как источник правды в
системе технического учета Netbox

Выполнил: Морозов Матвей
Группа К3320

Цель работы

С помощью Ansible и Netbox собрать всю возможную информацию об устройствах и сохранить их в отдельном файле

Ход работы

Клонируем репозиторий NetBox

Git clone -b master -depth 1 <https://github.com/netbox-community/netbox.git>

```
Cloning into '.'...
remote: Enumerating objects: 1924, done.
remote: Counting objects: 100% (1924/1924), done.
remote: Compressing objects: 100% (1700/1700), done.
remote: Total 1924 (delta 337), reused 821 (delta 170), pack-reused 0 (from 0)
Receiving objects: 100% (1924/1924), 16.60 MiB | 1.10 MiB/s, done.
Resolving deltas: 100% (337/337), done.
```

Создаем виртуальное окружение, устанавливаем зависимости и настраиваем конфигурационный файл configuration.py. Генерируем секретный ключ

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python3 generate_secret_key.py
```

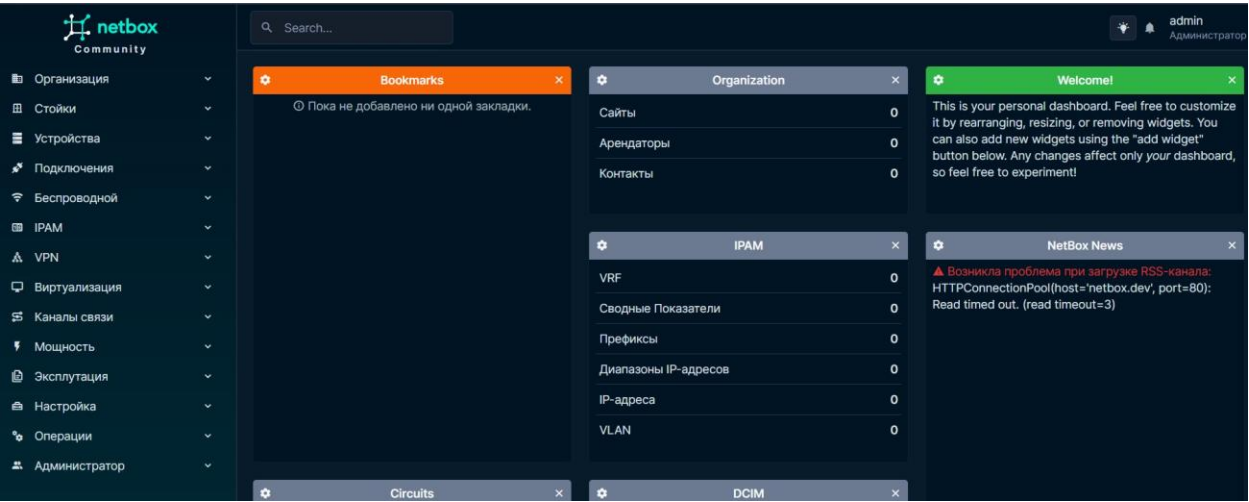
Инициализируем БД и создаем суперпользователя

```
python3 manage.py migrate
python3 manage.py createsuperuser
```

Устанавливаем Gunicorn - это WSGI-сервер, который запускает Python-приложение NetBox и обрабатывает запросы, и Nginx - веб-сервер, который используется как обратный прокси для Gunicorn. После этого запустим netbox и nginx

```
• netbox.service - NetBox WSGI Service
  Loaded: loaded (/etc/systemd/system/netbox.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-07-20 12:00:00 UTC; 1min 1s ago
    Docs: https://docs.netbox.dev/
  Main PID: 1160 (gunicorn)
    Tasks: 6 (limit: 2226)
  Memory: 507.3M
```

Перебрасываем порты в VB, переходим по ссылке <https://localhost:8000>



Во вкладке Устройства была внесена информация о двух роутерах R1 и R2 (роль, производитель, тип, IP-адрес)

Устройства

Результаты 2

Фильтры

Быстрый поиск

Настроить таблицу

ИМЯ	СТАТУС	АРЕНДАТОР	САЙТ	ЛОКАЦИЯ	СТОЙКА	РОЛЬ	ПРОИЗВОДИТЕЛЬ	ТИП	IP-АДРЕС	
R1	Активный	—	mrsh	—	—	Router	Mikrotik	RouterOS	172.20.10.5/28	<div></div>
R2	Активный	—	mrsh	—	—	Router	Mikrotik	RouterOS	172.20.10.2/28	<div></div>

показывая 1-2 из 2

Добавить компоненты

Изменить выбранное

Переименовать

Удалить выбранное

На страницу

Сохранение данных из NetBox в файл inventory.yaml

```
all:
  hosts:
    localhost:
      ansible_connection: local
```

Роль `fetch_info` (`roles/fetch_info/tasks/main.yaml`) для сбора данных и сохранения их в файл:

```
---
- name: NetBox Connect
  uri:
    url: "{{ netbox_url }}/api/dcim/devices/"
    headers:
      Authorization: "Token {{ netbox_token }}"
    validate_certs: false
    return_content: true
  register: netbox_response

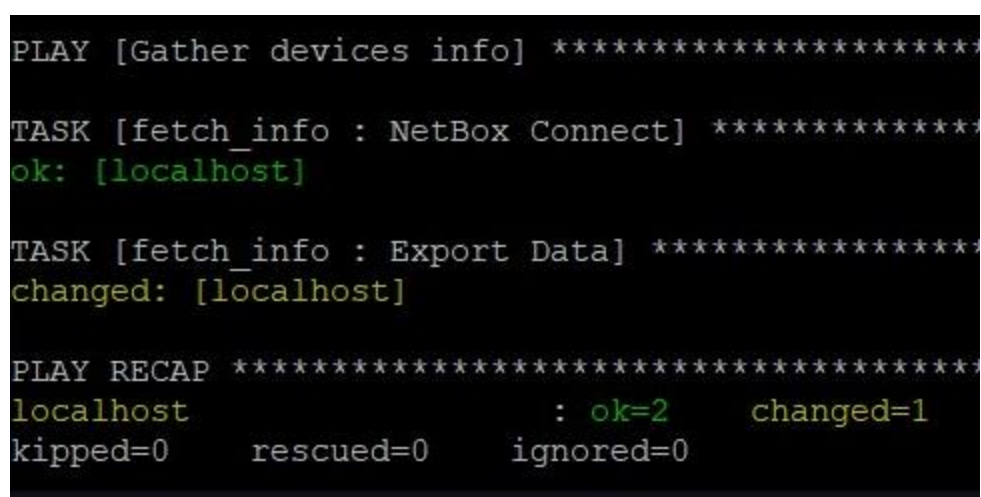
- name: Export Data
  copy:
    content: "{{ netbox_response.json | to_nice_json }}"
    dest: "{{ output_file }}"
```

Ansible плейбук `playbook.yaml`:

```
---
- name: Gather devices info
  hosts: localhost
  gather_facts: false
  vars:
    netbox_url: "https://10.0.2.15:443"
    netbox_token: "b0e1baf3819699b9f81b68096e13a811377f8f7a"
    output_file: "netbox_devices.json"
  roles:
    - fetch_info
```

Запуск выполнения плейбука:

```
ansible-playbook -i inventory.yaml playbook.yaml
```



```
PLAY [Gather devices info] *****

TASK [fetch_info : NetBox Connect] *****
ok: [localhost]

TASK [fetch_info : Export Data] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=2    changed=1
kipped=0    rescued=0    ignored=0
```

Получили файл `netbox_devices.json`

Изменение имени устройства и добавление IP адреса

Файл inventory.yaml

```
all:
  hosts:
    localhost:
      ansible_connection: local
    chr_host:
      ansible_host: 192.168.56.107
      ansible_user: admin
      ansible_connection: ansible.netcommon.network_cli
      ansible_network_os: community.routeros.routeros
      ansible_ssh_private_key_file: /root/.ssh/id_rsa
```

Создан плейбук для изменения настроек роутера в соответствии с данными из NetBox

```
---
- name: Fetch Info from NetBox
  hosts: localhost
  gather_facts: false
  vars:
    netbox_url: "https://10.0.2.15:443"
    netbox_token: "b0e1baf3819699b9f81b68096e13a811377f8f7a"

  tasks:
    - name: Fetch Info
      uri:
        url: "{{ netbox_url }}/api/dcim/devices"
        headers:
          Authorization: "Token {{ netbox_token }}"
        method: GET
        return_content: yes
        validate_certs: false
        register: device_data

    - name: Export Name and IP Address
      set_fact:
        device_name: "{{ device_data.json.results[1].name }}"
        netbox_ip_address: "{{ device_data.json.results[1].primary_ip.address }}"

- name: Configure CHR
  hosts: chr_host
  gather_facts: false

  tasks:
    - name: Change Name
      community.routeros.command:
        commands:
          - /system identity set name={{ hostvars['localhost'].device_name }}

    - name: Add IP address
      community.routeros.command:
        commands:
          - /ip address add address={{ hostvars['localhost'].netbox_ip_address }} interface=netbox disabled=no
```

После выполнения плейбука, проверяем изменения на роутере:

Изменение имени устройства с MikroTik на R2

```
CHR2 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
DfAc  172.20.10.0/28  ether1      0
DfAc  192.168.56.0/24 ether2      0
[admin@MikroTik] /ip/route> /
[admin@MikroTik] > ping 10.66.66.1
SEQ HOST                                SIZE TTL TIME          STATUS
0 10.66.66.1                            56  64 3ms224us
1 10.66.66.1                            56  64 1ms105us
2 10.66.66.1                            56  64 1ms525us
sent=3 received=3 packet-loss=0% min-rtt=1ms105us avg-rtt=1ms951us
max-rtt=3ms224us
[admin@MikroTik] > /ip address/print
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS NETWORK INTERFACE
0 10.66.66.3/24 10.66.66.0 wg-client
1 2.2.2.2/32 2.2.2.2 loopback
2 D 172.20.10.2/28 172.20.10.0 ether1
3 D 192.168.56.107/24 192.168.56.0 ether2
[admin@MikroTik] >
[admin@R2] >
[admin@R2] >
[admin@R2] >
[admin@R2] >
[admin@R2] >
```

Добавление IP-адреса 192.168.0.105/24

```
CHR2 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
# ADDRESS NETWORK INTERFACE
0 10.66.66.3/24 10.66.66.0 wg-client
1 2.2.2.2/32 2.2.2.2 loopback
2 D 172.20.10.2/28 172.20.10.0 ether1
3 D 192.168.56.107/24 192.168.56.0 ether2
[admin@R2] /interface> /ip address/print
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS NETWORK INTERFACE
0 10.66.66.3/24 10.66.66.0 wg-client
1 2.2.2.2/32 2.2.2.2 loopback
2 D 172.20.10.2/28 172.20.10.0 ether1
3 D 192.168.56.107/24 192.168.56.0 ether2
[admin@R2] /interface>
[admin@R2] /interface>
[admin@R2] /interface> /ip address/print
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS NETWORK INTERFACE
0 10.66.66.3/24 10.66.66.0 wg-client
1 2.2.2.2/32 2.2.2.2 loopback
2 D 172.20.10.2/28 172.20.10.0 ether1
3 D 192.168.56.107/24 192.168.56.0 ether2
4 192.168.0.105/24 192.168.0.0 netbox
[admin@R2] /interface>
```

Добавление серийного номера в NetBox

Сперва собираем информацию о серийном номере устройства

```
bash /system license print
```

Затем выполняем PATCH-запрос, чтобы обновить серийный номер устройства в Netbox

```
---
- name: CHR
  hosts: chr_host
  gather_facts: false

  tasks:
    - name: Get Serial Number
      community.routeros.command:
        commands:
          - /system license print
        register: serial_output

    - name: Parse Output
      set_fact:
        serial_number: "{{ serial_output.stdout_lines[0][0]| regex_search('system-id: (\\S+)', '\\1') }}"

    - name: Parse Output
      debug:
        var: serial_number

- name: NetBox
  hosts: localhost
  gather_facts: false
  vars:
    netbox_url: "https://10.0.2.15:443"
    netbox_token: "b0e1baf3819699b9f81b68096e13a811377f8f7a"

  tasks:
    - name: Update Serial Number in NetBox
      uri:
        url: "{{ netbox_url }}/api/dcim/devices/2/"
        method: PATCH
        headers:
          Authorization: "Token {{ netbox_token }}"
          Content-Type: "application/json"
        body:
          serial: "{{ hostvars['chr_host'].serial_number[0]| string }}"
        body_format: json
        validate_certs: no
        register: update_response
```

Проверка

Серийный номер

d4UM3N4aNqD