Automating Zabbix with Netbox and Ansible

About me

- Working @ AP Hogeschool
- Network and security expert
- Network automation enthusiast

@jefvantongerloo

Linkedin.com/in/jefvantongerloo

github.com/jefvantongerloo/zabbix-automation-demo

Network automation

***What*, Why & How?**

- Use of programmable logic and code to describe the network
  - Intent based networking
  - Predictable state

Network automation

***What*, Why & How?**

- Use of programmable logic and code to describe the network
  - Intent based networking
  - Predictable state
- Automation of network processes and operations
  - Configuration management
  - Device discovery and provisioning
  - Operational tasks

Network automation

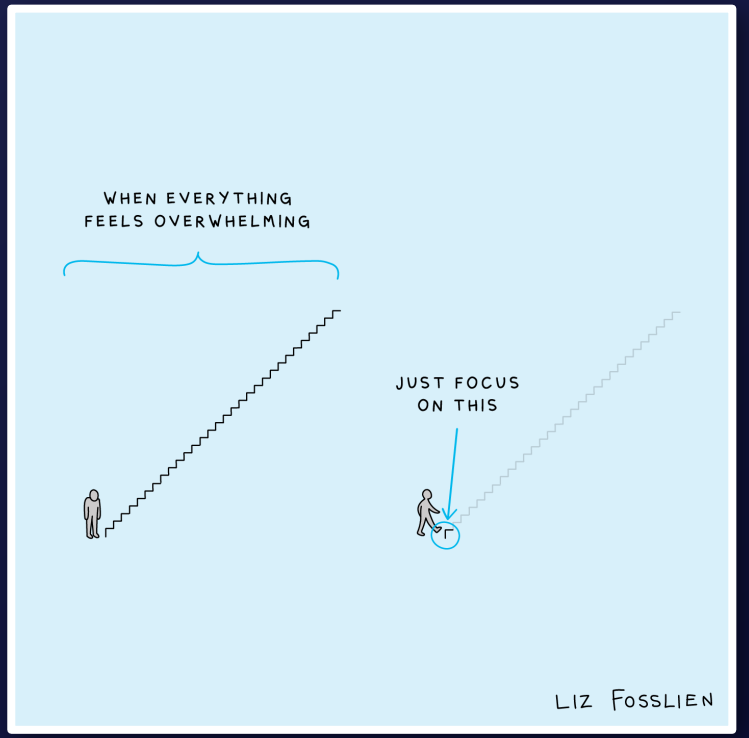
What, Why & How?

- Eliminates manual error prone tasks
- Improves efficiency
- Scales from cow to cattle
- Vendor independent

Network automation

What, Why & How?

- Think big, start small
- Start with read-only operations
- KISS – Keep it simple, stupid
  - Complexity lies into the problem, not the solution

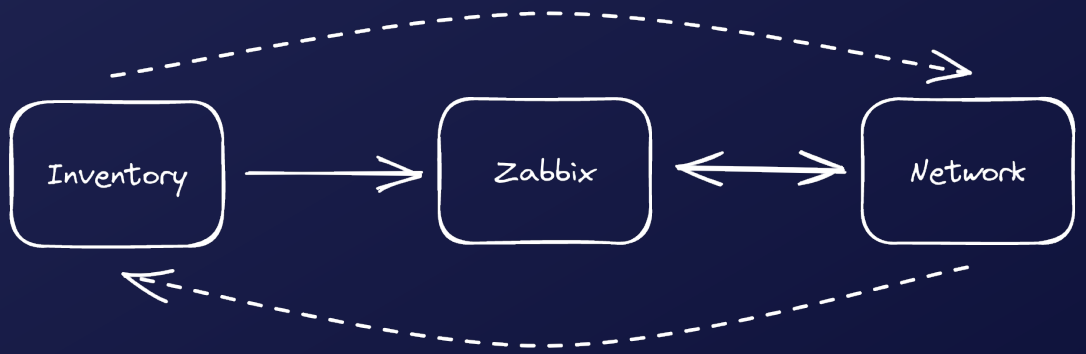


Liz Fosslien, https://fosslien.com

Case study

What do we want to achieve?

- An Automated flow for on-boarding, synchronizing and configuring network devices from our source of truth to Zabbix for enabling SNMP monitoring.
- A closed loop and hands-off approach, set and forget.



Case study

Automate all the things!

- Automated Zabbix host and interface creation
- Automated Zabbix templates and groups configuration
- Automated Network device SNMP configuration and validation
- Automated source of truth as inventory
- Automated callback to our inventory journal for logging

Case study

Open-source rules the world

- Infrastructure manager as source of truth
  - Network & device modeling
  - API integration
  - Device journal

Case study

Open-source rules the world

- Infrastructure manager as source of truth
  - Network & device modeling
  - API integration
  - Device journal

- Automating tasks and processes
  - Modular with Ansible collections
  - API interactions through libraries
  - Network CLI interaction capabilities
  - Configuration templating

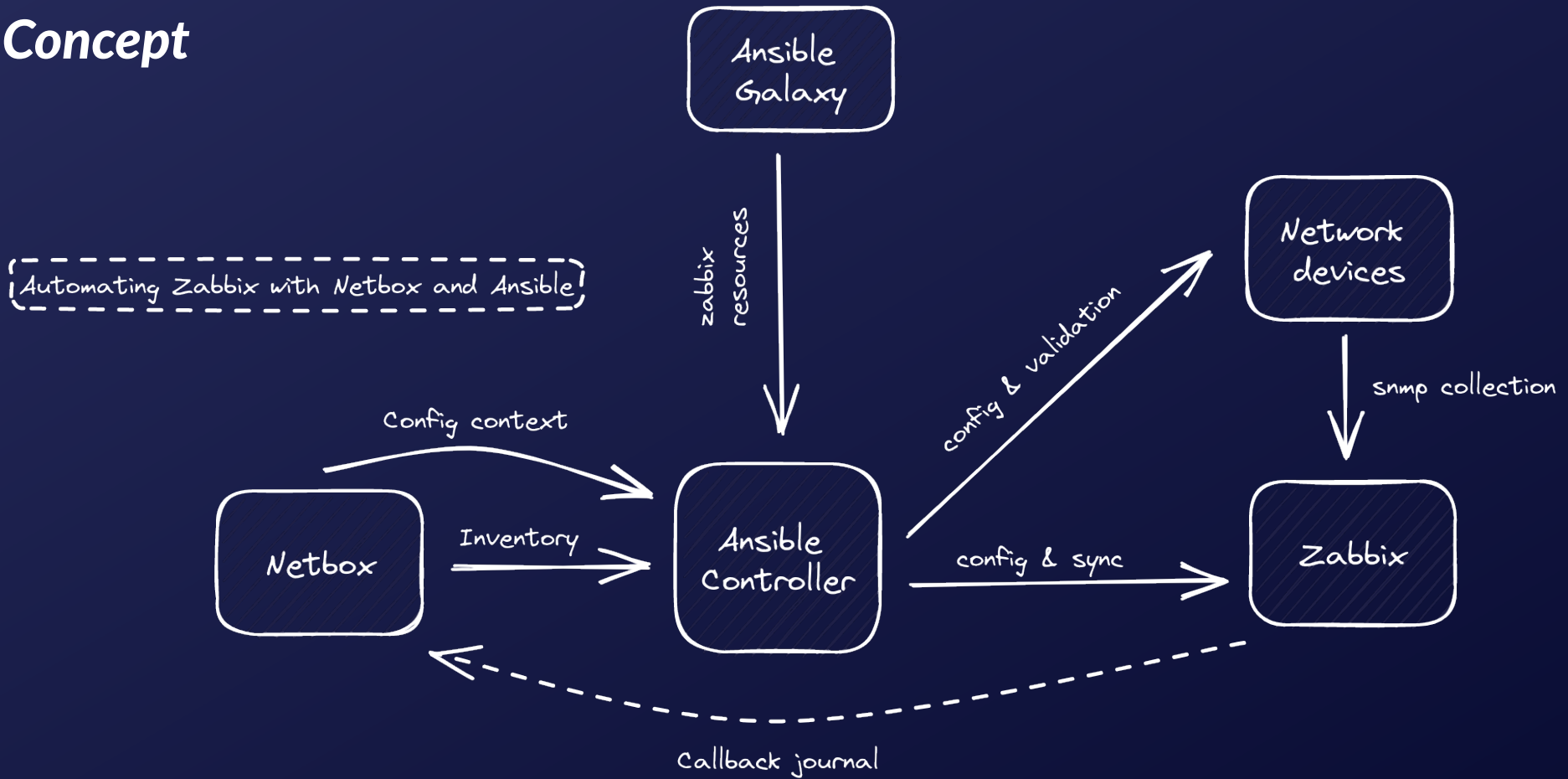
Case study

Open-source rules the world

- Infrastructure manager as source of truth
  - Network & device modeling
  - API integration
  - Device journal
- Automating tasks and processes
  - Modular with Ansible collections
  - API interactions through libraries
  - Network CLI interaction capabilities
  - Configuration templating
- Monitoring solution
  - Community Ansible collection
  - API integration

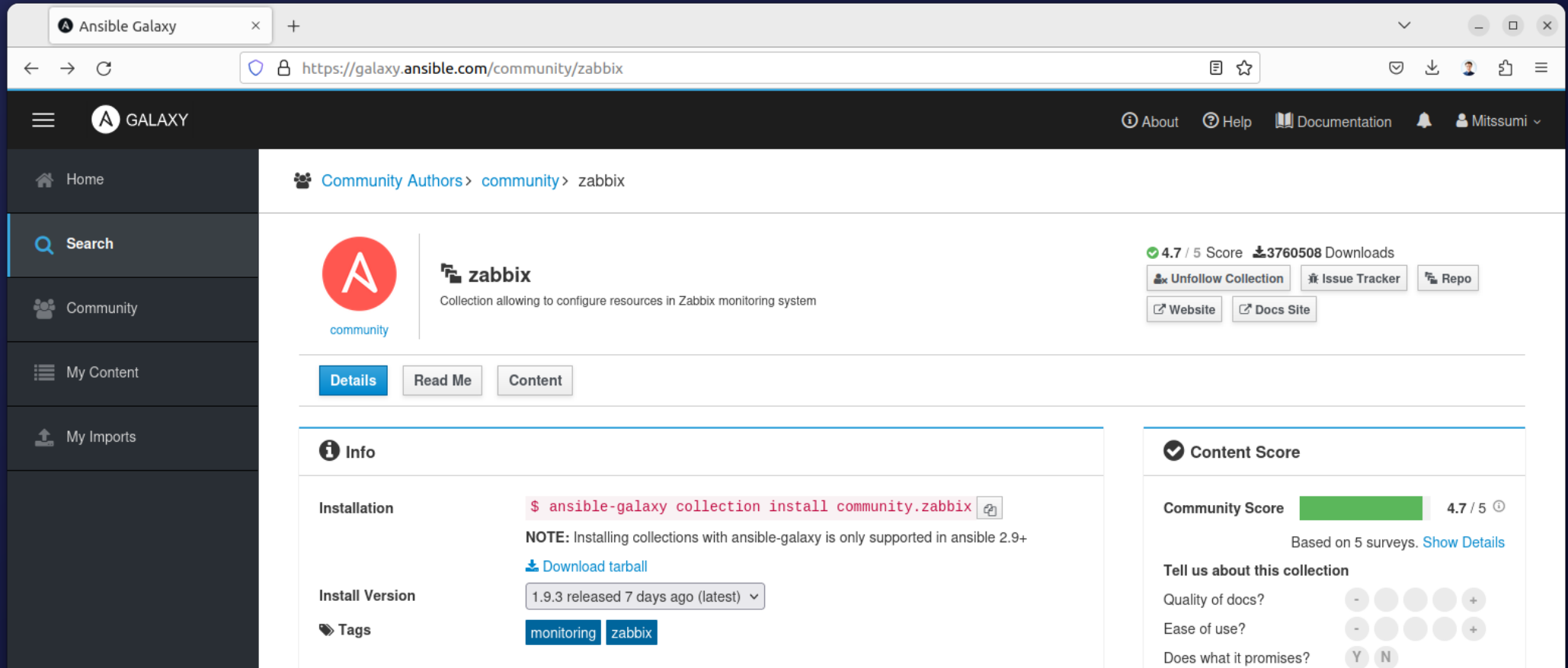
Case study

Concept



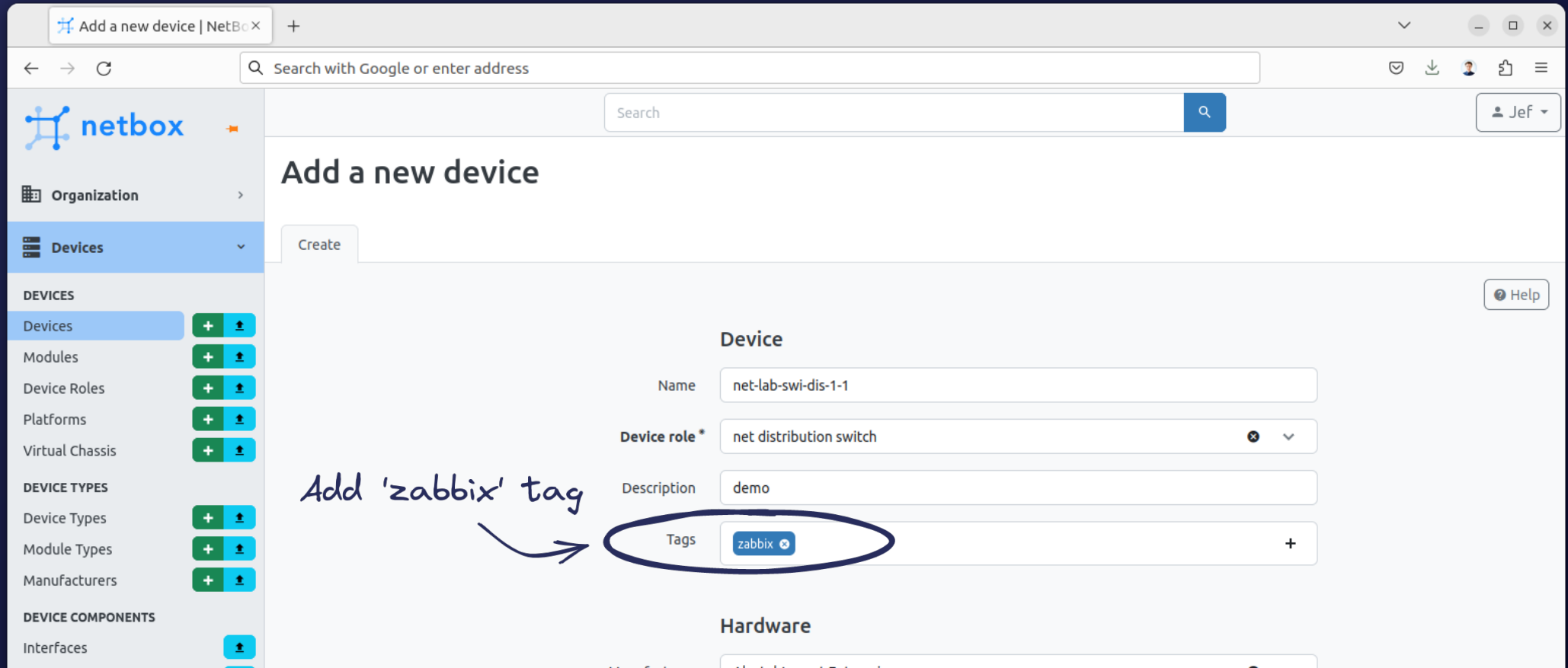
Case study

Ansible Galaxy



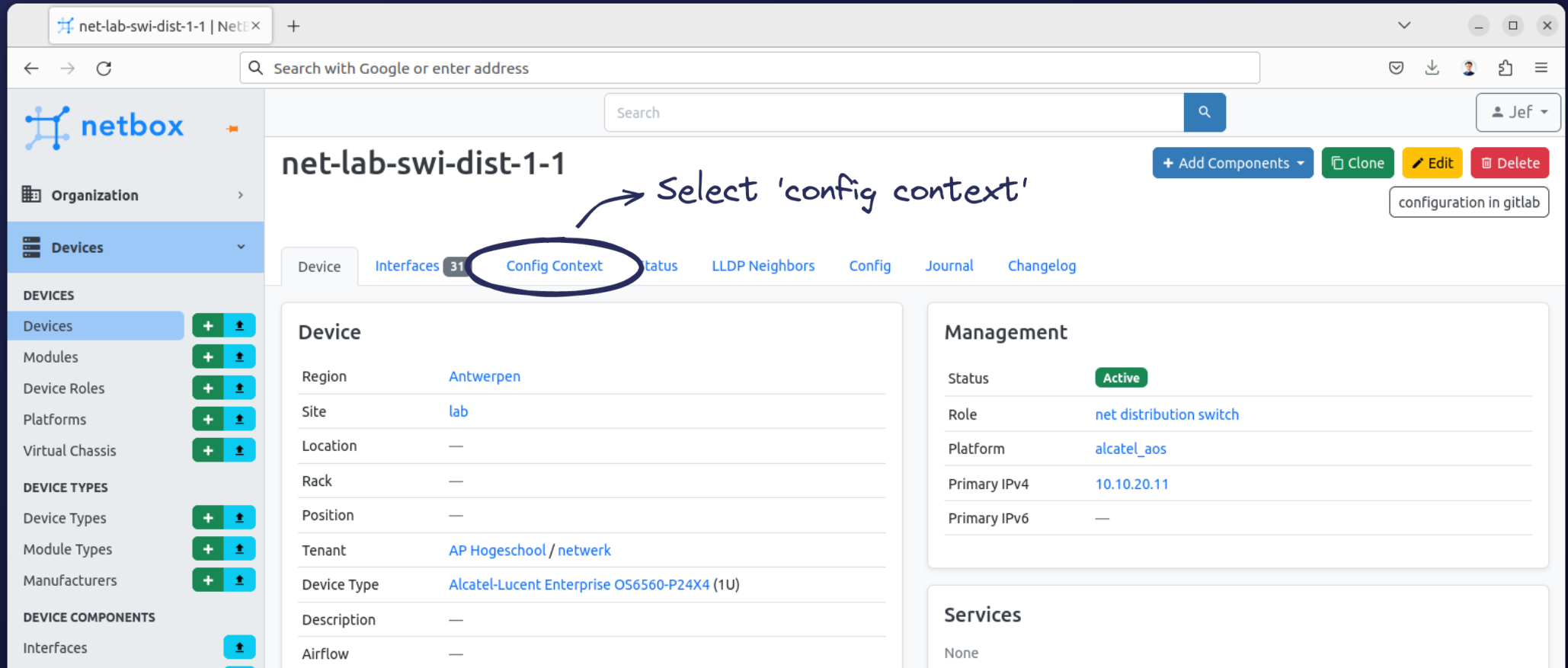
Demo

Add device to Netbox



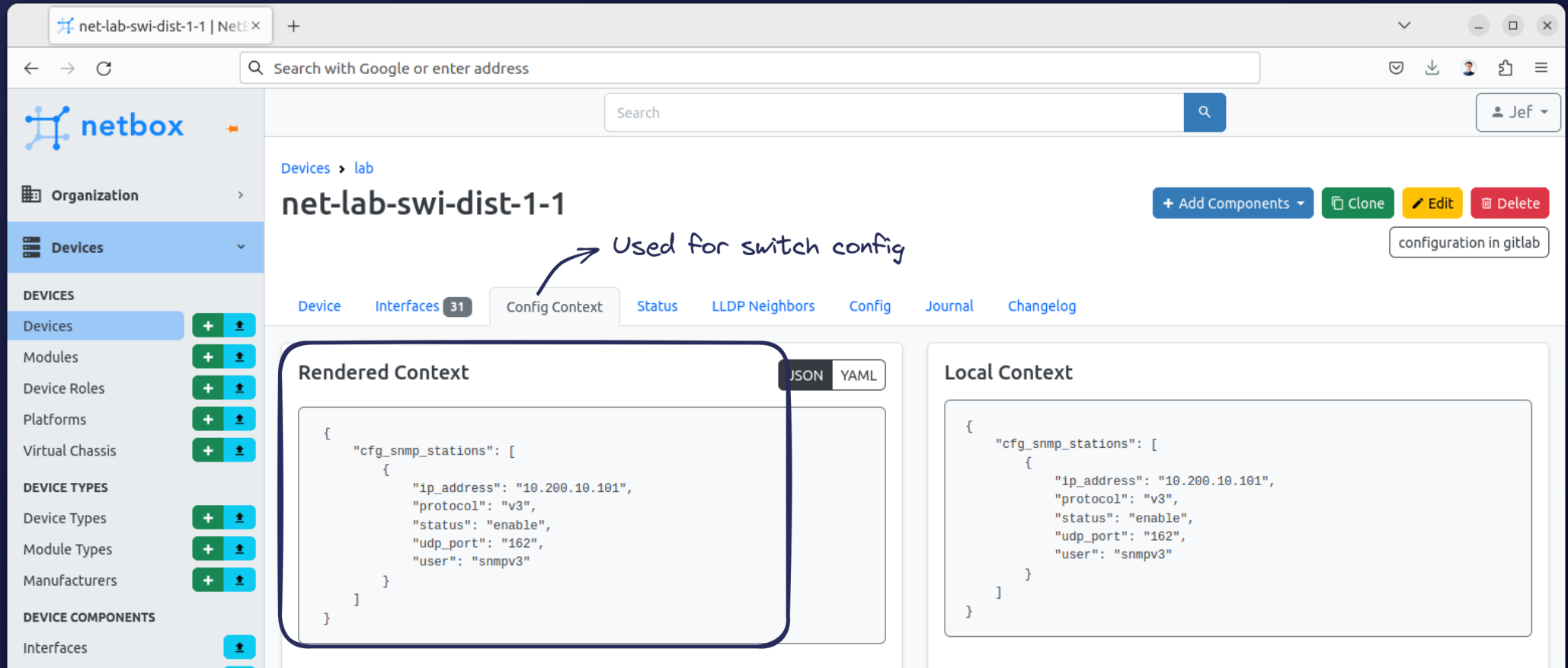
Demo

Add device to Netbox



Demo

Add device to Netbox



Demo

Ansible inventory

Netbox

netbox.yml

```
plugin: netbox.netbox.nb_inventory
validate_certs: false
config_context: true
flatten_config_context: true
compose:
  ansible_network_os: custom_fields.ansible_network_os
  device_id: id
  netbox_api: "{{ lookup('env', 'netbox_api') }}"
  netbox_user_id: '2'

group_names_raw: true
groups:
  netbox: ansible_host != ""
group_by:
  - sites

query_filters:
  - tag: zabbix
```

Ansible-inventory -i inventory/netbox.yml --list


```
{
    "_meta": {
        "hostvars": {
            "net-lab-swi-dist-1-1": {
                "ansible_host": "10.10.20.11",
                "cfg_snmp_stations": [
                    {
                        "ip_address": "10.200.10.101",
                        "protocol": "v3",
                        "status": "enable",
                        "udp_port": "162",
                        "user": "snmpv3"
                    }
                ]
            }
        }
    },
    "lab": {
        "hosts": [
            "net-lab-swi-dist-1-1"
        ]
    }
}
```

Demo

Switch configuration

switch-config.yml

```yaml
---
- name: Get >>> snmp stations
  ansible.utils.cli_parse:
    command: show snmp station
    parser:
      name: ansible.utils.textfsm
    set_fact: pre_snmp_stations

- name: Configure >>> snmp stations
  when: pre_snmp_stations != cfg_snmp_stations
  notify:
    - "Zabbix-sync : save configuration"
    - "Zabbix-sync : callback netbox journal - config diff"
  block:
    - name: Push >> no snmp stations
      ansible.netcommon.cli_command:
        command: "{{ lookup('template', 'no_snmp_stations.j2') }}"

    - name: Push >> snmp stations
      ansible.netcommon.cli_command:
        command: "{{ lookup('template', 'snmp_stations.j2') }}"

    - name: Info >> snmp configuration changes
      ansible.utils.fact_diff:
        before: "{{ pre_snmp_stations }}"
        after: "{{ cfg_snmp_stations }}"
      register: cfg_snmp_stations_diff
```

1

parse

aos_show_snmp_station.textfsm

```
Value ip_address ((?:[0-9]{1,3}\.){3}[0-9]{1,3}|(?:[a-f0-9]{1,4}:){7}[a-f0-9]{1,4})
Value udp_port (\d{1,5})
Value status (enable|disable)
Value protocol (v1|v2|v3)
Value user (.*)

Start
  ^ipAddress/port                                    status    protocol user$$
  ^[-+\+]+\s*$$ -> Station

Station
  ^${ip_address}\/${udp_port}\s+${status}\s+${protocol}\s+${user}$$ -> Record
  ^. -> Error
```

```yaml
parsed:
- ip_address: 10.200.10.11
  protocol: v3
  status: enable
  udp_port: '162'
  user: snmpv3
```

Demo

Switch configuration

switch-config.yml

```yaml
---
- name: Get >>> snmp stations
  ansible.utils.cli_parse:
    command: show snmp station
    parser:
      name: ansible.utils.textfsm
    set_fact: pre_snmp_stations

- name: Configure >>> snmp stations
  when: pre_snmp_stations != cfg_snmp_stations
  notify:
    - "Zabbix-sync : save configuration"
    - "Zabbix-sync : callback netbox journal - config diff"
  block:
    - name: Push >> no snmp stations
      ansible.netcommon.cli_command:
        command: "{{ lookup('template', 'no_snmp_stations.j2') }}"

    - name: Push >> snmp stations
      ansible.netcommon.cli_command:
        command: "{{ lookup('template', 'snmp_stations.j2') }}"

    - name: Info >> snmp configuration changes
      ansible.utils.fact_diff:
        before: "{{ pre_snmp_stations }}"
        after: "{{ cfg_snmp_stations }}"
      register: cfg_snmp_stations_diff
```

2

Source of truth

```yaml
parsed:
- ip_address: 10.200.10.101
  protocol: v3
  status: enable
  udp_port: '162'
  user: snmpv3
```

≠

Device state

```yaml
parsed:
- ip_address: 10.200.10.11
  protocol: v3
  status: enable
  udp_port: '162'
  user: snmpv3
```

Demo

Switch configuration

switch-config.yml

```yaml
---
- name: Get >>> snmp stations
  ansible.utils.cli_parse:
    command: show snmp station
    parser:
      name: ansible.utils.textfsm
    set_fact: pre_snmp_stations

- name: Configure >>> snmp stations
  when: pre_snmp_stations =≠ cfg_snmp_stations
  notify:
    - "Zabbix-sync : save configuration"
    - "Zabbix-sync : callback netbox journal - config diff"
  block:
    - name: Push >> no snmp stations
      ansible.netcommon.cli_command:
        command: "{{ lookup('template', 'no_snmp_stations.j2') }}"

    - name: Push >> snmp stations
      ansible.netcommon.cli_command:
        command: "{{ lookup('template', 'snmp_stations.j2') }}"

    - name: Info >> snmp configuration changes
      ansible.utils.fact_diff:
        before: "{{ pre_snmp_stations }}"
        after: "{{ cfg_snmp_stations }}"
      register: cfg_snmp_stations_diff
```

3

template

snmp_stations.j2

```
{% for snmp_station in cfg_snmp_stations %}
    snmp station {{ snmp_station.ip_address }}
    {{ snmp_station.udp_port }}
    {{ snmp_station.user }}
    {{ snmp_station.protocol }}
    {{ snmp_station.status }}
{% endfor %}
```

```
! SNMP:
snmp station 10.200.10.101 162 "snmpv3" v3 enable
```

Demo

Zabbix configuration

netbox.yml

1

```yaml
- name: Get >>> sites from netbox
  run_once: true
  block:
  - name: Get >>> netbox sites
    ansible.builtin.set_fact:
      netbox_sites: [
        "{{ query('netbox.netbox.nb_lookup',
        'sites',
        api_endpoint=netbox_api,
        token=netbox_token) }}"
```

2

```yaml
  - name: Add/update/remove >>> groups from netbox
    delegate_to: "{{ zabbix_api }}"
    community.zabbix.zabbix_group:
      state: "{{ netbox_state_map[item.value.status.value] }}"
      host_groups:
        - "net/{{ item.value.slug }}"
    loop: "{{ netbox_sites }}"
```

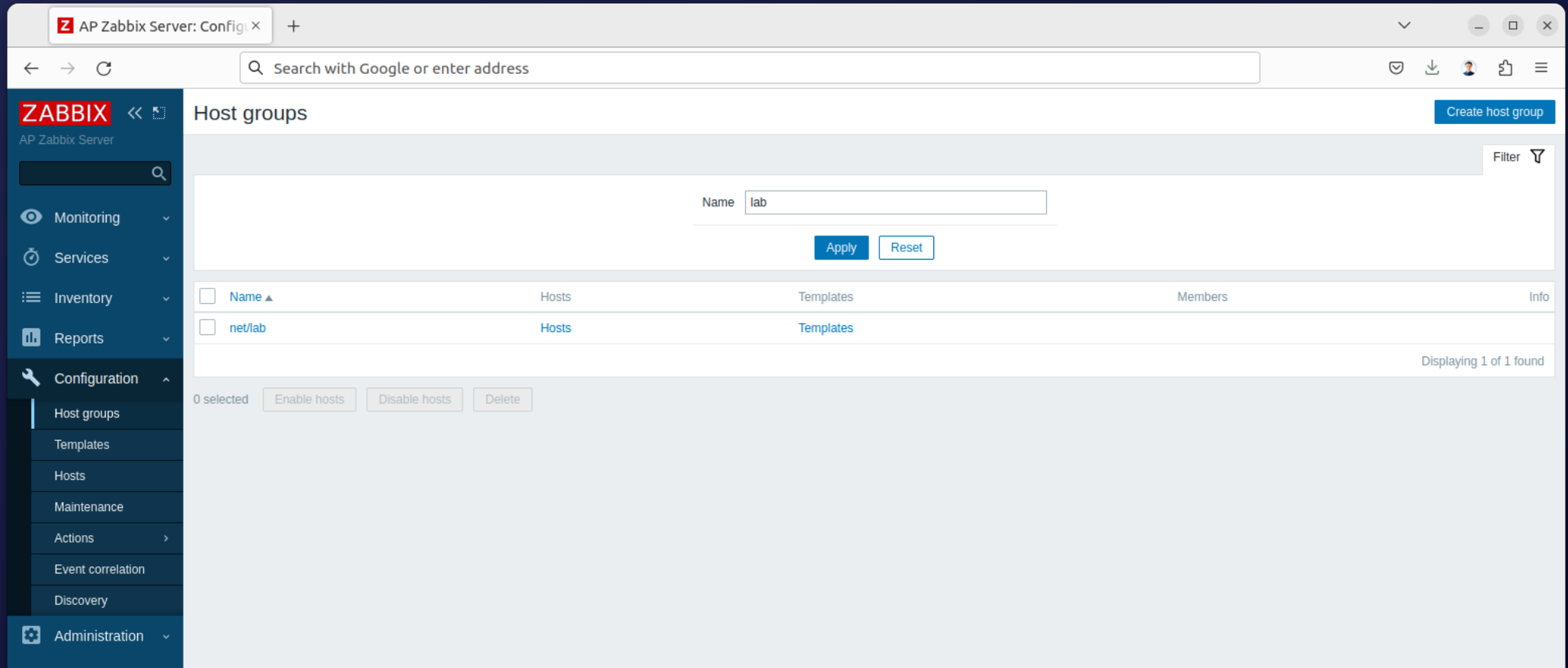
GET - HTTP 200 OK

Netbox:

```
Response body:
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "url": "api/dcim/sites/1/",
      "display": "lab",
      "name": "lab",
      "slug": "lab",
      "status": {
        "value": "active",
        "label": "Active"
...
```

Demo

Zabbix configuration



Demo

Zabbix configuration

netbox.yml

```yaml
- name: Add/update/delete >> templates
  run_once: true
  block:
    - name: Add/update >>> templates
      delegate_to: "{{ zabbix_api }}"
      community.zabbix.zabbix_template:
        template_json: "{{ lookup('file', item.template) | from_yaml | to_json }}"
        state: "{{ item.state }}"
      loop: "{{ zabbix_templates }}"
      when: item.state == "present"

    - name: Remove >>> templates
      delegate_to: "{{ zabbix_api }}"
      community.zabbix.zabbix_template:
        template_name: "{{ item.name }}"
        state: absent
      loop: "{{ zabbix_templates }}"
      when: item.state == "absent"
```

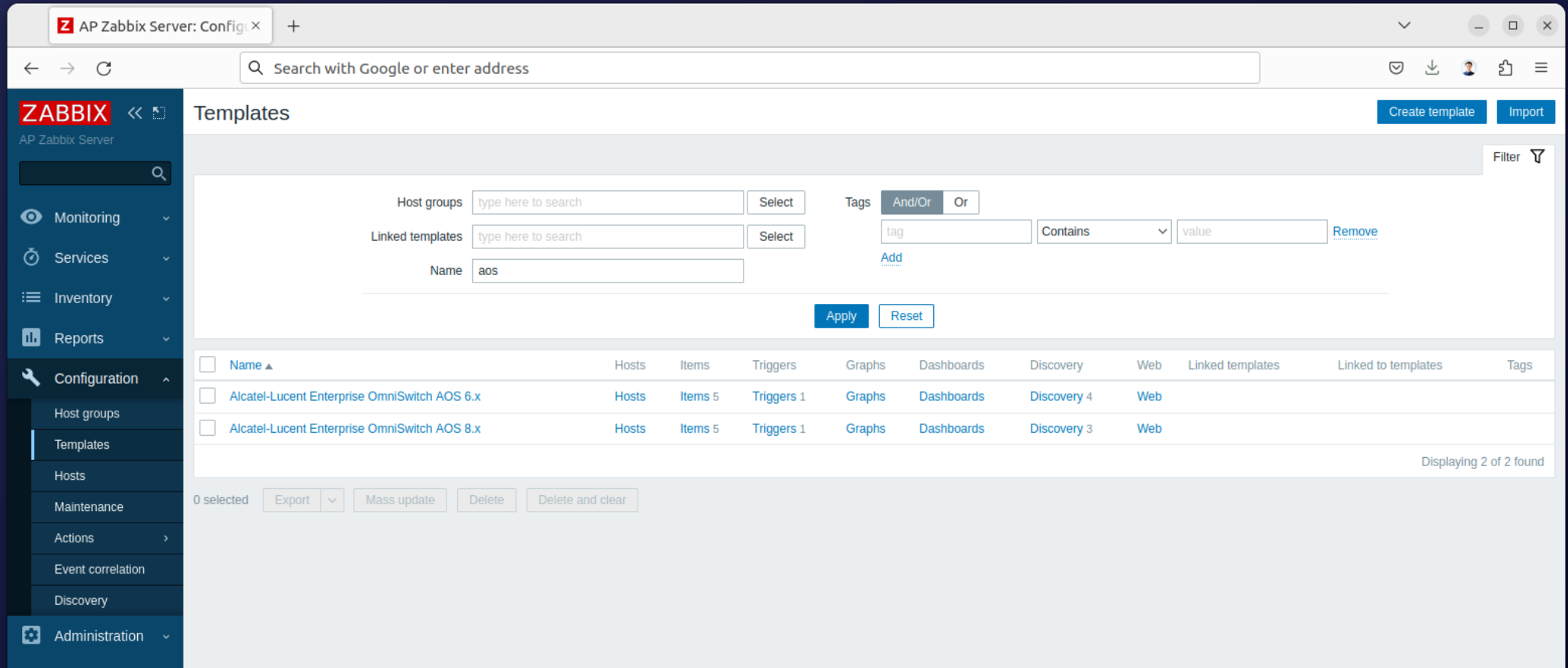
3

Ansible variable:

```yaml
zabbix_templates:
  - template: template_alcatel-lucent...
    name: Alcatel-Lucent Enterprise...
    platform: aos8
    state: present
```

Demo

Zabbix configuration



Demo

Zabbix hosts

zabbix-hosts.yml

```yaml
- name: Add/update >>> hosts to zabbix
  when: status.value in ["active", "offline", "planned", "staged"]
  notify: 'Zabbix-sync : callback netbox journal - added'
  block:
    - name: Add/update >>> hosts to zabbix
      delegate_to: "{{ zabbix_api }}"
      community.zabbix.zabbix_host:
        state: present
        host_name: "{{ inventory_hostname }}"
        visible_name: "{{ inventory_hostname }}"
        interfaces:
          - type: 2
            main: 1
            useip: 1
            ip: "{{ primary_ip4 }}"
            details:
              authpassphrase: "{{ cfg_snmp_passphrase }}"
              authprotocol: 1
              securityname: snmpv3
              privpassphrase: "{{ cfg_snmp_passphrase }}"
              privprotocol: 1
              securitylevel: 2
              version: 3
        host_groups:
          - "net/{{ sites[0] }}"
...
```

1

zabbix-hosts.yml

```yaml
...
    - name: Add/update >>> linked templates
      delegate_to: "{{ zabbix_api }}"
      community.zabbix.zabbix_host:
        host_name: "{{ inventory_hostname }}"
        link_templates:
          - Alcatel-Lucent Enterprise OmniSwitch AOS 8.x
      when: platforms[0] == "alcatel_aos"

    - name: Add/update >>> host status
      notify: 'Zabbix-sync : callback netbox journal - status changed'
      delegate_to: "{{ zabbix_api }}"
      community.zabbix.zabbix_host:
        host_name: "{{ inventory_hostname }}"
        status: "{{ zabbix_status_map[status.value] }}"
      when: status.value in ["offline", "planned", "staged"]

    - name: Delete >>> hosts
      notify: 'Zabbix-sync : callback netbox journal - deleted'
      delegate_to: "{{ zabbix_api }}"
      community.zabbix.zabbix_host:
        state: absent
        host_name: "{{ inventory_hostname }}"
      when: status.value in ["inventory", "decommissioning"]
```

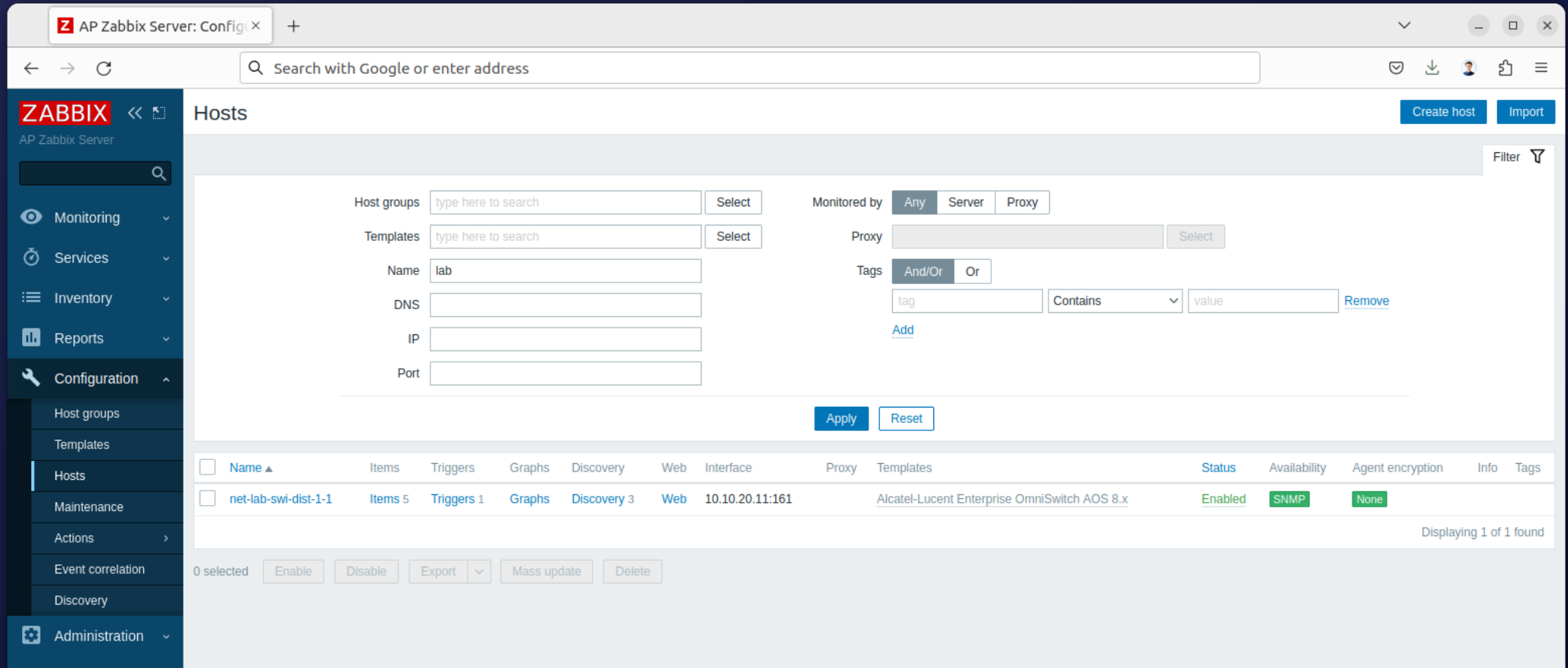
2

3

4

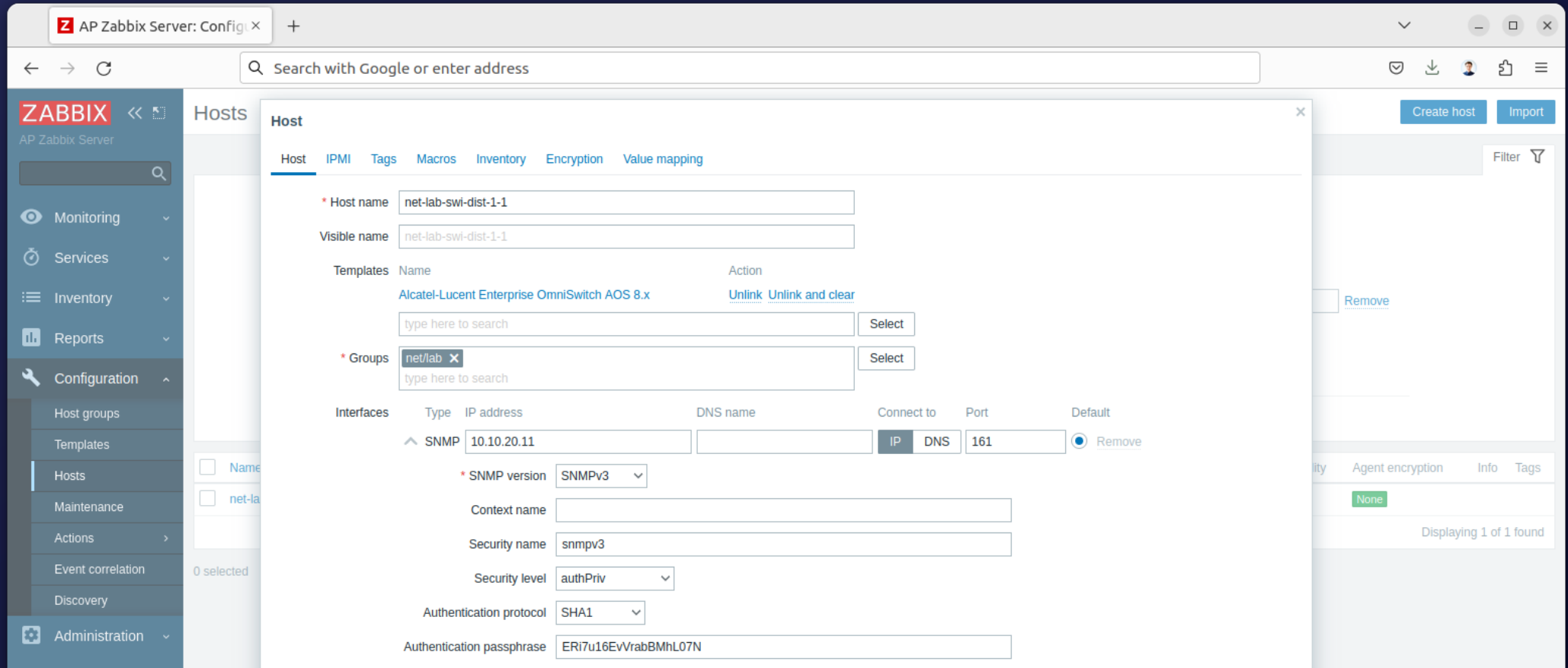
Demo

Zabbix configuration



Demo

Zabbix configuration



Demo

Ansible handlers

handlers.yml

1

```yaml
- name: 'Zabbix-sync : save configuration'
  vars:
    ansible_connection: ansible.netcommon.network_cli
    ansible_network_os: jefvantongerloo.alcatel.aos
  ansible.netcommon.cli_command:
    command: "write memory flash-synchro"
```

On task 'changed':

```yaml
notify:
  - "Zabbix-sync : save configuration"
```

2

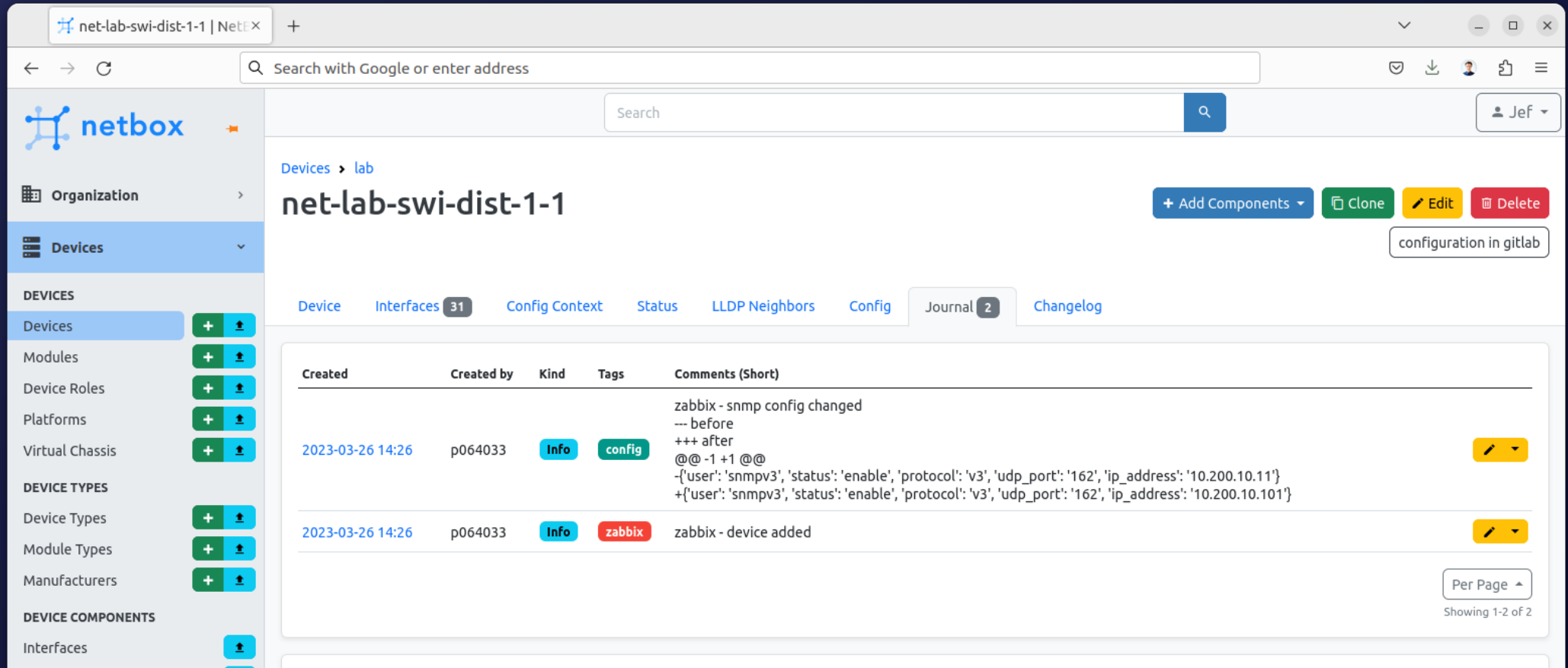
```yaml
- name: 'Zabbix-sync : callback netbox journal - added'
  ansible.builtin.uri:
    url: "{{ netbox_api }}/api/extras/journal-entries/"
    method: POST
    headers:
      Accept: application/json
      Authorization: "Token {{ netbox_token }}"
    status_code: 201
    body_format: json
    body:
      assigned_object_type: dcim.device
      assigned_object_id: '{{ device_id }}'
      created_by: '{{ netbox_user_id }}'
      kind: info
      tags:
        - name: zabbix
      comments: |
        zabbix - device added
```

On task 'changed':

```yaml
notify:
  - 'Zabbix-sync : callback netbox journal - added'
```

Demo

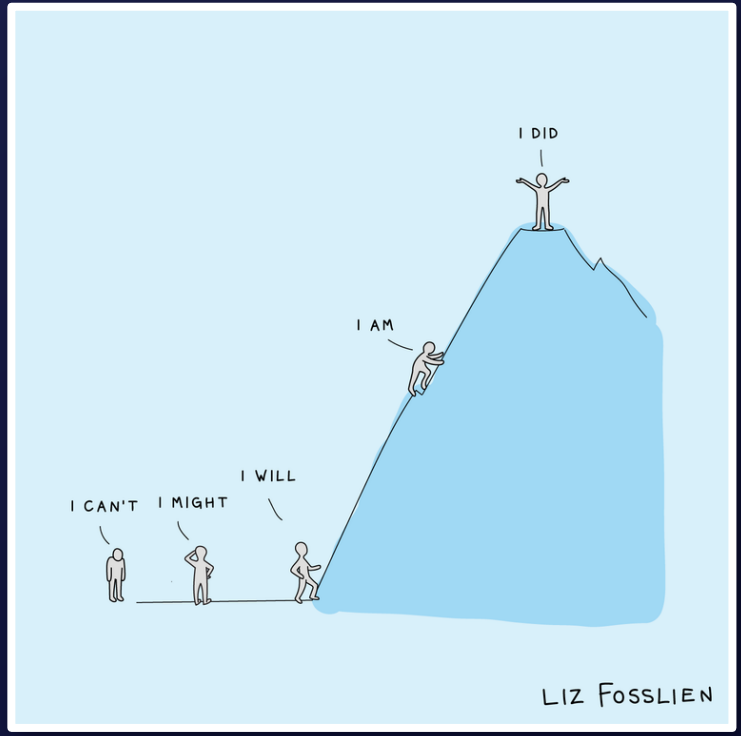
Netbox callback



What's next?

Integration

- Integrate in CI/CD pipeline services
  - Github, GitLab, Drone CI, ...
- Integrate into task manager
  - AWX, Jenkins, ...
- Sit back and relax



Liz Fosslien, https://fosslien.com

Questions?

 @jefvantongerloo

 Linkedin.com/in/jefvantongerloo

 github.com/jefvantongerloo/zabbix-automation-demo