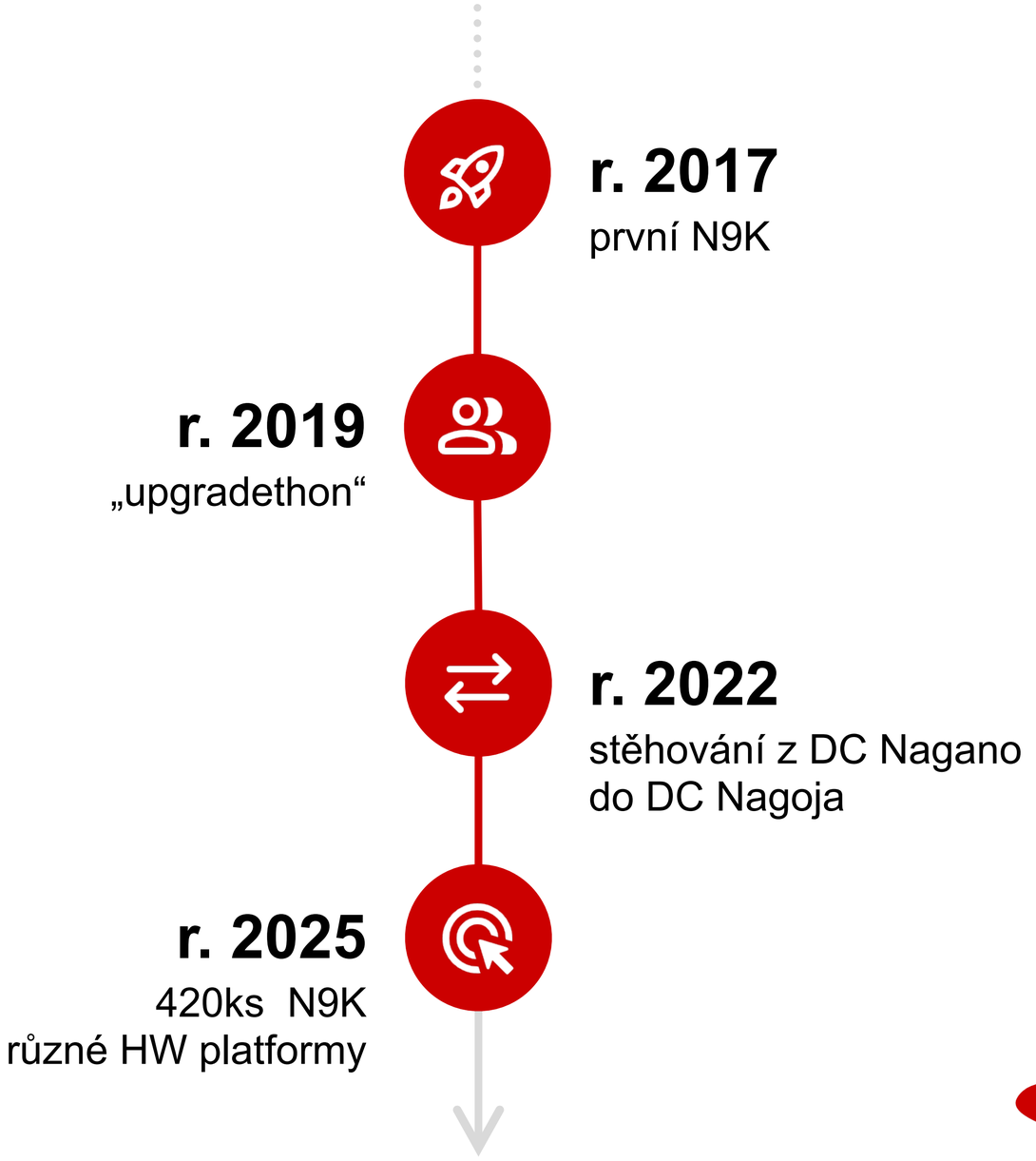
Když už to nejde „ručně“
-
automatizovaný upgrade network OS v DC pomocí Ansible

Tomáš Procházka
Network Team Leader, Seznam.cz

Jak šel s upgrady čas

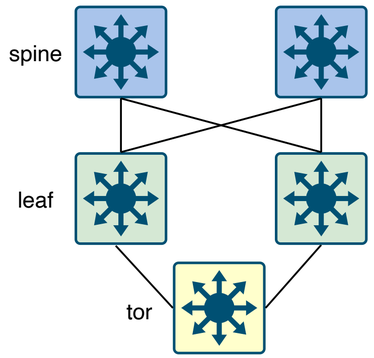
- první upgrady = stará dobrá klasika
- „upgradethon“ - manuální paralelizace ve více lidech
- stěhování mezi DC - první semiautomatizace (po jednom, pouze upgrade task)
- maintenance, obměna HW platforem, nové instalace (postupně - neceloplošně)



Výchozí stav pro automatizaci

access vrstva (TOR)

- neredundantní, největší (316ks), různé verze NXOS a EPLD, mezi upgrade
- min. downtime, paralelizace, požadavky od compute infrateamu
- = ručně to už prostě nejde!



leaf vrstva

- redundantní, bez „uživatelů“
- značně menší
- umět odstavit/vrátit zařízení z/do provozu

spine vrstva

- nezajímavý počet pro automatizaci
- novější NXOS proti leaf/TOR

Příprava je potřeba

precheck_upgrade.yml

- dostat seznam pro plánování
- roztřídit podle AZ

prepare_upgrade.yml

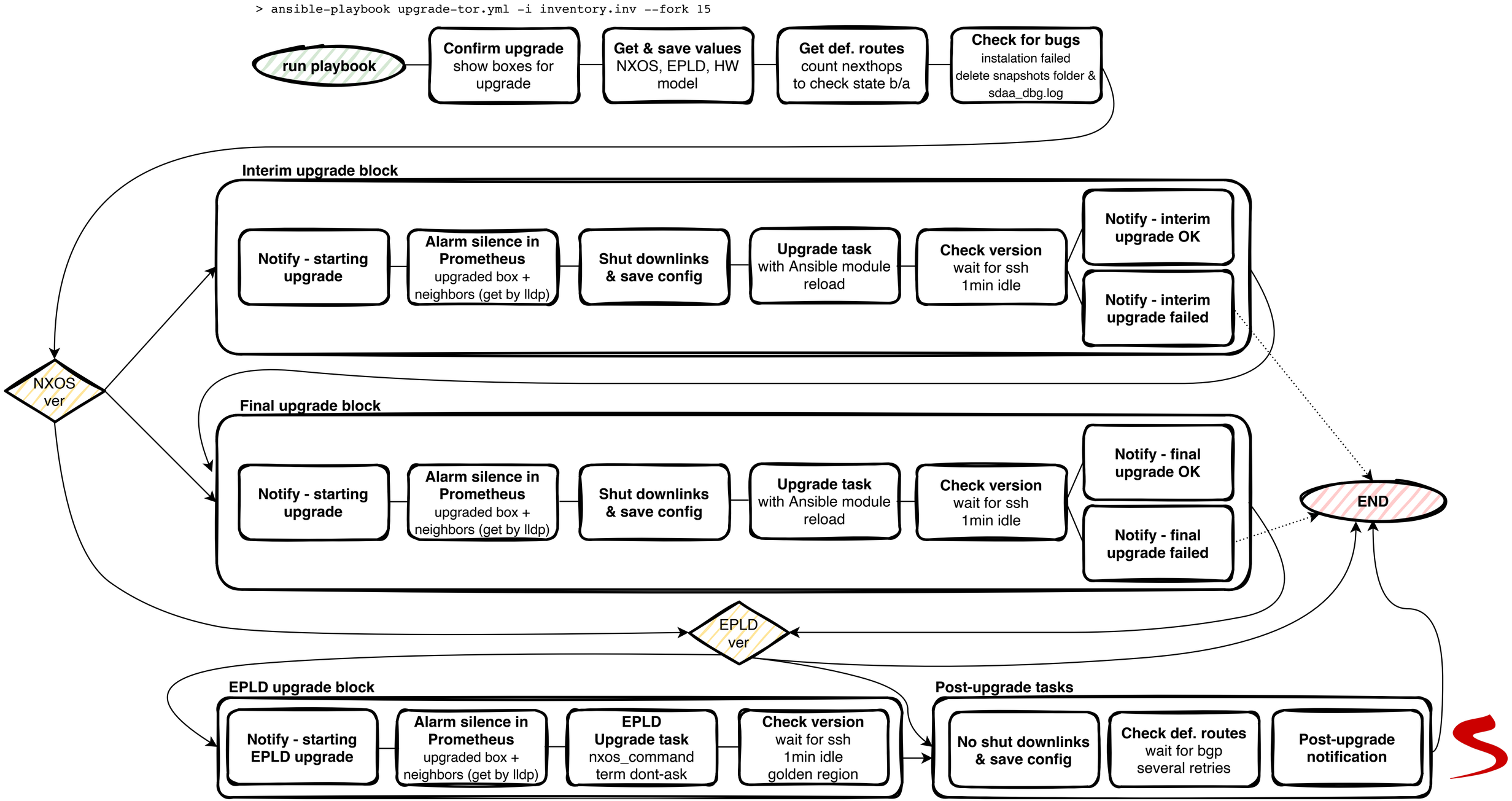
- smazat co nepotřebuji
- nahrát image NXOS, EPLD
- zkontrolovat checksum

devices_info.csv

```
HOST; AZ; NXOS; HW_MODEL; EPLD_CURRENT; EPLD_WILL_UPGRADE; UPGRADE_TIME
tor-c-1-ko.net.iszn.cz; az1; 9.3(7); N9K-C93180YC-EX; 0x4,0x15; False; 40
tor-c-10-ko.net.iszn.cz; az3; 9.3(7); N9K-C93180YC-EX; 0x4,0x15; False; 40
tor-c-11-ko.net.iszn.cz; az3; 9.3(7); N9K-C93180YC-EX; 0x4,0x15; False; 40
tor-d-11-ko.net.iszn.cz; az3; 9.3(7); N9K-C93108TC-EX; 0x2,0x10; True; 60
...
```

```
TASK [Delete .bin and .img files, excluding the current boot image]
ok: [10.255.8.99] => (item=n9000-epld.10.3.6.M.img)
ok: [10.255.8.99] => (item=nxos.9.3.7.bin)
ok: [10.255.8.99] => (item=nxos64-cs.10.2.4.M.bin)
skipping: [10.255.8.99] => (item=nxos64-cs.10.3.6.M.bin)
```

Upgrade access vrstvy


> ansible-playbook upgrade-tor.yml -i inventory.inv --fork 15
run playbook
Confirm upgrade
show boxes for upgrade
Get & save values
NXOS, EPLD, HW model
Get def. routes
count nexthops to check state b/a
Check for bugs
instalation failed
delete snapshots folder & sdaa_dbg.log
Interim upgrade block
Notify - starting upgrade
Alarm silence in Prometheus
upgraded box + neighbors (get by lldp)
Shut downlinks & save config
Upgrade task
with Ansible module reload
Check version
wait for ssh
1min idle
Notify - interim upgrade OK
Notify - interim upgrade failed
NXOS ver
Final upgrade block
Notify - starting upgrade
Alarm silence in Prometheus
upgraded box + neighbors (get by lldp)
Shut downlinks & save config
Upgrade task
with Ansible module reload
Check version
wait for ssh
1min idle
Notify - final upgrade OK
Notify - final upgrade failed
END
EPLD ver
EPLD upgrade block
Notify - starting EPLD upgrade
Alarm silence in Prometheus
upgraded box + neighbors (get by lldp)
EPLD Upgrade task
nxos_command
term dont-ask
Check version
wait for ssh
1min idle
golden region
Post-upgrade tasks
No shut downlinks & save config
Check def. routes
wait for bgp
several retries
Post-upgrade notification

Check for bugs
instalation failed
delete snapshots folder &
sdaa_dbg.log

```yaml
- name: Check for bug CSCwn58834 and CSCwj18822
  block:
    - name: Enable feature bash-shell
      cisco.nxos.nxos_config:
        lines:
          - feature bash-shell

    - name: Check the owner of .snapshots folder
      cisco.nxos.nxos_command:
        commands:
          - run bash ls -la /bootflash/ | grep '\.snapshots'
      register: _snapshot_ls_out

    - name: Delete .snapshots folder if owner is network-operator
      cisco.nxos.nxos_command:
        commands:
          - run bash rm -rf /bootflash/.snapshots
      when: ("network-operator" in _snapshot_ls_out.stdout[0])

    - name: Get flash usage
      cisco.nxos.nxos_command:
        commands:
          - show system internal flash | json
      register: _flash_usage

    - name: Set /nxos/tmp usage variable # noqa: jinja[spacing]
      ansible.builtin.set_fact:
        tmp_usage: "{{ _flash_usage.stdout[0].TABLE_flash.ROW_flash
                       | selectattr('Mounted-on', 'equalto', '/nxos/tmp')
                       | map(attribute='Use-percent') | first }}"

    - name: Delete log /nxos/tmp/sdaa_dbg.log if /nxos/tmp is full
      cisco.nxos.nxos_command:
        commands:
          - run bash rm /nxos/tmp/sdaa_dbg.log
      when: tmp_usage | default('0') | int == 100

    - name: Disable feature bash-shell
      cisco.nxos.nxos_config:
        lines:
          - no feature bash-shell
        save_when: changed
```

Upgrade task
with Ansible module
reload

```yaml
- name: Run final upgrade NXOS to {{ image_nxos_target }}
  cisco.nxos.nxos_install_os:
    system_image_file: "{{ image_nxos_target }}"

- name: Wait for device to come back up after final NXOS upgrade
  ansible.builtin.wait_for:
    port: 22
    state: started
    timeout: 500
    delay: 120
    host: '{{ inventory_hostname }}'
```

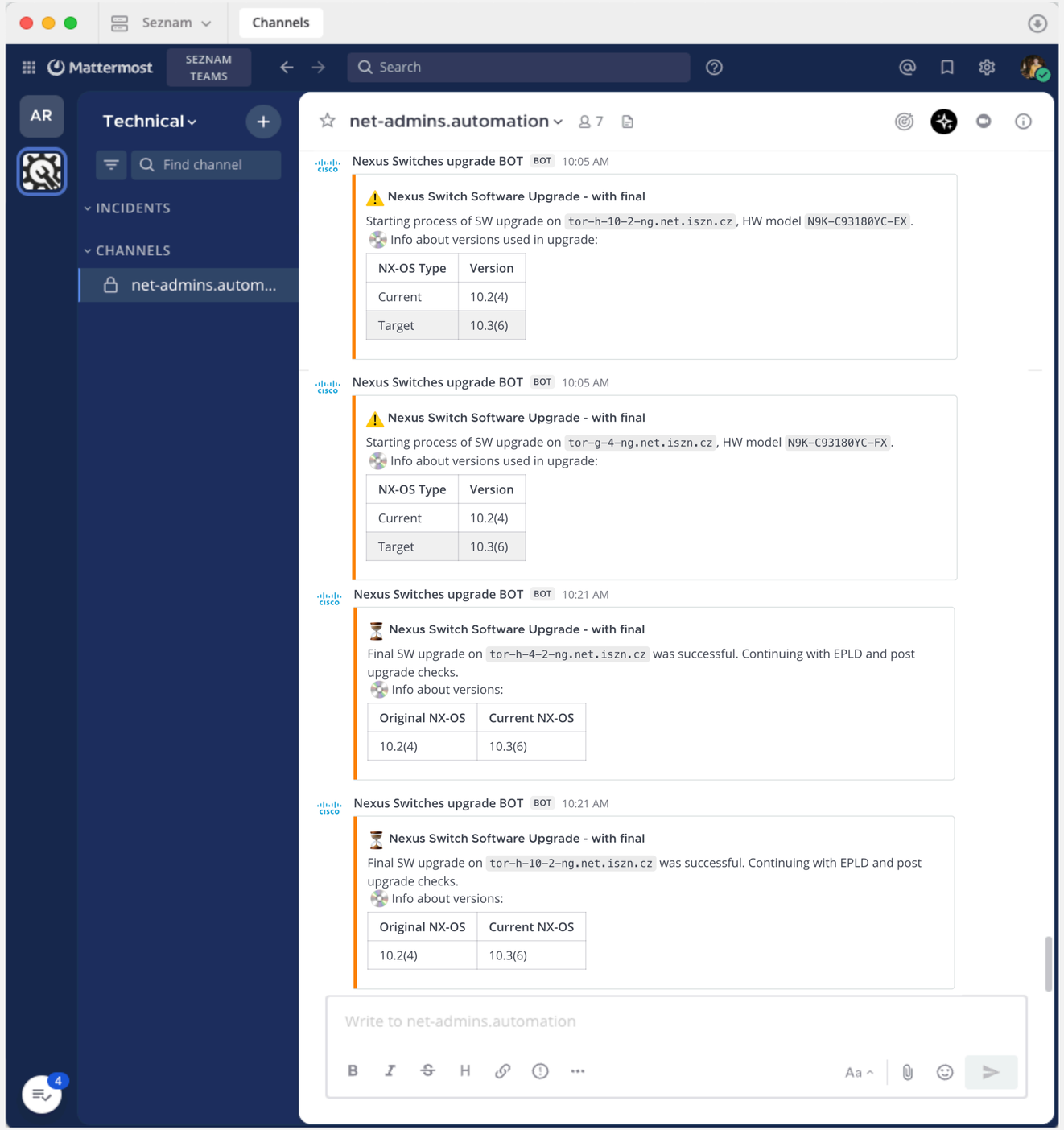
EPLD
Upgrade task
nxos_command
term dont-ask

```yaml
- name: Run EPLD upgrade for PRIMARY region to {{ image_epld_target }}
  cisco.nxos.nxos_command:
    commands:
      - terminal dont-ask
      - install epld {{ image_epld_target }} module all

- name: Wait for device to come back up after PRIMARY region EPLD upgrade
  ansible.builtin.wait_for:
    port: 22
    state: started
    timeout: 500
    delay: 120
    host: '{{ inventory_hostname }}'
```

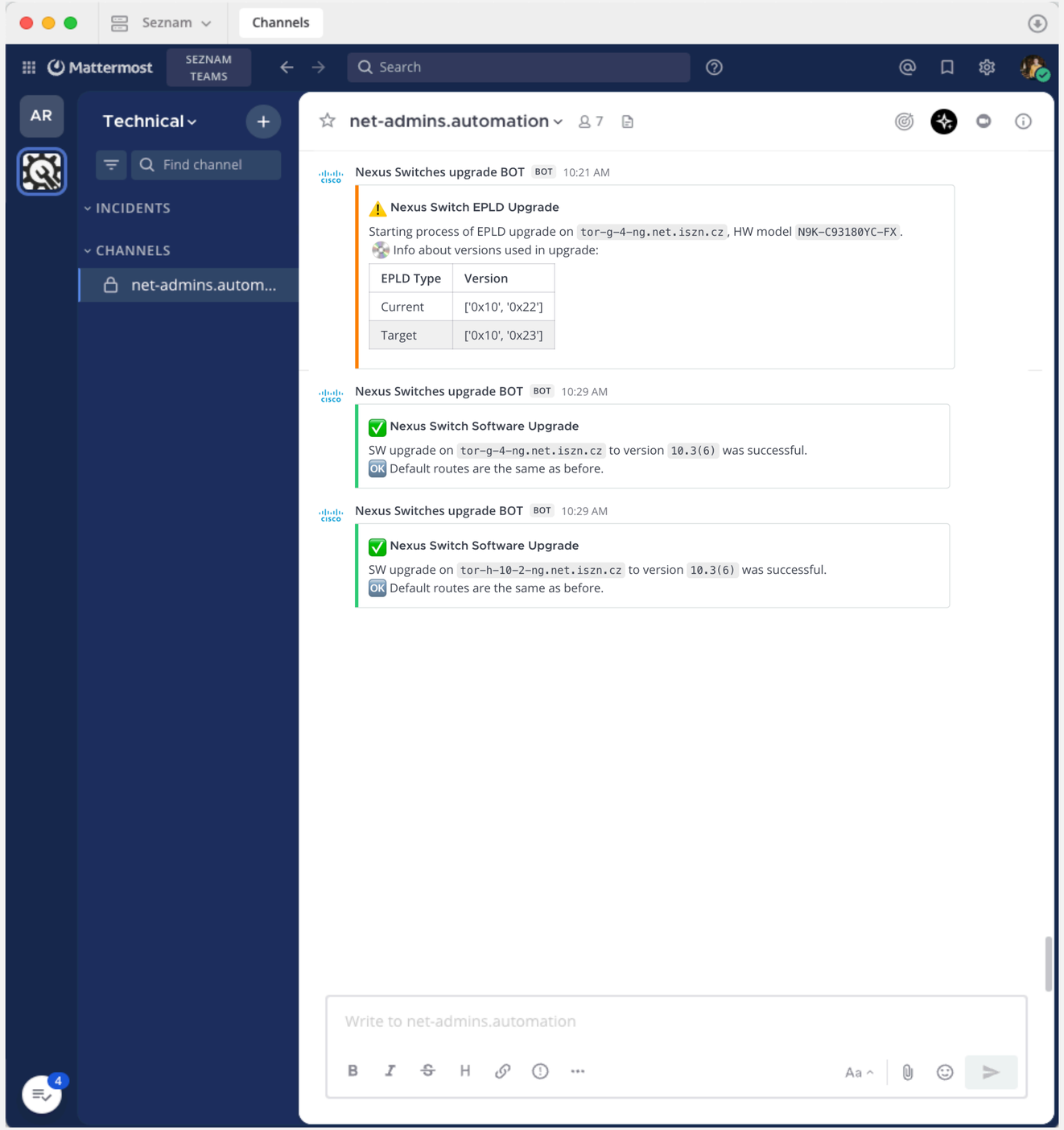
Notify - starting upgrade

```yaml
- name: Final upgrade NXOS to {{ image_nxos_target }}
  when:
    - expected_current_version not in current_version
    - expected_target_version not in current_version
  block:
    - name: Pre-upgrade tasks only when interim NXOS upgrade was not performed
      when: nxos_upgrade_interim is false
      block:
        - name: Send start notification to Mattermost - final
          ansible.builtin.uri:
            url: "https://teams.szn.cz/hooks/{{ mm_token }}"
            method: POST
            body:
              username: "Nexus Switches upgrade BOT"
              attachments:
                - color: "#FF8000"
                  title: "⚠️ Nexus Switch Software Upgrade - with final"
                  text: "Starting process of SW upgrade on `{{ inventory_hostname }}`
                        Info about versions used in upgrade:\n\n\
                        | NX-OS Type | Version |\n\
                        | --- | --- |\n\
                        | Current | {{ current_version }} |\n\
                        | Target | {{ expected_target_version }} |"
              icon_emoji: "cisco"
            body_format: json
            headers:
              Content-Type: application/json
```



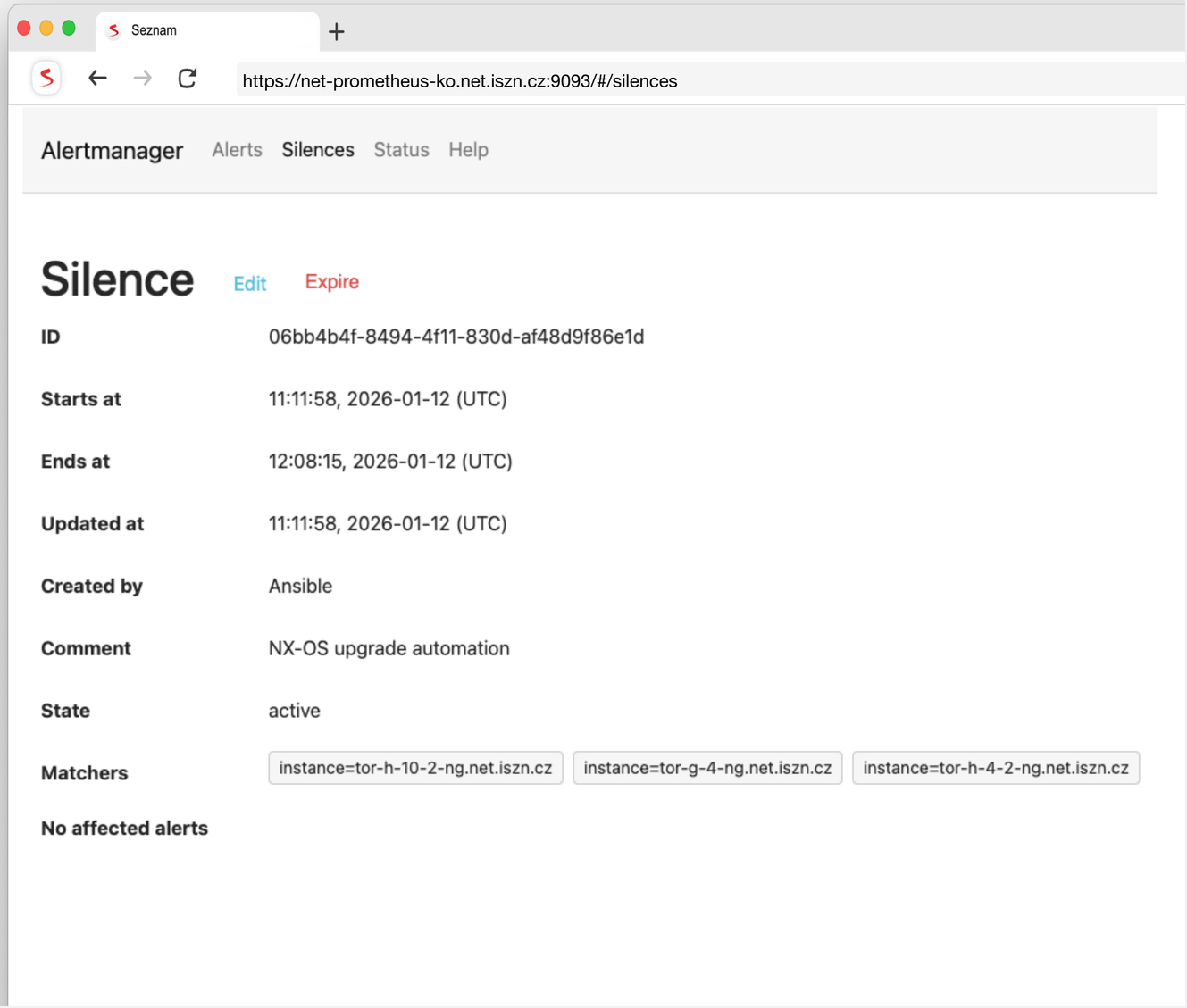
Notify - starting EPLD upgrade

```yaml
- name: EPLD upgrade needed. Starting with PRIMARY region image to {{ image_epld_target }}
  when: epld_upgrade_target is true
  block:
    - name: Send start notification to Mattermost - EPLD
      ansible.builtin.uri:
        url: "https://teams.szn.cz/hooks/{{ mm_token }}"
        method: POST
        body:
          username: "Nexus Switches upgrade BOT"
          attachments:
            - color: "#FF8000"
              title: "⚠️ Nexus Switch EPLD Upgrade"
              text: "Starting process of EPLD upgrade on `{{ inventory_hostname }}`, HW model
                    💠 Info about versions used in upgrade:\n\n\
                    | EPLD Type | Version |\n\
                    | --- | --- |\n\
                    | Current | {{ original_epld_version_output }} |\n\
                    | Target | {{ epld_result_expected }} |"
          icon_emoji: "cisco"
        body_format: json
        headers:
          Content-Type: application/json
```

Alarm silence in Prometheus
upgraded box + neighbors (get by lldp)

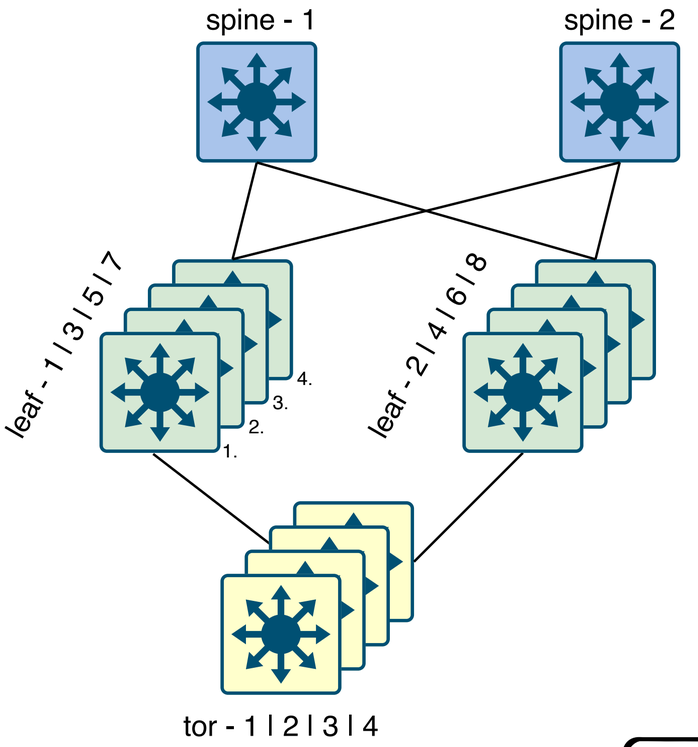
```yaml
- name: Silence alarms for device
  ansible.builtin.uri:
    url: "{{ silencer_url }}"
    method: POST
    body:
      matchers:
        - name: "instance"
          value: "{{ inventory_hostname }}"
          isRegex: false
      startsAt: "{{ alarm_start_time }}"
      endsAt: "{{ alarm_end_time }}"
      createdBy: "Ansible"
      comment: "NX-OS upgrade automation"
      id: null
    body_format: json
    headers:
      Content-Type: application/json
```



Upgrade leaf boxů

Redundantní vrstva

- 19 párů ve třech datacentrech
- mírně odlišné vlastnosti
  (např. rychlost nehraje roli)
- pokud se vše řeší „samo“



Odlišné požadavky

- kontrola stavů před a po upgradu
  - interfaces
  - BGP sessions
- odstavení přes maintenance mode
- před dalším upgradem čekej 5min

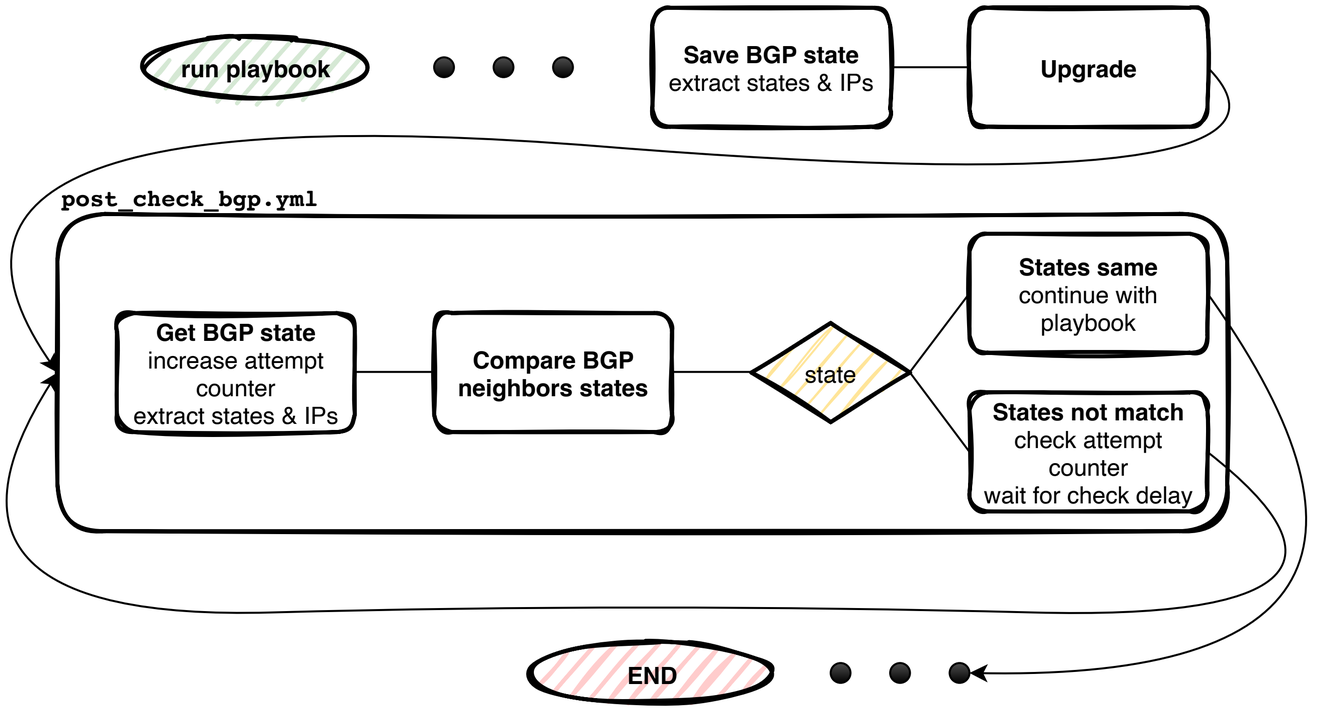
Maintanance mode tasks

```
- name: Configure custom maintenance profile
  cisco.nxos.command:
    commands:
      - no configure maintenance profile maintenance-mode
      - configure maintenance profile maintenance-mode
      - router bgp {{ bgp_as_num }}
      - isolate include-local
      - sleep instance 0 30
      - system interface shutdown

- name: Put switch into maintenance
  cisco.nxos.nxos_gir:
    system_mode_maintenance_dont_generate_profile: true
```

Kontrola stavu před a po upgradu

- u TORů (access vrstva) - kontrola default route
  - počet nexthopů a konkrétní nexthop IP
- u leafů navíc kontrola stavu interfaců a BGP sessions



Post upgrade tasks

```yaml
- name: Post-upgrade tasks
  when: >
      (epld_upgrade_target is true) or
      (nxos_upgrade_interim is true) or
      (nxos_upgrade_final is true)
  block:
    - name: Enable interfaces
      cisco.nxos.nxos_config:
        lines:
          - no system interface shutdown
        save_when: changed

    - name: Wait 1m for interfaces to come up
      ansible.builtin.wait_for:
        timeout: 60

    - name: Wait until interface status will match
      ansible.builtin.include_tasks: post_check_interfaces.yml
      vars:
        check_retries: 11
        check_delay: 30

    - name: Wait until BGP state will match
      ansible.builtin.include_tasks: post_check_bgp.yml
      vars:
        check_retries: 11
        check_delay: 30

    - name: Wait until default routes will match
      ansible.builtin.include_tasks: post_check_default_route.yml
      vars:
        check_retries: 11
        check_delay: 30
```

Get BGP state
extract states & IPs

```yaml
- name: Get BGP neighbors status post-upgrade
  cisco.nxos.nxos_command:
    commands:
      - "show bgp all summary | json"
  register: _bgp_post

- name: Extract post-upgrade IPv4 BGP neighbors states
  ansible.builtin.set_fact:
    bgp_states_post: "{{ bgp_states_post | default({}) | combine({item.neighborid: item.state}) }}"
  when:
    - item.state == "Established"
    - item.neighborid != "10.245.3.70"
  with_items: "{{
              (_bgp_post.stdout[0].TABLE_vrf.ROW_vrf.TABLE_af.ROW_af[0].TABLE_saf.ROW_saf.TABLE_neighbor.ROW_neighbor | default([])
              if (_bgp_post.stdout[0].TABLE_vrf.ROW_vrf.TABLE_af.ROW_af[0].TABLE_saf.ROW_saf.TABLE_neighbor.ROW_neighbor | default([])
                | type_debug == 'list')
              else [_bgp_post.stdout[0].TABLE_vrf.ROW_vrf.TABLE_af.ROW_af[0].TABLE_saf.ROW_saf.TABLE_neighbor.ROW_neighbor] | default([]))
              }}"
```

Jak jsme dopadli?

21dní
v cca 1,5 měsíci

4-12 TORů
souběžně

T_{min} 11m
T_{max} 45m

228ks

dopolední a odpolední slot na upgrade | úmrtnost: 2 z 228ks

Co si z toho odnést

- access - musíme ještě zrychlit - celé AZ/DC současně?
- nebo sériový upgrade bez účasti člověka?
- udržovat verze pro přímý upgrade

- vhodné u velkých a jednoduchých topologií
- použitelné pro různé NOS
- není potřeba vendor specific orchestrátorů
- flexibilní


SEZNAM.CZ