Ansible para Network – criando scripts de automação para configuração em massa de switchs.

Nas últimas semanas tenho dedicado parte do meu tempo para me aprofundar no estudo e aplicação de scripts de automação através de Ansible. Pensando em minimizar o tempo de trabalho com tarefas repetitivas do dia-a-dia e coleta de logs de vários equipamentos de uma vez o resultado é maior produtividade com tarefas que realmente demandam tempo do admin da rede.

O Uso de playbooks e modulos, providência uma maior flexibilidade e segurança na hora de aplicar as configurações, tendo como requisito apenas o equipamento final estar com IP de gerência e SSH habilitado o ansible é capaz de acessa-lo e aplicar as configurações pré-definidas no script.

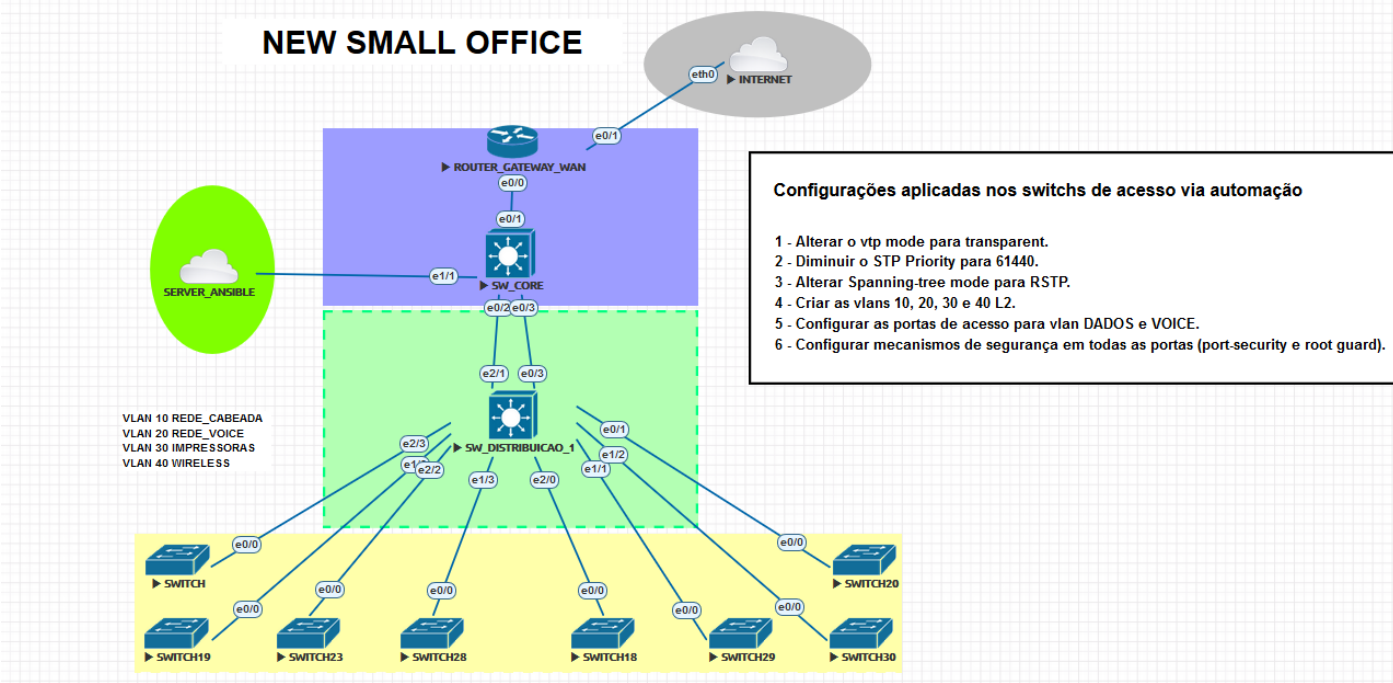
Hoje irei demonstrar a utilização do ansible para automatizar a configuração de novos switchs em um pequeno escritório.

Small Office:

> Objetivo: Retirar um backup das configurações pré-janela, aplicar as configurações abaixo em todos os switchs de acesso e salvar um novo backup pós-script.

1 - Alterar o vtp mode para transparent.
2 - Diminuir o STP Priority para 61440.
3 - Alterar Spanning-tree mode para RSTP.
4 - Criar as vlans 10, 20, 30 e 40 L2.
5 - Configurar as portas de acesso para vlan DADOS e VOICE.
6 - Configurar mecanismos de segurança em todas as portas (port-security e root guard).

Topologia:



ANSIBLE

Arquivo hosts:

```
 [SWITCH_ACESSO]
10.10.10.11
10.10.10.12
10.10.10.13
10.10.10.14
10.10.10.15
10.10.10.16
10.10.10.17
10.10.10.18
```

Playbook:

```
---

- hosts: SWITCH_ACESSO
  gather_facts: true
  connection: local

# Variaveis de autenticação Usuario e senha
```

```
  vars_prompt:
  - name: "mgmt_username"
    prompt: "Username"
    private: no
  - name: "mgmt_password"
    prompt: "Password"

  tasks:

# Modulo de autenticação
   - name: SYS | Define provider
     set_fact:
       provider:
        host: "{{ inventory_hostname }}"
        username: "{{ mgmt_username }}"
        password: "{{ mgmt_password }}"
        port: 22

# Comando show run no alvo
   - name: backup pré-janela
     ios_command:
       provider: "{{ provider }}"
       commands:
        - show run
     register: backup

# Salvando saída show run
   - name: save backup
     copy:
      content: "{{ backup.stdout[0] }}"
      dest: "/etc/ansible/backup/Pre_janela/BACKUP_{{ inventory_hostname }}.txt"

# renomeando arquivo backup com a data
   - name: copy and rename file
     shell:
       cmd: cp /etc/ansible/backup/Pre_janela/BACKUP_{{ inventory_hostname }}.txt
 /etc/ansible/backup/Pre_janela/bkp_{{ inventory_hostname }}_$(date +"%d-%m-%Y-
%H-%M").txt

# removendo arquivo backup antigo
   - name: remove old file
     shell:
       cmd: rm /etc/ansible/backup/Pre_janela/BACKUP_{{ inventory_hostname }}.txt

# Configurações
```

```
  - name: Deploy config
    ios_config:
     provider: "{{ provider }}"
     lines:
     - vtp mode transparent
     - spanning-tree vlan 1,10,20,30,40 priority 61440
     - spanning-tree mode rapid-pvst
     - vlan 10
     - vlan 20
     - vlan 30
     - vlan 40
     - interface range ethernet 0/1 - 3
     - switchport mode access
     - switchport access vlan 10
     - switchport voice vlan 20
     - switchport port-security
     - switchport port-security maximum 2
     - do wr


 # Gerando arquivo de configuração pós mudanças
  - name: arquivo de configuração pós janela
    ios_command:
      provider: "{{ provider }}"
      commands:
       - show run
       - show vlan brief
       - show ip int b
    register: backup2

  # Salvando nova config
  - name: save new config
    copy:
     content: "{{ backup.stdout[0] }}"
     dest: "/etc/ansible/backup/Pos_janela/BACKUP_NEW_CONFIG_{{ inventory_hostna
me }}.txt"

  # renomeando arquivo backup com a data
  - name: copy and rename file
    shell:
      cmd: cp /etc/ansible/backup/Pos_janela/BACKUP_NEW_CONFIG_{{ inventory_host
name }}.txt /etc/ansible/backup/Pos_janela/BACKUP_NEW_CONFIG_{{ inventory_hostnam
e }}_$(date +"%d-%m-%Y-%H-%M").txt

  # removendo arquivo backup antigo
```

```
   - name: remove old file
     shell:
       cmd: rm /etc/ansible/backup/Pos_janela/BACKUP_NEW_CONFIG_{{ inventory_host
name }}.txt
```

Configuração dos switch antes da execução do script:

```
Building configuration...

Current configuration : 1150 bytes
!
! Last configuration change at 17:20:53 EET Fri Mar 27 2020
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service compress-config
!
hostname 172.26.6.11
!
boot-start-marker
boot-end-marker
!
!
enable secret 4 7PyC4WCHwF0BBk8PDTJNAXP5JLEimHFZiuydgd0xrDM
!
username local privilege 15 password 0 local
no aaa new-model
clock timezone EET 2 0
!
ip cef
!
!
ip domain-name ansible
no ipv6 cef
ipv6 multicast rpf use-bgp
!
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
```

```
!
vlan internal allocation policy ascending
!
ip ssh version 2
!
!
!
!
!
!
!
!
!
interface Ethernet0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
 duplex auto
!
interface Ethernet0/1
 duplex auto
!
interface Ethernet0/2
 duplex auto
!
interface Ethernet0/3
 duplex auto
!
interface Vlan172
 ip address 172.26.6.11 255.255.255.0
!
ip default-gateway 172.26.6.1
!
no ip http server
!
ip route 0.0.0.0 0.0.0.0 172.26.6.1
!
!
!
!
control-plane
!
!
line con 0
```

```
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login local
 transport input ssh
 transport output ssh
!
end
```

Executando o script e inserindo o usuario e senha.

```
deploy_switchs.yml
root@eve-ng:/etc/ansible/tasks_ansible# ansible-playbook -vvv deploy_switchs.yml
ansible-playbook 2.8.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible-playbook
  python version = 2.7.12 (default, Dec  4 2017, 14:50:18) [GCC 5.4.0 20160609]
Using /etc/ansible/ansible.cfg as config file
host_list declined parsing /etc/ansible/hosts as it did not pass it's verify_file() method
script declined parsing /etc/ansible/hosts as it did not pass it's verify_file() method
auto declined parsing /etc/ansible/hosts as it did not pass it's verify_file() method
Parsed /etc/ansible/hosts inventory source with ini plugin

PLAYBOOK: deploy_switchs.yml ***************************************************************************************************
1 plays in deploy_switchs.yml
Username:
```

Script executado com sucesso:

```
PLAY RECAP *********************************************************************************************************************
10.10.10.11                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.12                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.13                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.14                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.15                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.16                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.17                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.10.10.18                : ok=11   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Configurações aplicadas:

```
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service compress-config
!
hostname SW_10.10.10.11
!
boot-start-marker
boot-end-marker
!
!
enable secret 4 7PyC4WCHwF0BBk8PDTJNAXP5JLEimHFZiuydgd0xrDM
!
username local privilege 15 password 0 local
no aaa new-model
clock timezone EET 2 0
vtp mode transparent
!
ip cef
!
!
ip domain-name ansible
no ipv6 cef
ipv6 multicast rpf use-bgp
!
!
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
spanning-tree vlan 1,10,20,30,40 priority 61440
!
!
!
!
```

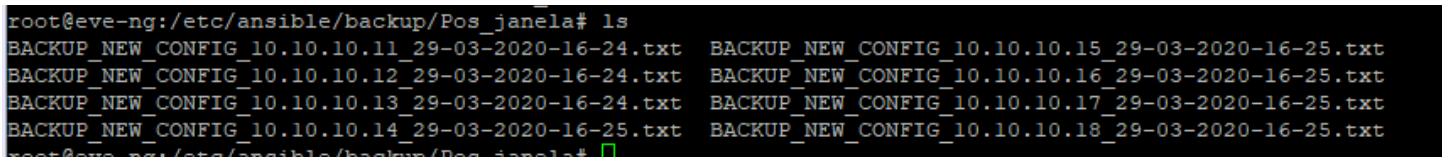
```
vlan internal allocation policy ascending
!
vlan 10,20,30,40
!
vlan 172
 name GERENCIA
!
ip ssh version 2
!
!
!
!
!
!
!
!
!
interface Ethernet0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
 duplex auto
!
interface Ethernet0/1
 switchport access vlan 10
 switchport mode access
 switchport voice vlan 20
 switchport port-security maximum 2
 switchport port-security
 duplex auto
!
interface Ethernet0/2
 switchport access vlan 10
 switchport mode access
 switchport voice vlan 20
 switchport port-security maximum 2
 switchport port-security
 duplex auto
!
interface Ethernet0/3
 switchport access vlan 10
 switchport mode access
 switchport voice vlan 20
 switchport port-security maximum 2
 switchport port-security
```

Arquivos de backup pré e pós script

Backup pré-janela

```
root@eve-ng:/etc/ansible/backup/Pre_janela# ls
bkp_10.10.10.11_29-03-2020-16-24.txt  bkp_10.10.10.14_29-03-2020-16-24.txt  bkp_10.10.10.17_29-03-2020-16-24.txt
bkp_10.10.10.12_29-03-2020-16-24.txt  bkp_10.10.10.15_29-03-2020-16-24.txt  bkp_10.10.10.18_29-03-2020-16-24.txt
bkp_10.10.10.13_29-03-2020-16-24.txt  bkp_10.10.10.16_29-03-2020-16-24.txt
```

Backup pós-janela



A Automação de scripts de configuração vai muito além de apenas configuração de devices em massa, o ansible é capaz de gerenciar arquivos JSON, XML interagir com APIs REST e muito mais.

Estarei disponibilizando todo o conteúdo de demonstração no meu github, acessa lá:

Espero que tenham gostado.

Allyson Galdino

IN https://www.linkedin.com/in/allyson-galdino/