Network Automation: El bueno, el feo y el malo

Ansible para ingenieros de red

Fernando García Fernández - 25/10/2018

Una pequeña bio

- Uno de los viejos del lugar
- 20 años programador
- 18 años ingeniero de redes
- Apple, Digital Domain (uno de los primeros ISPs), Tecnocom, BICS, Tuenti (Aka Telefonica Digital España)

La automatización que soñamos

- Quiero la ruta 192.0.2.0/24 en este cliente
  - El sistema configura las VLANs
  - El sistema configura los accesos ADSL/FTTH
  - El sistema configura el BGP
  - Todo de forma automágica

BUENO, PUES ESTO NO (AL MENOS DE MOMENTO)

La automatización que soñamos

- Un sistema que magicamente supervise la red
  - Diga donde falla
  - Lo solucione o palíe el problema

BUENO, PUES ESTO TAMPOCO (AL MENOS DE MOMENTO)

REAL AUTOMATIZACION (de redes)

- Configurar creando programas (puede haber más cosas, pero otro día)
- pero los routers también se programan
- Con un repositorio de la config de todos los routers
- Osea, un backup de todas las configuraciones
- No. Defines cosas de forma mas sencilla
  - Sin la complejidad del lenguaje de un router
  - Con configuraciones comunes a todos los routers
- Pues no lo entiendo
- Espera, que te pongo un ejemplo

EJEMPLOS (usando pseudo lenguaje)

Cambiar el password de toooodos los routers

```
- name: Change Password for User admin
  ios_user:
    name: admin
    configured_password: "{{ new_password }}"
    update_password: always
    state: present
```

Configurar las VLANs de un [illegible]

```
vlans:
  [illegible]:
    name MGMT
  112:
    name DB
```

Configurar una interfaz (Cisco? Juniper?)

```
port-channel6:
  description: "FWINT-2[illegible].Bo[illegible]1
  admin_state: [illegible]p
  swmode: t[illegible]
  st[illegible]present
  [illegible]mbers:
    - Eth1/26
  vpc: 26
  vlans:
    - 121
    - 124
    - 125
  native: 1
```

ESTO ES EL LENGUAJE REAL (YAML)

tuenti

EL BUENO

LAS VENTAJAS

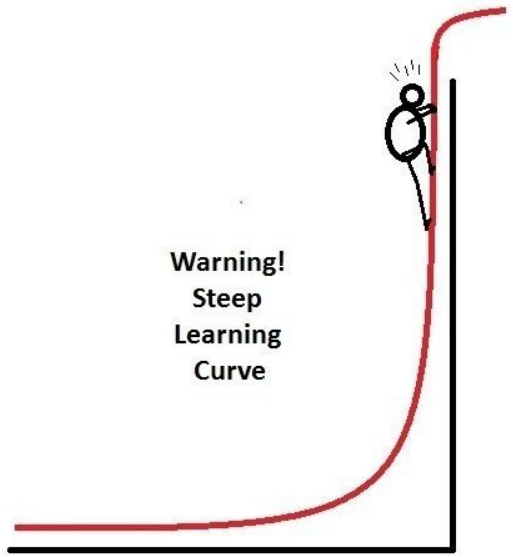
- Configuración mas sencilla
  - Nos centramos en lo que queremos, no en el lenguaje del router
- Idempotente
  - Al pasarlo, siempre se obtiene la misma configuración (en teoria)
- Evitamos errores de tecleado, copy/paste
  - Anti Finger Defined Networks
- Permite Code Reviews de forma sencilla
  - Es IAAC (Infrastructure As A Code), infraestructura como código
- Open Source: Gratuito

tuenti

EL FEO

INCONVENIENTES

- Curva de aprendizaje empinada
- Cambio de mentalidad
- Obliga a sentarse y pensar.
  - No puede ser "aquí te pillo aquí te mato"
  - ¿Pero eso es un inconveniente?
- Lento en pasar configuraciones
  - Puede ser incluso horas en un DC complejo
- Aprender nuevo lenguaje, nuevos procedimientos



tuenti

EL MALO

EN TODAS LAS HERRAMIENTAS
Y ANSIBLE EN ESPECIAL

- I
- N
- N
- N
- C

Ios

- ios_banner - Manage multiline banners on Cisco IOS devices
- ios_command - Run commands on remote devices running Cisco IOS
- ios_config - Manage Cisco IOS configuration sections
- ios_facts - Collect facts from remote devices running Cisco IOS
- ios_interface - Manage Interface on Cisco IOS network devices
- ios_l2_interface - Manage Layer-2 interface on Cisco IOS devices.
- ios_l3_interface - Manage L3 interfaces on Cisco IOS network devices.
- ios_linkagg - Manage link aggregation groups on Cisco IOS network devices
- ios_lldp - Manage LLDP configuration on Cisco IOS network devices.
- ios_logging - Manage logging on network devices
- ios_ping - Tests reachability using ping from Cisco IOS network devices
- ios_static_route - Manage static IP routes on Cisco IOS network devices
- ios_system - Manage the system attributes on Cisco IOS devices
- ios_user - Manage the aggregate of local users on Cisco IOS device
- ios_vlan - Manage VLANs on IOS network devices
- ios_vrf - Manage the collection of VRF definitions on Cisco IOS devices

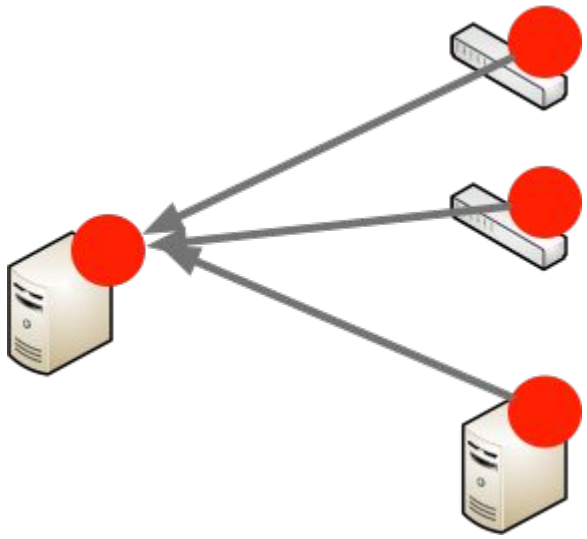
tuenti

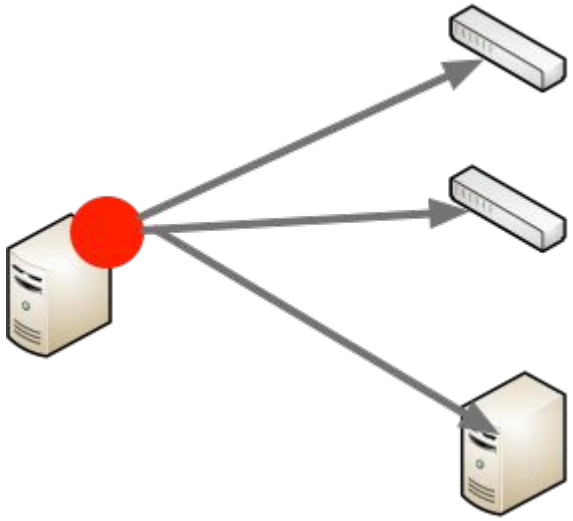
Un poco de base

Varias herramientas, fondo común

- Todas usan mismo paradigma
  - Configurar desde punto central
  - Por eso una red OOB
- Todas surgieron primero para sistemas
  - La opción de red más en pañales

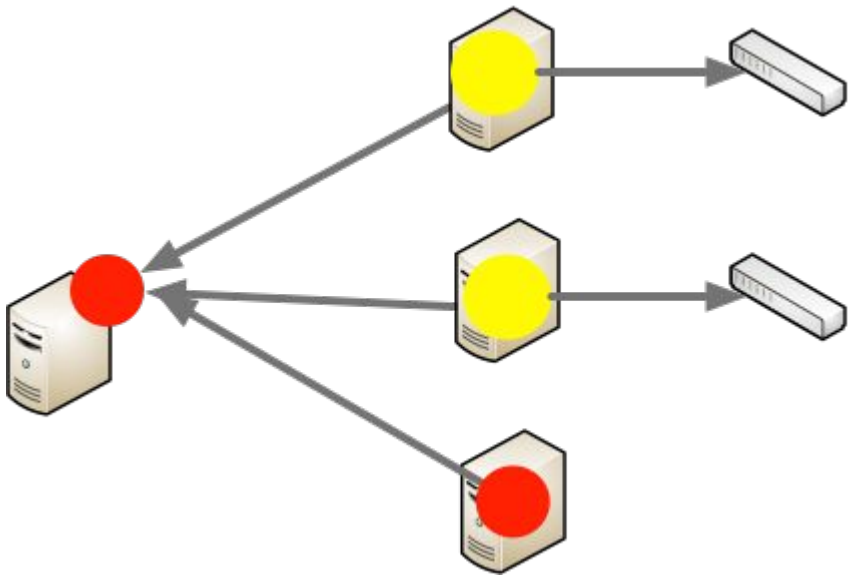
Agent vs Agentless





Las herramientas

- Puppet
  - AGENT
  - Compleja
  - Potente
- Ansible
  - AGENTLESS
  - Bastante sencilla
  - Muchos modulos de red
- SALT/CHEFF
  - AGENT
- STACKSTORM
- AGENT y AGENTLESS
- Las heramientas AGENT pueden funcionar AGENTLESS usando minions



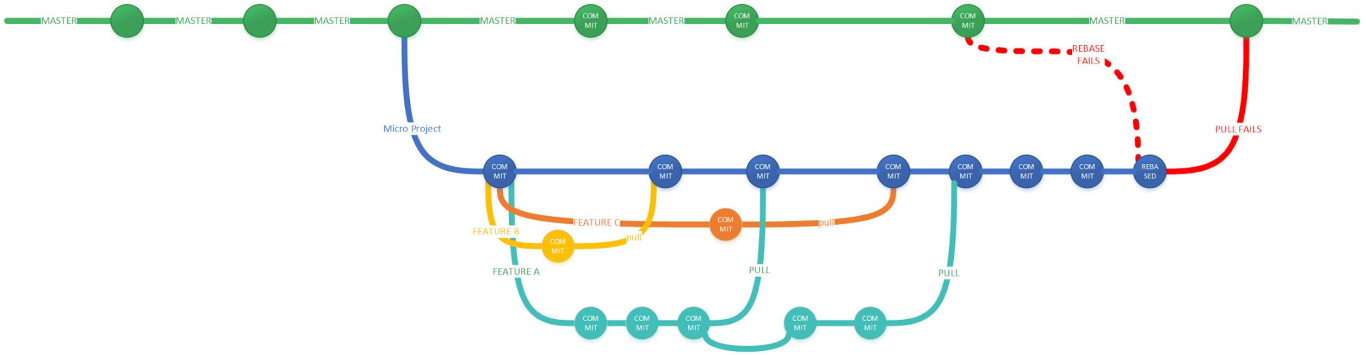
Un repositorio común

- Todos los ingenieros tienen que poder editar la configuración
- Pues la ponemos en un directorio compartido
- Pero...probablemente varios ingenieros quieran editarla a la vez
- ¿Por qué?
- Porque es de toda la red y uno quiere editar un router de Madrid y otro un switch de Barcelona
- O lo que es peor, los dos quieren editar routers de Valencia, uno el password y otro una VLAN
- hmmm

GIT AL RESCATE

¿GIT? ESO ES DE PROGRAMADORES

- Sistema de control de versiones
- Rama principal (master)
- Ramas secundarias
  - Que se pueden integrar en la principal
- Permite edición distribuida
- Incluso del mismo archivo



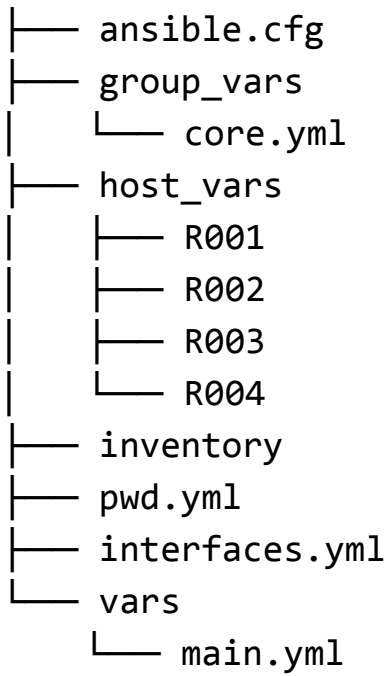
Ansible

Que es Ansible

- Herramienta escrita en Python
- Utiliza programas y configs. en YAML
  - (YAML Ain't Markup Language)
- Más sencillo que Puppet, etc.
- Agentless
  - Crea código python que descarga al cliente
  - Para routers el codigo se ejecuta en el servidor

Estructura de Ansible

- Recomendación
- Variables
  - Globales, grupo, host
  - Config de los equipos
- Playbooks
  - Programas en si



SuperGlue

```
├── ansible.cfg
├── group_vars
│   └── core.yml
├── host_vars
│   ├── R001
│   ├── R002
│   ├── R003
│   └── R004
├── inventory
├── pwd.yml
├── interfaces.yml
└── vars
    └── main.yml
```

Inventory

```
[core]
R001
R002
R003
R004
[clientes]
R005
R006
```

Host (R001)

```
ansible_ssh_host: 192.168.1.1
ansible_network_os: ios
ansible_user: ansible
ansible_password: YYYYYY
```

Variables

Globales

```
---
users:
  admin:
    password: xxxyyy
    state: present
  despedido:
    state: absent
```

Grupo

```
---
vlans:
  109:
    name: MGMT
  112:
    name DB
```

Host

```
---
interfaces:
  port-channel6:
    description: "FWINT-201:Bond1"
    admin_state: up
    swmode: trunk
    state: present
    members:
      - Eth1/26
    vpc: 26
    vlans:
      - 121
      - 125
    native: 1
```

Playbook usuarios (1)

```
---
- name: Configure users
  hosts: core

  tasks:
  - name: create user
    connection: network_cli
    ios_user:
      name: "{{ item.key }}"
      configured_password: "{{ item.value.password }}"
      update_password: always
      state: present
    when: item.value.state == 'present'
    with_dict: "{{ users }}"
```

```
---
users:
  admin:
    password: xxxyyy
    state: present
  despedido:
    state: absent
```

Playbook usuarios (2)

```
- name: delete user
    connection: network_cli
    ios_user:
      name: "{{ item.key }}"
      state: absent
    when: item.value.state == 'absent'
    with_dict: "{{ users }}"
  - name: Save config
    connection: network_cli
    ios_command:
      commands:
        - command: 'write'
```

```
---
users:
  admin:
    password: xxxyyy
    state: present
  despedido:
    state: absent
```

Playbook VLANs

```
---
- name: Configure vlans
  hosts: core

  tasks:
- name: "Create vlans"
  connection: network_cli
  ios_vlan:
    vlan_id: "{{ item.key }}"
    name: "{{ item.value.name }}"
    state: "{{ item.value.state }}"
  with_dict: "{{ vlans }}"
```

```
---
vlans:
  109:
    name: MGMT
  112:
    name DB
```

Playbook Interfaces (1)

```
---
- name: Configure routers
  hosts: core
  gather_facts: False

  tasks:
  - name: Configurar una interfaz fisica
    connection: network_cli
    ios_interface:
        name: "{{ item.key }}"
        enabled: "{{ (item.value.estado == 'up') |
ternary(true, false) }}"
        description: "{{ item.value.description }}"
    with_dict: "{{ interfaces }}"
```

```
---
interfaces:
  GigabitEthernet0/3:
    description: Test
interface
    estado: down
    ipv4: '10.20.30.1/30'
```

Playbook Interfaces (2)

```
- name: Configurar IP en una interfaz
  connection: network_cli
  ios_l3_interface:
      name: "{{ item.key }}"
      state: present
      ipv4: "{{ item.value.ipv4 }}"
  with_dict: "{{ l3_interfaces }}"
```

```
---
interfaces:
  GigabitEthernet0/3:
    description: Test
interface
    estado: down
    ipv4: '10.20.30.1/30'
```

Ejecución

```
ansible-playbook -v -i ./inventory pwd.yml


PLAY [Configure routers]
*********************************************************************************
TASK [create user]
*********************************************************************************
changed: [R003] => (item={'value': {u'state': u'present', u'password': u'XXXXXXX'},
'key': ADMIN}) => {"changed": true, "commands": ["username admin secret ********"],
"item": {"key": "admin", "value": {"password": "XXXXXXX", "state": "present"}}}
changed: [R003] => (item={'value': {u'state': u'absent', u'password': u'YYYYYYY'},
'key': u'despedido'}) => {"changed": true, "commands": ["username despedido secret
********"], "item": {"key": "despedido", "value": {"password": "YYYYYYY", "state":
"absent"}}}
```

Enlaces útiles

Ansible

https://www.ansible.com/resources/get-started

Google: Ansible tutorial

Módulos de red

https://docs.ansible.com/ansible/2.7/modules/list_of_network_modules.html

Git

https://github.com/git/git/blob/master/INSTALL

tuenti

¡Mañana taller!

THANKS!