



# Webinars DIY

## Programmabilité et automatisation dans les environnements LAN Datacenter

François COUDERC  
CCIE # 3435  
Principal Architect  
Business & Solution Expertise – Datacenter  
[fcouderc@cisco.com](mailto:fcouderc@cisco.com)



v1.0 – 17 février 2022

# Modèle DIY et architectures LAN Datacenter Cisco

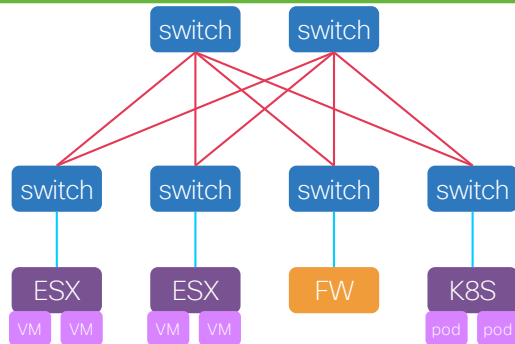
## Open NX-OS / DIY

opérateur

source of truth

ansible / autre

Focus de ce webinar



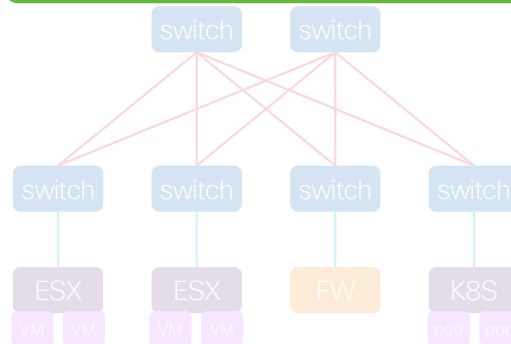
## NDFC

opérateur

source of truth

ansible / autre

Youtube – ACI & Infrastructure as Code  
ACI Days / VXLAN NDFC Days  
Devnet



## ACI

opérateur

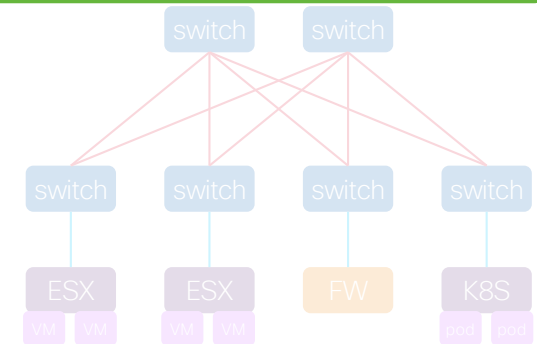
source of truth

ansible / autre

service now

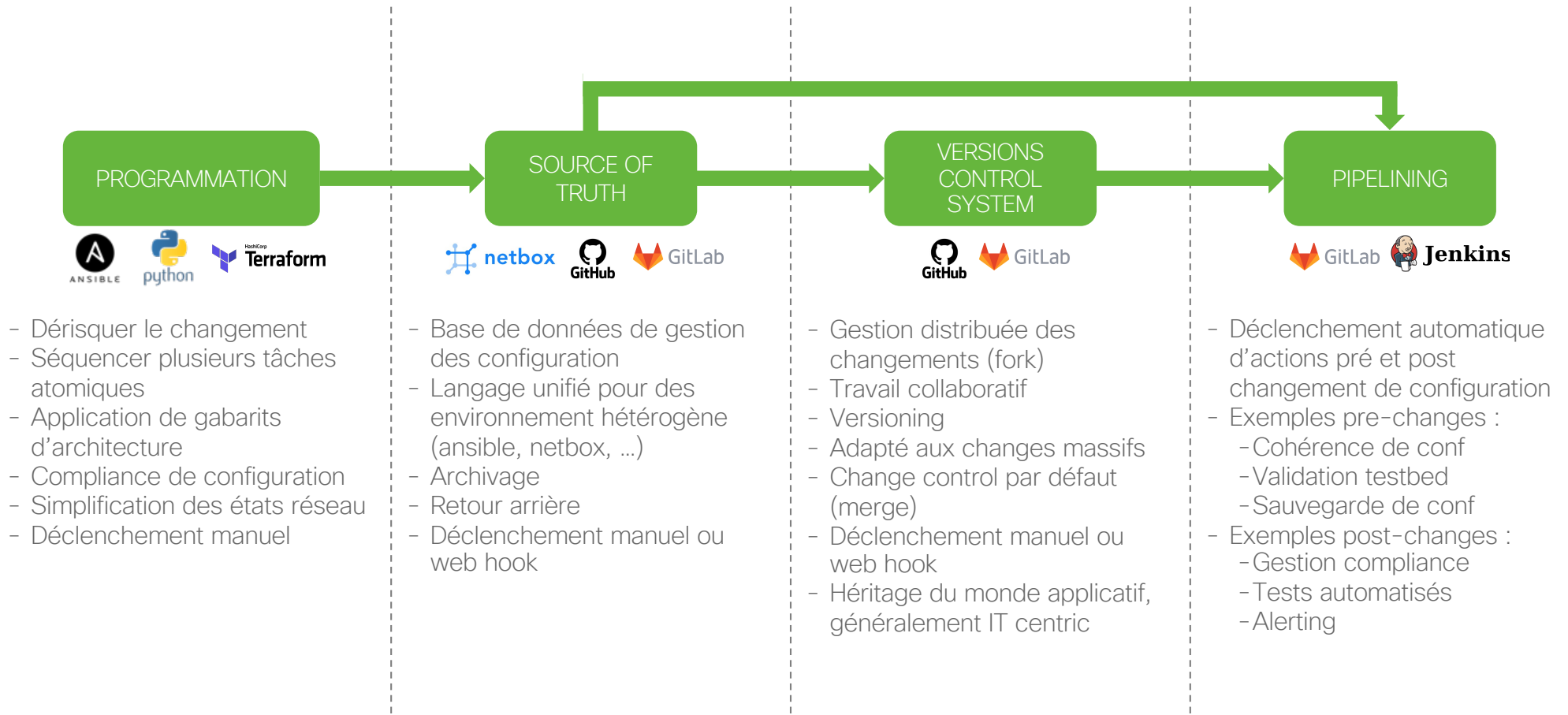
panorama

control

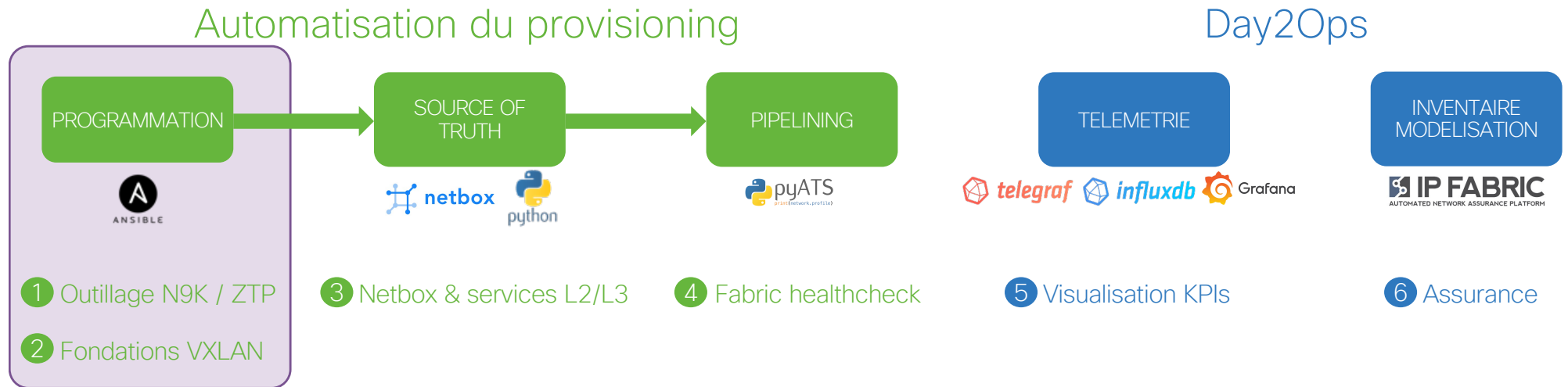


# Comment mettre en œuvre DIY ?

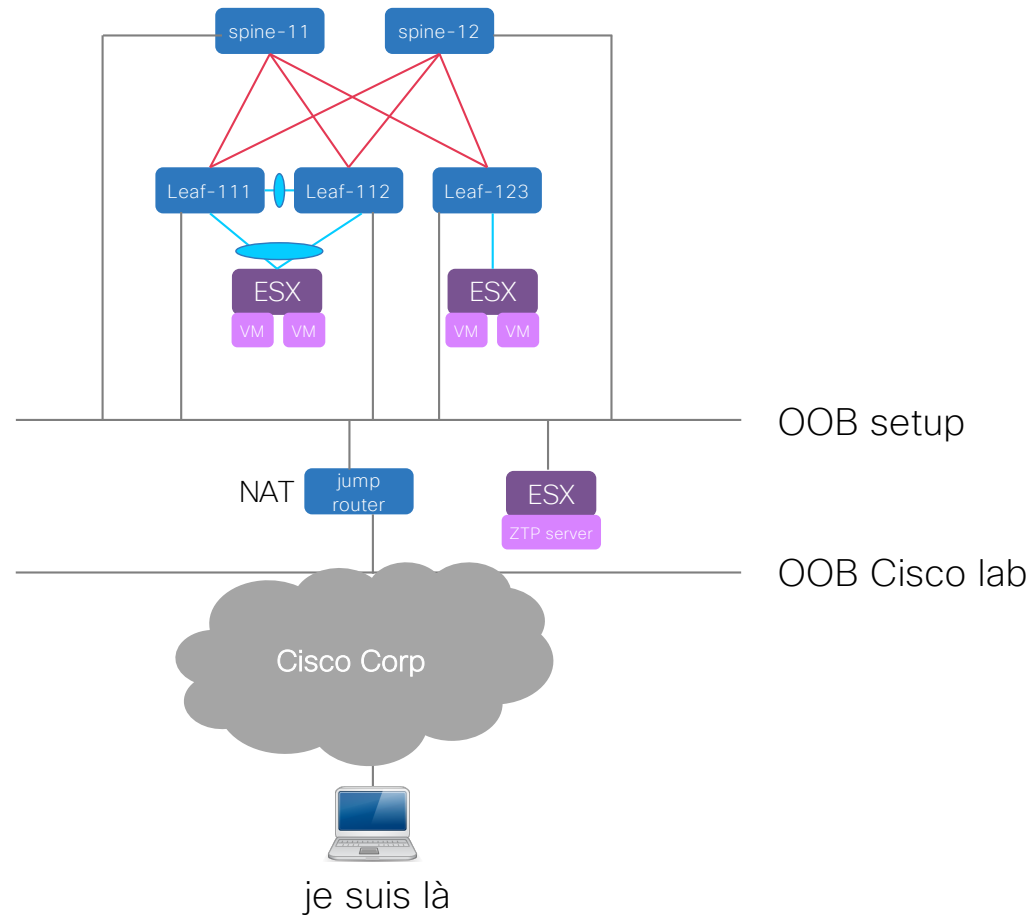
## Phasing netdevops



# Exemples de mise en route



# Démo setup



# Quick Win : Zero Touch Provisioning (aka POAP)

Avancer de piloter un Nexus 9000, il lui faut une adresse IP



<https://github.com/datacenter/nexus9000/blob/master/nx-os/poap/poap.py>

/etc/dhcp/dhcpd.conf

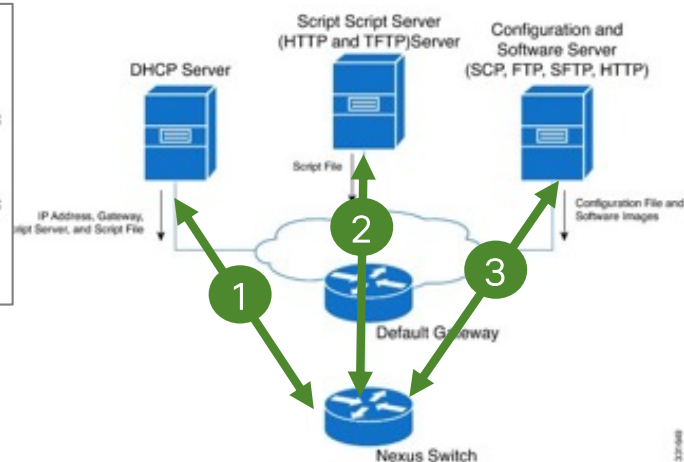
```
subnet 192.168.123.0 netmask 255.255.255.0 {
  group {
    option bootfile-name "poap_script/poap.py";
    server-name "http://192.168.123.252";
    option routers 192.168.123.254;
    host Leaf-101 {
      option dhcp-client-identifier "\000FD022281FP1";
      fixed-address 192.168.123.111;
      option host-name "Leaf-101";
    }
    host Leaf-102 {
      option dhcp-client-identifier "\000FD0223506V0";
      fixed-address 192.168.123.112;
      option host-name "Leaf-102";
    }
  }
}
```

```
options = {
  "hostname": "192.168.123.252",
  "transfer_protocol": "http",
  "mode": "hostname",
  "target_system_image": "nxos.9.3.1.bin",
  "config_path": "/configs",
  "target_image_path": "/images",
  "destination_path": "/bootflash/",
}
```

/config/conf\_Leaf-101.cfg

```
switchname Leaf-101
no password strength-check
username admin password 0 cisco123 role network-admin
vrf context management
  ip route 0.0.0.0/0 192.168.123.254
interface mgmt0
  vrf member management
  ip address 192.168.123.111/24
boot nxos bootflash:/nxos.9.3.1.bin
<snip>
```

/images/nxos.9.3.1.bin

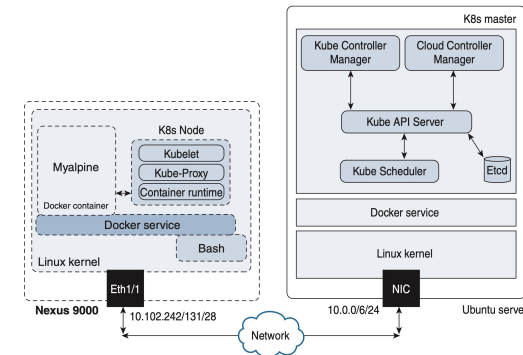


Facilite l'enrôlement d'un nouveau switch dans une fabrique  
Changement d'un hardware défectueux sans présence locale dans le DC  
Inband / Out of Band / USB / container Nexus 9000

- 1 Le switch récupère en DHCP :
  - IP/GW
  - l'URL pour télécharger le script POAP
- 2 Le switch récupère en HTTP le script POAP et l'exécute (interpréteur python natif dans NXOS)
- 3 Le switch récupère en HTTP sa configuration et son image (si nécessaire)

# N9K : fonctions de programmation avancées

- Piloter le switch (NX-OS) différemment avec ses propres commandes
  - Interpréteur Python, Guest Shell, Environnement python, Docker Engine, Kubernetes, NX-SDX



- Accéder au switch à distance à travers une API
  - Facilite l'outillage plus haut niveau à travers un langage intuitif pour les développeurs
  - Modèles Open facilitant interopérabilité (IETF, OpenConfig)
    - Définis en Yang
    - Respect des standard (IETF, ITU, OpenConfig)
    - Protocoles d'accès standards (Netconf, Restconf)
  - Modèles Yang NX-OS spécifiques
    - Générés à partir du Device Model NX-OS
    - Protocoles d'accès standards (Netconf, Restconf)
  - Modèles NX-OS DME
    - Défini comme MO (Managed Object) XML/JSON
    - Modèles spécifiques à NX-OS
    - Accès aux API RESTful





- Logiciel libre de gestion des configurations
- Automatise le déploiement et la livraison continue des mises à jours
- Historiquement pensé pour le système et les applications
- Pas d'agent sur les switches
- Gestion d'une Fabrique Nexus 9000
  - Gestion via templates CLIs et/ou nombreux modules intégrés à Ansible
  - Les modules NXOS (collection) sont gérés nativement par RedHat sous partenariat Cisco



# Démo ansible

## Fondations VXLAN

```
! Leaf-111.yaml ×
Users > fcouderc > Documents > FY22 > DIY > foundations > host_vars > ! Leaf-111.yaml
1  loopback_interfaces:
2    - {port: "loopback0", IP: "1.1.1.111", description: "*** Router ID ***"}
3    - {port: "loopback1", IP: "1.0.0.111", description: "*** VTEP IP address ***"}
4
5  fabric_interfaces:
6    - {port: "Ethernet1/51", IP: "1.11.111.111", mask: "24", description: "*** To Spine-11 - e1/1 ***"}
7    - {port: "Ethernet1/52", IP: "1.12.111.111", mask: "24", description: "*** To Spine-12 - e1/1 ***"}
```

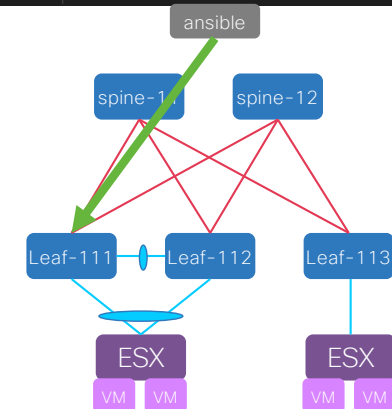
```
leaves.j2 ×
Users > fcouderc > Documents > FY22 > DIY > foundations > CLI templates > leaves.j2
108 interface {{ fabric_interface.port }}
109   description {{ fabric_interface.description }}
110   mtu {{ MTU }}
111   ip address {{ fabric_interface.IP }}/{{ fabric_interface.mask }}
112   no ip redirects
113   ip ospf network point-to-point
114   ip router ospf {{ OSPF_name }} area {{ OSPF_area }}
115   no shutdown
116 {% endfor %}
```

1 Pour chaque switch  
je récupère les variables ...

2 ... que j'utilise pour remplir  
les blancs de mon template ...

- 3 ... et ainsi consolider le fichier  
de configuration...
- 4 ... que je pousse au switch

```
Leaf-111.txt ×
Users > fcouderc > Documents > FY22 > DIY > foundations
82
83 interface Ethernet1/51
84   description *** To Spine-11 - e1/1 ***
85   mtu 9216
86   ip address 1.11.111.111/24
87   no ip redirects
88   ip ospf network point-to-point
89   ip router ospf underlay area 0.0.0.0
90   no shutdown
91
92 interface Ethernet1/52
93   description *** To Spine-12 - e1/1 ***
94   mtu 9216
95   ip address 1.12.111.111/24
96   no ip redirects
97   ip ospf network point-to-point
98   ip router ospf underlay area 0.0.0.0
99   no shutdown
```



# Les prochains webinars

<https://gblogs.cisco.com/fr/>

**Webinar 1** : Cas d'usage et bénéfices DIY – Jeudi 10 février de 10:00 à 12:00 : [inscription](#)

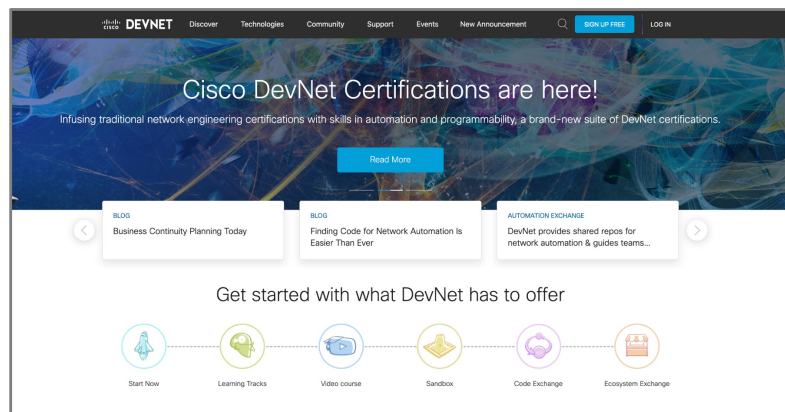
**Webinar 2** : Zero Touch Provisionning – Jeudi 17 février de 10:00 à 12:00 : [inscription](#)

**Webinar 3** : Provisioning de bout en bout – Jeudi 10 mars de 10:00 à 12:00 : [inscription](#)

**Webinar 4** : PyATS – Jeudi 17 mars de 10:00 à 12:00 : [inscription](#)

**Webinar 5** : Télémétrie et intégration IP Fabric – Jeudi 24 mars de 10:00 à 12:00 : [inscription](#)

developper.cisco.com



Contenu DevNet pour le Datacenter :

