TecnoCampus Escola Superior Politècnica

Centre adscrit a la upf. Universitat Pompeu Fabra Barcelona

Grau en Enginyeria Informàtica de Gestió i Sistemes d'Informació

AUTOMATITZACIÓ DE XARXES

TecnoCampus Mataró-Maresme

ARNAU AMARGANT DOMENE
TUTOR: PERE BARBERAN AGUT

DATA
07-07-2021

Índex

Índex

Canvis…

- Les xarxes han incrementat en complexitat i dimensions.
- Noves tecnologies, aplicacions al Cloud, IoT, centres de dades, etc.

Inconvenients…

- Xarxes més difícils d'escalar i gestionar.
- Principal font d'errors són errors humans de configuracions.

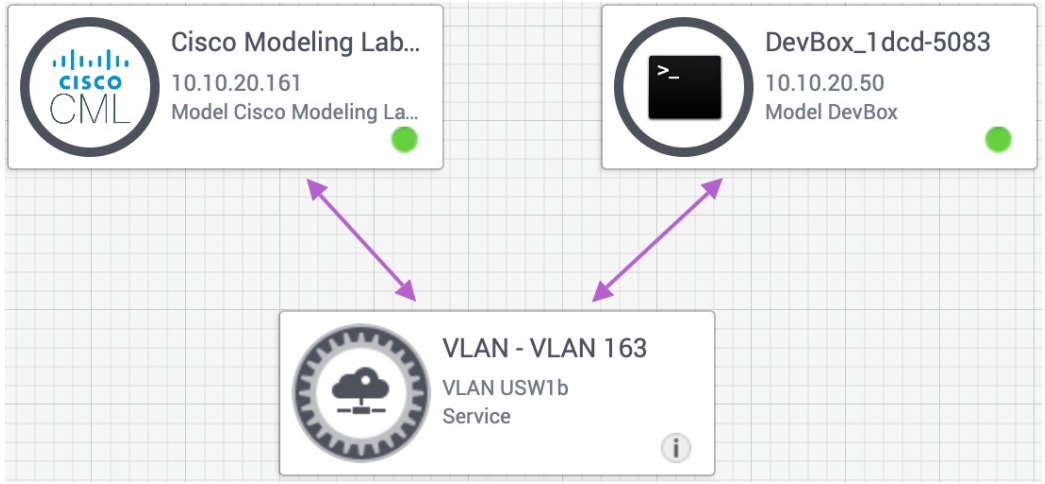
Índex

- Dissenyar, automatitzar i monitorar una xarxa de tipus campus.
- Comprovar els avantatges que s'obtenen en temps operacional, temps d'implementacions, instal·lacions i gestió de diferents protocols i paquets.
- Conèixer i implementar diferents mecanismes i eines per automatitzar una xarxa com Ansible, RESTCONF i NETCONF.

Índex

Cisco modeling labs

- Eina de Cisco per la simulació de xarxes virtuals.
- Connexió a través de SSH als nodes de la topologia.
- Connexió del node CML a la DevBox(CentOS) a través del node ”external_conection” de la topologia.



Formats de dades

- Estructures de dades.
- Comprensible per les persones i ordinadors.
- Es defineixen en forma de clau i valor.
- Representació d'objectes I llistes.
- Els més comuns i utilitzats són JSON, XML I YAML.

core1# show int

```
Ethernet1/1 is up
admin state is up, Dedicated Interface
  Description: Core1 to Core2 interface
  Internet Address is 20.0.0.101/30
```

Format de dades XML

```
<interface>
    <name>eth1/1</name>
    <description>Core1 to Core2 interface</description>
    <enabled>true</enabled>
    <ipv4>
        <address>
            <ip>20.0.0.101</ip>
            <netmask>255.255.255.0</netmask>
        </address>
    </ipv4>
</interface>
```

Formats de dades - XML

- Va ser dissenyat per l'internet.
- Ús d'etiquetes i elements per representar la funció de clau i valor.
- Els espai en blanc son irrellevants.

```
<interface>
    <name>eth1/1</name>
    <description>Core1 to Core2 interface</description>
    <enabled>true</enabled>
    <ipv4>
        <address>
            <ip>20.0.0.101</ip>
            <netmask>255.255.255.0</netmask>
        </address>
    </ipv4>
</interface>
```

Formats de dades - JSON

- Fa servir l'estructura clau i valor.
- Utilitza les {} per objectes i [] per llistes.
- Els espais en blanc són irrellevants.

```
{
  "interface": {
    "name": "eth1/1",
    "description": "Core1 to Core2 interface",
    "enabled": "true",
    "ipv4": {
      "address": [
        {
          "ip": "20.0.0.101",
          "netmask": "255.255.255.0"
        }
      ]
    }
  }
}
```

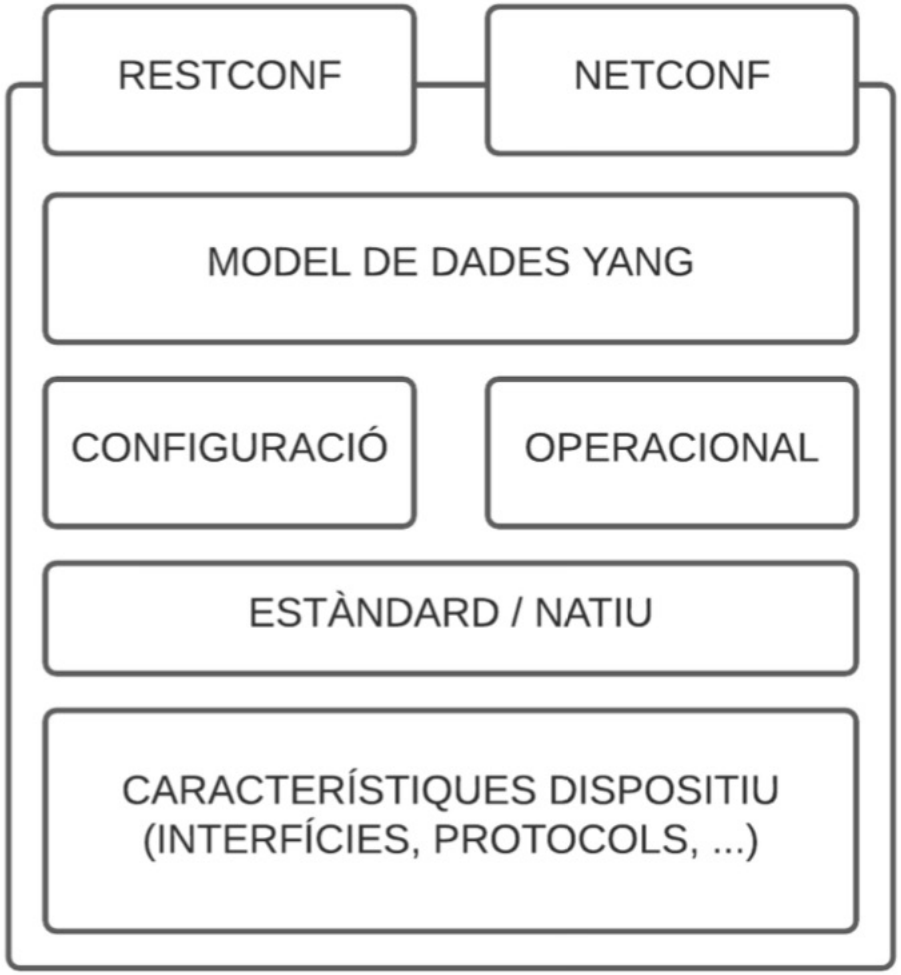
Formats de dades - YAML

- Format minimalista.
- Fa servir l'estructura clau i valor.
- Els espais en blanc defineixen l'estructura.

```
---
interface:
  name: eth1/1
  description: Core1 to Core2 interface
  enabled: "true"
  ipv4:
    address:
      - ip: 20.0.0.101
        netmask: 255.255.255.0
```

YANG

- És un llenguatge de modelatge de dades.
- YANG defineix i descriu les característiques d'un dispositiu.
- Utilitza contenidors i llistes per identificar nodes.



YANG

- Estructura del model de dades YANG.
- Part de configuració i operacional.

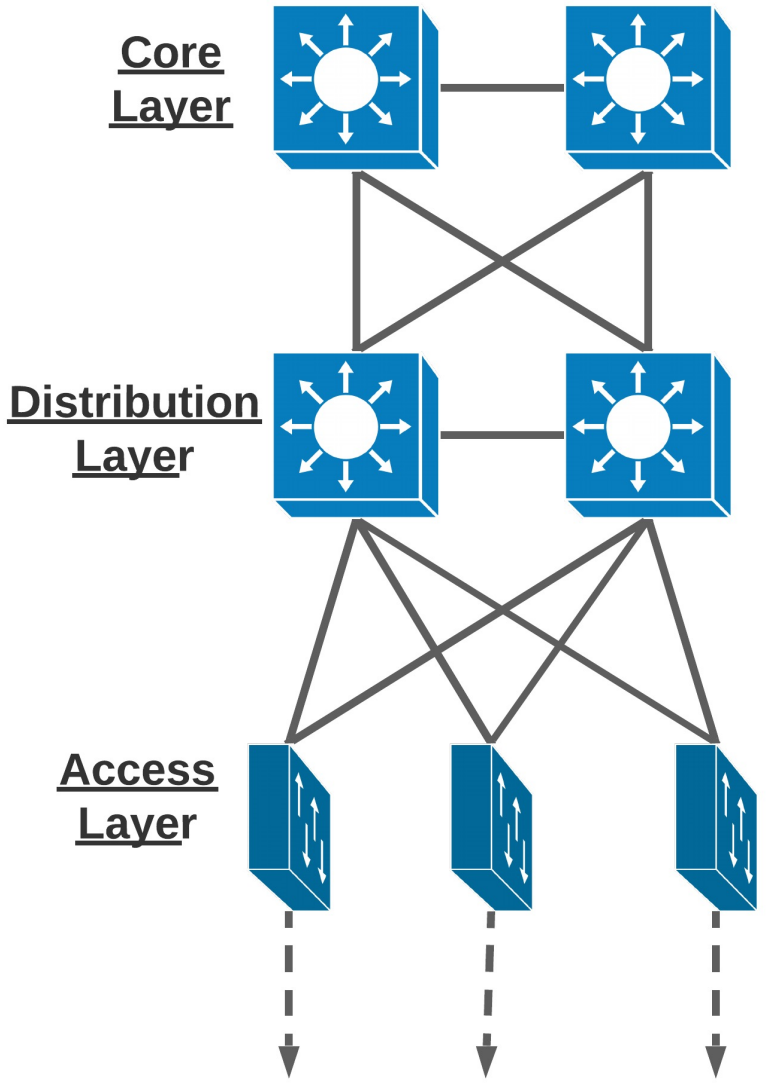
```
container interfaces {
      description
        "Top level container for interfaces, including configuration
        and state data.";


      list interface {
        key "name";

        description
          "The list of named interfaces on the device.";

        leaf name {
          type leafref {
            path "../config/name";
          }
```

```
$ pyang -f tree openconfig-interfaces.yang
```

```
module: openconfig-interfaces
  +--rw interfaces
     +--rw interface* [name]
        +--rw name                  -> ../config/name
        +--rw config
        |  +--rw name?              string
        |  +--rw type               identityref
        |  +--rw mtu?               uint16
        |  +--rw loopback-mode?     boolean
        |  +--rw description?       string
        |  +--rw enabled?           boolean
        +--ro state
        |  +--ro name?              string
        |  +--ro type               identityref
        |  +--ro mtu?               uint16
        |  +--ro loopback-mode?     boolean
        |  +--ro description?       string
        |  +--ro enabled?           boolean
```

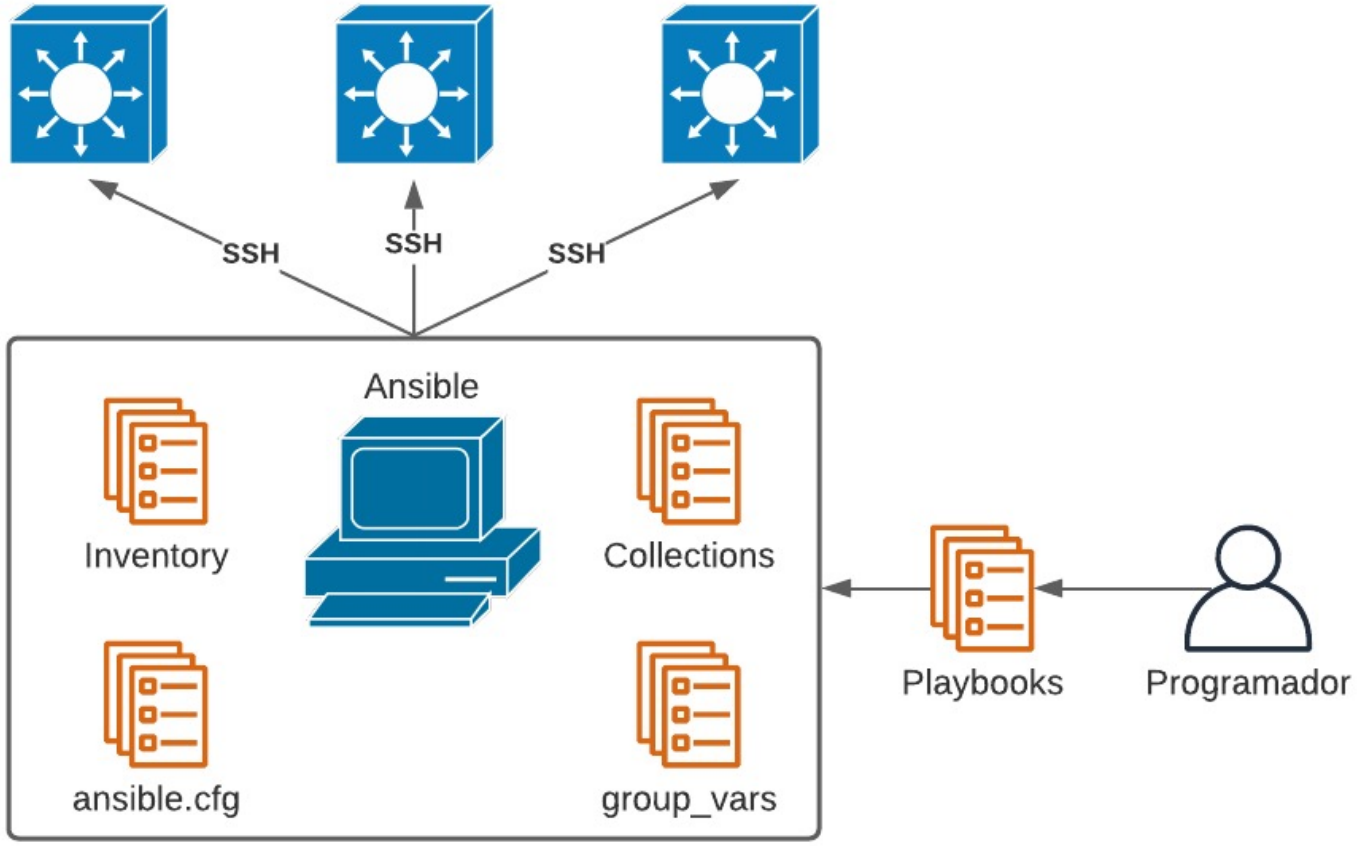
Topologia

- Topologia de tipus campus.
- Dispositius Nexus switch a la capa del nucli i distribució.
- Dispositiu IOSvL2 a la capa d'accés.
- Configuració inicial (Interfície de management i SSH).
- Importació de la topologia amb "topologySetUp.py".



Ansible

- Eina per l'automatització de dispositius.
- Connexió a través de SSH.
- Ús de plantilles en format YAML.



Ansible

Playbook “core.yml”

```yaml
---
- name: Configure Core Group
  hosts: core
  tasks:

  - name: Enable RESTCONF
    nxos_feature:
      feature: restconf
      state: enabled

  - name: Enable NETCONF
    nxos_feature:
      feature: netconf
      state: enabled
```

Inventari

```
[all:vars]
username=cisco
password=cisco

[core:children]
core1
core2
core3
core4

[core1]
10.10.20.177

[core2]
10.10.20.178

[core3]
10.10.20.179

[core4]
10.10.20.180
```

Arxiu group_vars/core.yml

```yaml
ansible_network_os: nxos
```

Arxiu group_vars/all.yml

```yaml
ansible_connection: network_cli
ansible_user: "{{username}}"
ansible_ssh_pass: "{{password}}"
```

Arxiu requirements.yml

```yaml
---
collections:
  - name: cisco.ios
    version: 2.0.0
    source: https://galaxy.ansible.com
  - name: cisco.nxos
    version: 2.0.0
    source: https://galaxy.ansible.com
  - name: cisco.asa
    version: 2.0.2
    source: https://galaxy.ansible.com
```

RESTCONF

- Protocol amb connexió a través de HTTP.
- Interfície programàtica per accedir a dades definides en YANG.
- Suporta els formats de dades JSON i XML.
- Suporta les sol·licituds GET, POST, PUT, PATCH i DELETE.

```
def get_request(node):
  url = "https://" + node["address"] + "/restconf/data/Cisco-NX-OS-device:System/ipv4-items/inst-items/dom-items/Dom-list"

  payload= ""
  headers = {
    'Content-Type': 'application/yang.data+json',
    'Accept': 'application/yang.data+xml',
  }

  response = requests.request("GET", url, auth=('cisco', 'cisco'), headers=headers, data=payload,  verify=False)
```

Configuració amb RESTCONF

Arxiu restconf_ip_route_conf.py

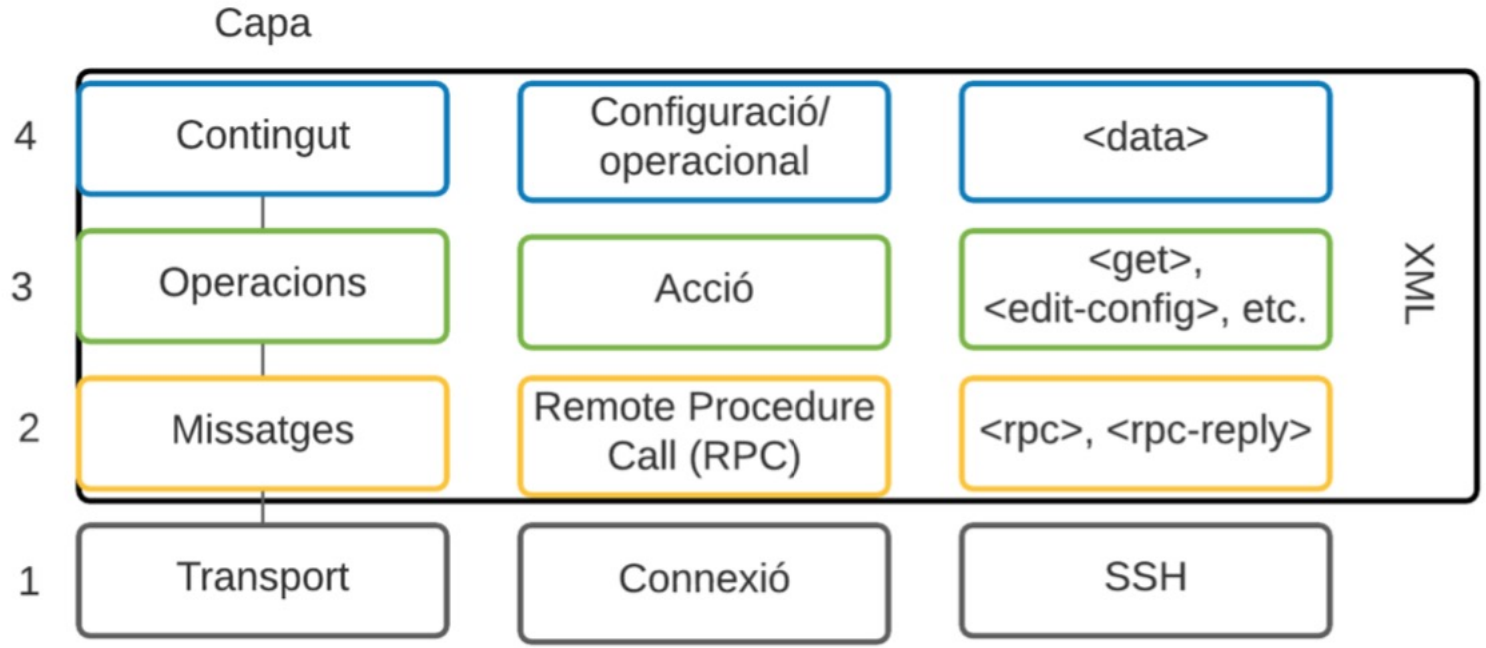
```python
def get_request(node):
  url = "https://" + node["address"] + "/restconf/data/Cisco-NX-OS-device:System"

  payload= open("ip_routing_conf.json").read()
  headers = {
    'Content-Type': 'application/yang.data+json',
    'Accept': 'application/yang.data+json',
  }

  response = requests.request("PATCH", url, auth=('cisco', 'cisco'), headers=headers,
                              data=payload,  verify=False)
```

Arxiu ip_routing_conf.json

```json
{
  "ipv4-items": {
    "inst-items": {
      "dom-items": {
        "Dom-list": {
          "name": "default",
          "rt-items": {
            "Route-list": {
              "prefix": "1.1.1.1",
              "nh-items": {
                "Nexthop-list": [
                  {
                    "nhIf": "eth1/1",
                    "nhAddr": "2.2.2.2",
                    "nhVrf": "default"
                  },
                  {
                    "nhIf": "eth1/2",
                    "nhAddr": "3.3.3.3",
                    "nhVrf": "default"
}]}}}}}}}}}
```

NETCONF

- Protocol amb connexió a través de SSH.
- Interfície programàtica per accedir a dades definides en YANG.
- Suporta només el format XML.



NETCONF – Automatització amb ncclient (Configuració)

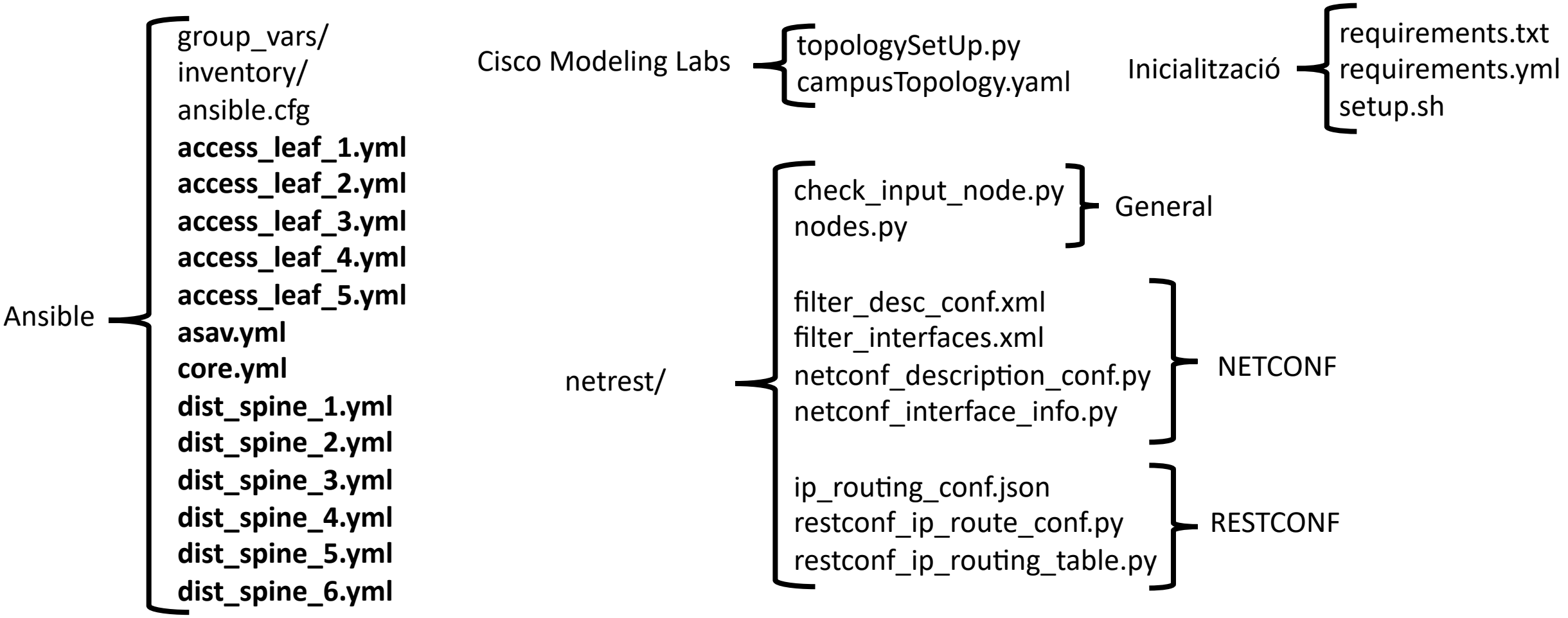
xml_filter

```
<config>
    <interfaces xmlns="http://openconfig.net/yang/interfaces">
                <interface>
                <name>eth1/1</name>
                    <config>
                        <description>new description</description>
                    </config>
                </interface>
    </interfaces>
</config>
```

```
from ncclient import manager
```

```
def get_request(xml_filter, node):
    with manager.connect(host=node["address"], port=node["port"],
        username= node["username"], password=node["password"],
        hostkey_verify=False) as device:

        netconf_reply = device.edit_config(xml_filter, target = 'running')
```

Arxius del repositori: https://github.com/aamargant/net_automation_aamargant



Índex

Índex

- Les eines d'automatització milloren la visibilitat i temps d'aprovisionament de la xarxa.
- Són essencials per l'escalabilitat i optimització de la xarxa.
- Els objectius s'han complert i s'han explicat i implementat les diferents eines d'automatització de xarxes com Ansible, NETCONF i RESTCONF.