

The logo for NIC.br features the text 'nic.br' in a bold, sans-serif font. The 'nic' is in black, and '.br' is in a light green color. Below the logo, the full name 'Brazilian Network Information Center' is written in a smaller, black, sans-serif font.

**nic.br**  
Brazilian Network  
Information Center

The logo for EGI.br features the text 'egi.br' in a bold, sans-serif font. The 'egi' is in black, and '.br' is in a light green color. Below the logo, the full name 'Brazilian Internet Steering Committee' is written in a smaller, black, sans-serif font.

**egi.br**  
Brazilian Internet  
Steering Committee

A horizontal row of logos for various Brazilian internet services. From left to right: 'registro.br' (registro in black, .br in green), 'cert.br' (cert in black, .br in green), 'cetic.br' (cetic in black, .br in green), 'ceptro.br' (ceptro in black, .br in green), 'ceweb.br' (ceweb in black, .br in green), and 'ix.br' (ix in black, .br in green).

registro.br cert.br cetic.br ceptro.br ceweb.br ix.br

# Automatizando tarefas com NetBox e Nornir

William Prado  
IX.br Engineering

nie.br

# Agenda

01

Por que  
Automatizar?

02

Desafios

03

Nornir  
O que é?  
Plugins, Tasks e  
exemplos.

04

Napalm

O que é?  
Exemplos com nornir.

05

NetBox

O que é?  
Inventário e exemplos  
com nornir.

06

NetBox como  
Source of Truth

07

Curiosidades

08

Conclusão

# Por que automatizar?

- Até 95% das alterações de rede são feitas manualmente;
- Mudanças manuais levam a erros de configuração e inconsistências na rede;
- Tarefas repetitivas demandam muito tempo de um profissional;
- Tempo elevado para solucionar problemas - troubleshooting;
- Tempo elevado para implantação de equipamentos na rede;
- Tempo elevado para ativação de serviços na rede;

# Desafios

- Ter uma rede padronizada faz muita diferença;
- Rede bem documentada - Inventário;
- Conhecer a infraestrutura e seus protocolos;
- Conhecimento em outras áreas além de desenvolvimento de software;



# Nornir - o que é?

Inventory: hosts.yaml

- Framework de automação 100% em python (debugging);
- Open-source;
- Estrutura multithread;
- Gerenciamento de inventário;
- Suporta YAML e JINJA2 através de plugins;
- Muito rápido;

```
1 from nornir import InitNornir
2 from nornir_rich.functions import print_result
3 from nornir_netmiko.tasks.netmiko_send_command import netmiko_send_command
4
5
6 nr = InitNornir(
7     runner={"plugin": "threaded", "options": {"num_workers": 20}},
8     config_file="hosts.yaml")
9
10
11 print_result(nr.run(netmiko_send_command, command_string="show interfaces brief"))
12
```

```
1 R1:
2   hostname: '192.168.246.94'
3   port: 22
4   username: 'ixforum'
5   password: 'ixforum'
6   platform: 'ios'
7 R2:
8   hostname: '192.168.246.95'
9   port: 22
10  username: 'ixforum'
11  password: 'ixforum'
12  platform: 'ios'
13 R3:
14  hostname: '192.168.246.96'
15  port: 22
16  username: 'ixforum'
17  password: 'ixforum'
18  platform: 'ios'
19 R4:
20  hostname: '192.168.246.102'
21  port: 22
22  username: 'ixforum'
23  password: 'ixforum'
24  platform: 'ios'
25 R5:
26  hostname: '192.168.246.103'
27  port: 22
28  username: 'ixforum'
29  password: 'ixforum'
30  platform: 'ios'
31
```



Function

Task

<https://nornir.readthedocs.io/en/latest/>

# Nornir Plugins

Plugin é um código utilizado para aumentar e melhorar as funcionalidades de um software;

Nornir permite adicionar funcionalidades através de plugins;

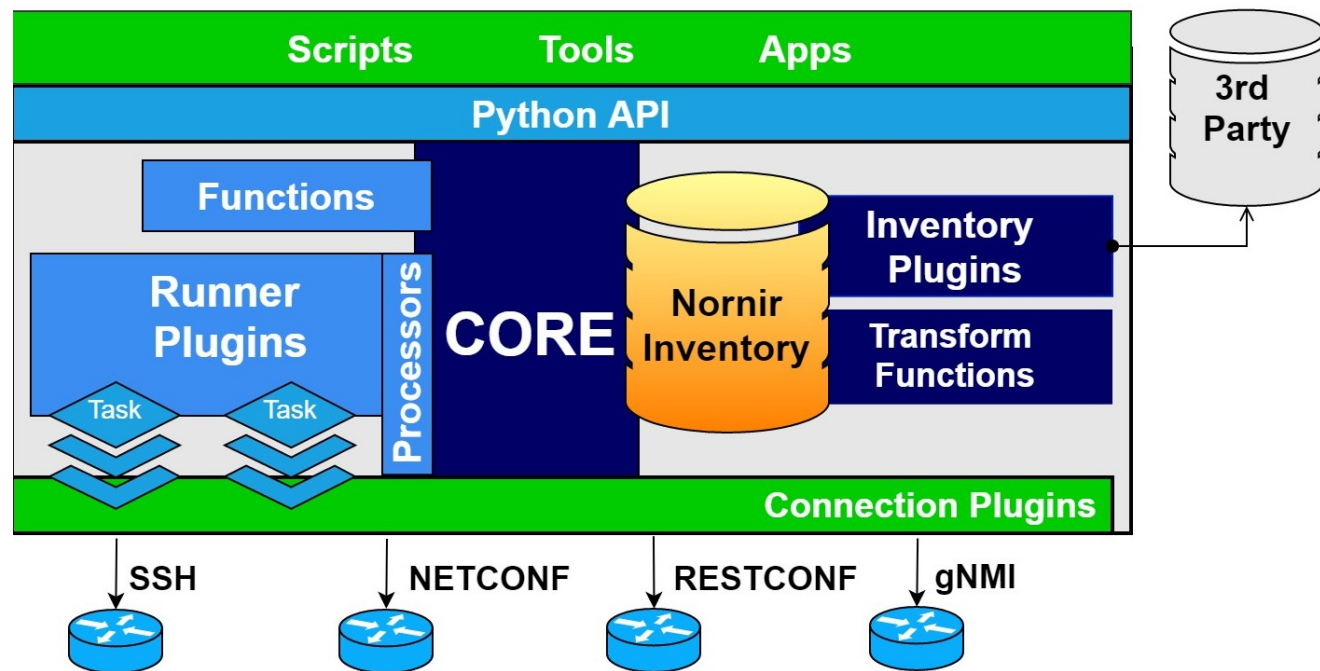
Plugins podem ser instalados usando pip:

```
pip install nornir_napalm
```

Nesta apresentação vamos usar:

- nornir\_napalm
- nornir\_netbox
- nornir\_utils
- nornir\_netmiko
- nornir\_rich

## Nornir Plugins Architecture



<https://nornir.tech/nornir/plugins/>

# Nornir Tasks

- Task define as ações que serão executadas no host;
- Task é uma função python;
- Para executar uma task usamos **run**;
- Tasks são marcadas com failed em caso de exceções na execução;

```
1 from nornir import InitNornir
2 from nornir_rich.functions import print_result
3 from nornir.core.task import Task, Result
4
5 def hello_world(task: Task) -> Result:
6     return Result(host=task.host, result=f"{task.host.name} diga olá mundo!")
7
8 nr = InitNornir(
9     runner={"plugin": "threaded", "options": {"num_workers": 20}},
10    config_file="hosts.yaml")
11
12 result = nr.run(task=hello_world)
13
14 print_result(result)
15
```

```
hello_world
R1 diga olá mundo!
```

```
hello_world
R2 diga olá mundo!
```

```
hello_world
R3 diga olá mundo!
```

```
hello_world
R4 diga olá mundo!
```

```
hello_world
R5 diga olá mundo!
```





# Nornir - Task 1: Netmiko - results

```
R1 | netmiko_send_command
netmiko_send_command

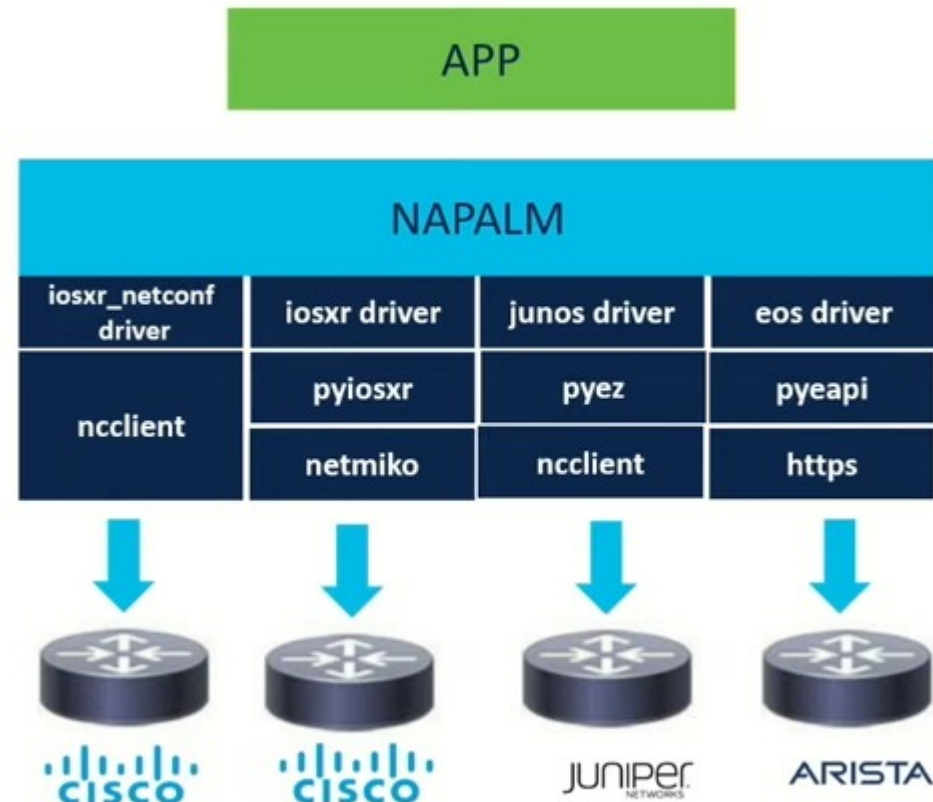
Fri Nov 24 21:14:30.349 UTC

  Intf      Intf      LineP      Encap      MTU      BW
  Name      State     State      Type        (byte)   (Kbps)
-----
    Lo0      up        up         Loopback    1500      0
  Lo1000    up        up         Loopback    1500      0
    Nu0      up        up         Null        1500      0
Mg0/RP0/CPU0/0 up        up         ARPA        1514      0
  Gi0/0/0/0 up        up         ARPA        1514    1000000
  Gi0/0/0/1 up        up         ARPA        1514    1000000
  Gi0/0/0/2 up        up         ARPA        1514    1000000
  Gi0/0/0/3 up        up         ARPA        1514    1000000
```

# Napalm – O que é?

- Biblioteca Python que disponibiliza funções para interagir com diferentes equipamentos de rede;
- Cisco IOS-XR, Cisco IOS, Cisco NX-OS, Junos e Arista EOS;
- Open Source;
- Camada de abstração para programação e Automação de Redes;
- Outras plataformas: <https://github.com/napalm-automation-community>

Functions	Functions	Functions	Functions
get_facts	get_route_to	get_arp_table	get_environment
get_interfaces	get_snmp_information	get_ntp_peers	cli
get_interfaces_counters	get_probes_config	get_ntp_servers	get_firewall_policies
get_interfaces_ip	get_probes_results	get_ntp_stats	get_ipv6_neighbors_table
get_bgp_config	traceroute	get_lldp_neighbors	get_network_instances
get_bgp_neighbors	get_users	get_lldp_neighbors_detail	get_optics
get_bgp_neighbors_detail	get_config	get_mac_address_table	ping



<https://napalm.readthedocs.io/en/latest/>

# Napalm – Estrutura de Dados

## IOS-XR

```
{  
  "uptime": 35457914,  
  "vendor": "Cisco",  
  "hostname": "edge01.tab",  
  "fqdn": "edge01.tab01",  
  "os_version": "5.3.1",  
  "serial_number": "FOX171",  
  "model": "ASR-9904-AC",  
  "interface_list": [  
    "TenGigE0/0/0/13",  
    "TenGigE0/0/0/14",  
    "TenGigE0/0/0/24"  
  ]  
}
```

## IOS

```
{  
  "uptime": 16676160,  
  "vendor": "Cisco",  
  "hostname": "NS2903",  
  "fqdn": "NS2903-ASW",  
  "os_version": "15.0(2)",  
  "serial_number": "FOC1",  
  "model": "WS-C2960G",  
  "interface_list": [  
    "Vlan1",  
    "GigabitEthernet0/1",  
    "GigabitEthernet0/5"  
  ]  
}
```

## JUNOS

```
{  
  "uptime": 4380,  
  "vendor": "Juniper",  
  "hostname": "vsrx",  
  "fqdn": "vsrx",  
  "os_version": "12.1X4",  
  "serial_number": "beb91",  
  "model": "FIREFLY",  
  "interface_list": [  
    "ge-0/0/0",  
    "ge-0/0/1",  
    "ge-0/0/2"  
  ]  
}
```

## EOS

```
{  
  "uptime": 123456,  
  "vendor": "Arista",  
  "hostname": "localhost",  
  "fqdn": "localhost",  
  "os_version": "4.15.5M",  
  "serial_number": "",  
  "model": "vEOS",  
  "interface_list": [  
    "Ethernet1",  
    "Ethernet2",  
    "Ethernet3",  
    "Management1"  
  ],  
}
```

# Nornir - Task 2: Napalm

```
1 from nornir import InitNornir
2 from nornir_napalm.plugins.tasks import napalm_get
3 from nornir_rich.functions import print_result
4
5 nr = InitNornir(
6     runner={"plugin": "threaded", "options": {"num_workers": 20}},
7     config_file="hosts.yaml")
8
9 r2 = nr.filter(name="R2")
10
11 print_result(r2.run(task=napalm_get, getters=["get_interfaces"]), vars=["result"])
12
```

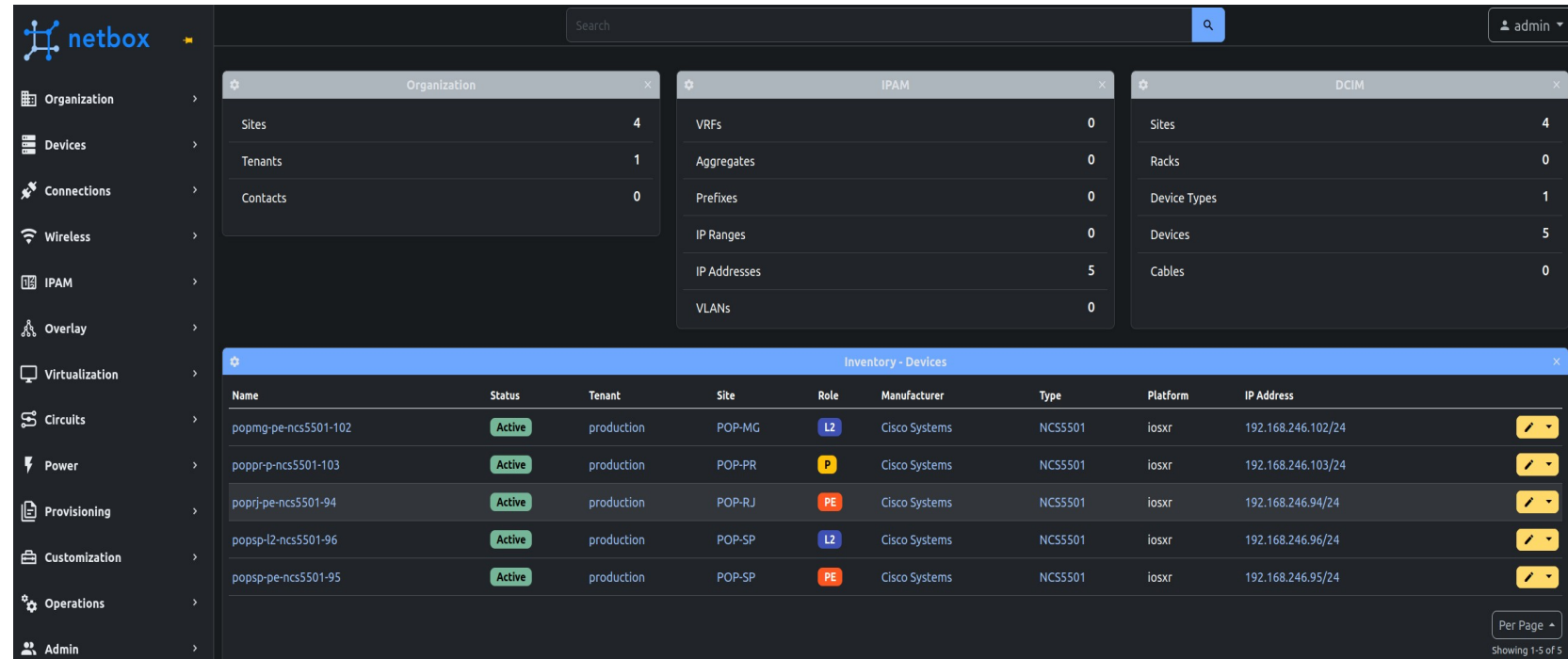


# Nornir - Task 2: Napalm - results

```
result = {
  'get_interfaces': {
    'Loopback0': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '', 'last_flapped': -1.0, 'mtu': 1500, 'speed': 0.0},
    'Loopback1000': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '', 'last_flapped': -1.0, 'mtu': 1500, 'speed': 0.0},
    'Null0': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '', 'last_flapped': -1.0, 'mtu': 1500, 'speed': 0.0},
    'MgmtEth0/RP0/CPU0/0': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '50:03:00:03:00:00', 'last_flapped': -1.0, 'mtu': 1514, 'speed': 0.0},
    'GigabitEthernet0/0/0/0': {
      'is_enabled': False,
      'is_up': False,
      'description': '',
      'mac_address': '50:03:00:03:00:03',
      'last_flapped': -1.0,
      'mtu': 9198,
      'speed': 1000.0
    },
    'GigabitEthernet0/0/0/1': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '50:03:00:03:00:04', 'last_flapped': -1.0, 'mtu': 1514, 'speed': 1000.0},
    'GigabitEthernet0/0/0/2': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '50:03:00:03:00:05', 'last_flapped': -1.0, 'mtu': 1514, 'speed': 1000.0}
  }
}
```

# O que é NetBox?

- Open-source IPAM/DCIM;
- Desenvolvido em Python (Django);
- Permite documentar sua infraestrutura;
- Peça chave em Automação de Redes;
- Atua como “Source of Truth” para sua infraestrutura/rede;
- Ideal para todos os times da empresa;
- API Support
  - REST and GraphQL;
  - Webhooks;
  - Pynetbox;



The screenshot displays the NetBox web interface. On the left is a navigation sidebar with categories like Organization, Devices, Connections, Wireless, IPAM, Overlay, Virtualization, Circuits, Power, Provisioning, Customization, Operations, and Admin. The main content area features three summary cards: Organization (4 Sites, 1 Tenants, 0 Contacts), IPAM (0 VRFs, 0 Aggregates, 0 Prefixes, 0 IP Ranges, 5 IP Addresses, 0 VLANs), and DCIM (4 Sites, 0 Racks, 1 Device Types, 5 Devices, 0 Cables). Below these is a table titled 'Inventory - Devices' with columns for Name, Status, Tenant, Site, Role, Manufacturer, Type, Platform, and IP Address. The table lists five devices, all with 'Active' status and 'production' tenant, manufactured by 'Cisco Systems' and of type 'NCS5501' on 'iosxr' platform. The bottom right corner shows 'Per Page' and 'Showing 1-5 of 5'.

<https://docs.netbox.dev/en/stable/>  
<https://github.com/netbox-community/netbox-docker>

# NetBox - Inventory

<input type="checkbox"/> Name	Status	Tenant	Site	Role	Manufacturer	Type	Platform	IP Address	
<input type="checkbox"/> popmg-pe-ncs5501-102	Active	production	POP-MG	L2	Cisco Systems	NCS5501	iosxr	192.168.246.102/24	
<input type="checkbox"/> poppr-pe-ncs5501-103	Active	production	POP-PR	PE	Cisco Systems	NCS5501	iosxr	192.168.246.103/24	
<input type="checkbox"/> poprj-pe-ncs5501-94	Active	production	POP-RJ	PE	Cisco Systems	NCS5501	iosxr	192.168.246.94/24	
<input type="checkbox"/> popsp-l2-ncs5501-96	Active	production	POP-SP	L2	Cisco Systems	NCS5501	iosxr	192.168.246.96/24	
<input type="checkbox"/> popsp-pe-ncs5501-95	Active	production	POP-SP	PE	Cisco Systems	NCS5501	iosxr	192.168.246.95/24	

Per Page ▲

Showing 1-5 of 5

# Nornir - Task 3: Napalm + Netbox

```
6 nr = InitNornir(  
7     runner={"plugin": "threaded", "options": {"num_workers": 20}},  
8     inventory={  
9         "plugin": "NetBoxInventory2",  
10        "options": {  
11            "nb_url": os.getenv("NETBOX_URL"),  
12            "nb_token": os.getenv("NETBOX_TOKEN"),  
13            "filter_parameters": {"tenant": "production",  
14                                  "role": "l2",  
15                                  "region": "mg",  
16                                  "status": "active",  
17                                  "platform": "iosxr"},  
18            "ssl_verify": False}  
19        })  
20  
21 nr.inventory.defaults.username = os.getenv("USER")  
22 nr.inventory.defaults.password = os.getenv("PASSWORD")  
23 nr.inventory.defaults.port = os.getenv("SSH_PORT")  
24  
25 print_result(nr.run(task=napalm_get, getters=["get_interfaces"]), vars=["result"])  
26
```

# Nornir - Task 3: Napalm + NetBox - results

```
result = {
  'get_interfaces': {
    'Loopback0': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '', 'last_flapped': -1.0, 'mtu': 1500, 'speed': 0.0},
    'Loopback1000': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '', 'last_flapped': -1.0, 'mtu': 1500, 'speed': 0.0},
    'Null0': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '', 'last_flapped': -1.0, 'mtu': 1500, 'speed': 0.0},
    'MgmtEth0/RP0/CPU0/0': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '50:03:00:03:00:00', 'last_flapped': -1.0, 'mtu': 1514, 'speed': 0.0},
    'GigabitEthernet0/0/0/0': {
      'is_enabled': False,
      'is_up': False,
      'description': '',
      'mac_address': '50:03:00:03:00:03',
      'last_flapped': -1.0,
      'mtu': 9198,
      'speed': 1000.0
    },
    'GigabitEthernet0/0/0/1': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '50:03:00:03:00:04', 'last_flapped': -1.0, 'mtu': 1514, 'speed': 1000.0},
    'GigabitEthernet0/0/0/2': {'is_enabled': True, 'is_up': True, 'description': '', 'mac_address': '50:03:00:03:00:05', 'last_flapped': -1.0, 'mtu': 1514, 'speed': 1000.0}
  }
}
```



# NetBox – Source of Truth

Devices > POP-PR

dcim.device:4

## poppr-pe-ncs5501-103

+ Add Components ▾

Bookmark

Clone

Edit

Delete

Created 2023-09-11 13:11 · Updated 1 hour, 38 minutes ago

Device

Interfaces **5**

Config Context

Render Config

Contacts

Journal

Changelog

Quick search



Configure Table

<input type="checkbox"/> Name ▾	× Type	Enabled	MTU	IP Addresses	MAC address	Description	Cable	
<input type="checkbox"/> GigabitEthernet0/0/0/0	SFP (1GE)	✓	1514		50:03:00:05:00:03	—	#14	<input type="checkbox"/>
<input type="checkbox"/> GigabitEthernet0/0/0/1	SFP (1GE)	✓	1514		50:03:00:05:00:04	—	#16	<input type="checkbox"/>
<input type="checkbox"/> GigabitEthernet0/0/0/2	SFP (1GE)	✓	1514		50:03:00:05:00:05	—	#15	<input type="checkbox"/>
<input type="checkbox"/> GigabitEthernet0/0/0/3	SFP (1GE)	✗	1514		50:03:00:05:00:06	—	#17	<input type="checkbox"/>
<input type="checkbox"/> Mg0/RP0/CPU0/0	1000BASE-T (1GE)	✓	1514	192.168.246.103/24	50:03:00:05:00:00	—	—	<input type="checkbox"/>

Per Page ▾

Showing 1-5 of 5

# Informações do NetBox – GraphQL API

```
{
  "data": {
    "interface_list": [
      {
        "name": "GigabitEthernet0/0/0/0",
        "enabled": true,
        "mac_address": "50:03:00:05:00:03",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "GigabitEthernet0/0/0/1",
        "enabled": true,
        "mac_address": "50:03:00:05:00:04",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "GigabitEthernet0/0/0/2",
        "enabled": true,
        "mac_address": "50:03:00:05:00:05",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "GigabitEthernet0/0/0/3",
        "enabled": false,
        "mac_address": "50:03:00:05:00:06",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "Mg0/RP0/CPU0/0",
        "enabled": true,
        "mac_address": "50:03:00:05:00:00",
        "description": "",
        "mtu": 1514
      }
    ]
  }
}
```

# Informações do Device:

```
name = 'napalm_get'
result = {
  'get_interfaces': {
    'GigabitEthernet0/0/0/0': {'is_enabled': True, 'is_up': True, 'mac_address': '50:03:00:05:00:03', 'description': '', 'speed': 1000.0, 'last_flapped': -1.0, 'mtu': 1514},
    'GigabitEthernet0/0/0/1': {'is_enabled': True, 'is_up': True, 'mac_address': '50:03:00:05:00:04', 'description': '', 'speed': 1000.0, 'last_flapped': -1.0, 'mtu': 1514},
    'GigabitEthernet0/0/0/2': {'is_enabled': True, 'is_up': True, 'mac_address': '50:03:00:05:00:05', 'description': '', 'speed': 1000.0, 'last_flapped': -1.0, 'mtu': 1514},
    'GigabitEthernet0/0/0/3': {
      'is_enabled': False,
      'is_up': False,
      'mac_address': '50:03:00:05:00:06',
      'description': '',
      'speed': 1000.0,
      'last_flapped': -1.0,
      'mtu': 1514
    },
    'Loopback0': {'is_enabled': True, 'is_up': True, 'mac_address': '', 'description': '', 'speed': 0.0, 'last_flapped': -1.0, 'mtu': 1500},
    'MgmtEth0/RP0/CPU0/0': {'is_enabled': True, 'is_up': True, 'mac_address': '50:03:00:05:00:00', 'description': '', 'speed': 0.0, 'last_flapped': -1.0, 'mtu': 1514},
    'Null0': {'is_enabled': True, 'is_up': True, 'mac_address': '', 'description': '', 'speed': 0.0, 'last_flapped': -1.0, 'mtu': 1500}
  }
}
```

# Nornir - Task 5: Napalm + NetBox + Netmiko

```
18 def nornir_netmiko_configure(task):
19     try:
20         data_device = task.run(task=napalm_get, getters=["get_interfaces"])
21         data_device = data_device.result['get_interfaces']
22
23         url_netbox = str(os.getenv("NETBOX_URL"))+"/api/dcim/interfaces/?device="+str(task.host)
24         response_device = requests.request("GET", url_netbox, headers=headers, verify=False)
25
26         for interface_netbox in response_device.json()['results']:
27             configurations=["interface "+str(interface_netbox['name'])]
28             if data_device[interface_netbox['name']]['is_enabled'] != interface_netbox['enabled']:
29                 if interface_netbox['enabled'] == True:
30                     configurations.append("no shutdown")
31                 if interface_netbox['enabled'] == False:
32                     configurations.append("shutdown")
33             if data_device[interface_netbox['name']]['description'] != interface_netbox['description']:
34                 if interface_netbox['description'] == "":
35                     configurations.append("no description")
36                 else:
37                     configurations.append("description "+str(interface_netbox['description']))
38             if data_device[interface_netbox['name']]['mtu'] != interface_netbox['mtu']:
39                 configurations.append("mtu "+str(interface_netbox['mtu']))
40             if len(configurations) > 1:
41                 command = task.run(netmiko_send_config, config_commands=configurations)
42                 print_result(command)
43                 commit = task.run(netmiko_commit)
44                 print_result(commit)
45     except Exception as err:
46         print(err)
47
```



# NetBox – Alterações

Devices > POP-PR

dcim.device:4

## poppr-pe-ncs5501-103

Created 2023-09-11 13:11 · Updated 3 hours, 10 minutes ago

- + Add Components
- Bookmark
- Clone
- Edit
- Delete

- Device
- Interfaces **5**
- Config Context
- Render Config
- Contacts
- Journal
- Changelog

Quick search



Configure Table

<input type="checkbox"/> Name	Type	Enabled	MTU	IP Addresses	MAC address	Description	Cable	
<input type="checkbox"/> GigabitEthernet0/0/0/0	SFP (1GE)	✗	1600		50:03:00:05:00:03	CUSTOMER_A	#14	+ 🔗 🗑️ 📄 ✎
<input type="checkbox"/> GigabitEthernet0/0/0/1	SFP (1GE)	✗	1600		50:03:00:05:00:04	UPLINK_SWX	#16	+ 🔗 🗑️ 📄 ✎
<input type="checkbox"/> GigabitEthernet0/0/0/2	SFP (1GE)	✗	1600		50:03:00:05:00:05	UPLINK_SWY	#15	+ 🔗 🗑️ 📄 ✎
<input type="checkbox"/> GigabitEthernet0/0/0/3	SFP (1GE)	✓	1600		50:03:00:05:00:06	CUSTOMER_B	#17	+ 🔗 🗑️ 📄 ✎
<input type="checkbox"/> Mg0/RP0/CPU0/0	1000BASE-T (1GE)	✓	1514	192.168.246.103/24	50:03:00:05:00:00	—	—	+ 🔗 🗑️ 📄 ✎

Per Page

Showing 1-5 of 5



# Nornir - Task 5 – results (Configuração Automatizada)

netmiko\_send\_config

```
configure terminal
```

```
Tue Nov 28 19:38:32.431 UTC
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config)#interface GigabitEthernet0/0/0/0
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#shutdown
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#description CUSTOMER_A
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#mtu 1600
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

netmiko\_commit

```
commit
```

```
Tue Nov 28 19:38:33.290 UTC
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

netmiko\_send\_config

```
configure terminal
```

```
Tue Nov 28 19:25:18.936 UTC
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config)#interface GigabitEthernet0/0/0/1
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#shutdown
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#description UPLINK_SWX
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#mtu 1600
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

netmiko\_commit

```
commit
```

```
Tue Nov 28 19:38:35.126 UTC
```

```
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

# Nornir - Task 5 – results (Configuração Automatizada)

```
netmiko_send_config
configure terminal
Tue Nov 28 19:25:20.888 UTC
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config)#interface GigabitEthernet0/0/0/2
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#shutdown
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#description UPLINK_SWY
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#mtu 1600
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

```
netmiko_commit
commit
Tue Nov 28 19:38:36.955 UTC
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

```
netmiko_send_config
configure terminal
Tue Nov 28 19:38:38.051 UTC
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config)#interface GigabitEthernet0/0/0/3
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#no shutdown
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#description CUSTOMER_B
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#mtu 1600
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

```
netmiko_commit
commit
Tue Nov 28 19:38:38.913 UTC
RP/0/RP0/CPU0:poppr-p-ncs5501-103(config-if)#
```

# Informações do Device (após alterações):

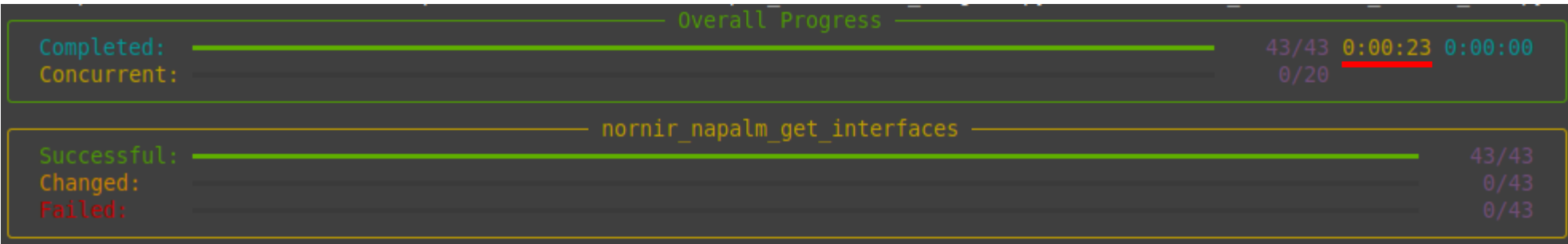
```
name = 'napalm_get'
result = {
  'get_interfaces': {
    'GigabitEthernet0/0/0/0': {
      'is_enabled': False,
      'is_up': False,
      'mac_address': '50:03:00:05:00:03',
      'description': 'CUSTOMER_A',
      'speed': 1000.0,
      'last_flapped': -1.0,
      'mtu': 1600
    },
    'GigabitEthernet0/0/0/1': {
      'is_enabled': False,
      'is_up': False,
      'mac_address': '50:03:00:05:00:04',
      'description': 'UPLINK_SWX',
      'speed': 1000.0,
      'last_flapped': -1.0,
      'mtu': 1600
    },
    'GigabitEthernet0/0/0/2': {
      'is_enabled': False,
      'is_up': False,
      'mac_address': '50:03:00:05:00:05',
      'description': 'UPLINK_SWY',
      'speed': 1000.0,
      'last_flapped': -1.0,
      'mtu': 1600
    },
    'GigabitEthernet0/0/0/3': {'is_enabled': True, 'is_up': True, 'mac_address': '50:03:00:05:00:06', 'description': 'CUSTOMER_B', 'speed': 1000.0, 'last_flapped': -1.0, 'mtu': 1600},
    'Loopback0': {'is_enabled': True, 'is_up': True, 'mac_address': '', 'description': '', 'speed': 0.0, 'last_flapped': -1.0, 'mtu': 1500},
    'MgmtEth0/RP0/CPU0/0': {'is_enabled': True, 'is_up': True, 'mac_address': '50:03:00:05:00:00', 'description': '', 'speed': 0.0, 'last_flapped': -1.0, 'mtu': 1514},
    'Null0': {'is_enabled': True, 'is_up': True, 'mac_address': '', 'description': '', 'speed': 0.0, 'last_flapped': -1.0, 'mtu': 1500}
  }
}
```

# Informações do NetBox – GraphQL API (Antes x Depois)

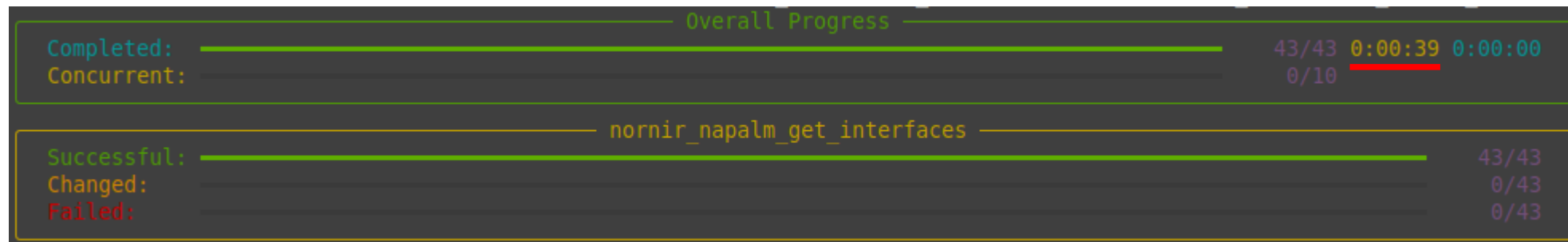
```
{
  "data": {
    "interface_list": [
      {
        "name": "GigabitEthernet0/0/0/0",
        "enabled": true,
        "mac_address": "50:03:00:05:00:03",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "GigabitEthernet0/0/0/1",
        "enabled": true,
        "mac_address": "50:03:00:05:00:04",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "GigabitEthernet0/0/0/2",
        "enabled": true,
        "mac_address": "50:03:00:05:00:05",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "GigabitEthernet0/0/0/3",
        "enabled": false,
        "mac_address": "50:03:00:05:00:06",
        "description": "",
        "mtu": 1514
      },
      {
        "name": "Mg0/RP0/CPU0/0",
        "enabled": true,
        "mac_address": "50:03:00:05:00:00",
        "description": "",
        "mtu": 1514
      }
    ]
  }
}
```

```
{
  "data": {
    "interface_list": [
      {
        "name": "GigabitEthernet0/0/0/0",
        "enabled": false,
        "mac_address": "50:03:00:05:00:03",
        "description": "CUSTOMER_A",
        "mtu": 1600
      },
      {
        "name": "GigabitEthernet0/0/0/1",
        "enabled": false,
        "mac_address": "50:03:00:05:00:04",
        "description": "UPLINK_SWX",
        "mtu": 1600
      },
      {
        "name": "GigabitEthernet0/0/0/2",
        "enabled": false,
        "mac_address": "50:03:00:05:00:05",
        "description": "UPLINK_SWY",
        "mtu": 1600
      },
      {
        "name": "GigabitEthernet0/0/0/3",
        "enabled": true,
        "mac_address": "50:03:00:05:00:06",
        "description": "CUSTOMER_B",
        "mtu": 1600
      },
      {
        "name": "Mg0/RP0/CPU0/0",
        "enabled": true,
        "mac_address": "50:03:00:05:00:00",
        "description": "",
        "mtu": 1514
      }
    ]
  }
}
```

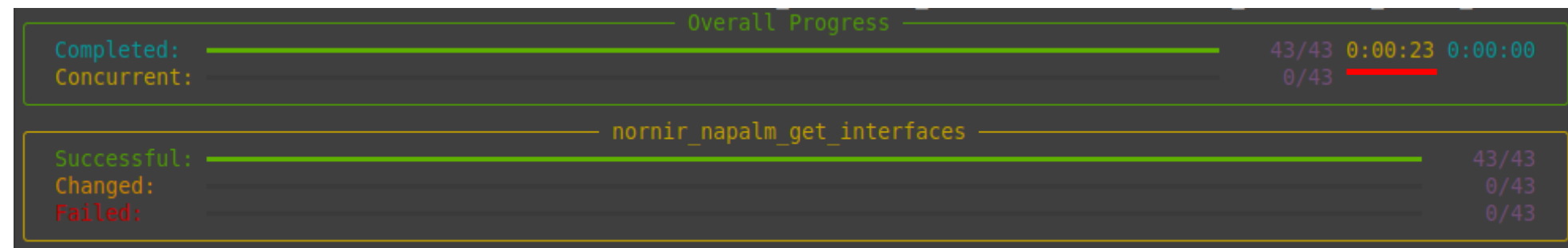
# Curiosidades: Nornir Workers



20



10



50



# CONCLUSÕES

- Nornir é um framework de automação de redes com gerenciamento de inventário que ajuda a operar vários elementos de rede;
- A estrutura multithread do Nornir permite gerenciar configurações de múltiplos devices simultaneamente;
- Ter a documentação da Rede em sistemas como o NetBox facilita o processo de automação;
- Usar o NetBox como Source of Truth aumenta as possibilidades de automação;
- Tarefas automatizadas podem ser executadas muito mais rápido do que quando operadas em redes manualmente;
- Tarefas automatizadas diminuem o número de erros de configuração e também os custos de operação;
- Tarefas automatizadas aprimoram a escalabilidade da rede;

# Repositório no GITHUB:

The screenshot shows a GitHub repository page for 'wsdoprado / ixforum17'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows a series of commits, with the most recent one being 'Create hosts.yml' by 'wsdoprado' 2 minutes ago. The commit message is 'ed451aa 2 minutes ago' and it has 16 commits in total. The file list includes README.md, hosts.yml, and seven Python files (nornir1.py through nornir7.py).

wsdoprado / ixforum17 Public

<> Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags Go to file Code

**wsdoprado** Create hosts.yml ed451aa 2 minutes ago 🕒 16 commits

README.md	Update README.md	6 minutes ago
hosts.yml	Create hosts.yml	2 minutes ago
nornir1.py	Create nornir1.py	1 hour ago
nornir2.py	Create nornir2.py	1 hour ago
nornir3.py	Create nornir3.py	1 hour ago
nornir4.py	Create nornir4.py	1 hour ago
nornir5.py	Create nornir5.py	1 hour ago
nornir6.py	Create nornir6.py	1 hour ago
nornir7.py	Create nornir7.py	1 hour ago

<https://github.com/wsdoprado/ixforum17/>

# Obrigado

[www.ix.br](http://www.ix.br)

@wprado@nic.br

7 de dezembro de 2023

nic.br egi.br

[www.nic.br](http://www.nic.br) | [www.cgi.br](http://www.cgi.br)