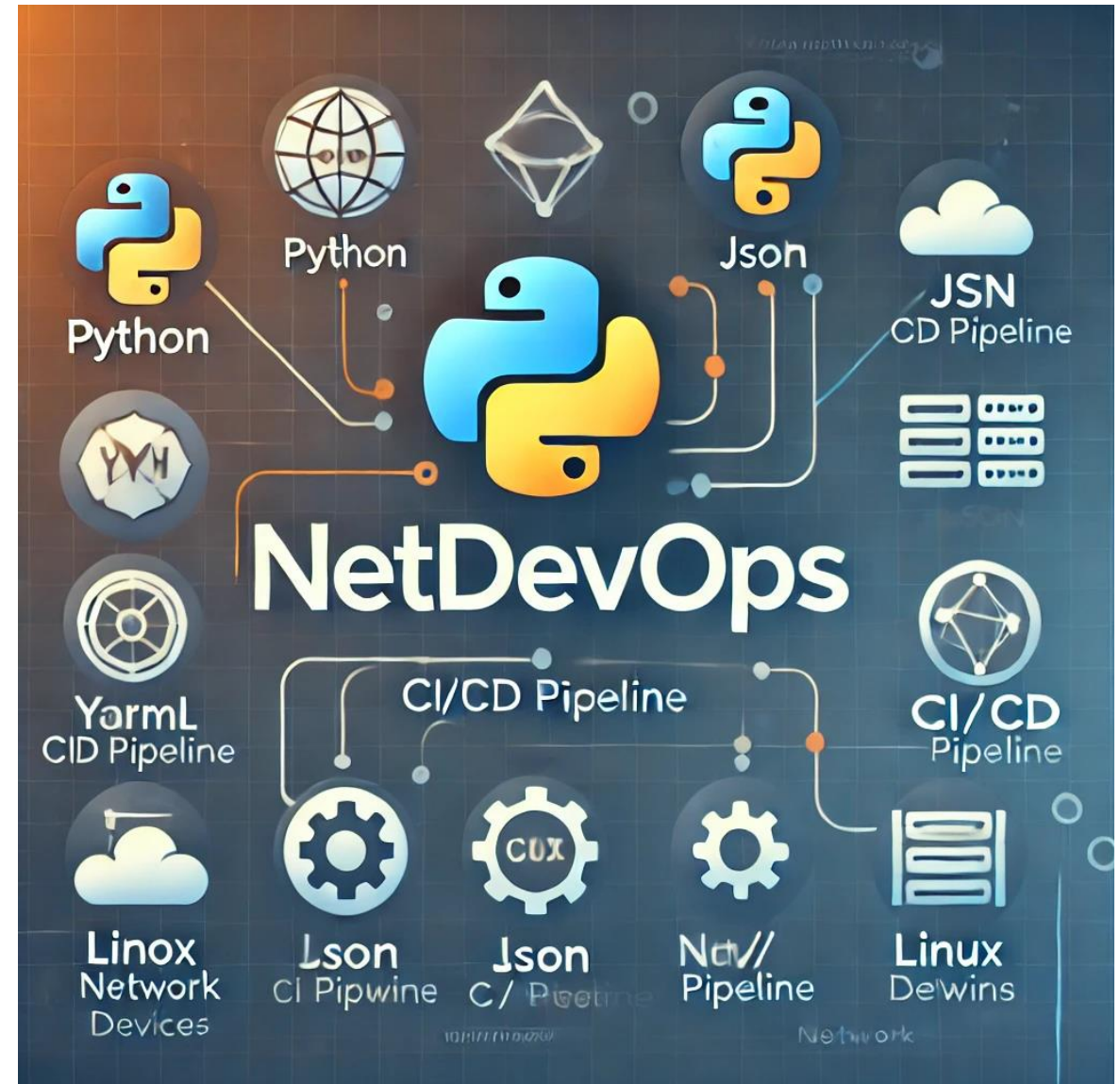


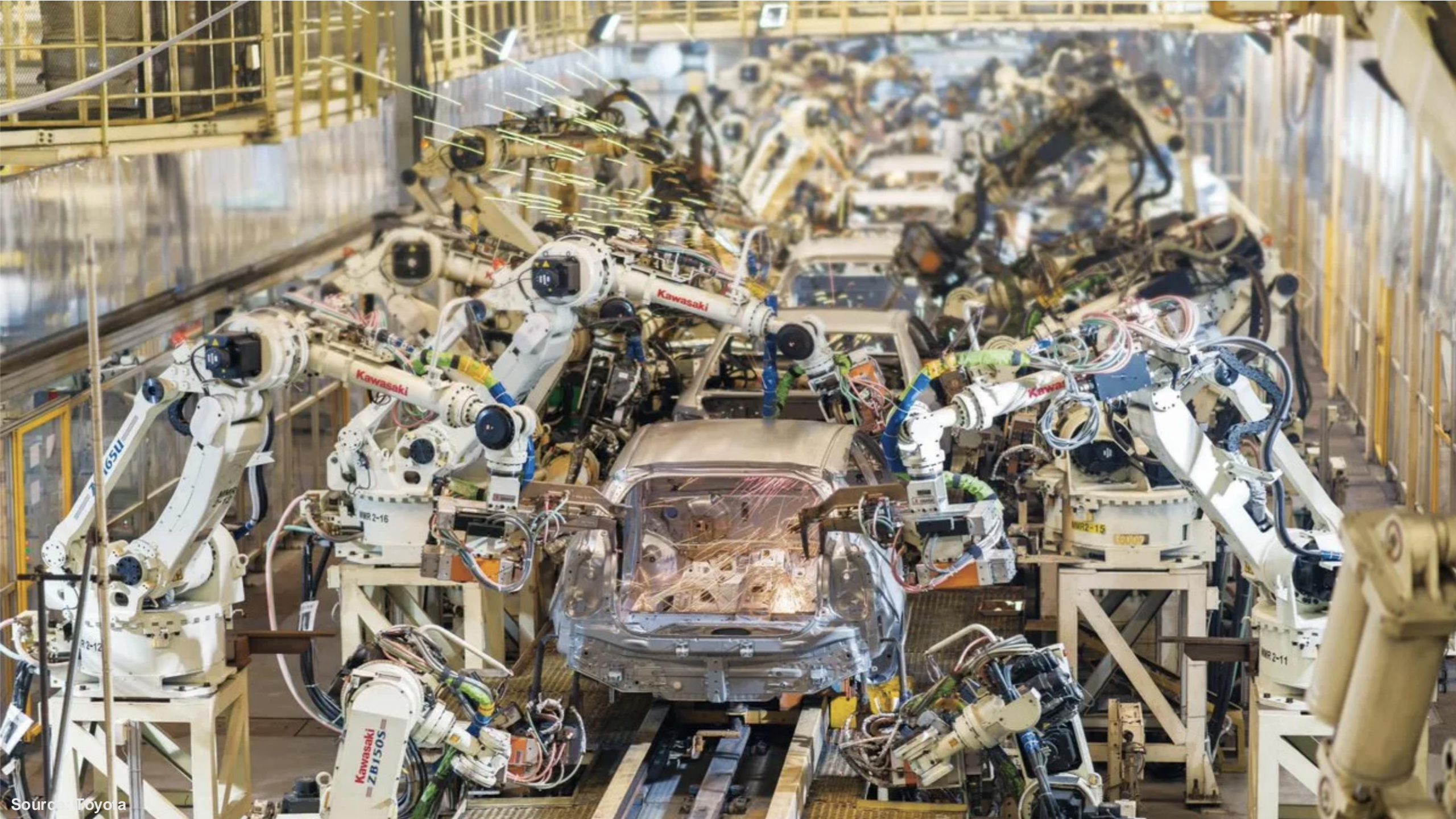
# Remessa 2.0: uma perspectiva NetDevOps

Humberto Galiza

Network Development Engineer



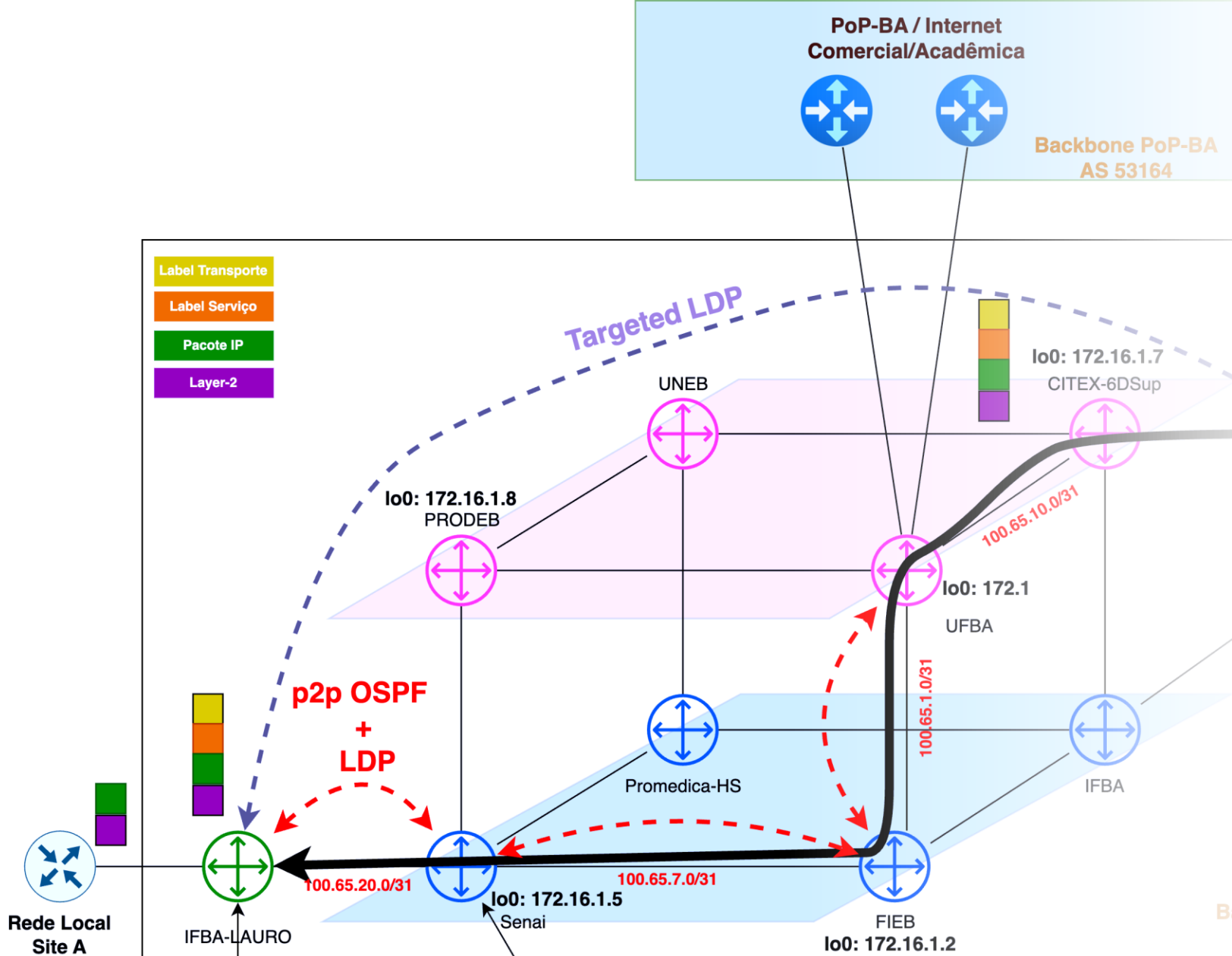




# Redes modernas não tem janelas de manutenção

---





## A Remessa 2.0 é um mar de desafios e oportunidades

- Novos conceitos arquiteturais com impacto no design físico e lógico
- Novo hardware / fabricante
- Novas tecnologias
- Novo modelo operacional
- Novas oportunidades de serviços

**FIB**  
 destination: 172.16.1.7  
 IP next-hop: 100.65.20.1  
 OutIF: XgigabitEthernet 0/0/48  
 Oper: Push Label 32 (LDP)

**LFIB**  
 destination: Label 32  
 OutIF: XgigabitEthernet 0/0/44  
 Oper: Swap Label 32 with 95 (LDP)

# Alguns desafios de implementação da Remessa 2.0

Como **padronizar** as nossas configurações e ter certeza que estão aderentes à intenção do design?

Por que precisamos continuar configurando os dispositivos de rede da mesma forma que temos feito **nos últimos 15 anos**?

Não é estranho que ainda tenhamos que fazer login em **cada caixa individual** e usar comandos CLI para executar quaisquer alterações ou verificações?

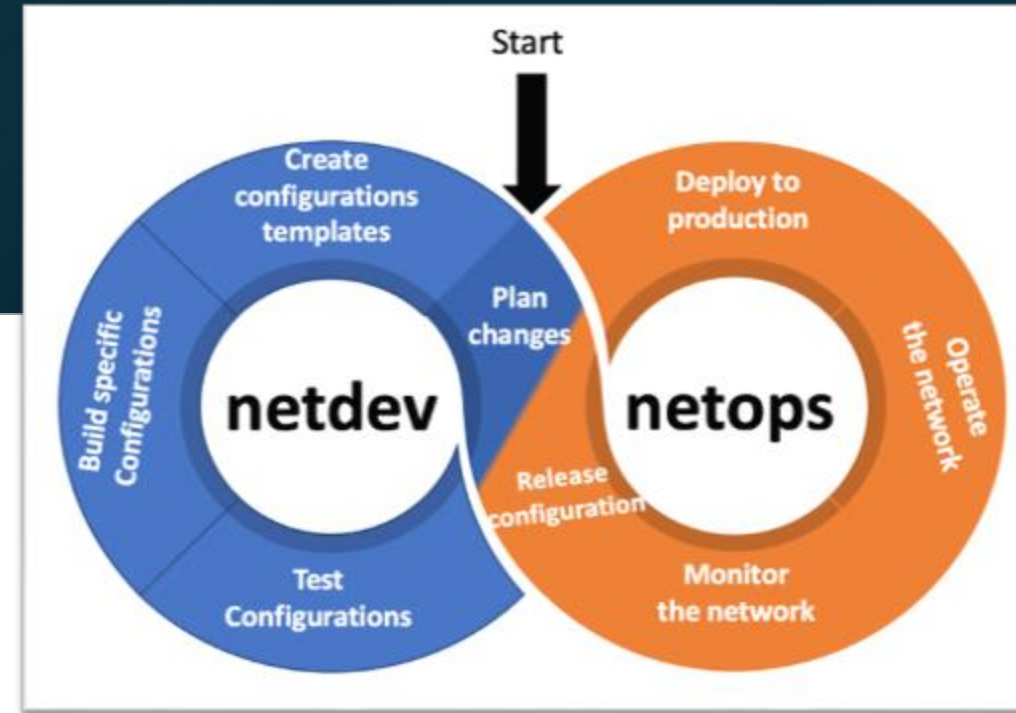
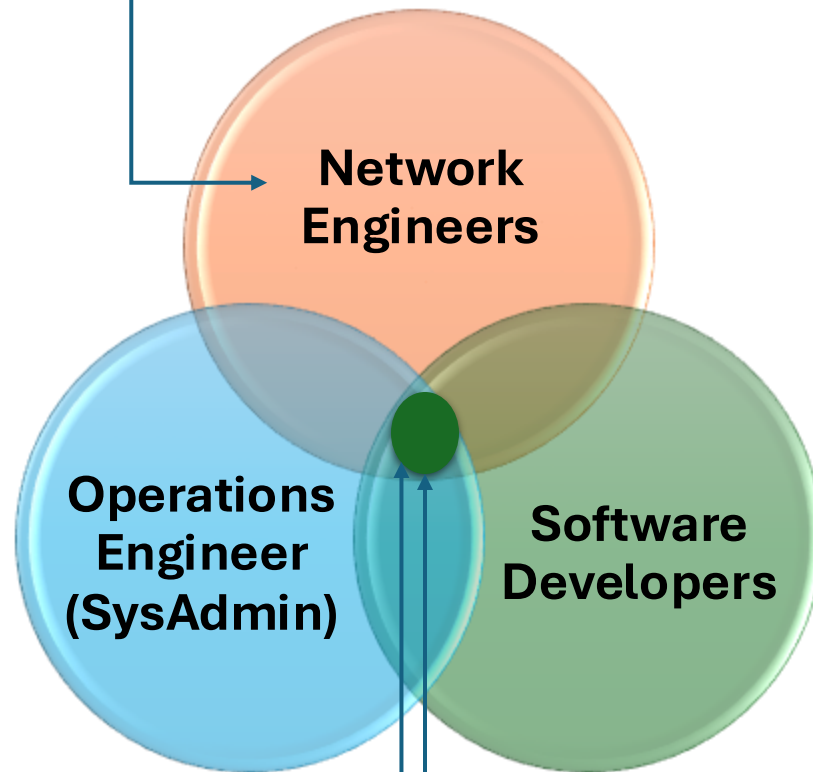
Como posso ter certeza de que minha mudança "**simple**s" não quebrará toda a rede? Posso testá-la com segurança antes da produção?

Como ter uma abordagem mais "**proativa**" ao invés de "**reativa**" no monitoramento dos equipamentos e enlaces, e bem como no envio de alertas?

É possível migrar os clientes da Remessa 1.0 para a 2.0 **sem causar downtime**?

(...)

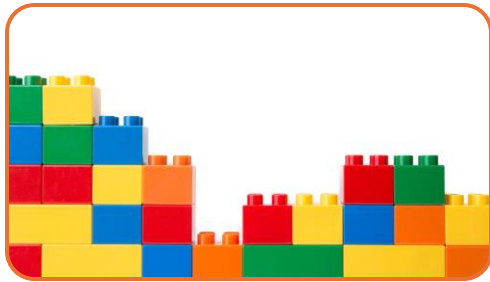
# O que é NetDevOps?



**DevOps** é uma estratégia de desenvolvimento de software que preenche a lacuna entre as áreas de Desenvolvimento (Dev) e times de Operações (Ops), possibilitando a construção, testes, e implantação de softwares de maneira mais rápida e confiável.

**NetDevOps** traz a cultura, métodos técnicos, estratégias, e melhores práticas do mundo **DevOps** para a área de redes (*networking*).

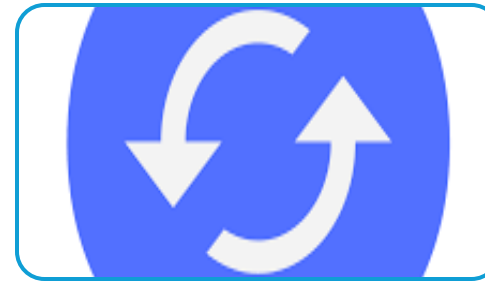
# Princípios do NetDevOps



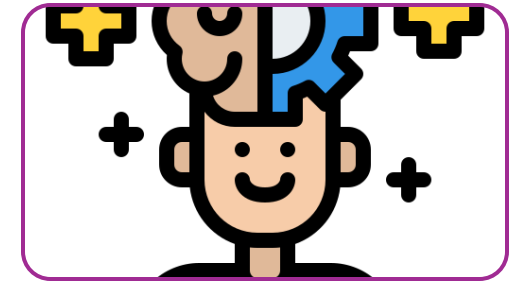
Modularização da  
Rede (*Network  
Segmentation*)



Automação e  
Infraestrutura  
como Código  
(*Network as  
Code*)



Desenvolvimento  
contínuo (CI/CD)



Mudança Cultural



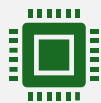
# Remessa Network Automation (RNA)

---



## Planejamento Inicial

Requisitos de rede e estratégias de automação com a definição da visão de curto (migração MPLS) e longo prazo



## Configuração automática de protocolos e serviços

Codificação de “*metadados*” com as características e parâmetros sobre a rede (*intent*)

Modelagem do serviço L2VPN



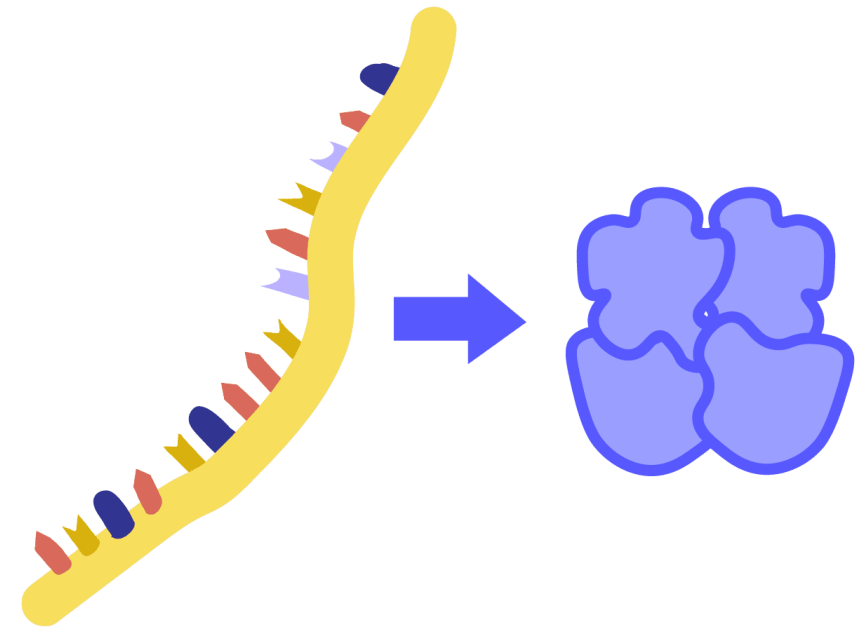
## Casos de uso prioritários

Automação do provisionamento inicial


Gerenciamento e auditoria das configurações com controle de versão

Provisionamento automático de circuitos L2VPN

Troubleshooting básico



The primary function of RNA is **to create proteins via translation**. RNA carries genetic information that is translated by ribosomes into various proteins necessary for cellular processes. mRNA, rRNA, and tRNA are the three main types of RNA involved in protein synthesis. 29 Jul 2023

 National Institutes of Health (NIH) (.gov)  
<https://www.ncbi.nlm.nih.gov/books/NBK558999>

[Biochemistry, RNA Structure - StatPearls - NCBI Bookshelf](#)

# Triângulo de fogo da automação



# Escolha de ferramentas



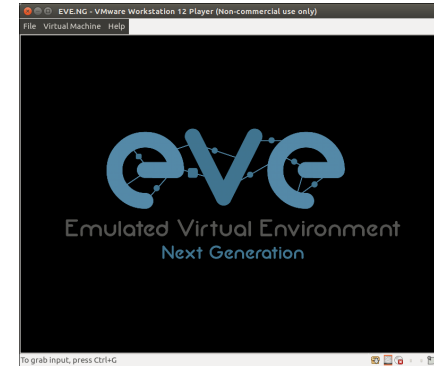
Infraestrutura como código



Orquestração e automação



Ambiente de Testes (pré-produção)



# Vantagens do Nornir



Provê uma abstração para o inventário (hosts e grupos) e execução de tarefas em paralelo



Escrito em Python – é só mais uma biblioteca Python



Facilidade de extensão e flexibilidade



Super rápido, graças ao uso de threads



Suporta diversas bibliotecas de conexão aos equipamentos (netmiko, napalm, ncclient, etc)



Pode ser integrada com frameworks Python tais como Flask, Django, pytest, etc



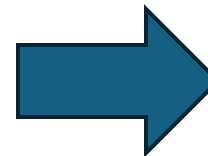
# Single Source of Truth (SSoT)

- Escrevemos um plugin para o Nornir que ingere uma planilha em Excel com a lista de sites, equipamentos, enlaces de backbone etc., (NetBox está em processo de adoção...)



equipamentos

	A	B	C	D	E				
1	hostname	loopback_v4	loopback_v6	mgmt_ip	platform				
2	sdr-ond-popba-1	172.1/32	2804:1f50/1/128	169.1/24	huawei_vrp				huawe
3	sdr-stp-fieb-1	172.2/32	2804:1f50/2/128	169.2/24	huawei_vrp				huawe
4	sdr-cbl-uneb-1	172.3/32	2804:1f50/3/128	169.3/24	huawei_vrp				huawe
5	sdr-pia-senai-1	172.4/32	2804:1f50/4/128	169.4/24	huawei_vrp				huawe
6	sdr-bar-ifba-1	172.5/32	2804:1f50/5/128	169.5/24	huawei_vrp				huawe
7	sdr-per-promed-1	172.5/32	2804:1f50/6/128	169.6/24	huawei_vrp				huawe
8	sdr-cab-prodeb-1	172.7/32	2804:1f50/7/128	169.7/24	huawei_vrp				huawe
9	sdr-2dj-eb-1	172.8/32	2804:1f50/8/128	169.8/24	huawei_vrp				huawe
10	sdr-trb-unijorge-1	172.9/32	2804:1f50/9/128	169.9/24	huawei_vrp				huawe
11	sdr-ond-cogel-1	172.10/32	2804:1f50/10/128	169.10/24	huawei_vrp				huawe
12	sdr-sus-cprm-1	172.11/32	2804:1f50/11/128	169.11/24	huawei_vrp				huawe



circuitos

A	B	C	D	E	F	G	H	I
#	layer	left	right	left_port	right_port	left_ip	right_ip	iftype
3	inner_core	sdr-cbl-uneb-1	sdr-per-promed-1	100GE0/0/4	100GE0/0/3	100.31	100.31	l3
12	inner_core	sdr-cbl-uneb-1	sdr-2dj-eb-1	100GE0/0/6	100GE0/0/3	100.31	100.31	l3
15	outer_core	sdr-cbl-uneb-1	sdr-cbl-eb-1	XGigabitEthernet0/0/24	XGigabitEthernet0/0/23	100.31	100.31	l3
20	access	sdr-aer-rnp-1	ifs-iti-ifba-1	XGigabitEthernet0/0/24	XGigabitEthernet0/0/23	100.31	100.31	l3
33	outer_core	sdr-cbl-uneb-1	sdr-imb-unifacs-1	XGigabitEthernet0/0/23	XGigabitEthernet0/0/23	100.31	100.31	l3

```
In [3]: nr.inventory.hosts["sdr-ond-popba-1"].dict()
Out[3]:
{'name': 'sdr-ond-popba-1',
 'connection_options': {},
 'groups': ['huawei_6730'],
 'data': {'loopback_v4': '172.x.x.x/32',
 'loopback_v6': '2804:1f50:x:x:x/128',
 'mgmt_ip': '169.x.x.x/24',
 'model': 'huawei_6730',
 'cidade': 'sdr',
 'bairro': 'ond',
 'site': 'popba',
 'site_description': 'Ponto de Presença da RNP na Bahia',
 'site_address': 'Sede',
 'hostname_rnp': 'ba-remessa-nc-sw-sdr-ond-popba-1',
 'hostname_antigo': 'sw-bkp-fibra',
 'platform': 'huawei_vrp',
 'data': {'os_family': 'huawei',
 'device_type': 'router',
 'model': 's6730',
 'os_version': 's6730-h_v200r022c00@spc500.cc',
 'os_patch': 's6730-h_v200r022sph220.pat',
 'vendor': 'huawei',
 'collect_commands': ['display current-configuration']},
 'interfaces': {'100GE0/0/1': {'remote': 'sdr-stp-fieb-1',
 'remote_port': '100GE0/0/1',
 'local_ip': '100.x.x.x/31',
 'remote_ip': '100.x.x.x/31',
 'srlg': 1,
 'speed': '100Gbps',
 'layer': 'inner_core',
 'iftype': 'l3',
 'lag': '',
 'remote_lag': '',
 'vlan': 0},
```

# Templates



- Jinja2 renderiza qualquer tipo de configuração
- Recebe variáveis da plataforma de automação (Nornir)
- Facilita a construção lógica de blocos (loops, condicionais, etc)
- Templates Jinja2 por funcionalidade (Baseline, interfaces, mpls, routing\_ospf, etc)

```
interface {{ primary_loopback }}
  description "CORE|Loopback-Interface|MPLS|BGP|LDP"
  ipv6 enable
  ip address {{primary_ipv4 | ip }} {{primary_ipv4 | netmask }}
  ipv6 address {{ primary_ipv6 | ip | upper }}/{{ primary_ipv6 | prefixlen_v6 }}
quit
{# BASELINE MPLS CONFIG #}
{{'#'}}
mpls lsr-id {{ primary_ipv4 }}
router id {{ primary_ipv4 }}
mpls
quit
{{'#'}}
{% if mpls_te_config.enabled %}
mpls te
{% endif %}
```

```
{# MPLS PW TEMPLATES #}
{{'#'}}
{{'#'}}
{% for dst_node in ns.hosts if dst_node["hostname"] != name %}
pw-template {{ dst_node["hostname"] }}
  peer-address {{ dst_node["loopback_v4"] | ip }}
  mtu {{ defaults.mtu_l2vc }}
  control-word
  tnl-policy LOAD-BALANCE
quit
#
{% endfor %}
{{'#'}}
{# MPLS LDP PEERS #}
{% for dst_node in ns.hosts if dst_node["hostname"] != name %}
mpls ldp remote-peer {{ dst_node["hostname"] }}
  remote-ip {{ dst_node["loopback_v4"] | ip }}
quit
{{'#'}}
{% endfor %}
```

# Metadados



Metadados são utilizados para descrever características padrão da rede, ex: métricas IGP, valores default de MTU, etc.

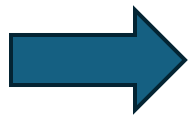


Durante a renderização dos templates esses valores são lidos e incluídos na configuração final (texto claro)

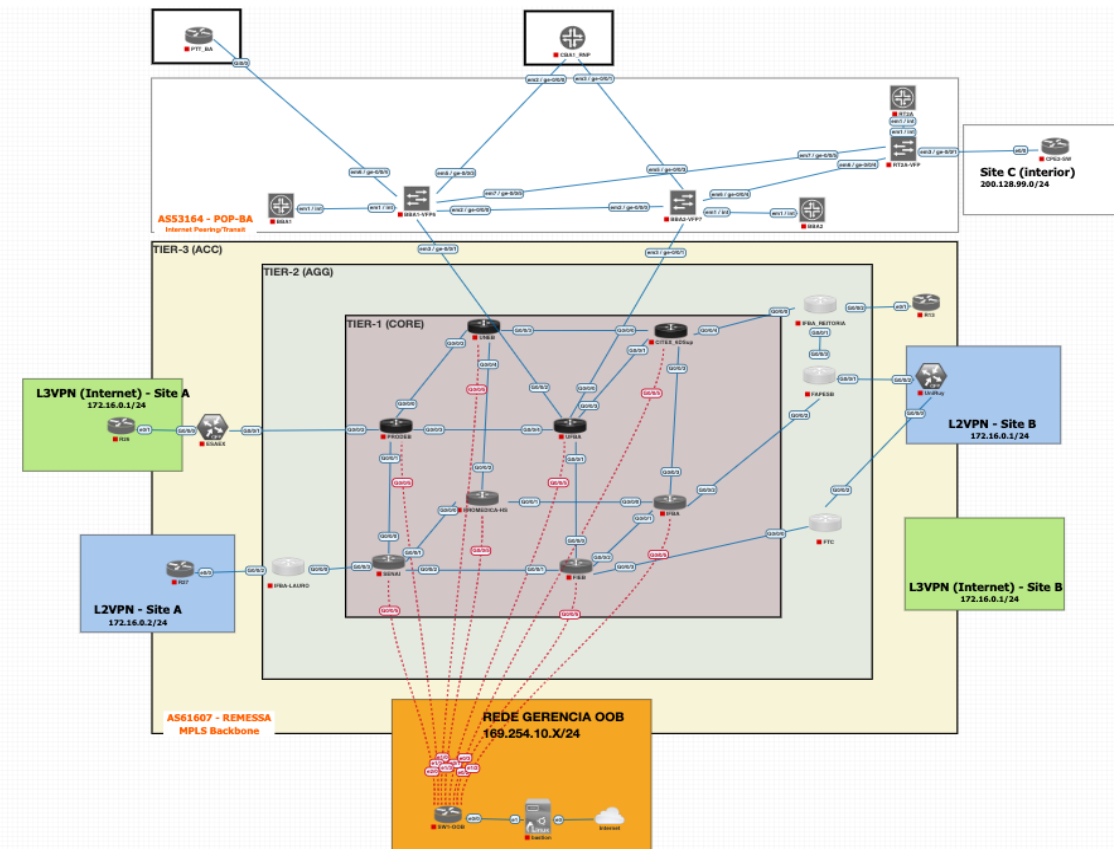
```
---
defaults:
  local_asn: 61607
  domain: remessa.net.br
  mtu_physical: 12288
  mtu_layer3: 9216
  mtu_mpls: 9100
  mtu_l2vc: 9000
routing:
  static:
    sdr-ond-popba-1:
      - net: "2804:1F50::/32"
        next_hop: NULL0
  mpls_te:
    enabled: False
  ospf:
    areas:
      - backbone:
          name: 0.0.0.0
          ref_bw: 1000000
    metrics:
      inner_core:
        400Gbps: 12
        100Gbps: 20
        40Gbps: 35
        10Gbps: 110
        1Gbps: 1010
```

```
{#Case 1: L3 link, no VLAN #}
  {% if values.iftyp == 'l3' %}
interface {{ src_port }}
  undo portswitch
  description to->{{ dst_node }}:{{ dst_port }}
  ip address {{ src_ip | ip }} {{ src_ip | netmask }}
  mtu {{ defaults.mtu_layer3 }}
  ospf authentication-mode hmac-sha256 1 cipher {{ ospf_config.areas[0].cipher }}
  ospf network-type p2p
  ospf bfd enable
  ospf cost {{ ospf_config.metrics[layer][speed] }}
  ospf enable 1 area 0.0.0.0
  ospf ldp-sync
  mpls
  mpls mtu {{ defaults.mtu_mpls }}
  {% if mpls_te_config.enabled %}
  mpls te
  mpls te bandwidth max-reservable-bandwidth 8000000
  mpls te bandwidth bc0 8000000
  mpls rsvp-te
  mpls rsvp-te bfd enable
  mpls rsvp-te hello
  {% endif %}
  mpls ldp
  quit
  {{ '#' }}
  {# End case 1: L3 link, no VLAN #}
  {% endif %}
```

# Ambiente de Testes Virtualizado



- Eve-ng para simular a topologia do mundo real
- Validação sintática e semântica da configuração gerada
- Bug/fix do design (ex: convergência, métricas IGP, etc)
- Validação dos serviços (L2VPN/L3VPN)





# Objetivo 1: geração da configuração “alvo”

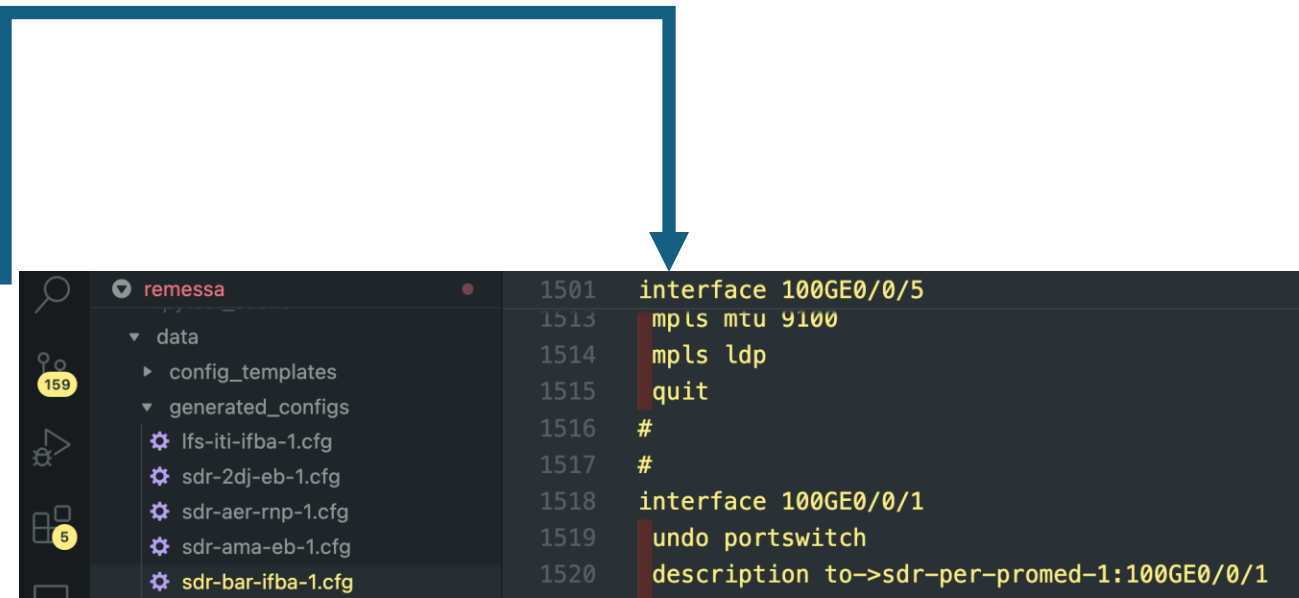
```
from nornir.core import Nornir

from remessa.utils.inventory import load_inventory
from remessa.utils.template_helper import TemplateHelper
from remessa.utils.utils import export_config_file

def generate_config(nr: Nornir) -> None:
    t = TemplateHelper()
    config_template = t.template_env.get_template("huawei/main.j2")
    for host in nr.inventory.hosts:
        device_config = config_template.render(nr.inventory.hosts[host].dict())
        export_config_file(host, device_config)

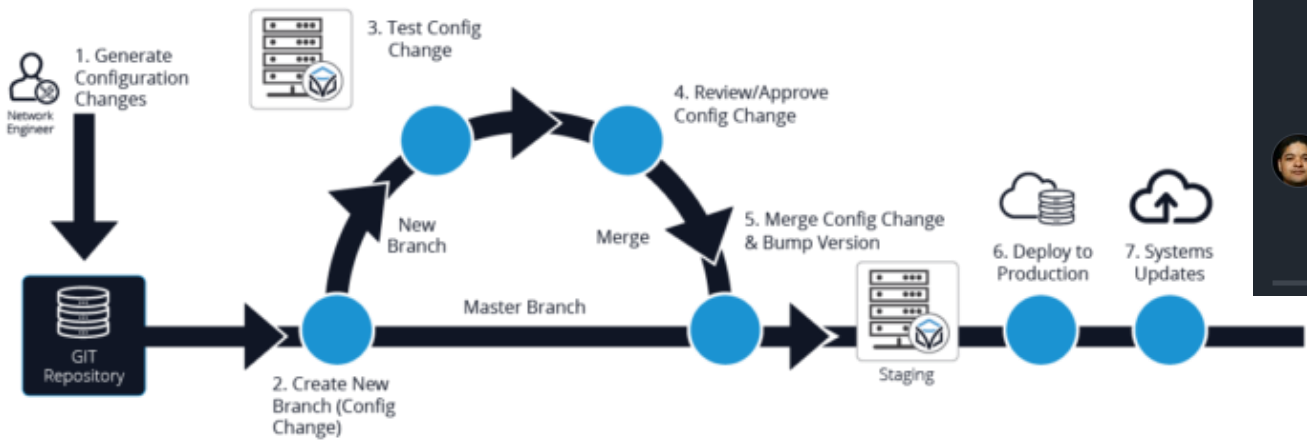
# Get inventory from nornir
devices_data = load_inventory()

# Run config generation for the device inventory
generate_config(devices_data)
```



```
remessa 1501 interface 100GE0/0/5
1513 mpls mtu 9100
1514 mpls ldp
1515 quit
1516 #
1517 #
1518 interface 100GE0/0/1
1519 undo portswitch
1520 description to->sdr-per-promed-1:100GE0/0/1
```

Com poucas linhas de código conseguimos gerar de maneira padronizada as configurações finais de todos os equipamentos presentes em nosso inventário



# Infra updates #5

Merged humbertogaliza merged 4 commits into mainline from infra-updates on Aug 6

Conversation 0 Commits 4 Checks 0 Files changed 101

adriaviriato commented on Aug 6

No description provided.

Adriana Ribeiro and others added 4 commits last month

- add server-source in ssh template f81b97b
- include user\_interface.j2 template in main.j2 bf2af64
- Create reverse diff function fb9a387
- new configurations 8e9a421

humbertogaliza approved these changes on Aug 6 [View reviewed changes](#)

humbertogaliza merged commit aa50a4a into mainline on Aug 6 [Revert](#)

Source: <https://www.italent.com/blog/company/infrastructure-as-code/building-a-netdevops-mindset/>

# Workflow

# Processo fim-a-fim

# Workflow: diff da configuração no PR

```
26 data/generated_configs/sdr-2dj-eb-1.cfg [Viewed] ...
@@ -165,6 +165,7 @@ user-interface vty 16 20
165 dsa local-key-pair create
166 #
167 ssh server port
168 ssh server cipher aes256_ctr aes128_ctr
169 ssh server hmac sha2_256
170 ssh server dh-exchange min-len 2048
+ ssh server-source all-interface
169 ssh server cipher aes256_ctr aes128_ctr
170 ssh server hmac sha2_256
171 ssh server dh-exchange min-len 2048
@@ -1725,4 +1726,27 @@ user-interface vty 16 20
1725 ssh server acl 3998
1726 snmp-agent acl SERVERS-SNMP
1727 snmp-agent acl-ipv6 SERVERS-SNMP-v6
1728 - # END SECURITY.J2 MODULE #
+ # END SECURITY.J2 MODULE ## BEGIN USER_INTERFACE.J2 MODULE #
+ #
+ user-interface con 0
+ authentication-mode aaa
+ history-command max-size 128
+ quit
+ #
+ user-interface vty 0 4
+ authentication-mode aaa
+ history-command max-size 128
```

## Objetivo 2: auditoria e troubleshooting básico



```
from remessa.utils.inventory import load_inventory
from remessa.orchestrator.tasks.huawei.mpls import get_l2vc

nr = load_inventory()

def is_l2vc_circuit_up(vcid: int, target_device: str) -> None:
    targets = nr.filter(F(name__in=[target_device]))
    l2vc_info = targets.run(task=get_l2vc(vcid))
    print_result(l2vc_info)
```



```
def is_device_healthy(task: Task) -> Result:
    """
    1. Run Environment checks (uptime, psu, fans, etc)
    2. Run LLDP link verification (intended vs actual)
    3. Run OSPF checks for each backbone interface
    4. Run LDP checks for each backbone interface
    """
    uptime = task.run(get_system_uptime)
    psu_check = task.run(get_power_supply_status)
    fan_check = task.run(get_fan_status)
    lldp_checks = task.run(check_lldp_neighbors)
    ospf_checks = task.run(check_ospf_neighbors)
    ldp_checks = task.run(check_ldp_neighbors)

    tasks_results = [
        uptime.failed,
        psu_check.failed,
        fan_check.failed,
        lldp_checks.failed,
        ospf_checks.failed,
        ldp_checks.failed
    ]

    if all(tasks_results) is False:
        return Result(task.host, result="Device is healthy", failed=False)
    return Result(task.host, result="Device is NOT healthy", failed=True)
```

# Objetivo 3: provisionamento de circuitos L2VPN [wip]

```
services:
```

```
- name: "SENAI"
```

```
description: "L2VPN VPWS/VLL service for SENAI"
```

```
customer_id: 1001
```

```
pseudowires:
```

```
- name: "vFIEBSenai_l2vc_tkt60806"
```

```
pw_id: 28052401
```

```
left:
```

```
pe: "sdr-pia-senai-1"
```

```
ac:
```

```
type: "vlan_svi"
```

```
pe_vid: 5
```

```
interface: "XGigabitEthernet0/0/3"
```

```
right:
```

```
pe: "sdr-ond-popba-1"
```

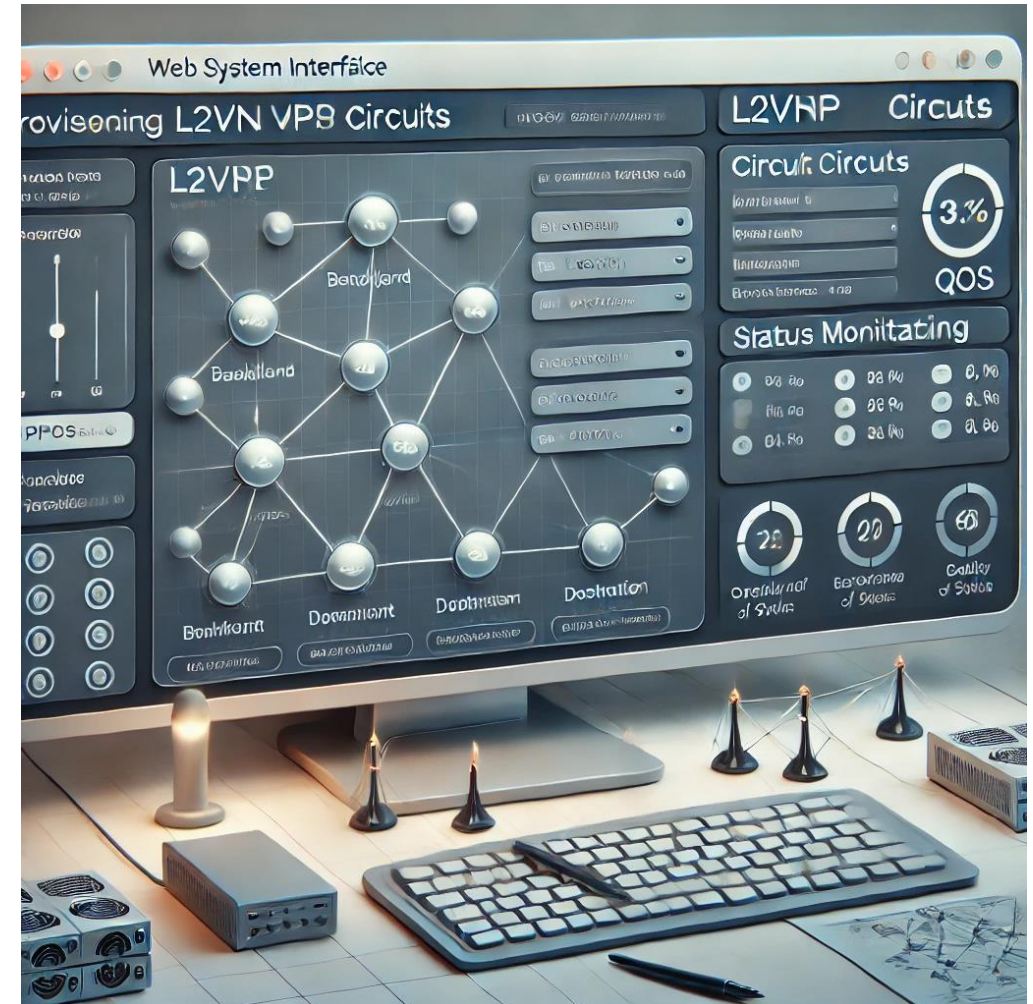
```
ac:
```

```
type: "lag_qinq"
```

```
pe_vid: 5
```

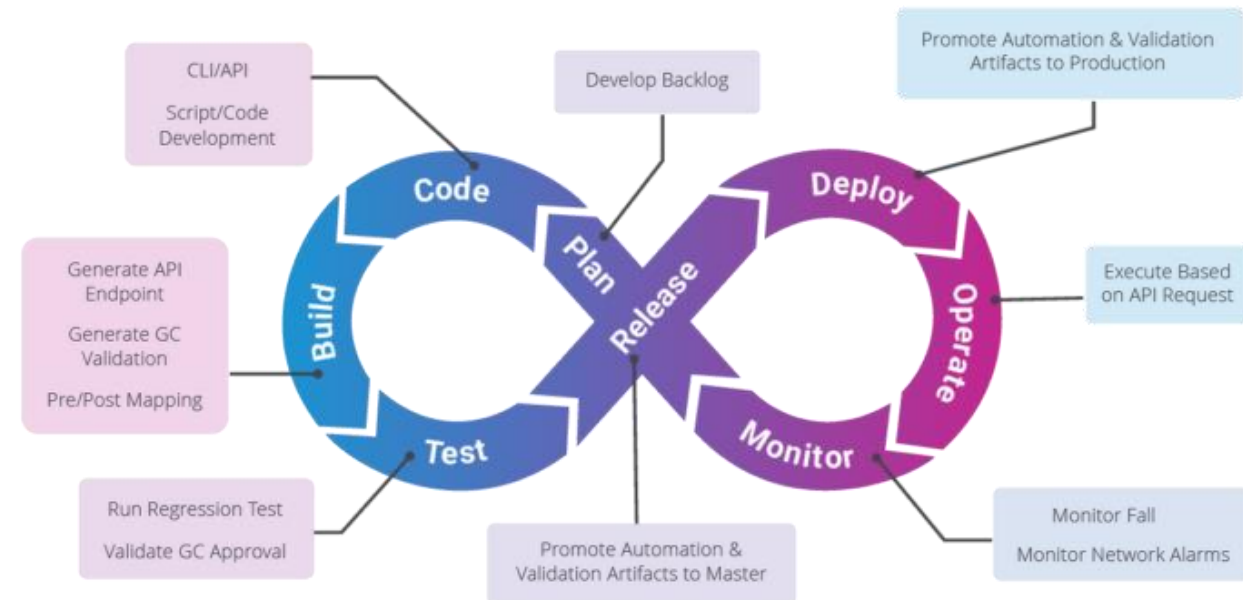
```
ce_vid: "1 to 4094"
```

```
interface: "Eth-Trunk1"
```



# Direção futura

- Shift-left cultural
  - *Education, hands on, knowledge sharing*
- Finalização da automação do ciclo de vida de configuração (geração, testes, liberação, e implantação na produção)
  - Integração com CI/CD rodando testes em ambiente virtualizado
- Auto-remediação
  - Circuitos
  - Caixas
  - Monitoramento/Alertas
  - Log parsing
- Projetos de TCC/Estágio/Mestrado envolvendo automação/netdevops



Obrigado!

Perguntas/Reclamações?

SCAN ME

