

The logo for NIC.br, featuring the text "nic.br" in a bold, sans-serif font. The ".br" is a bright green color, while "nic" is black.

Brazilian Network
Information Center

The logo for CGI.br, featuring the text "cgi.br" in a bold, sans-serif font. The ".br" is a bright green color, while "cgi" is black.

Brazilian Internet
Steering Committee

A row of six logos for Brazilian internet registries: "registro.br", "cert.br", "cetic.br", "ceptro.br", "ceweb.br", and "ix.br". Each logo has the ".br" in green and the rest in black.

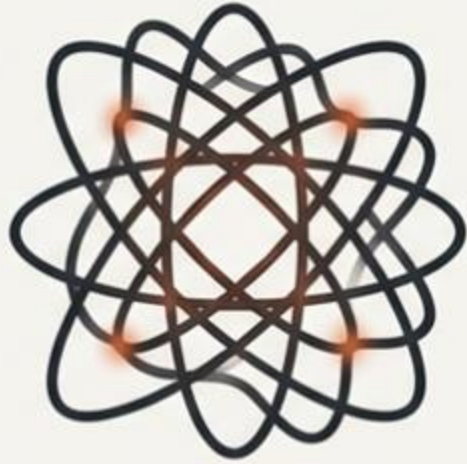
Redes como Código

Rumo ao NETDEVOPS

William Prado <wprado@nic.br>



O Desafio das Redes Tradicionais



Config Drift

Configurações não documentadas e mudanças manuais causam divergências silenciosas entre a intenção e o estado real da rede.



Falta de Rastreabilidade

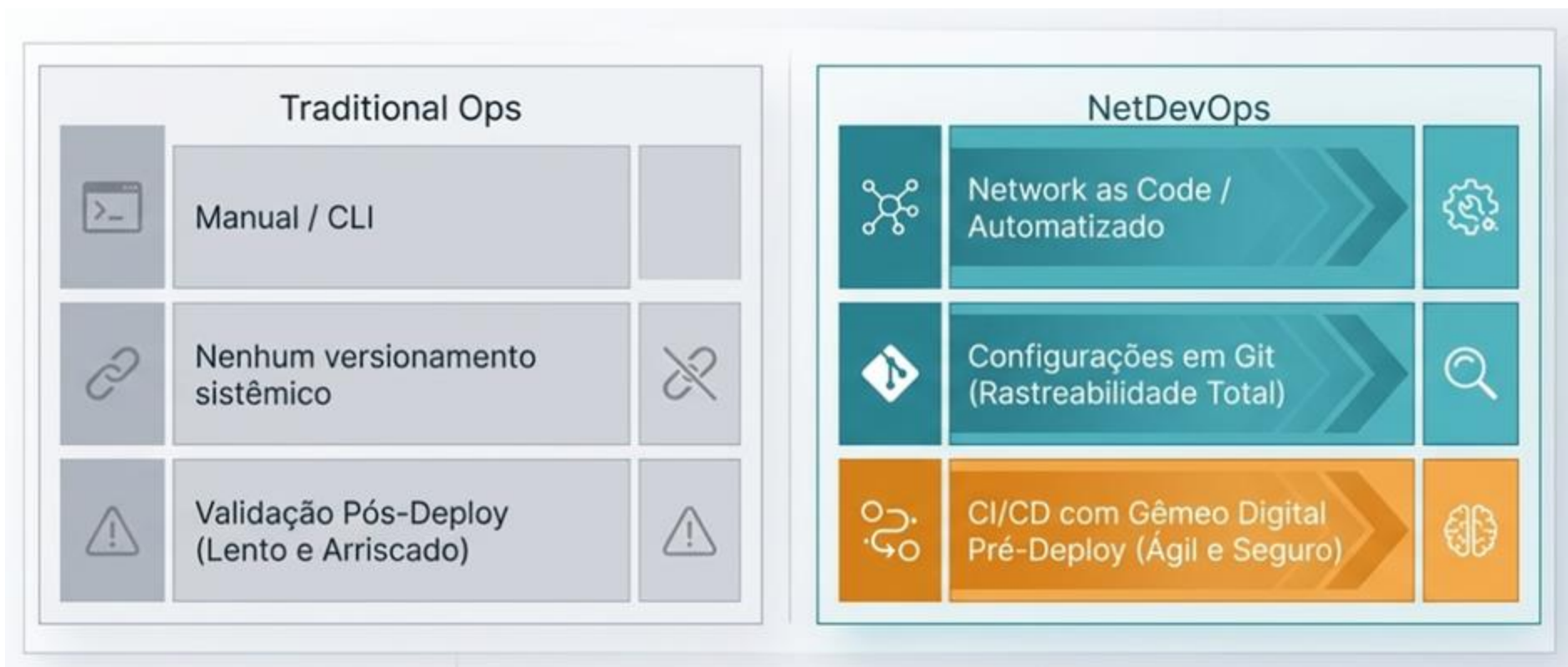
A caixa preta operacional. Sem versionamento, as perguntas 'Quem mudou?', 'Quando?' e 'Por quê?' ficam sem resposta durante incidentes.



Validação Pós-Deploy

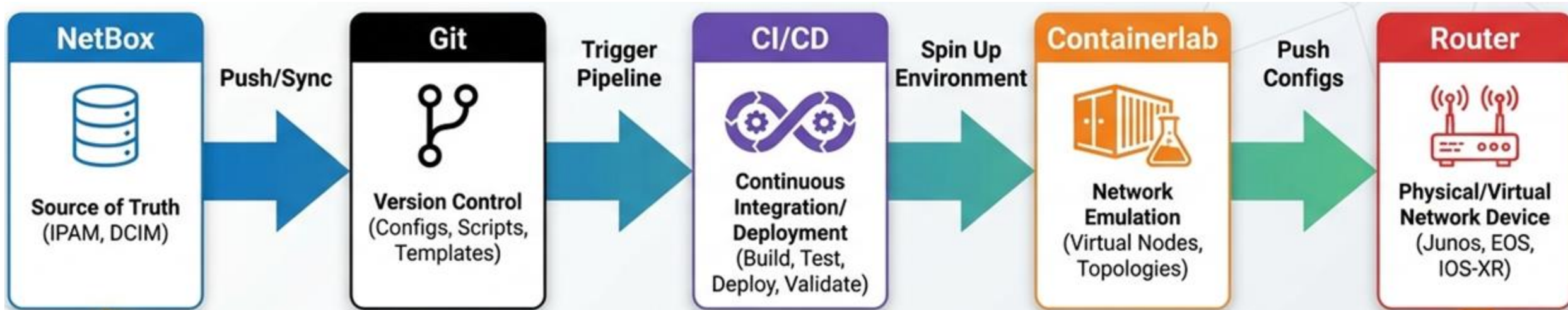
Testando na sorte. Descobrir problemas de protocolo apenas quando a rede já está em produção, aumentando o risco de downtime.

A Mudança de Paradigma: Cultura NetDevOps

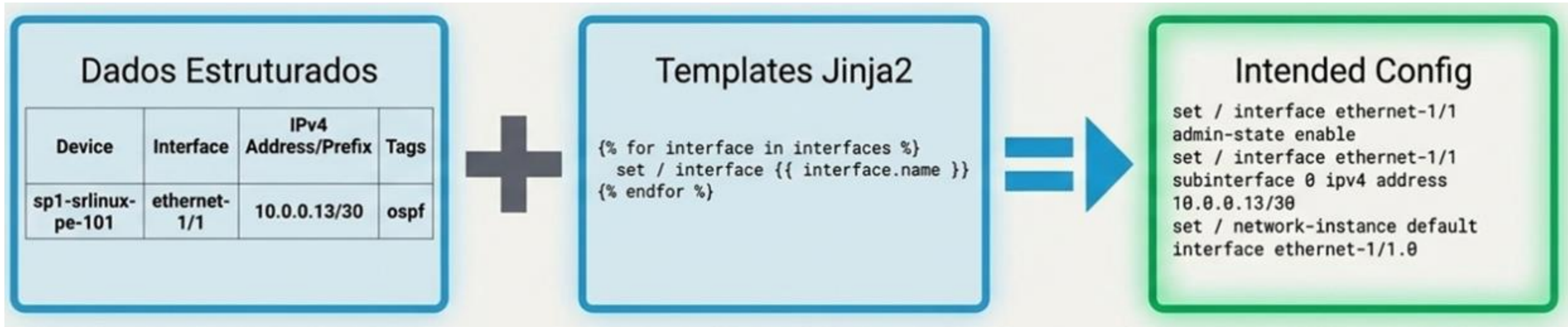


Aplica metodologias de desenvolvimento de software à infraestrutura: cada mudança é gerada por templates, testada e validada antes de tocar em um roteador.

Visão Geral do Pipeline End-to-End



Fase 1: Network Source of Truth (NSoT)



Script Python consolida dados e templates, gerando o .cfg e realizando Git Push.

Fase 1: Templates jinja + Config Context

53 lines | 2.5 KiB | Django/Jinja

Raw

Permalink

Blame

History

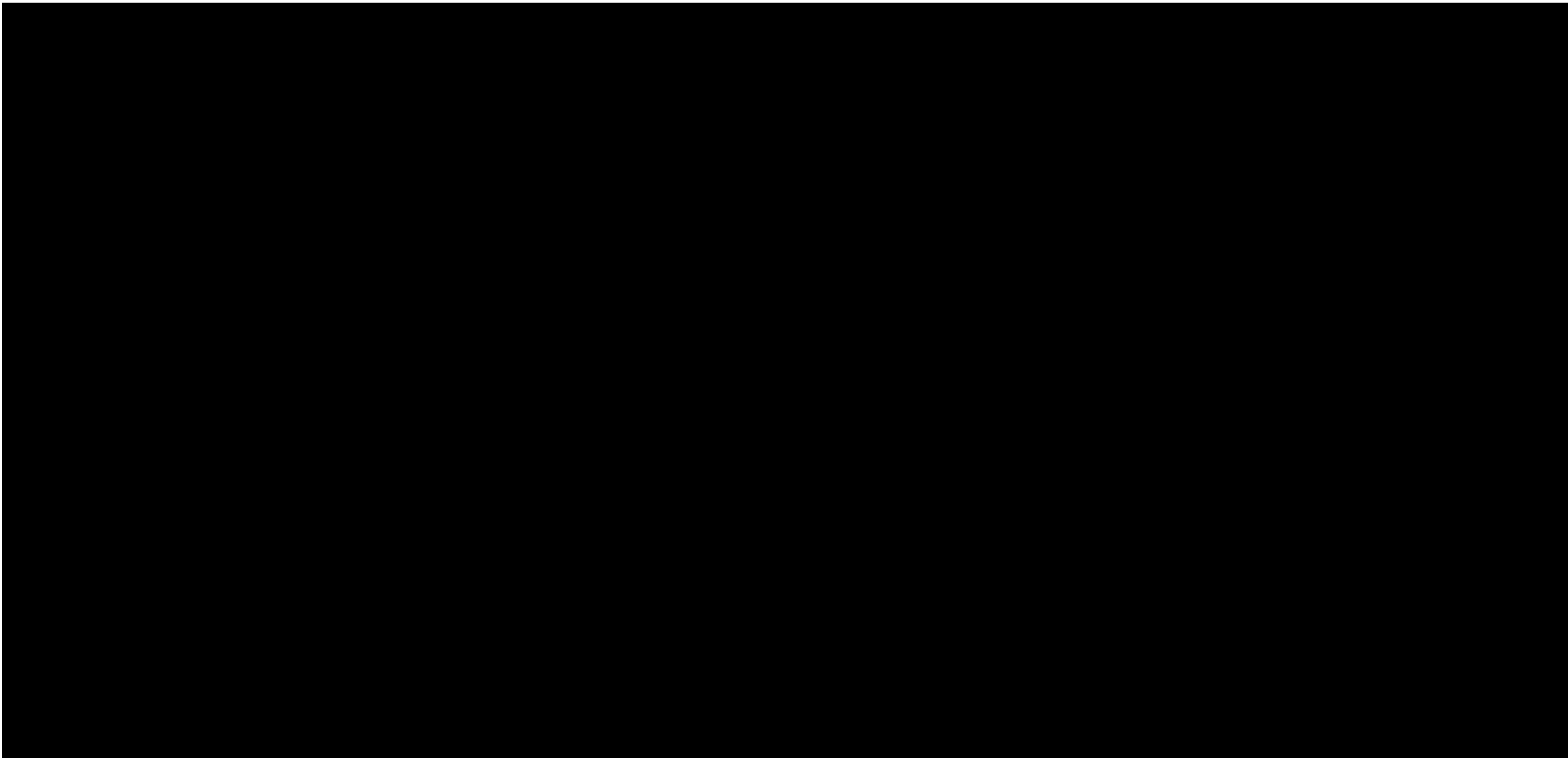


```
1 set / network-instance default type default
2
3 {%- for iface in device.interfaces.all() %}
4 {%- set parts = iface.name.split('.') %}
5 {%- if iface.name == "lo0.0" %}
6 set / interface {{ parts[0] }} admin-state {{ 'enable' if iface.enabled else 'disable' }}
7
8 {%- for ip in iface.ip_addresses.all() %}
9 set / interface {{ parts[0] }} subinterface {{ parts[1] }} ipv4 address {{ ip.address }}
10
11 {%- endfor %}
12 set / interface {{ parts[0] }} subinterface {{ parts[1] }} ipv4 admin-state {{ 'enable' if iface.enabled else 'disable' }}
13 set / network-instance default interface {{ iface.name }}
14
15 {%- endif %}
16 {%- endfor %}
17 {%- for iface in device.interfaces.all() %}
18 {%- set parts = iface.name.split('.') %}
19 {%- if iface.name not in ["lo0.0", "mgmt0"] %}
20 {%- if parts|length == 1 %}
21 set / interface {{ iface.name }} admin-state {{ 'enable' if iface.enabled else 'disable' }}
22 {%- elif parts|length == 2 %}
```

8 lines | 116 B | JSON

```
1 {
2   "ospf": {
3     "enabled": "enable",
4     "instance": "main",
5     "version": "ospf-v2",
6     "area": "0.0.0.0"
7   }
8 }
```

Fase 1: Intended Config



Fase 2: O Git como Data Source

networkautomation / wtr-popba-2026

<> Code Issues Pull Requests Actions Packages Projects Releases Wiki Activity Settings

development wtr-popba-2026 / netbox-data-source

Go to file Add File ... History

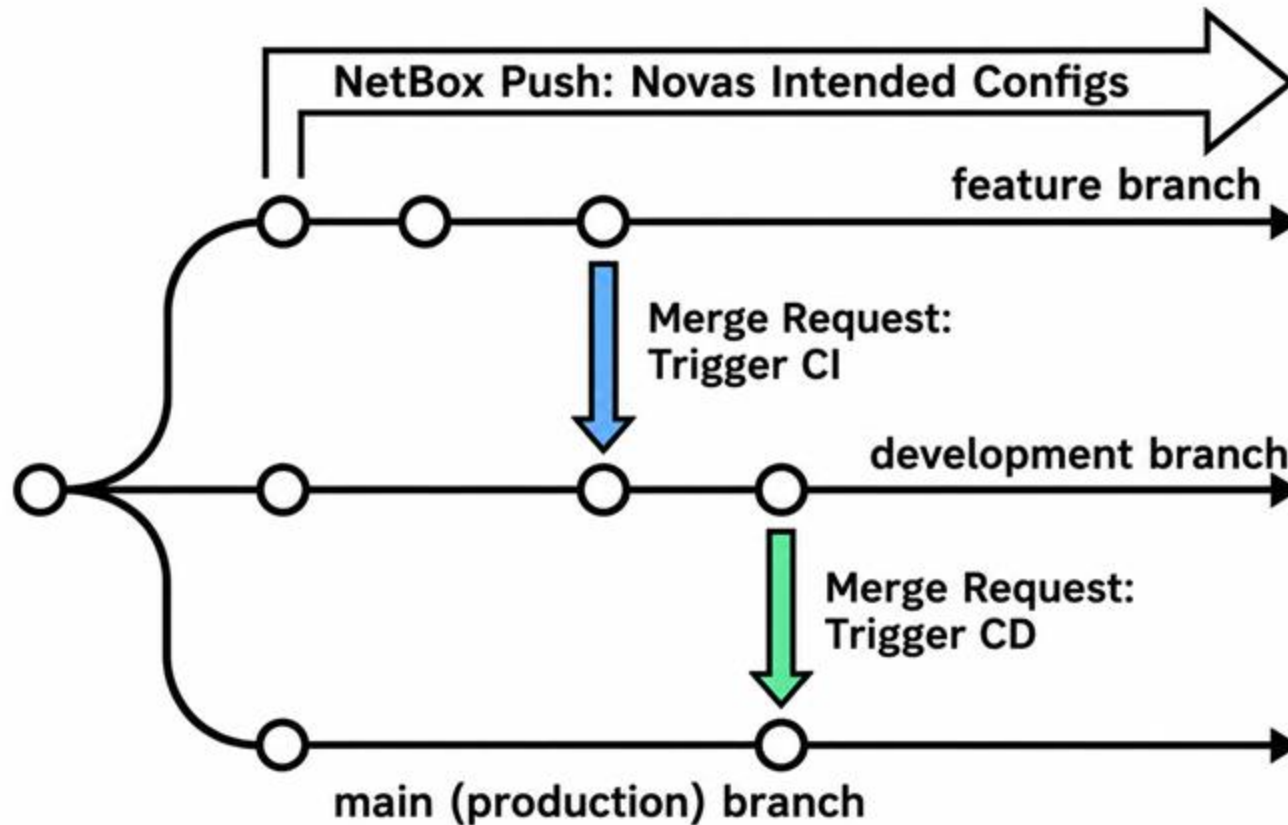
wsdoprado c775ef3edd ✓ Add netbox-data-source README 22 minutes ago

..		
config-context	Fix empty OSPF router-id: fall back to lo0.0 address; fix ospf.json	yesterday
intended	NetBox: sp1-srlinux-pe-101 - 2026-09-25 16:19:38	40 minutes ago
scripts	Stop tracking __pycache__	yesterday
templates	Fix empty OSPF router-id: fall back to lo0.0 address; fix ospf.json	yesterday
README.md	Add netbox-data-source README	22 minutes ago

networkautomation 6746086267 ✓ NetBox: sp1-srlinux-pe-101 - 2026-09-25 16:19:38 42 minutes ago

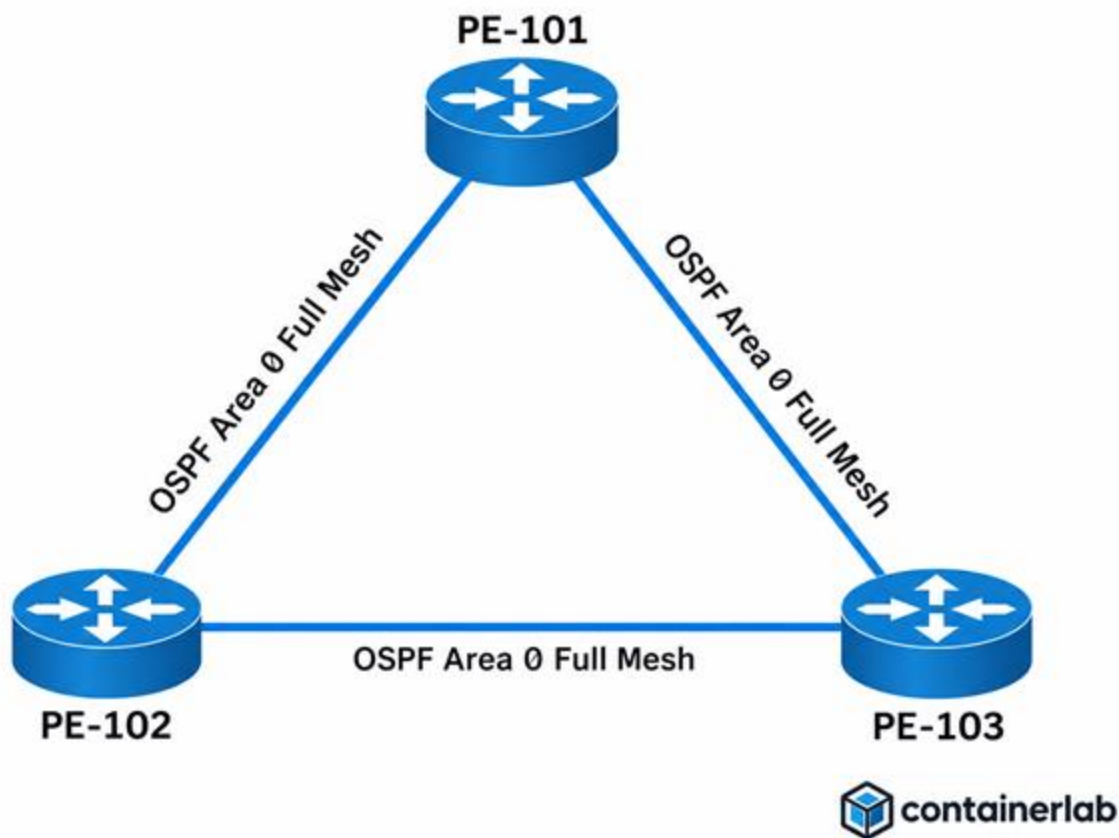
..		
sp1-srlinux-pe-101.cfg	NetBox: sp1-srlinux-pe-101 - 2026-09-25 16:19:38	42 minutes ago
sp2-srlinux-pe-102.cfg	Fix empty OSPF router-id: fall back to lo0.0 address; fix ospf.json	yesterday
sp3-srlinux-pe-103.cfg	Fix empty OSPF router-id: fall back to lo0.0 address; fix ospf.json	yesterday

Fase 2: O Git como Motor para Automação



Fase 2: Pipeline Cl.yml

Fase 3 & 4: O Gêmeo Digital no Pipeline CI



Intended configs carregadas automaticamente como startup-config em containers Docker (Nokia SR Linux).

Gitea Actions: Sequência do Runner

- ✓ Clone do Repositório
- ✓ Setup Python
- ✓ Deploy ContainerLab
- ✓ Testes Automatizados (NUTS)
- ✓ Destroy & Cleanup (Efêmero)

Fase 3 & 4: O Gêmeo Digital no Pipeline CI

```
- name: Setup Python venv
  run: |
    cd /tmp/wtr-popba-2026-dev-${{ github.run_id }}
    uv venv .venv
    source .venv/bin/activate
    uv sync

- name: Clean up old labs
  run: |
    cd /tmp/wtr-popba-2026-dev-${{ github.run_id }}
    containerlab destroy --all -y || true
    sleep 20

- name: Deploy lab
  run: |
    cd /tmp/wtr-popba-2026-dev-${{ github.run_id }}
    containerlab deploy -t containerlab/wtrpopba.clab.yaml --reconfigure

- name: Wait for lab to be ready
  run: sleep 30
```

```
name: wtrpopba

mgmt:
  network: wtrpopba-mgmt
  ipv4-subnet: 172.24.0.0/24
  ipv6-subnet: 2001:db8:cafe::/64

topology:
  nodes:
    sp1-srlinux-pe-101:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:24.7.2
      mgmt-ipv4: 172.24.0.101
      mgmt-ipv6: 2001:db8:cafe::101
      startup-config: ../netbox-data-source/intended/sp1-srlinux-pe-101.cfg

    sp2-srlinux-pe-102:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:24.7.2
      mgmt-ipv4: 172.24.0.102
      mgmt-ipv6: 2001:db8:cafe::102
      startup-config: ../netbox-data-source/intended/sp2-srlinux-pe-102.cfg

    sp3-srlinux-pe-103:
      kind: nokia_srlinux
      image: ghcr.io/nokia/srlinux:24.7.2
      mgmt-ipv4: 172.24.0.103
      mgmt-ipv6: 2001:db8:cafe::103
      startup-config: ../netbox-data-source/intended/sp3-srlinux-pe-103.cfg

  links:
    - endpoints: ["sp1-srlinux-pe-101:e1-1", "sp2-srlinux-pe-102:e1-1"]
    - endpoints: ["sp1-srlinux-pe-101:e1-2", "sp3-srlinux-pe-103:e1-2"]
    - endpoints: ["sp2-srlinux-pe-102:e1-2", "sp3-srlinux-pe-103:e1-1"]
```

Fase 3 & 4: O Gêmeo Digital no Pipeline CI

✓ Deploy lab

1m35s

```
1 13:41:46 INFO Containerlab started version=0.79.0
2 13:41:46 INFO Parsing & checking topology file=wtrpopba.clab.yaml
3 13:41:47 INFO Removing directory path=/tmp/wtr-popba-2026-dev-46/containerlab/clab-wtrpopba
4 13:41:47 INFO Creating docker network name=wtrpopba-mgmt IPv4 subnet=172.24.0.0/24 IPv6 subnet=2001:db8:cafe::/64 MTU=0
5 13:41:47 INFO Creating lab directory path=/tmp/wtr-popba-2026-dev-46/containerlab/clab-wtrpopba
6 13:41:48 INFO Creating container name=sp1-srlinux-pe-101
7 13:41:48 INFO Creating container name=sp2-srlinux-pe-102
8 13:41:48 INFO Creating container name=sp3-srlinux-pe-103
9 13:41:49 INFO Running postdeploy actions kind=nokia_srlinux node=sp3-srlinux-pe-103
10 13:41:50 INFO Created link: sp1-srlinux-pe-101:e1-2 --- sp3-srlinux-pe-103:e1-2
11 13:41:50 INFO Running postdeploy actions kind=nokia_srlinux node=sp1-srlinux-pe-101
12 13:41:50 INFO Created link: sp1-srlinux-pe-101:e1-1 --- sp2-srlinux-pe-102:e1-1
13 13:41:50 INFO Created link: sp2-srlinux-pe-102:e1-2 --- sp3-srlinux-pe-103:e1-1
14 13:41:50 INFO Running postdeploy actions kind=nokia_srlinux node=sp2-srlinux-pe-102
15 13:43:21 INFO Adding host entries path=/etc/hosts
16 13:43:21 INFO Adding SSH config for nodes path=/etc/ssh/ssh_config.d/clab-wtrpopba.conf
```

Name	Kind/Image	State	IPv4/6 Address
clab-wtrpopba-sp1-srlinux-pe-101	nokia_srlinux	running	172.24.0.101
	ghcr.io/nokia/srlinux:24.7.2		2001:db8:cafe::101
clab-wtrpopba-sp2-srlinux-pe-102	nokia_srlinux	running	172.24.0.102
	ghcr.io/nokia/srlinux:24.7.2		2001:db8:cafe::102
clab-wtrpopba-sp3-srlinux-pe-103	nokia_srlinux	running	172.24.0.103
	ghcr.io/nokia/srlinux:24.7.2		2001:db8:cafe::103

Fase 5: Validação Robusta com NUTS



Testes de Interface	Testes de Conectividade	Testes de Protocolo (OSPF)
✓ - Status (UP) ✓ - Velocidade configurada ✓ - MTU correto ✓ - Endereçamento IP	✓ - ICMP Loopback local ✓ - Ping entre PEs ✓ - Conectividade P2P	✓ - Neighbors UP ✓ - Confirmação Area 0 ✓ - Estado FULL

NUTS (Network Unit Testing System) substitui a inspeção manual, garantindo precisão cirúrgica

Fase 5: Validação Robusta com NUTS

Fase 5: Validação Robusta com NUTS - ospf.yaml

15 lines | 404 B | YAML | Executable File

Raw

Permalink

Blame

History



```
1 ---
2 - test_module: custom_tests.sr_linux_ospf_napalm
3   test_class: TestSRLinuxOSPFNeighbor
4   test_data:
5     - host: SR1
6       instance: main
7       neighbor_id: 10.255.255.2
8       expected_interface: ethernet-1/1.0
9       description: "SR1 - Neighbor SR2"
10    - host: SR1
11      instance: main
12      neighbor_id: 10.255.255.3
13      expected_interface: ethernet-1/2.0
14      description: "SR1 - Neighbor SR3"
```

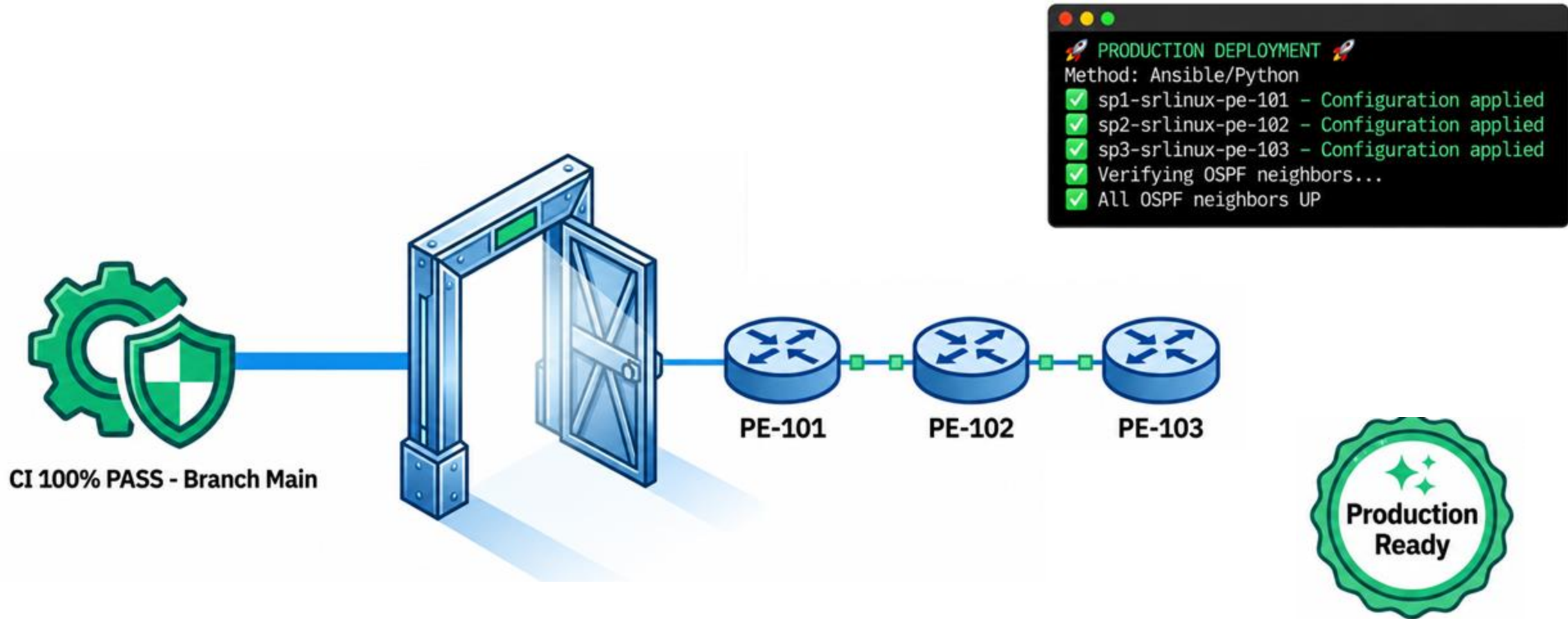
Deep Dive: O Valor da Falha Pré-Deploy

```
FAILED:  
TestSRLinuxOSPFNeighbor.test_n  
eighbor_exists[SR1_0]  
E  AssertionError: Neighbor  
2.2.2.2 não encontrado no  
host SR1. Neighbors  
Neighbors disponíveis:  
['10.255.255.2',  
'10.255.255.3']
```



```
test_data:  
- host: SR1  
  instance: main  
  neighbor_id: 10.255.255.2  
  expected_interface:  
  ethernet-1/1.0  
- host: SR1  
  instance: main  
  neighbor_id: 10.255.255.3  
  expected_interface:  
  ethernet-1/2.0
```

Fase 6: Deploy em Produção (CD Pipeline)



Fase 6: Deploy em Produção (CD Pipeline)

deploy
Success

> Set up job 1s

> Checkout repository 0s

▼ Production Deploy 7s

```
1
2
3
4  🚀 PRODUCTION DEPLOY 🚀
5
6
7
8  📁 Project: WTR POPBA 2026 Network Automation Lab
9  🔧 Deployment Method: Ansible/Python
10
11  🚫 Deploying configurations...
12
13  🔧 [Ansible] Connecting to devices...
14  ✅ sp1-srlinux-pe-101 - Configuration applied
15  ✅ sp2-srlinux-pe-102 - Configuration applied
16  ✅ sp3-srlinux-pe-103 - Configuration applied
17
18  🔍 Verifying OSPF neighbors...
19  ✅ All OSPF neighbors UP
20
21
22  ✅ DEPLOY SUCCESSFUL! ✅
23
24
25
26
27  🚀 Production environment updated!
28  🌟 Ready for WTR POPBA 2026 demonstration!
```

> Cleanup 1s

> Complete job 0s

O Ciclo Completo: Rastreabilidade Total

Quem e Quando?

Autor e Timestamp registrados no commit do NetBox.
(Ex: NetBox: sp1-srlinux...)

O Quê?

O Diff exato do código versionado e imutável no Git.

Por Quê?

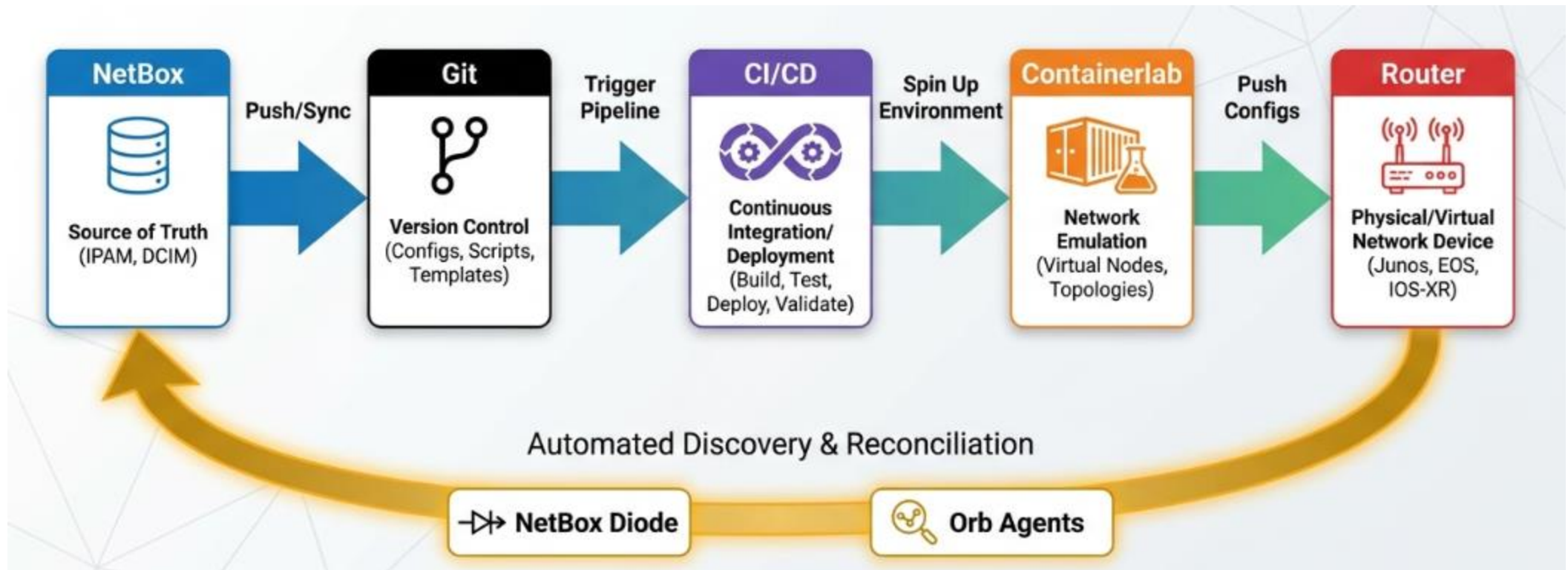
Histórico de Pull Requests e validação do Digital Twin (NUTS).

Onde?

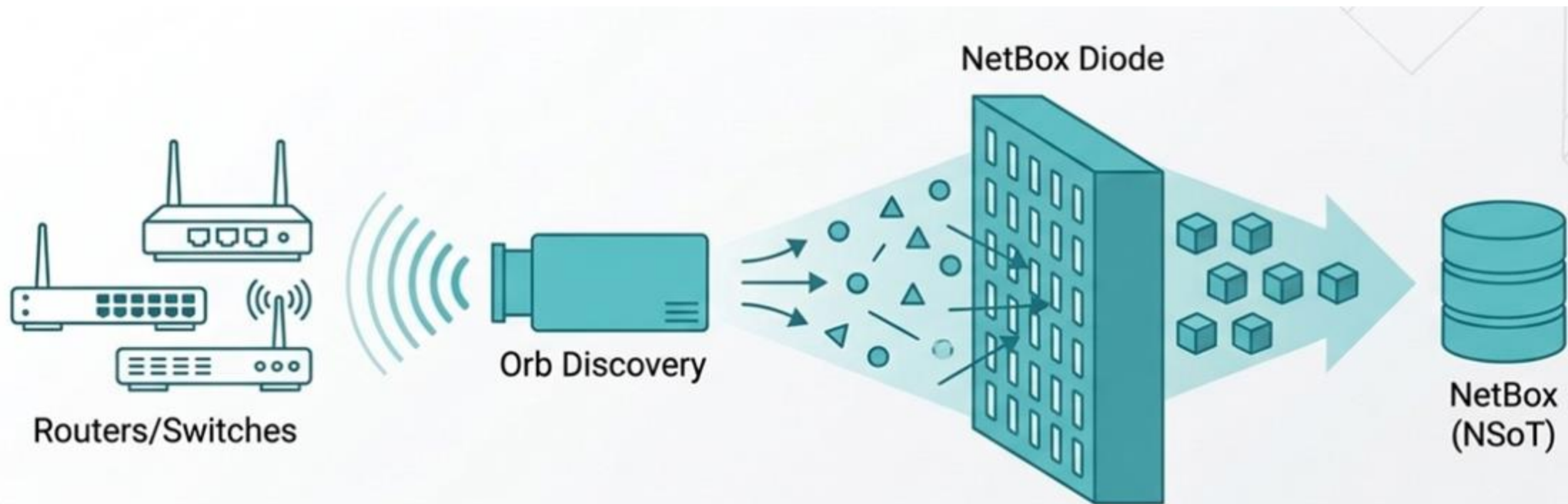
Aplicação cirúrgica no roteador via pipeline CD com logs definitivos.

A configuração está no roteador. O pipeline funcionou 100%. Mas, e se alguém acessar o CLI e realizar uma mudança?

Closed-Loop



Netbox Discovery - Orb Agent

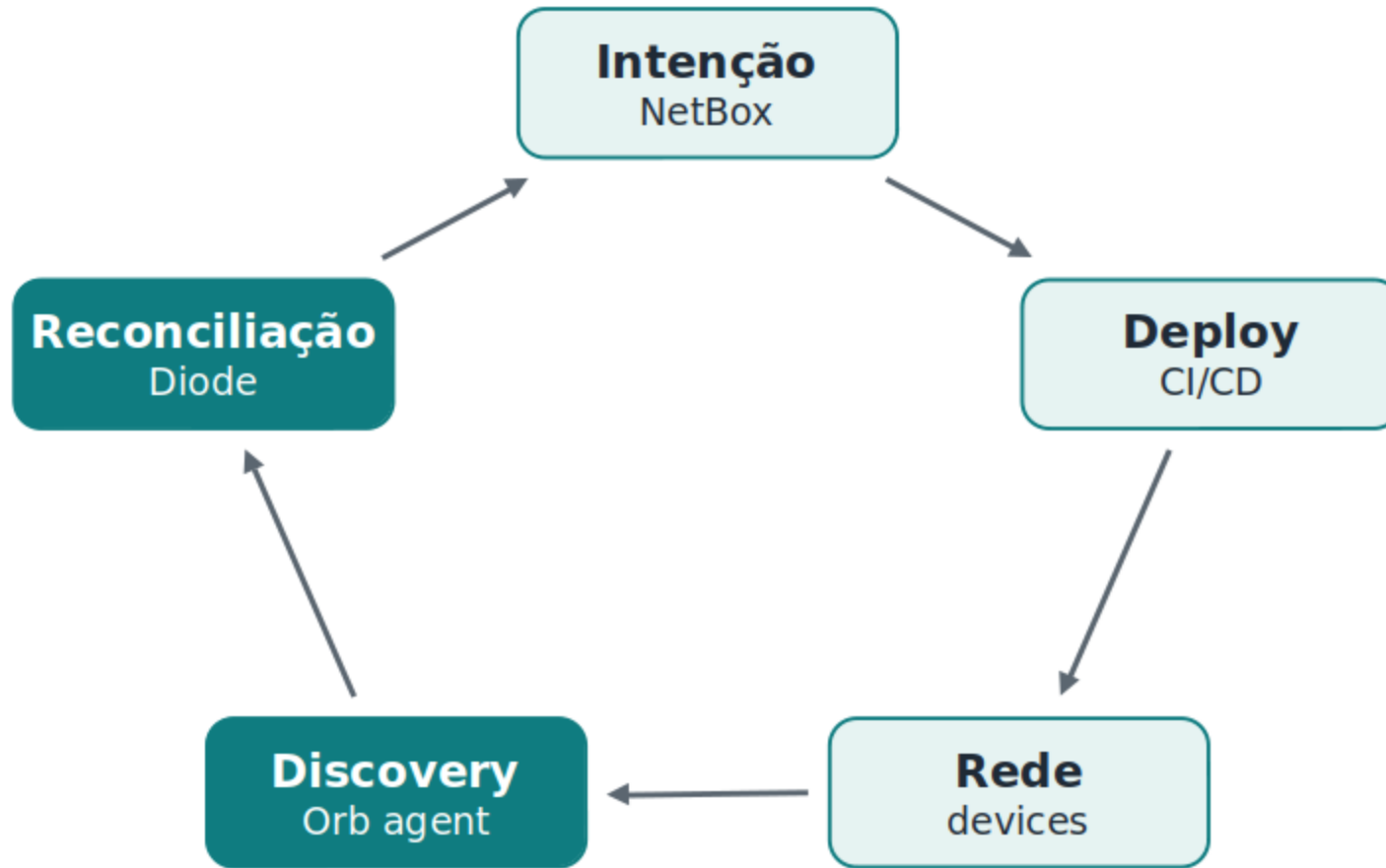


Objetivo: Descoberta dinâmica de ativos físicos e coleta do estado real da infraestrutura.

Mecanismo: Varredura passiva e ativa (SNMP/LLDP) orquestrada pelo Orb.

Integração Segura: Inserção estruturada de dados de inventário na NSoT exclusivamente via NetBox Diode. O Diode atua como uma barreira protetora, normalizando os dados e impedindo a corrupção do Source of Truth por telemetria desestruturada.

Netbox Diode - Reconciliation



Ingestão via gRPC

O agente envia o que descobriu ao Diode, não direto na API do NetBox



Reconciliação

Diode compara com o NetBox e identifica drift e desvios



NetBox atualizado

Aplica o que for relevante e reduz entrada manual



Qualquer script vira produtor

Com o SDK do Diode, Nornir e validadores alimentam o mesmo fluxo

NetDevOps não é só “empurrar/push” configuração: é fechar o loop entre intenção e a realidade.

NetDevOps: Transformando Riscos em Confiabilidade



Segurança

Validação prévia obrigatória em Digital Twin antes de aplicar em produção.



Velocidade

Deploy automatizado reduz vertiginosamente tempos de janela e elimina erros manuais.



Rastreabilidade

Cada alteração possui histórico completo do banco de dados ao roteador.



Auditoria

Registro imutável no Git detalhando exatamente quem alterou, quando e o porquê.



Consistência

Configurações rigorosamente padronizadas por templates Jinja2; fim da configuração artesanal.



Confiabilidade

Cobertura de testes unitários garante que interfaces e OSPF operem perfeitamente.

Obrigado
www.ix.br

Setembro, 29, 2026

nic.br **cgi.br**
www.nic.br | www.cgi.br