
nic.br
Brazilian Network
Information Center
cgi.br
Brazilian Internet
Steering Committee
registro.br
cert.br
cetic.br
ceptro.br
ceweb.br
ix.br

GTER54

Do GIT ao Router

From GIT to the Router

William Prado <wprado@nic.br>

O Desafio

Config Drift

Configurações não documentadas e mudanças manuais causam divergências.

Falta de Rastreabilidade e Versionamento

Quem mudou? Quando? Por quê? Essas perguntas ficam sem resposta.

Validação Pós-Deploy

Descobrir problemas apenas em produção aumenta riscos e downtime.

NetDevOps: DevOps para Redes

Aplicando DevOps à Infraestrutura de Rede

NetDevOps traz automação, versionamento e testes para configuração e operação de redes

📝 Network as Code

Configurações versionadas em Git, geradas por templates, testadas antes do deploy.

🤖 Automação

CI/CD pipelines validam e aplicam mudanças automaticamente em devices.

🧪 Testing

Digital twins e testes automatizados garantem qualidade antes da produção.

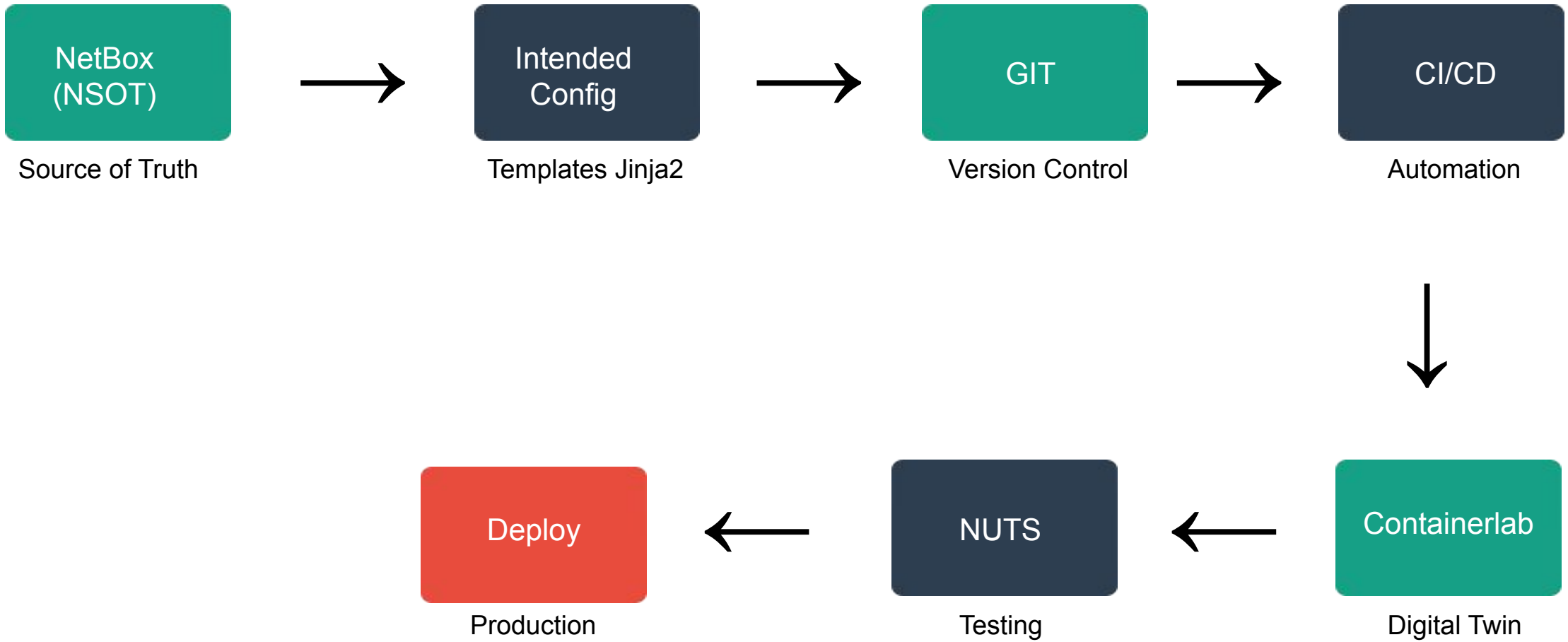
🔄 Traditional Network Ops

Manual, CLI, sem versionamento, validação pós-deploy.

⚡ NetDevOps

Automatizado, versionado, testado, validação pré-deploy.

Pipeline End-to-End



NetBox: Network Source of Truth

📊 Dados Estruturados

Devices, interfaces, IPs, VLANs, etc

🔧 Config Contexts

Estrutura de Dados personalizada para representar sua Rede (ospf.json)

📝 Templates Jinja2

Renderização das configurações do NetBox. (nokia.j2)

📊 Branching

Estados da Rede representados por branches: main, staging, lab, etc

✨ Resultado

Intended Configs geradas automaticamente via Script

GIT: Data Source + Trigger

📂 netbox-data-source

Config Contexts

config-context/ospf.json

Templates

templates/nokia.j2

Intended Config

intended/*.cfg

Scripts

generate_intended.py

🚀 CI/CD - Control Plane

Intended config: git push → branch feature

Pull Request: branch feature → development

Start CI pipeline - validation.yml

Pull Request: development → main

Start CD pipeline - deploy.yml

Rastreabilidade Total

Commit → Config → Router

ix.br nic.br

ContainerLAB: Digital Twin

Por que Digital Twin?

Validar mudanças em ambiente idêntico à produção antes de aplicar nos routers reais.

🔧 Topologia

3x Nokia SR Linux (PE-101, PE-102, PE-103)

🌐 Protocols

OSPF Area 0 com full mesh

⚡ Startup Config

Intended configs carregadas automaticamente

gter54.clab.yaml

```
topology:
  nodes:
    sp1-srlinux-pe-101:
      kind: nokia_srlinux
      startup-config:
../netbox-data-source/intended/sp1-srlinux-pe-101.cfg
```

NUTS - Network Unit Testing System

Framework de Testes Automatizados

- Validação completa do Digital Twin antes do deploy em Produção;
- Utiliza Nornir no backend para acesso aos devices;
- Existem varios test cases prontos, mas é possível criar os testes personalizados;
- SR Linux utiliza NAPALM-SRLINUX com GNMI para testes;

Interface Tests

Status operacional
Velocidade configurada
MTU correto
Endereçamento IP

ICMP Tests

Ping loopback local
Ping entre PEs
P2P connectivity

OSPF Tests

Neighbors UP
Area 0.0.0.0
FULL STATE

Garante a qualidade do deploy.

Se os testes passam no Digital Twin → Seguro para produção

Workflow

1. **NetBox** gera intended configs a partir de dados estruturados e templates.
2. **Git** versiona configurações e config-contexts, sendo source of truth e trigger.
3. **CI/CD** é chamado/trigger automaticamente em push/merge para validar ou fazer o deploy.
4. **ContainerLab** cria Digital Twin idêntico à produção com configurações aplicadas.
5. **NUTS** executa testes automatizados validando interfaces, conectividade e OSPF.
6. **Deploy** Real aplica configurações nos routers apenas se todos os testes passarem.

Fluxo de Branches

🧪 CI - Continuous Integration

Deploy ContainerLab + NUTS Testing

🚀 CD - Continuous Deployment

Deploy automático nos routers

CI/CD Pipeline com Gitea Actions

DEVELOPMENT

- ✓ Clone do repositório
- ✓ Setup Python + UV
- ✓ Deploy ContainerLab
- ✓ Testes: Interfaces
- ✓ Testes: ICMP
- ✓ Testes: OSPF
- ✓ Destroy + Cleanup

MAIN

- ✓ Deploy
- ✓ Ansible/Nornir
- ✓ Aplicação nos devices

✨ **Production Ready**

Self-hosted runner: Gitea + act_runner em Docker ou Nativo com acesso privilegiado para ContainerLab

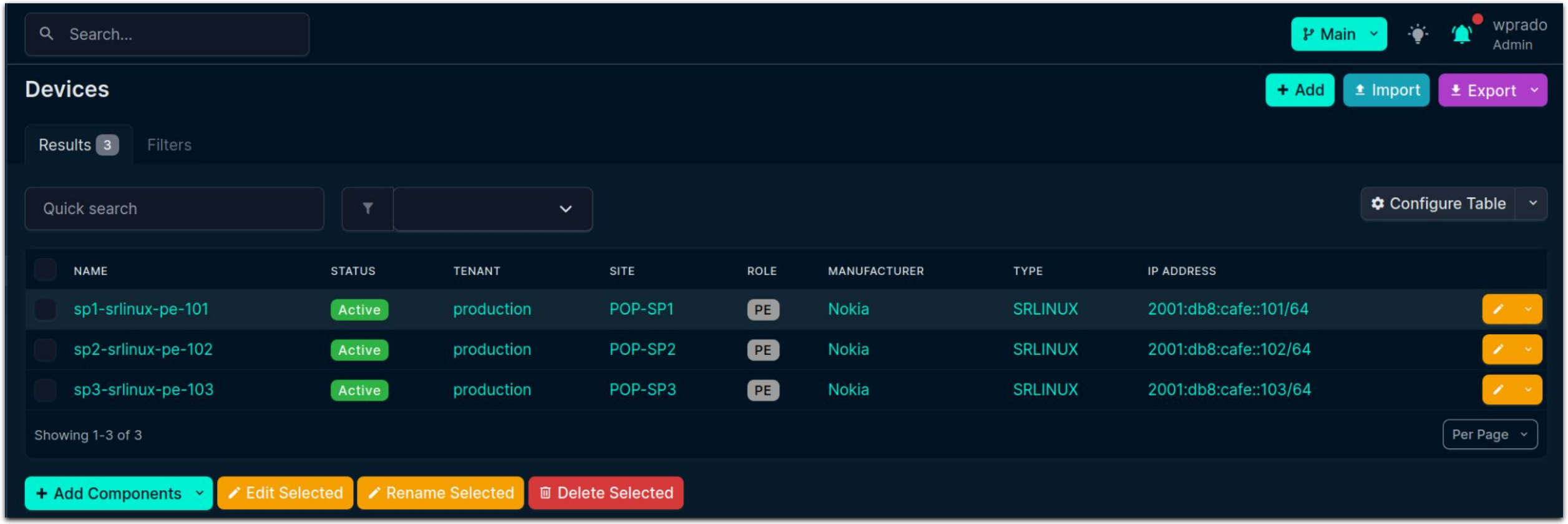
E na prática?

Video no youtube:

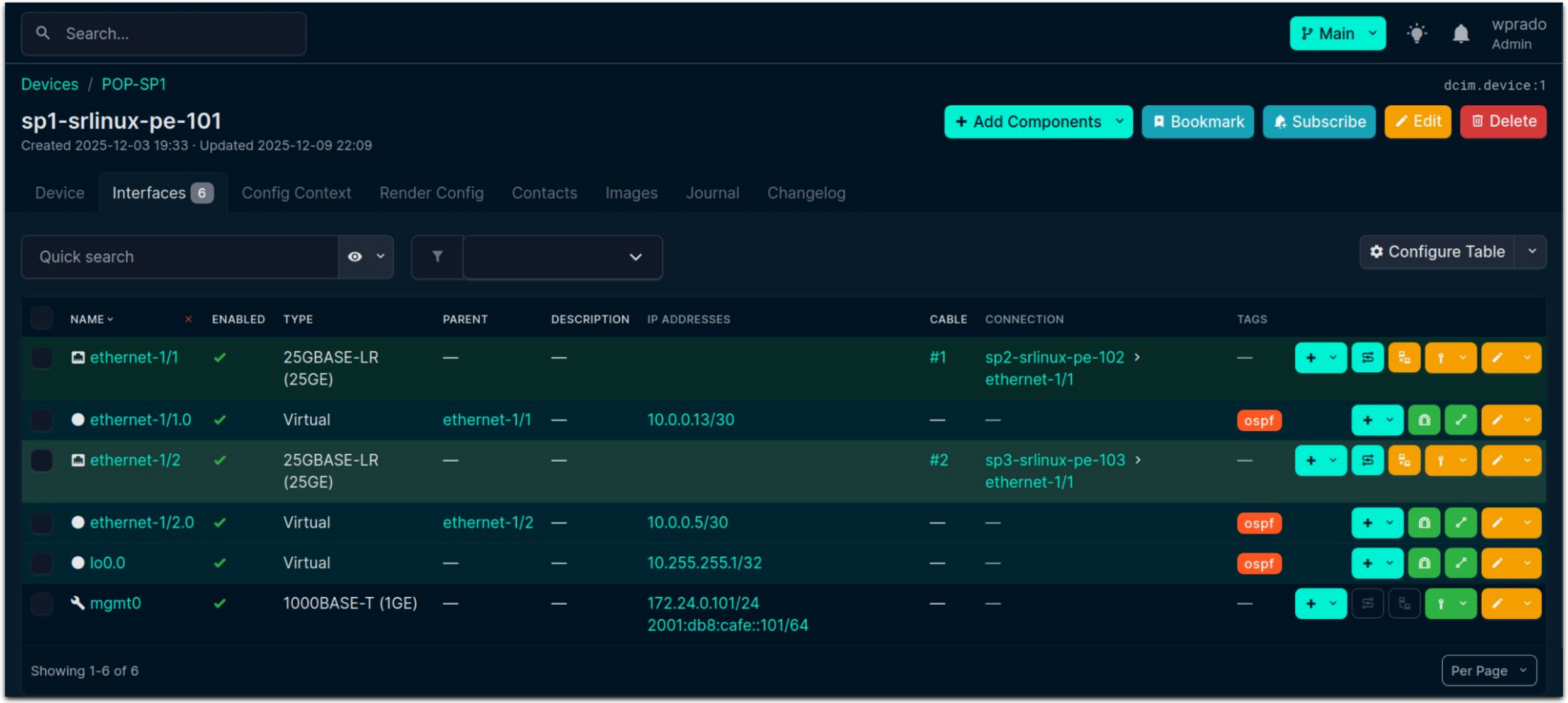
@NetworkAutomationBR

ix.br nic.br

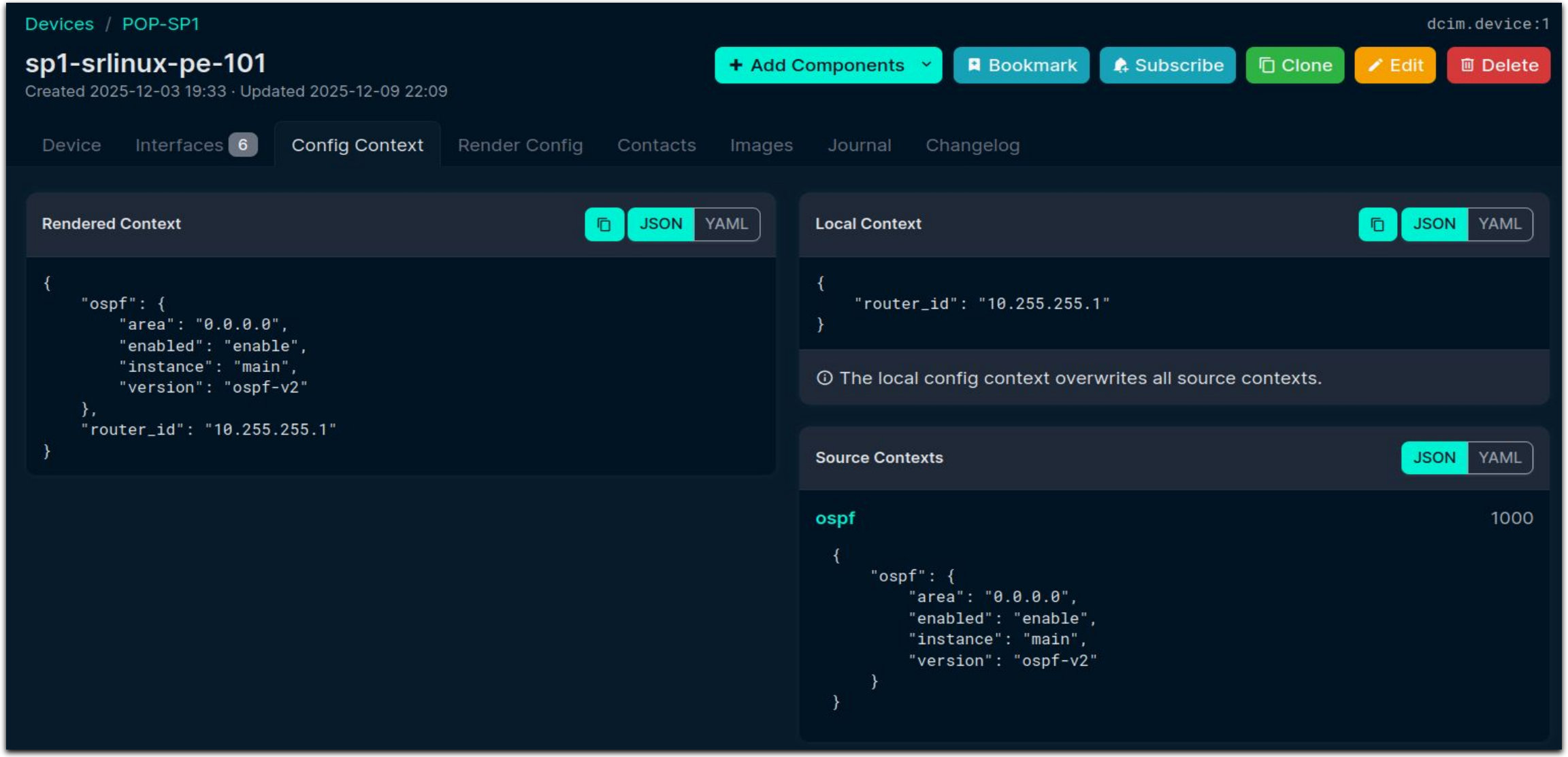
Netbox - Devices



Netbox - Interfaces



Netbox - Config Context



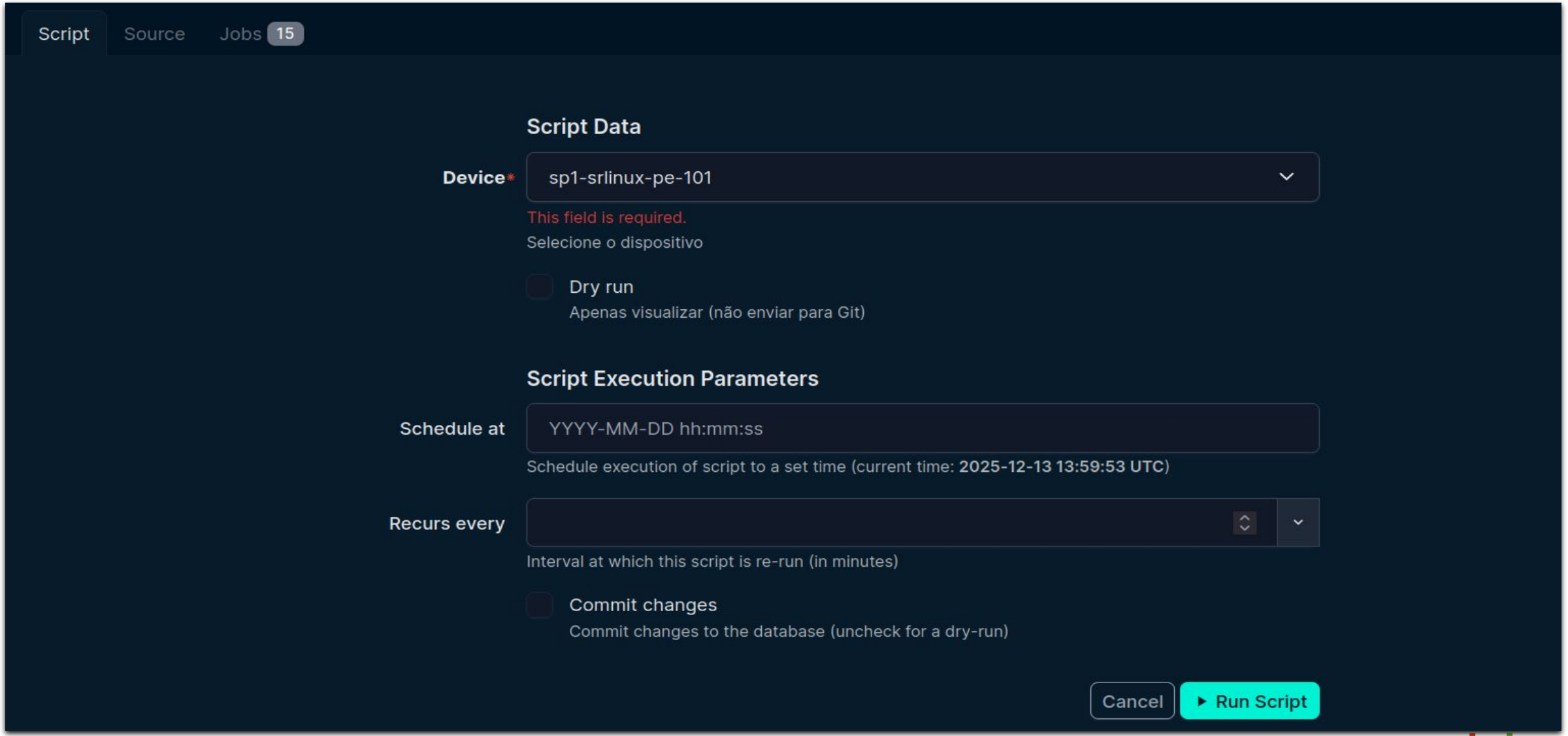
Netbox - Render Config

Rendered Config

Download

```
set / network-instance default type default
set / interface lo0 admin-state enable
set / interface lo0 subinterface 0 ipv4 address 10.255.255.1/32
set / interface lo0 subinterface 0 ipv4 admin-state enable
set / network-instance default interface lo0.0
set / interface ethernet-1/1 admin-state enable
set / interface ethernet-1/1 subinterface 0 ipv4 address 10.0.0.13/30
set / interface ethernet-1/1 subinterface 0 ipv4 admin-state enable
set / network-instance default interface ethernet-1/1.0
set / interface ethernet-1/2 admin-state enable
set / interface ethernet-1/2 subinterface 0 ipv4 address 10.0.0.5/30
set / interface ethernet-1/2 subinterface 0 ipv4 admin-state enable
set / network-instance default interface ethernet-1/2.0
set / network-instance default protocols ospf instance main admin-state enable
set / network-instance default protocols ospf instance main version ospf-v2
set / network-instance default protocols ospf instance main router-id 10.255.255.1
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/1.0 admin-state enable
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/1.0 interface-type point-to-point
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/2.0 admin-state enable
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/2.0 interface-type point-to-point
set / network-instance default protocols ospf instance main area 0.0.0.0 interface lo0.0 admin-state enable
set / network-instance default protocols ospf instance main area 0.0.0.0 interface lo0.0 interface-type point-to-point
set / network-instance default protocols ospf instance main area 0.0.0.0 interface lo0.0 passive true
```

Netbox Scripts - Intended Config



Netbox Scripts - Intended Config

LINE	TIME	LEVEL	OBJECT	MESSAGE
1	2025-12-13 14:01:14	Info	—	==========
2	2025-12-13 14:01:14	Info	—	📡 Device: sp1-srlinux-pe-101
3	2025-12-13 14:01:14	Info	—	Site: POP-SP1
4	2025-12-13 14:01:14	Info	—	Tipo: SRLINUX
5	2025-12-13 14:01:14	Info	—	==========
6	2025-12-13 14:01:14	Success	—	✓ Config Template: Nokia - SR Linux
7	2025-12-13 14:01:14	Info	—	🔄 Renderizando configuração...
8	2025-12-13 14:01:18	Success	—	✓ Configuração renderizada com sucesso!
9	2025-12-13 14:01:18	Info	—	————————————————————
10	2025-12-13 14:01:18	Info	—	Total de linhas: 23
11	2025-12-13 14:01:18	Info	—	==========
12	2025-12-13 14:01:18	Info	—	📤 ENVIANDO PARA GITEA
13	2025-12-13 14:01:18	Info	—	==========
14	2025-12-13 14:01:18	Info	—	🔗 URL: http://192.168.246.95:3000
15	2025-12-13 14:01:18	Info	—	📁 Repo: wprado/gter54
16	2025-12-13 14:01:18	Info	—	🌿 Branch: feature
17	2025-12-13 14:01:18	Info	—	📄 Arquivo: netbox-data-source/intended/sp1-srlinux-pe-101.cfg

GITEA - Intended Config - Branch feature

23 lines | 1.7 KiB | INI

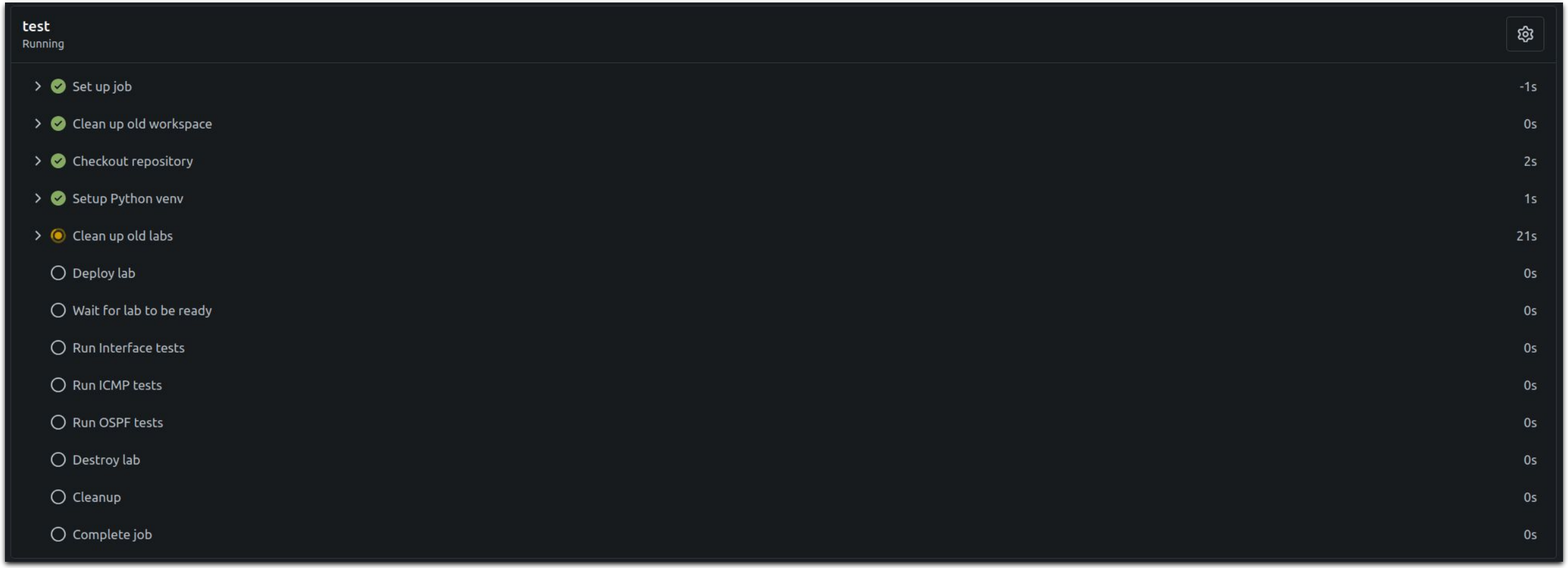
Raw | Permalink | Blame | History

```
set / network-instance default type default
set / interface lo0 admin-state enable
set / interface lo0 subinterface 0 ipv4 address 10.255.255.1/32
set / interface lo0 subinterface 0 ipv4 admin-state enable
set / network-instance default interface lo0.0
set / interface ethernet-1/1 admin-state enable
set / interface ethernet-1/1 subinterface 0 ipv4 address 10.0.0.1/30
set / interface ethernet-1/1 subinterface 0 ipv4 admin-state enable
set / network-instance default interface ethernet-1/1.0
set / interface ethernet-1/2 admin-state enable
set / interface ethernet-1/2 subinterface 0 ipv4 address 10.0.0.5/30
set / interface ethernet-1/2 subinterface 0 ipv4 admin-state enable
set / network-instance default interface ethernet-1/2.0
set / network-instance default protocols ospf instance main admin-state enable
set / network-instance default protocols ospf instance main version ospf-v2
set / network-instance default protocols ospf instance main router-id 10.255.255.1
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/1.0 admin-state enable
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/1.0 interface-type point-to-point
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/2.0 admin-state enable
set / network-instance default protocols ospf instance main area 0.0.0.0 interface ethernet-1/2.0 interface-type point-to-point
set / network-instance default protocols ospf instance main area 0.0.0.0 interface lo0.0 admin-state enable
set / network-instance default protocols ospf instance main area 0.0.0.0 interface lo0.0 interface-type point-to-point
set / network-instance default protocols ospf instance main area 0.0.0.0 interface lo0.0 passive true
```

GITEA - Pull Request para Branch Development

GITEA - Pull Request para Branch Development

Pipeline CI - validation.yml [JOBS]



Pipeline CI - validation.yml - Interface Tests [PASSED]

Run Interface tests 10s

```
============================= test session starts ==============================
platform linux -- Python 3.14.0, pytest-7.4.4, pluggy-1.6.0 -- /tmp/gter54-dev-208/.venv/bin/python3
cachedir: .pytest_cache
rootdir: /tmp/gter54-dev-208/nuts
configfile: pytest.ini
plugins: nuts-3.5.1
collecting ...
collecting 45 items
collected 45 items

tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR1_0] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [  2%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR1_1] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [  4%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR1_2] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [  6%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR2_0] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [  8%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR2_1] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 11%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR2_2] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 13%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR3_0] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 15%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR3_1] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 17%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_enabled[SR3_2] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 20%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR1_0] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 22%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR1_1] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 24%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR1_2] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 26%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR2_0] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 28%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR2_1] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 31%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR2_2] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 33%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR3_0] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 35%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR3_1] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 37%]
tests/interfaces/test_sr_interfaces.yaml::TestNapalmInterfaces::test_is_up[SR3_2] <- ../.venv/lib/python3.14/site-packages/nuts/base_tests/napalm_interfaces.py PASSED [ 40%]
```

Pipeline CI - validation.yml - ICMP Tests [PASSED]

Run ICMP tests 1m34s

```
============================= test session starts ==============================
platform linux -- Python 3.14.0, pytest-7.4.4, pluggy-1.6.0 -- /tmp/gter54-dev-208/.venv/bin/python3
cachedir: .pytest_cache
rootdir: /tmp/gter54-dev-208/nuts
configfile: pytest.ini
plugins: nuts-3.5.1
collecting ...
collecting 3 items
collecting 12 items
collecting 24 items
collecting 36 items
collected 45 items

tests/icmp/test_sr1_loopback_local.yaml::TestSRLinuxPing::test_ping_success[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [  2%]
tests/icmp/test_sr1_loopback_local.yaml::TestSRLinuxPing::test_ping_no_loss[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [  4%]
tests/icmp/test_sr1_loopback_local.yaml::TestSRLinuxPing::test_ping_max_drop[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [  6%]
tests/icmp/test_sr1_loopback_to_sr2.yaml::TestSRLinuxPing::test_ping_success[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [  8%]
tests/icmp/test_sr1_loopback_to_sr2.yaml::TestSRLinuxPing::test_ping_no_loss[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 11%]
tests/icmp/test_sr1_loopback_to_sr2.yaml::TestSRLinuxPing::test_ping_max_drop[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 13%]
tests/icmp/test_sr1_loopback_to_sr3.yaml::TestSRLinuxPing::test_ping_success[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 15%]
tests/icmp/test_sr1_loopback_to_sr3.yaml::TestSRLinuxPing::test_ping_no_loss[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 17%]
tests/icmp/test_sr1_loopback_to_sr3.yaml::TestSRLinuxPing::test_ping_max_drop[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 20%]
tests/icmp/test_sr1_to_sr2_p2p.yaml::TestSRLinuxPing::test_ping_success[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 22%]
tests/icmp/test_sr1_to_sr2_p2p.yaml::TestSRLinuxPing::test_ping_no_loss[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 24%]
tests/icmp/test_sr1_to_sr2_p2p.yaml::TestSRLinuxPing::test_ping_max_drop[SR1_] <- custom_tests/sr_linux_ping_napalm.py PASSED [ 26%]
```

Pipeline CI - validation.yml - OSPF Tests [FAILED]

test
Failure

Run OSPF tests

```
1  ============================= test session starts ==============================
2  platform linux -- Python 3.14.0, pytest-7.4.4, pluggy-1.6.0 -- /tmp/gter54-dev-182/.venv/bin/python3
3  cachedir: .pytest_cache
4  rootdir: /tmp/gter54-dev-182/nuts
5  configfile: pytest.ini
6  plugins: nuts-3.5.1
7  collecting ...
   collecting 6 items
   collected 18 items
8
9  tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR1_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [  5%]
10 tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR1_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 11%]
11 tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR1_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 16%]
12 tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR1_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 22%]
13 tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR1_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 27%]
14 tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR1_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 33%]
15 tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR2_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 38%]
16 tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR2_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 44%]
17 tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR2_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 50%]
18 tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR2_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 55%]
19 tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR2_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 61%]
20 tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR2_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 66%]
21 tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR3_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 72%]
22 tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR3_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 77%]
23 tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR3_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 83%]
24 tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR3_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 88%]
25 tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR3_0] <- custom_tests/sr_linux_ospf_napalm.py FAILED [ 94%]
26 tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR3_1] <- custom_tests/sr_linux_ospf_napalm.py FAILED [100%]
27
28 =================================== FAILURES ===================================
29 _____________ TestSRLinuxOSPFNeighbor.test_neighbor_exists[SR1_0] ______________
30 custom_tests/sr_linux_ospf_napalm.py:82: in test_neighbor_exists
31     assert neighbor_id in single_result.result, \
32 E   AssertionError: Neighbor 2.2.2.2 não encontrado no host SR1. Neighbors disponiveis: ['10.255.255.2', '10.255.255.3']
```

ix.br nic.br

NUTS - test_data - Error

development | gter54 / nuts / tests / ospf / test_sr1_ospf_neighbors.yaml

wprado 3a082d6f56 NUTS

15 lines | 394 B | YAML | Executable File

```yaml
---
- test_module: custom_tests.sr_linux_ospf_napalm
  test_class: TestSRLinuxOSPFNeighbor
  test_data:
    - host: SR1
      instance: main
      neighbor_id: 2.2.2.2
      expected_interface: ethernet-1/1.0
      description: "SR1 - Neighbor SR2"
    - host: SR1
      instance: main
      neighbor_id: 3.3.3.3
      expected_interface: ethernet-1/2.0
      description: "SR1 - Neighbor SR3"
```

NUTS - test_data - Error

development | gter54 / nuts / tests / ospf / test_sr1_ospf_neighbors.yaml

wprado 3a082d6f56 NUTS

15 lines | 394 B | YAML | Executable File

```yaml
---
- test_module: custom_tests.sr_linux_ospf_napalm
  test_class: TestSRLinuxOSPFNeighbor
  test_data:
    - host: SR1
      instance: main
      neighbor_id: 2.2.2.2
      expected_interface: ethernet-1/1.0
      description: "SR1 - Neighbor SR2"
    - host: SR1
      instance: main
      neighbor_id: 3.3.3.3
      expected_interface: ethernet-1/2.0
      description: "SR1 - Neighbor SR3"
```

ESPERADO:
- 10.255.255.2
- 10.255.255.3

NUTS - Atualizando o teste_data

development | gter54 / nuts / tests / ospf / test_sr1_ospf_neighbors.yaml

wprado 655d1a6f27 ✕ Update nuts/tests/ospf/test_sr1_ospf_neighbors.yaml

15 lines | 404 B | YAML | Executable File

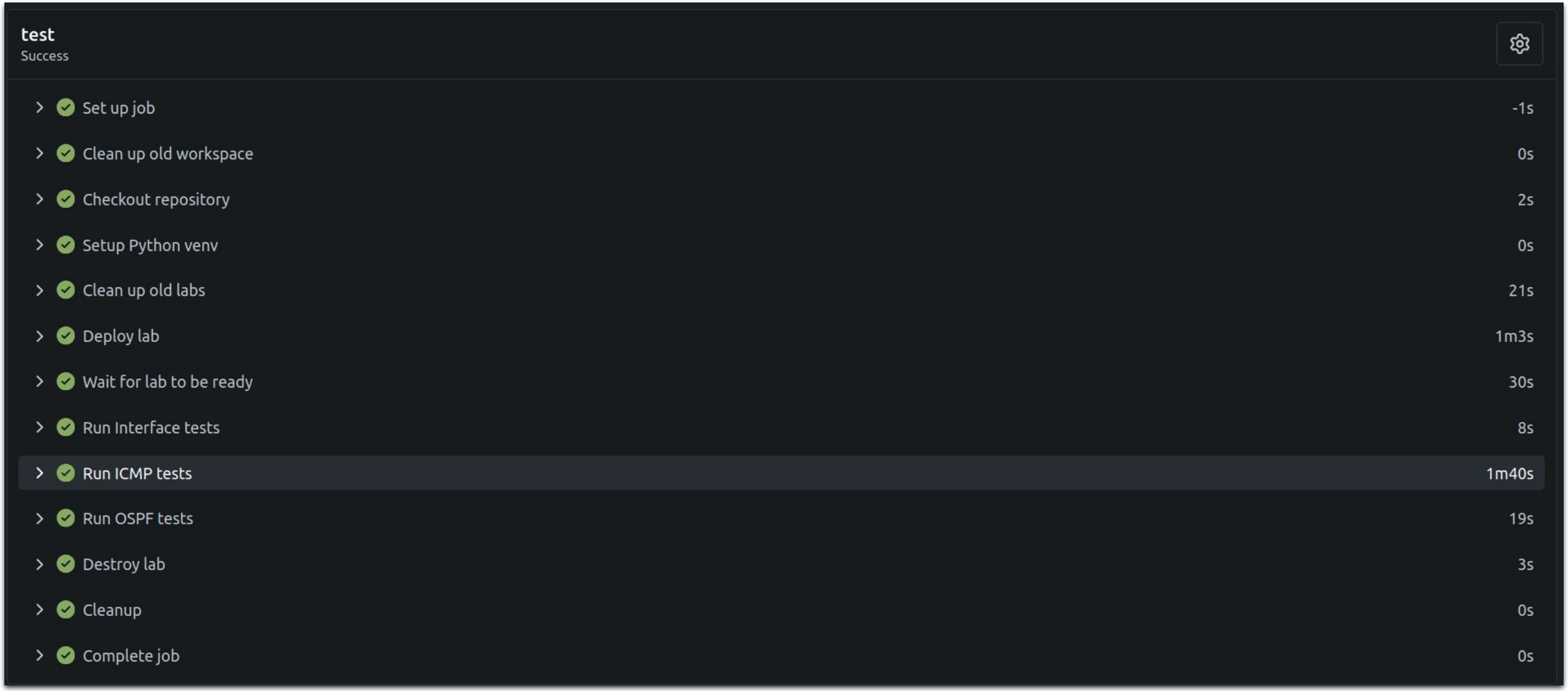
```yaml
---
- test_module: custom_tests.sr_linux_ospf_napalm
  test_class: TestSRLinuxOSPFNeighbor
  test_data:
    - host: SR1
      instance: main
      neighbor_id: 10.255.255.2
      expected_interface: ethernet-1/1.0
      description: "SR1 - Neighbor SR2"
    - host: SR1
      instance: main
      neighbor_id: 10.255.255.3
      expected_interface: ethernet-1/2.0
      description: "SR1 - Neighbor SR3"
```

Pipeline CI - validation.yml - OSPF Tests [PASSED]

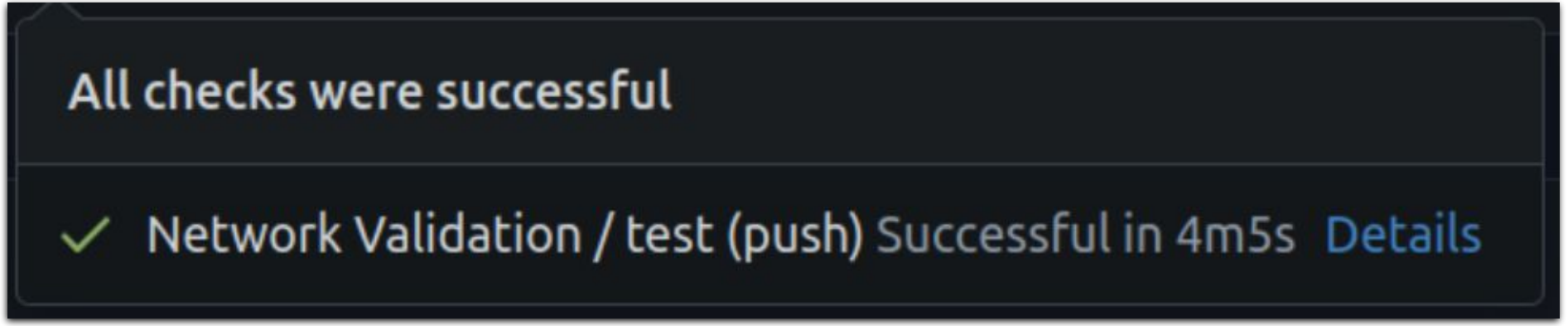
Run OSPF tests

```
============================= test session starts ==============================
platform linux -- Python 3.14.0, pytest-7.4.4, pluggy-1.6.0 -- /tmp/gter54-dev-185/.venv/bin/python3
cachedir: .pytest_cache
rootdir: /tmp/gter54-dev-185/nuts
configfile: pytest.ini
plugins: nuts-3.5.1
collecting ...
collecting 6 items
collected 18 items

tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR1_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [  5%]
tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR1_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 11%]
tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR1_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 16%]
tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR1_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 22%]
tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR1_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 27%]
tests/ospf/test_sr1_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR1_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 33%]
tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR2_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 38%]
tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR2_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 44%]
tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR2_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 50%]
tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR2_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 55%]
tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR2_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 61%]
tests/ospf/test_sr2_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR2_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 66%]
tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR3_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 72%]
tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_exists[SR3_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 77%]
tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR3_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 83%]
tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_state_full[SR3_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 88%]
tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR3_0] <- custom_tests/sr_linux_ospf_napalm.py PASSED [ 94%]
tests/ospf/test_sr3_ospf_neighbors.yaml::TestSRLinuxOSPFNeighbor::test_neighbor_interface[SR3_1] <- custom_tests/sr_linux_ospf_napalm.py PASSED [100%]

============================== 18 passed in 17.81s ==============================
```

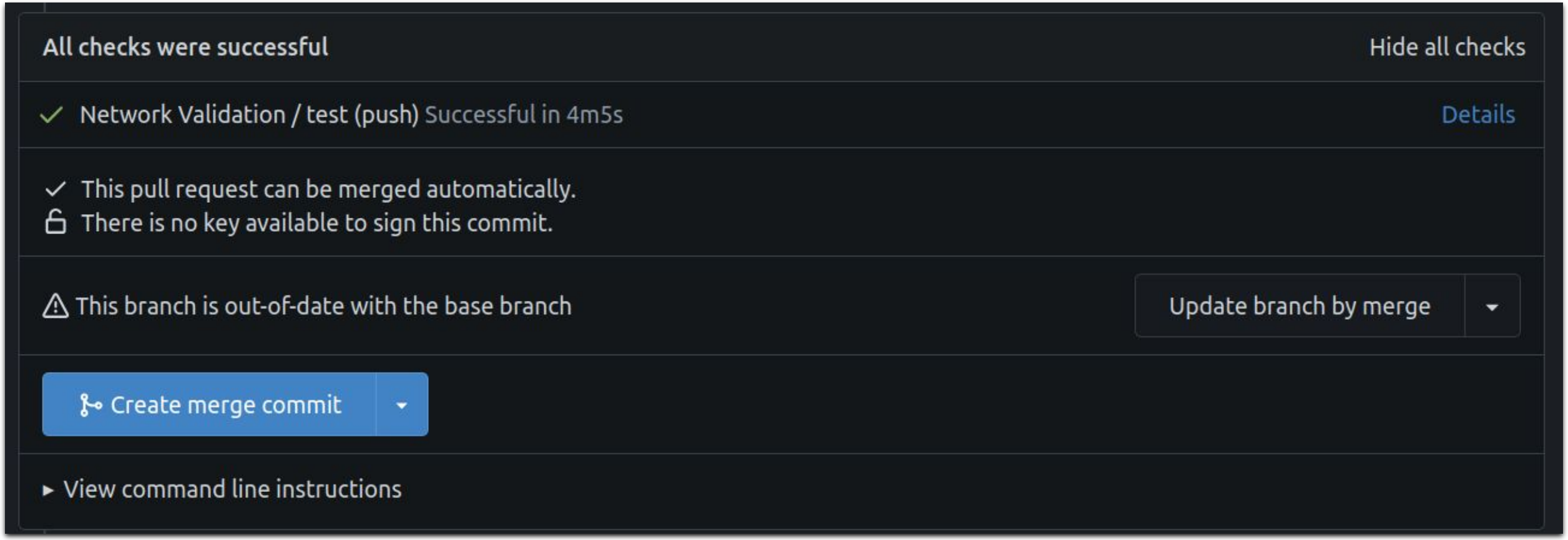
Pipeline CI - validation.yml - [ALL JOBS PASSED]



GITEA - Pull Request - Branch MAIN



GITEA - Pull Request - Branch MAIN

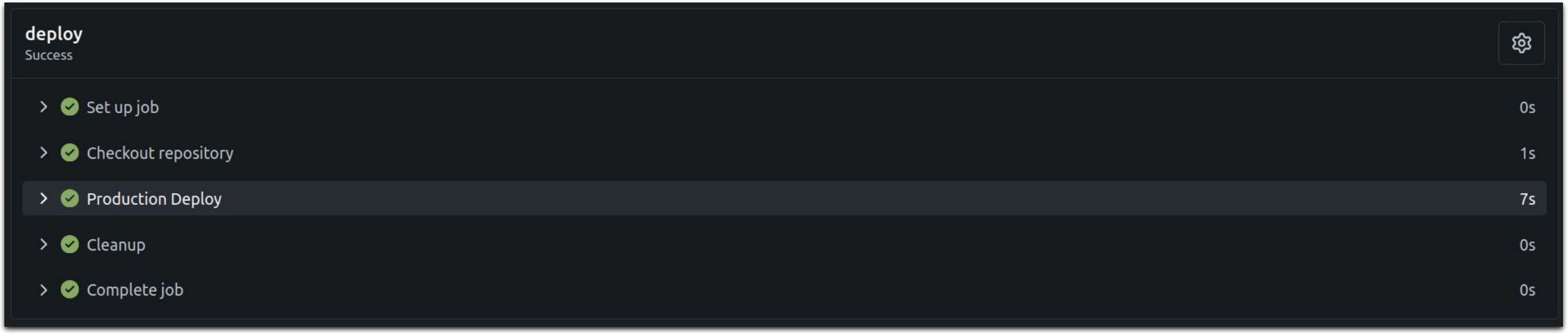


Pipeline CD - deploy.yml

Production Deploy 7s

```
1
2  ╔═══════════════════════════════════════════════════╗
3  ║                                                   ║
4  ║          🚀 PRODUCTION DEPLOYMENT 🚀              ║
5  ║                                                   ║
6  ╚═══════════════════════════════════════════════════╝
7
8  📦 Project: GTER54 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 ║                                                   ║
23 ║             ✅ DEPLOY SUCCESSFUL! ✅              ║
24 ║                                                   ║
25 ╚═══════════════════════════════════════════════════╝
26
27 🎉 Production environment updated!
28 ✨ Ready for GTER54 demonstration!
```

Pipeline CD - deploy.yml



Resumo

Rastreabilidade

Cada mudança rastreada do Git ao router com histórico completo.

Segurança

Validação prévia em Digital Twin antes de aplicar em produção.

Velocidade

Deploy automatizado reduz tempo e elimina erros manuais.

Auditoria

Registro completo de quem mudou, quando e por quê.

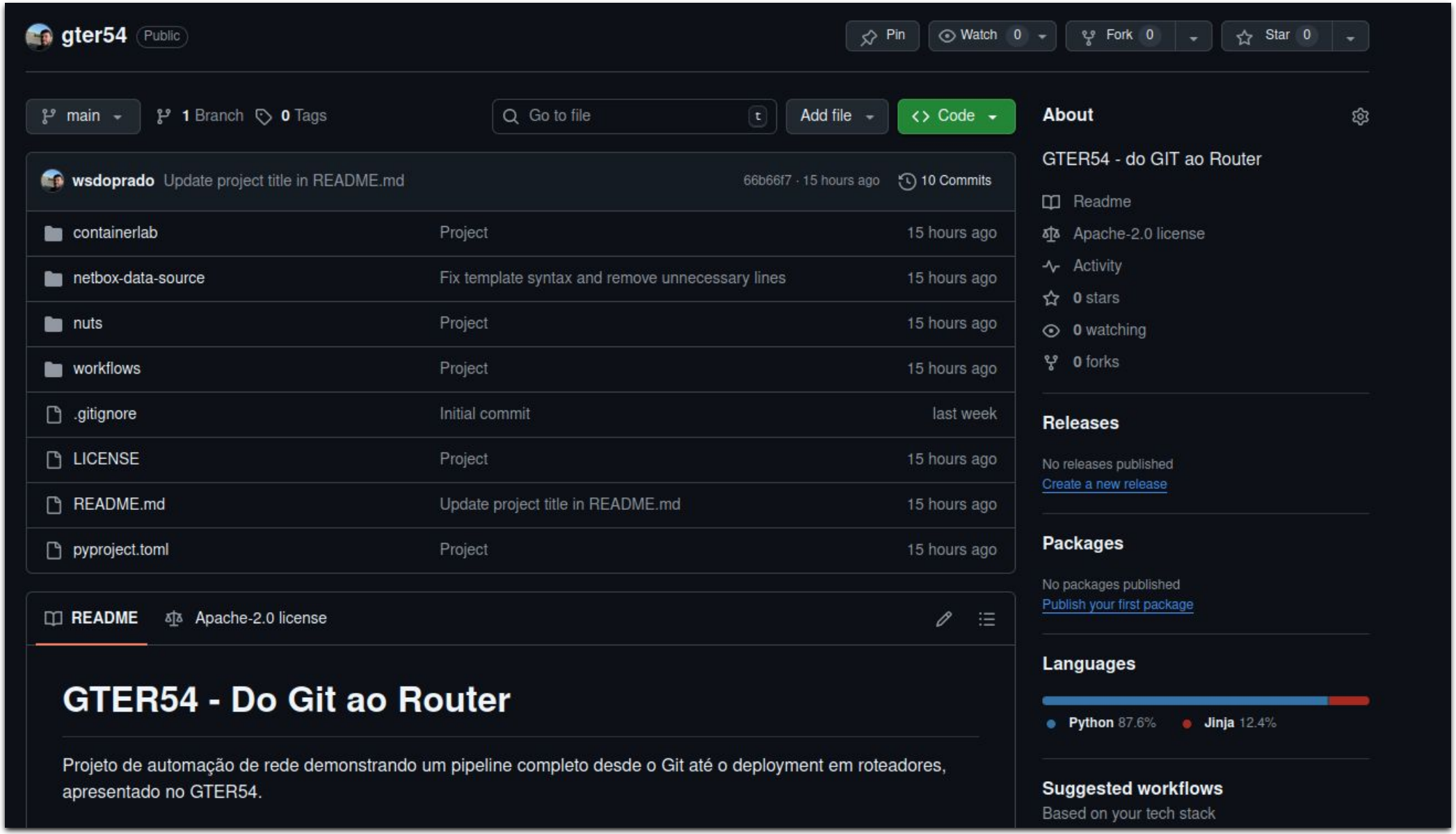
Consistência

Configurações padronizadas geradas por templates.

Confiabilidade

Testes automatizados garantem qualidade antes do deploy.

GITHUB - https://github.com/wsdoprado/gter54



Obrigado

www.ix.br

Dezembro, 16, 2025

nic.br cgi.br

www.nic.br | www.cgi.br