

NetBox

O que é, pra quê server e como implantar na sua infraestrutura



Lucas Rayan Guerra

Introdução

Em ambientes de rede modernos, a complexidade e a escala crescem exponencialmente. Gerenciar essa infraestrutura com planilhas, wikis e scripts desconexos é uma receita para ineficiência, erros e vulnerabilidades de segurança. A solução para esse caos é a adoção de uma Fonte da Verdade (Source of Truth - SoT), um repositório centralizado e autoritativo que contém toda a informação sobre o estado desejado da rede.

É aqui que entra o NetBox. Mais do que uma simples ferramenta de documentação, o NetBox é uma plataforma open-source projetada para ser o coração da sua infraestrutura de rede, combinando as disciplinas de IPAM (IP Address Management) e DCIM (Data Center Infrastructure Management).

Esta cartilha oferece um guia completo sobre o que é o NetBox, por que a documentação de rede é crucial, como ele potencializa a automação e, finalmente, como implementá-lo de forma segura em seu ambiente.

A Importância Crítica da Documentação de Redes

Uma documentação de rede precisa e atualizada não é um luxo, mas uma necessidade operacional. Ela é a base para:

- **Resolução Rápida de Problemas:** Em vez de rastrear cabos ou adivinhar configurações durante uma falha, uma documentação precisa permite identificar e isolar problemas rapidamente, minimizando o tempo de inatividade.
- **Planejamento e Expansão:** Permite visualizar a capacidade disponível, planejar a alocação de recursos (IPs, VLANs, portas) e projetar expansões de forma inteligente e sem conflitos.
- **Segurança e Conformidade:** Uma visão clara da infraestrutura é essencial para auditorias de segurança, análise de vulnerabilidades e garantia de conformidade com regulamentações (como LGPD, PCI-DSS, etc.).

- **Automação Eficaz:** A automação depende de dados confiáveis. Sem uma fonte da verdade, scripts de automação podem operar com informações desatualizadas, causando configurações incorretas ou até mesmo interrupções de serviço.
- **Colaboração e Onboarding:** Facilita a integração de novos membros na equipe e garante que todos os profissionais, de diferentes áreas, operem com a mesma visão da infraestrutura.

O que é o NetBox?

NetBox é uma plataforma que modela e documenta a infraestrutura de rede. Ele estabelece uma relação clara entre os diferentes componentes da rede, criando um modelo de dados rico e interconectado ¹.

| Principais Funcionalidades | |
|----------------------------|---|
| Categoria | Funcionalidades Principais |
| IPAM | Gerenciamento de prefixos e endereços IP (IPv4 e IPv6), VRFs e VLANs. |
| DCIM | Modelagem de sites, racks, dispositivos, cabos, conexões e circuitos. |
| Dispositivos | Inventário de tipos de dispositivos, modelos, componentes (interfaces, console, etc.) e status. |
| Virtualização | Gerenciamento de clusters, máquinas virtuais e suas interfaces. |
| Circuitos | Rastreamento de circuitos de provedores, incluindo tipo, ID e pontos de terminação. |
| Extensibilidade | Suporte a campos customizados, plugins e webhooks para integração com outros sistemas. |

O NetBox não é uma ferramenta de monitoramento, provisionamento ou gerenciamento de configuração. Em vez disso, ele é a fonte de dados autoritativa que alimenta essas outras ferramentas.

NetBox como Motor para Automação de Rede

O verdadeiro poder do NetBox é desbloqueado quando ele é usado como a fonte da verdade para a automação. Ele foi projetado com uma filosofia "API-first", o que significa que tudo o que pode ser feito através da interface web também pode ser feito via sua robusta API REST ².

Como o NetBox potencializa a automação:

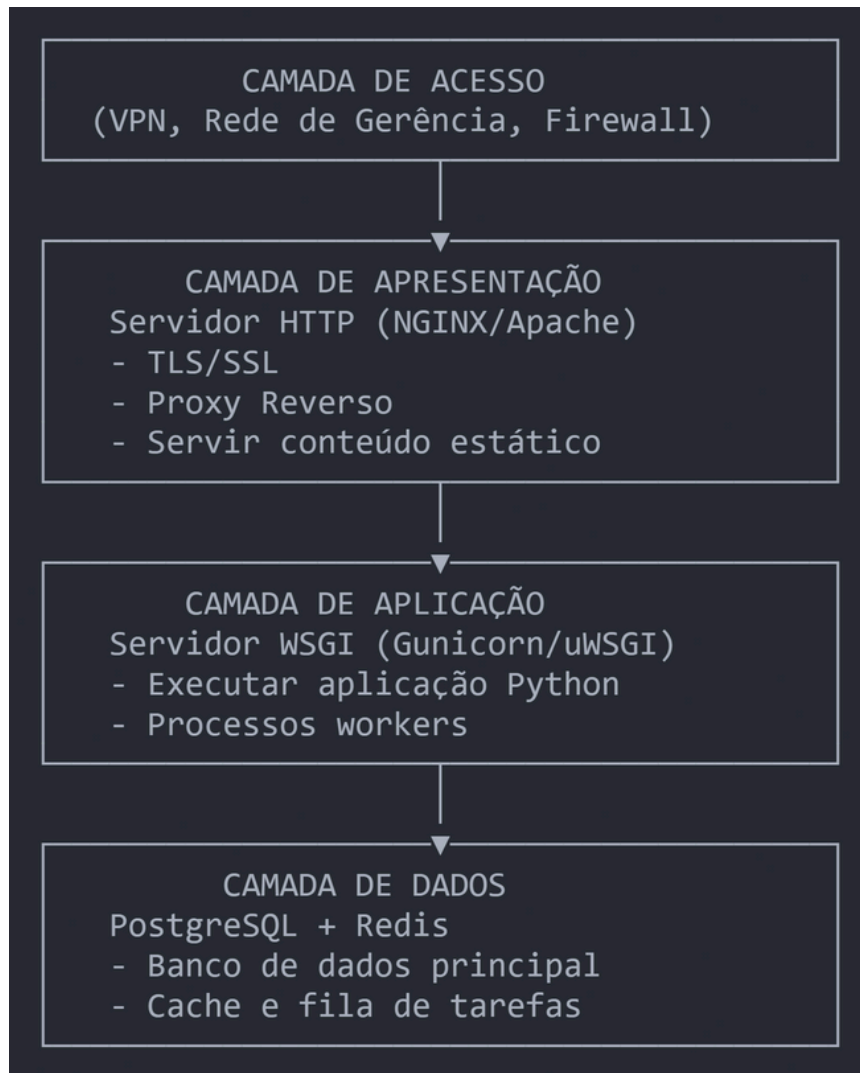
- **Inventário Dinâmico para Ansible:** Em vez de manter arquivos de inventário estáticos, o Ansible pode consultar a API do NetBox em tempo real para obter a lista de dispositivos a serem configurados. A coleção `netbox.netbox` para Ansible facilita essa integração ³.
- **Geração de Configuração:** Scripts (em Python, por exemplo) podem usar os dados do NetBox para gerar configurações de dispositivos de forma padronizada e livre de erros, usando templates (como Jinja2).
- **Validação de Rede:** A automação pode comparar o estado real da rede (coletado de dispositivos) com o estado desejado (documentado no NetBox) para identificar discrepâncias e garantir a conformidade.
- **Webhooks e Automação Orientada a Eventos:** O NetBox pode enviar webhooks para sistemas externos sempre que um objeto é criado, atualizado ou deletado. Isso permite a criação de fluxos de automação reativos. Por exemplo, ao adicionar uma nova VLAN no NetBox, um webhook pode acionar um playbook Ansible para configurar essa VLAN nos switches relevantes.

Implementando o NetBox de Forma Segura

A segurança da sua fonte da verdade é primordial. Uma implementação segura do NetBox segue as melhores práticas de segurança de aplicações web.

Arquitetura de Implantação

Uma instalação típica e segura do NetBox consiste em múltiplas camadas:



Hardware Recomendado Por Porte da Organização

Para organizações de pequeno porte, com algo entre 10 e 50 dispositivos de rede, um servidor virtualizado com configurações simples é suficiente para executar bem a aplicação e suas dependências de software:

CPU: 2 núcleos (2.0 GHz+)

RAM: 4GB

Disco: 50GB SSD

Rede: 100 Mbps

SO: Ubuntu 22.04 LTS / RHEL 8+ / Debian 11+

Já para organizações de médio porte, com algo entre 50 e 500 dispositivos de rede, já é mais interessante utilizar um servidor dedicado, e se possível, um outro apenas para o banco de dados:

CPU: 4-8 núcleos (2.5 GHz+)

RAM: 8-16GB

Disco: 100GB SSD

Rede: 1 Gbps

SO: Ubuntu 22.04 LTS / RHEL 8+ / Debian 11+

Considerar: Separação do banco de dados em servidor dedicado

Para organizações de grande porte, com mais de 500 dispositivos de rede, um cluster de servidores é uma opção mais segura e flexível, garantindo uma melhor disponibilidade dos serviços:

Aplicação:

- **CPU:** 8+ núcleos (3.0 GHz+)
- **RAM:** 16-32GB
- **Disco:** 100GB SSD NVMe

Banco de Dados (separado):

- **CPU:** 8+ núcleos
- **RAM:** 32-64GB
- **Disco:** 200GB+ SSD NVMe (RAID 10)

Load Balancer: HAProxy ou NGINX

Backup: Storage dedicado com replicação

Dependências de Software

- Python: 3.10 ou superior
- PostgreSQL: 13+ (recomendado 14 ou 15)
- Redis: 6.2+
- NGINX: 1.18+ ou Apache 2.4+
- Sistema Operacional: Kernel 5.x+

Plano de Migração Progressiva Dos Dados:

Semana 1-2: Auditoria Inicial

- └─ Planilhas de inventário de equipamentos
- └─ Diagramas de rede
- └─ Planilhas de alocação de IPs e sub-redes
- └─ Documentação de VLANs e segmentação
- └─ Contratos e SLAs de circuitos WAN

Semana 3-4: IPAM Básico

- └─ Redes principais (/16, /24)
- └─ VLANs críticas
- └─ Reservas de IPs importantes

Semana 5-6: Topologia Física

- └─ Sites e localizações
- └─ Racks e elevações
- └─ PDUs e infraestrutura de energia

Semana 7-8: Dispositivos Críticos

- └─ Core switches
- └─ Firewalls
- └─ Roteadores WAN
- └─ Load balancers

Mês 3: Expansão de Dispositivos

- └─ Distribution switches
- └─ Access switches
- └─ Wireless controllers

Mês 4: Conectividade

- └─ Cabos e patch panels
- └─ Conexões entre dispositivos
- └─ Transceivers e módulos

Mês 5: Serviços Avançados

- └─ Circuitos WAN
- └─ Virtualização (VMs)
- └─ Tenancy e separação lógica

Recomendações de Segurança

- **Não Exponha o NetBox à Internet Pública:** O NetBox deve ser tratado como uma ferramenta de gerenciamento interno. O acesso deve ser restrito à sua rede de gerência, preferencialmente via VPN ⁴.
- **Use HTTPS:** Configure seu servidor HTTP (NGINX/Apache) para usar TLS, garantindo que toda a comunicação com o NetBox seja criptografada.
- **Segregação de Contas:** Não permita o compartilhamento de contas de usuário. Cada administrador deve ter seu próprio login para garantir a rastreabilidade das ações.
- **Controle de Acesso ao Banco de Dados:** O banco de dados PostgreSQL deve ser configurado para aceitar conexões apenas do(s) servidor(es) NetBox. Use senhas fortes e, se possível, autenticação baseada em certificados.
- **Manter Tudo Atualizado:** Mantenha o sistema operacional, o NetBox e todas as suas dependências (Python, PostgreSQL, Redis) atualizados com os últimos patches de segurança.

Configuração Segura (configuration.py)

O arquivo **configuration.py** é o coração da configuração do NetBox. Atenção especial aos seguintes parâmetros:

SECRET_KEY: Deve ser uma chave longa e gerada aleatoriamente. Não a compartilhe e não a comite em repositórios de código.

CSRF_COOKIE_SECURE e SESSION_COOKIE_SECURE: Defina ambos como True em um ambiente de produção que usa HTTPS.

HSTS (HTTP Strict Transport Security): Os parâmetros **SECURE_HSTS_SECONDS**, **SECURE_HSTS_INCLUDE_SUBDOMAINS** e **SECURE_HSTS_PRELOAD** forcem o uso exclusivo de HTTPS pelo navegador, reduzindo riscos como SSL stripping. Ative apenas após validar corretamente o TLS em produção.

ALLOWED_HOSTS: Defina explicitamente os nomes de domínio ou endereços IP que podem ser usados para acessar sua instância do NetBox. Evite usar "*" em produção.

X_FRAME_OPTIONS e **SECURE_CONTENT_TYPE_NOSNIFF:** impedem que o NetBox seja carregado em iframes e reduzem riscos de ataques baseados em manipulação de conteúdo.

```
# ALLOWED_HOSTS: Lista explícita de domínios/IPs permitidos
# NUNCA use "*" em produção
ALLOWED_HOSTS = [
    'netbox.suaempresa.com',
    '10.0.0.100',
]

# SECRET_KEY: Gerar chave aleatória forte
# Método de geração:
# python3 -c "from django.core.management.utils import get_random_secret_key;
# print(get_random_secret_key())"
SECRET_KEY = 'sua_chave_secreta_muito_longa_e_aleatoria_aqui'

# Forçar cookies seguros (HTTPS apenas)
CSRF_COOKIE_SECURE = True
SESSION_COOKIE_SECURE = True

# HSTS (HTTP Strict Transport Security)
SECURE_HSTS_SECONDS = 31536000 # 1 ano
SECURE_HSTS_INCLUDE_SUBDOMAINS = True
SECURE_HSTS_PRELOAD = True

# Forçar HTTPS
SECURE_SSL_REDIRECT = True

# Prevenir clickjacking
SECURE_CONTENT_TYPE_NOSNIFF = True
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = 'DENY'

# Session timeout (em segundos)
SESSION_COOKIE_AGE = 3600 # 1 hora

# Timezone
TIME_ZONE = 'America/Recife'
USE_TZ = True

# Idioma
LANGUAGE_CODE = 'pt-br'

# Formatos brasileiros
DATE_FORMAT = 'd/m/Y'
SHORT_DATE_FORMAT = 'd/m/Y'
TIME_FORMAT = 'H:i:s'
SHORT_TIME_FORMAT = 'H:i'
DATETIME_FORMAT = 'd/m/Y H:i:s'
SHORT_DATETIME_FORMAT = 'd/m/Y H:i'
```

Configuração de Logs (configuration.py)

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'handlers': {
        'file': {
            'level': 'INFO',
            'class': 'logging.handlers.RotatingFileHandler',
            'filename': '/var/log/netbox/netbox.log',
            'maxBytes': 1024 * 1024 * 10, # 10 MB
            'backupCount': 5,
        },
    },
    'loggers': {
        'django': {
            'handlers': ['file'],
            'level': 'INFO',
            'propagate': True,
        },
        'netbox': {
            'handlers': ['file'],
            'level': 'DEBUG',
        },
    },
}
```

Conclusão

O NetBox é muito mais do que uma ferramenta de documentação; é uma plataforma estratégica que serve como a fundação para a automação e a operação inteligente de redes. Ao centralizar informações críticas de IPAM e DCIM, ele elimina a ambiguidade e os erros causados por dados descentralizados. Implementar o NetBox de forma segura e integrá-lo aos seus fluxos de automação não apenas aumenta a eficiência operacional, mas também fortalece a postura de segurança e a resiliência da sua infraestrutura.

Referências

¹ NetBox Labs. "World's Leading Network Source of Truth (NSoT)".

² Red Hat. "Using NetBox for Ansible Source of Truth".

³ Ansible. "netbox.netbox Collection".

⁴ NetBox Community. "Security Overview".

Apêndice A - Checklist de Validação

- ❑ **SECRET_KEY** é longa, aleatória e única.
- ❑ **SECRET_KEY** não está em repositórios Git.
- ❑ **ALLOWED_HOSTS** contém apenas hosts legítimos (sem "**")
- ❑ **CSRF_COOKIE_SECURE = True**
- ❑ **SESSION_COOKIE_SECURE = True**
- ❑ **SECURE_HSTS_SECONDS** configurado (mínimo 31536000)
- ❑ **SECURE_HSTS_INCLUDE_SUBDOMAINS = True**
- ❑ **X_FRAME_OPTIONS = 'DENY'**
- ❑ **SESSION_COOKIE_AGE** apropriado para seu ambiente
- ❑ **TIME_ZONE** correto para sua localização
- ❑ Certificado SSL válido e configurado
- ❑ HTTPS funcionando corretamente
- ❑ Redirecionamento HTTP → HTTPS ativo
- ❑ Testes de acesso realizados
- ❑ Logs configurados e funcionando