

NetBox For Newbies

NetSIG Presentation



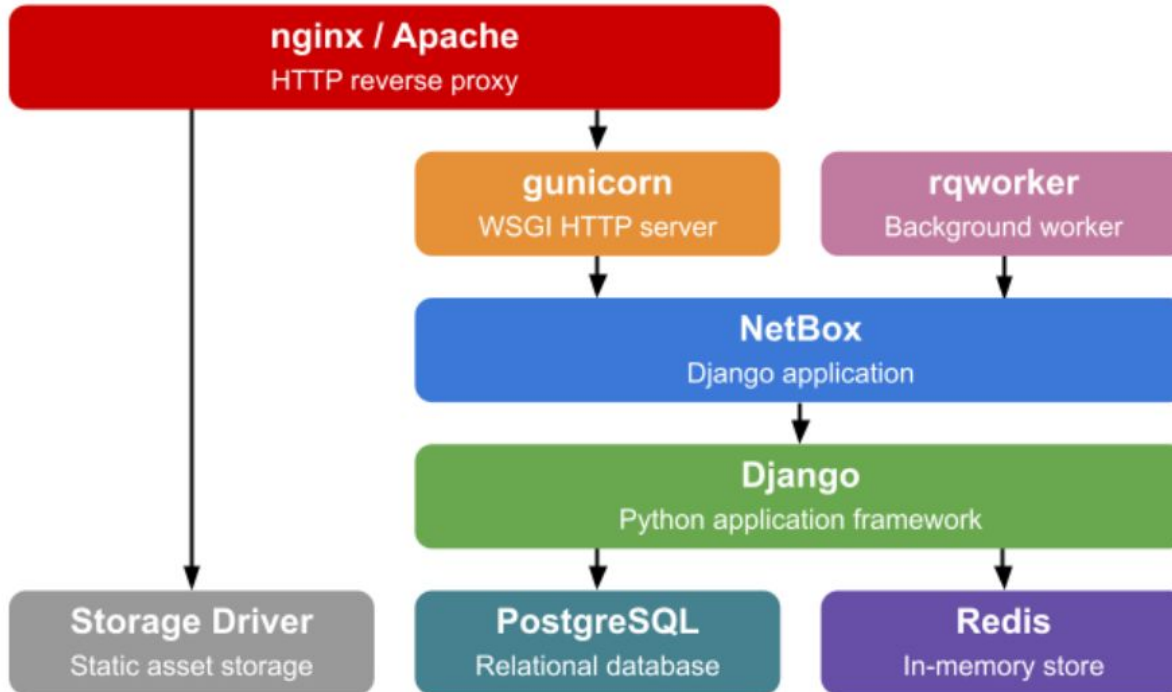
What is NetBox?



- Network inventory solution
- **Goal:** Be the network "**Source of Truth**"
 - ... And drive network automation
- Provides a physical and logical model of your entire network
 - Represents the desired state (not necessarily the operational state)
 - Purpose-built for IPAM and DCIM
 - Caters to network engineers and operators
- Easy to Extend
 - Plugins, custom data modeling, and APIs (REST, GraphQL)
- Open Source
 - No vendor lock-in
 - Community-driven
- Python-Based
 - Built on Django



NetBox Architecture



Key Features



- Physical inventory management
 - Devices, power, cabling, etc.
- Logical inventory management
 - IPs, VRFs, VLANs, VPNs, and ASNs, etc.
- IPAM and automated IP provisioning
- Customizability
 - Custom fields for data model extension
 - Custom validation & protection rules
 - Community supported plugin framework to add custom functionality
- Event-driven scripts & webhooks
- Network device configuration rendering
- Global search



NetBox is Not



- Network monitoring
- DNS server
- RADIUS server
- Configuration management
- However, NetBox can be used to drive these external systems



Demo - Local Docker Setup



- Note - Requires **docker** and **docker compose**

```
$ git clone https://github.com/netbox-community/netbox-docker.git
```

```
$ cd netbox-docker/
```

```
$ docker compose up -d
```

- Browse to <http://localhost:8000>
 - Username: **admin**
 - Password: **admin**
 - If you are having authentication issues, try this:

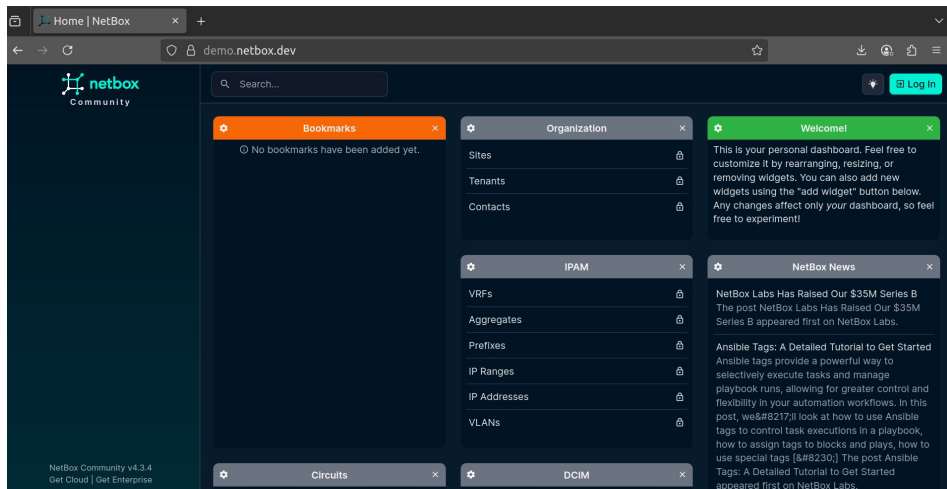
```
$ docker compose exec netbox \  
    /opt/netbox/netbox/manage.py changepassword admin
```



Demo - Alternate Setup



- Public NetBox demo site available on the Internet
- <https://demo.netbox.dev/login/>
 - Username: **admin**
 - Password: **admin**



Demo - Manual Provisioning



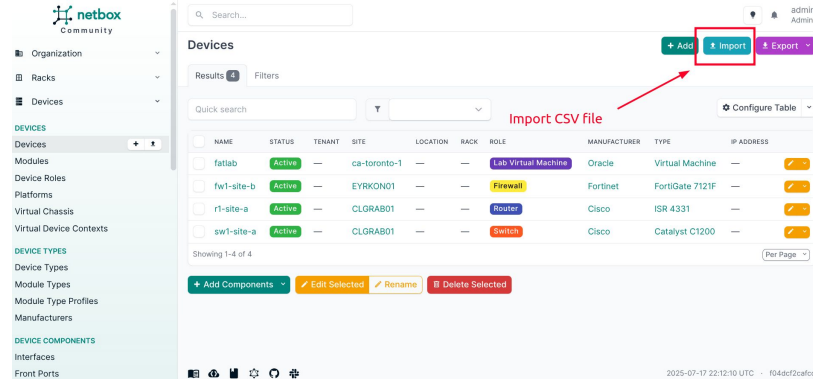
- NetBox documents both physical and logical components of your network
 - Physical: Racks, Devices, Connections, etc.
 - Logical: IPs, VPNs, VMs, etc.
 - <http://localhost:8000/>

A screenshot of the NetBox web interface. The left sidebar shows a navigation menu with categories like Organization, Racks, Devices, Connections, Wireless, IPAM, VPN, Virtualization, Circuits, Power, Provisioning, Customization, Operations, and Admin. The main content area has a search bar at the top and a user profile for 'admin'. Below the search bar, there are several widgets: 'Organization' showing counts for Sites (26), Tenants (13), and Contacts (3); 'Welcome!' with instructions on customizing the dashboard; 'NetBox News' with recent articles; 'DCIM' showing counts for Sites (26), Racks (45), Device Types (22), Devices (125), and Cables (115); 'IPAM' showing counts for VRFs (8), Aggregates (4), Prefixes (93), IP Ranges (5), IP Addresses (183), and VLANs (64); and 'Virtualization' showing counts for Clusters (32) and Virtual Machines (180).

Demo - CSV Imports



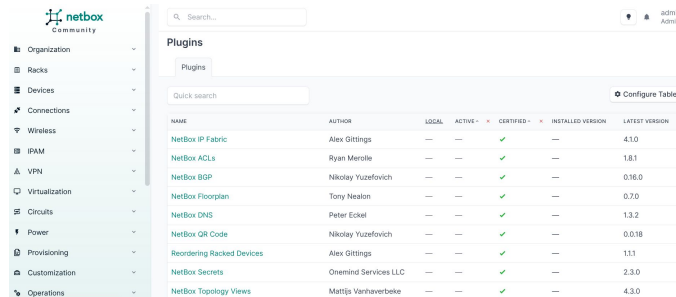
- CSV file imports are supported
- Try importing a list of devices:
 - <http://localhost:8000/dcim/devices/>
- Can be used to bulk load multiple inventory objects
- However, more complex load jobs should be scripted



Demo - Plugins



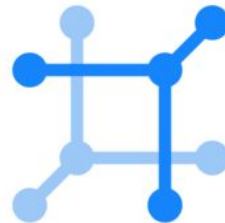
- Packaged Django apps that can be installed alongside NetBox
- Provides custom functionality not present in core NetBox application
- Plugin Architecture allows for:
 - New data models
 - New URLs and forms in the webapp
 - New navigation menu items for plugin content
- Plugins Available:
 - <http://localhost:8000/core/plugins/>



NAME	AUTHOR	LOCAL	ACTIVE	CERTIFIED	INSTALLED VERSION	LATEST VERSION
NetBox IP Fabric	Alex Gittings	—	—	✓	—	4.1.0
NetBox ACLs	Ryan Merille	—	—	✓	—	1.8.1
NetBox BGP	Nikolay Yuzefovich	—	—	✓	—	0.16.0
NetBox Floorplan	Tony Nealon	—	—	✓	—	0.7.0
NetBox DNS	Peter Eckel	—	—	✓	—	1.3.2
NetBox QR Code	Nikolay Yuzefovich	—	—	✓	—	0.0.18
Reordering Racked Devices	Alex Gittings	—	—	✓	—	1.1.1
NetBox Secrets	Onemind Services LLC	—	—	✓	—	2.3.0
NetBox Topology Views	Mattja Vanhaverbeke	—	—	✓	—	4.3.0



Feature - Configuration Rendering



- Render network configurations on the fly using jinja2 templating
- Use inventory updates to push network device configurations
- Example:

```
! Basic Cisco IOS config for {{ device.name }}

hostname {{ device.name }}

{% for iface in device.interfaces.all() %}
interface {{ iface.name }}
  description {{ iface.description or "Interface for " ~ device.name }}
  {%- for ip in iface.ip_addresses.all() %}
    {%- if ip.family == 4 %}
    ip address {{ ip.address }}
    {%- elif ip.family == 6 %}
    ipv6 address {{ ip.address }}
    {%- endif %}
  {%- endfor %}
{% endfor %}
```



Demo - Data Extraction



- NetBox supports a simple REPL (CLI)
- This can be used for ad-hoc queries, reports, scripts, etc.
- Example:

```
$ docker exec -it netboxdemo-netbox-1 python3 manage.py nbshell
```

```
...  
>>> from dcim.models import Site  
>>> all_sites = Site.objects.all()  
>>> for site in all_sites:  
...     print(site.name)
```

- Syntax also supported using the NetBox API



Demo - Ansible Inventory



- NetBox supports seamless ansible integration

```
$ ansible-galaxy collection install netbox.netbox
```

```
$ ansible-galaxy collection list | grep netbox
```

```
$ cat netbox_inventory.yml
```

```
plugin: netbox.netbox.nb_inventory
```

```
api_endpoint: https://your-netbox-url.com
```

```
token: YOUR_NETBOX_API_TOKEN
```

```
$ ansible-inventory -i netbox_inventory.yml --graph
```

```
$ ansible all -i netbox_inventory.yml -m ping
```



Feature - Custom Scripts



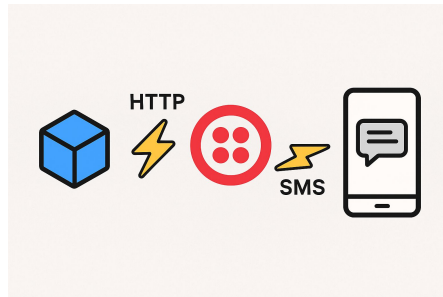
- Allows users to execute custom logic from within the NetBox UI
- Used to accomplish tasks such as:
 - Fetching data from an external source and import it to NetBox
 - Updating objects with invalid or incomplete data
 - Validating the integrity of data
 - Automatically populating new devices and device attributes
- GUI interface to drive custom scripts:
 - <http://localhost:8000/extras/scripts/>



Demo - Event Rules



- Offers a powerful way to automate tasks
- "Events" in NetBox can be processed by Event Rules
- These Event Rules can trigger webhooks, notifications or scripts
- Benefits
 - **Automation:** Trigger configuration jobs in Ansible, Nornir, etc.
 - **ChatOps:** Push real-time change alerts to Slack, G-Chat, etc.
 - **Data Sync:** Keep other systems like DNS instantly updated.
- <http://localhost:8000/extras/event-rules/>



NetBox Pros & Cons



Pros

- ✓ One single inventory database - Clears up spreadsheet sprawl
- ✓ Integration and automation are made easy
- ✓ Self-hosting is trivial

Cons

- ✗ Documentation is light - detective work required
- ✗ Challenging to troubleshoot custom script issues
- ✗ Overkill for most hobbyists



Summary

- NetBox is the single source of truth for your network infrastructure
- It models physical assets - racks, devices, cables
- It also models logical resources - IP addresses, VLANs
- It offers features that drive network automation
 - Keeps your live network synchronized with the desired state



Resources



- **Main NetBox Site** - <https://netboxlabs.com/>
- **NetBox on GitHub** - <https://github.com/netbox-community/netbox>
- **NetBox Docs** - <https://NetBoxlabs.com/docs/NetBox/>
- **Public NetBox Demo** - <https://demo.NetBox.dev/>
- **NetBox Video Series** - <https://www.youtube.com/@NetBoxLabs>
- **NetBox Community Slack** - <https://netdev.chat/>
- **Presentation Slides & Demo** - <https://github.com/netserf/netsig-presentation-netbox-for-newbies>



Questions

