Work and Progress

netbox migration

zielserver: SLES 15 SP7, kein direkter internetzugang -> nur über proxy mit einzeln freigeschalteten adressen.

setup

- postgresql 17 (datenbank)
- valkey 8 (redis-kompatibel, caching/background jobs)
- nginx (reverse proxy davor)
- gunicorn (wsgi server für die django-app)

→ das ist die von netbox offiziell empfohlene stack-kombi für native installs (kein docker)

warum kein podman/container? hatte ich erst überlegt, aber:

- plugins gehen über pip → bei container müsste ich bei jedem plugin ein neues image bauen
- hätte den proxy-freigabeprozess (siehe unten) nur noch mehr belastet → native installation gewählt

python versions-ärger

server hatte schon python 3.14 drauf (SLES bringt das mit). netbox braucht offiziell min. 3.10, aber:

- `strawberry-graphql` (graphql dependency von netbox) ist mit 3.14 nicht kompatibel
- migration bricht mit laufzeitfehler in der feld-initialisierung ab

→ fix: einfach die von SLES mitgelieferte python 3.11 genommen statt 3.14. lief dann sauber durch.

lesson learned: neueste version =! automatisch die beste

---

proxy shit

server ist nicht air-gapped --> proxy (freischaltung durch ticket)

genehmigungsprozess:

1. ticket an proxy-team → quellsystem, zieladresse, port 443 ohne auth
2. weil proxy SSL terminiert: zusätzliches ticket an infosec → SSL-bypass + freischaltung für `application/zip` und `application/gzip` mime-types
   - dazu noch nen risikobogen ausfüllen (auswirkung bei nichtverfügbarkeit, alternativen geprüft etc.)

konkrete probleme die aufgetreten sind:

problem	ursache	fix
407 bei `pip install`, curl ging aber	proxy-var nicht dauerhaft gesetzt / sudo killt env	in `/etc/environment` eintragen + `sudo -E` nutzen
407 bei internen suse-manager repos	interne adressen liefen fälschlich über externen proxy	intern in `no_proxy` aufnehmen
podman pull auf ghcr.io scheitert trotz freigabe	freigabe deckte nur registry-api ab, nicht den separaten blob-storage (`pkg-containers.githubusercontent.com`)	extra domain nachbeantragt
netbox source code nicht über pypi verfügbar	netbox gibt's nur als github release, kein pypi paket	einmal extern von jemandem mit internet geladen, per datenträger über sprungserver + SynaMan reingebracht

→ takeaway: jedes neue plugin = potenziell neues ticket

---

installation

- `psycopg` (postgres adapter) will kompilieren, braucht `pg_config` → fehlte, nachinstalliert: `postgresql17-server-devel`
- Pillow braucht jpeg dev-header → `libjpeg8-devel` nachinstalliert
- postgres/valkey wollten nicht auf `localhost` hören → server hat das erst auf `::1` (ipv6) aufgelöst, dienste lauschten aber nur auf ipv4 loopback → fix: überall `127.0.0.1` statt `localhost` in der config
- postgres auth: `ident` methode ging nicht (mein domänen-user ≠ db-user `netbox`) → auf `md5` umgestellt
- valkey config datei war gar nicht da, musste ich selbst minimal anlegen (nur loopback)

systemd service unter `netbox.service` angelegt → hat ausversehen die schon vorhandene datei vom netzwerkdienst `wicked` überschrieben. nach neustart → server komplett weg vom netz, nur noch über physische konsole erreichbar. □□ → merke: bei eigenen systemd units immer eindeutige namen die nicht mit bestehenden diensten kollidieren können

---

plugins

dns (F1)

- erst falsches paket installiert (`netbox-dns`) → passt nicht zur plugin-api von netbox 4.2, nutzt noch das alte `extras.plugins` modul das es nicht mehr gibt
- richtiges paket: `netbox-plugin-dns` v1.2.14 → über `PLUGINS` aktiviert
- microsoft-typische namen mit unterstrich (z.b. `_msdcs.domain.de`) sind nach RFC 1034 eigentlich nicht erlaubt → musste `tolerate_underscores_in_labels` setzen

octoDNS test:

- externes tool, liest zonen aus netbox (via `NetBoxDNSProvider` / `octodns-netbox-dns` paket) und kann die an andere systeme syncen
- testzone `test.lokal` angelegt (ns + a record), erfolgreich als yaml exportiert
- **nicht getestet:** gegen echten microsoft-dns server, dafür hätte ich nen provider gebraucht den's für das ökosystem nicht gab + schreibzugriff auf produktive windows infra → F1 = architektur-mäßig belegt, aber nicht produktiv verifiziert

dhcp (F2) - hat einfach nicht geklappt

drei anläufe, alle gescheitert:

1. `netbox-plugin-dhcp` (offiziell) → braucht netbox 4.5 + python 3.12, hab aber 4.2/3.11
2. `netbox-kea-dhcp` → gibt's nicht mehr auf pypi
3. `netbox-kea-ng` → installiert sich, aber migration bricht ab, weil's `core.exceptions.JobFailed` importiert, was's in 4.2.6 noch nicht gibt

→ fazit: DNS-plugin-ökosystem ist für die netbox version ok, DHCP ist einfach noch nicht reif genug. kein grundsätzliches "geht nicht", eher momentaufnahme.

---

was ich sonst noch getestet hab

- **rechte (F5):** testgruppe mit constraint (`tenant__name`) → nur präfixe vom zugewiesenen mandanten sichtbar. funktioniert.
- **vlan (F6):** gleiche VID (100) in zwei verschiedenen vlan-groups → geht problemlos. gleiche VID nochmal in derselben group → wird zurückgewiesen. genau wie erwartet.
- **custom fields (F9):** 3 felder angelegt (`domaene`, `sicherheitsbereichs_id`, `svn_id`) auf präfix-objekte. domaene erstmal als textfeld statt choice-set, weil ich die vollständige werteliste aus infoblox nicht hatte.
- **filter/export (F8):** präfixliste nach mandant/status gefiltert, als csv exportiert. läuft.
- **ldap (F7):** `django-auth-ldap` installiert, `REMOTE_AUTH_BACKEND` konfiguriert, verbindungsparameter vorbereitet — **aber nicht live getestet**, hatte keinen service-account und keinen zugriff aufs AD. offener punkt.

---

datenimport

- 572 standorte aus infoblox importiert (region/site group/site mapping)
- 4.262 präfixe importiert, über eigenes python-skript + netbox REST API
- standortzuordnung war nervig weil netbox numerische IDs will, infoblox hatte klartext-kurzbezeichnungen
  - direkt eindeutig zuordenbar: 63,1%
  - über zusatzmatching (kurzbezeichnung ↔ beschreibungsfeld): +17,0% → macht 80,1% insgesamt
  - rest (15,9%) ohne standort importiert, 4,1% hatten gar keine standort-id im quelldatensatz
- encoding-ärger: teile vom datensatz waren windows-1252 statt utf-8 → umlaute in ortsnamen kaputt, musste nachbearbeiten
- ~500 nameserver-einträge + ein paar dns-zonen mit importiert
- **nicht importiert:** einzelne IP-adressen (nur die präfixe/subnetze als ganzes). infoblox hat da schätzungsweise 10.000er bis 100.000er einträge, hätte entweder direkten API-zugriff aufs produktive infoblox gebraucht oder manuellen export pro subnetz — beides nicht machbar im rahmen der arbeit.

---

offene punkte

☐ ldap gegen echtes AD testen (braucht service-account)

☐ einzelne IP-adressen importieren für vollständige skalierbarkeits-aussage

☐ die ~16% unzugeordneten präfixe manuell nacharbeiten

☐ DHCP nochmal probieren sobald netbox 4.5 + python 3.12 möglich ist, oder netbox labs enterprise variante anschauen (soll DHCP vorkonfiguriert für offline-betrieb haben)

Revision #3
Created 2026-06-15 06:32:55 UTC by Denode
Updated 2026-07-14 11:59:16 UTC by Denode