

NetBox

- Utworzenie konfiguracji w docker
 - Pliki yaml do konfiguracji netbox

Utworzenie konfiguracji w docker

Pliki yaml do konfiguracji netbox

Instalację netboxa w dockerze opieramy na instrukcji:

[netboxcommunity/netbox - Docker Image | Docker Hub](#)

Polega na odpowiedniej zmianie pliku override - całość konfiguracji domyślnej jest klonowana z repozytorium git

Treść docker-compose-override.yml ze skonfigurowaną obsługą pluginów oraz połączeniem LDAP - AD

docker-compose-override.yml

```
services:

#Netbox config

netbox:
  image: netbox:latest-plugins
  pull_policy: never
  ports:
    - 8000:8080
  volumes:
    - /docker/apps/netbox/docker/nginx-unit.json:/etc/unit/nginx-unit.json:ro
      volumes:
    - /docker/apps/netbox/configuration:/etc/netbox/config:z,ro
    - /docker/apps/netbox/media:/opt/netbox/netbox/media:rw
    - /docker/apps/netbox/reports:/opt/netbox/netbox/reports:rw
    - /docker/apps/netbox/scripts:/opt/netbox/netbox/scripts:rw
  environment:
    REMOTE_AUTH_ENABLED: "True"
    REMOTE_AUTH_BACKEND: "netbox.authentication.LDAPBackend"
    AUTH_LDAP_SERVER_URI: "ldaps://domain.local"
```

```
AUTH_LDAP_BIND_DN: "CN=srv_netbox,OU=ServiceUsers,OU=Users,DC=domain,dc=local"
AUTH_LDAP_BIND_PASSWORD: "TrudneHasło"
AUTH_LDAP_USER_SEARCH_BASEDN: "OU=Users,OU=BCC,DC=domain,dc=local"
AUTH_LDAP_GROUP_SEARCH_BASEDN: "OU=Groups,OU=BCC,DC=domain,dc=local"
AUTH_LDAP_REQUIRE_GROUP_DN: "CN=netbox_access,OU=Netbox,OU=Groups,DC=domain,dc=local"
AUTH_LDAP_GROUP_TYPE: "NestedGroupOfNamesType"
AUTH_LDAP_IS_ADMIN_DN: "CN=netbox_admins,OU=Netbox,OU=Groups,DC=domain,dc=local"
AUTH_LDAP_IS_SUPERUSER_DN: "CN=netbox_superusers,OU=Netbox,OU=Groups,DC=domain,dc=local"
LDAP_IGNORE_CERT_ERRORS: "true"
```

build:

```
context: .
```

```
dockerfile: Dockerfile-Plugins
```

netbox-worker:

```
image: netbox:latest-plugins
```

```
pull_policy: never
```

netbox-housekeeping:

```
image: netbox:latest-plugins
```

```
pull_policy: never
```

```
restart: unless-stopped
```

#Netbox-Worker config

netbox-worker:

```
restart: unless-stopped
```

#DB config

postgres:

```
volumes:
```

```
- /docker/apps/netbox/postgres:/var/lib/postgresql/data
```

```
restart: unless-stopped
```

#REDIS config

redis:

```
volumes:
```

```
- /docker/apps/netbox/redis:/data
```

```
restart: unless-stopped
```

#REDIS config

```
redis-cache:  
volumes:  
  - /docker/apps/netbox/redis-cache:/data  
restart: unless-stopped
```

Konfiguracja instalacji pluginów. By pluginy instalowały się poprawnie tworzymy plik:

Dockerfile-Plugins

```
FROM netboxcommunity/netbox:latest  
  
COPY ./plugin_requirements.txt /opt/netbox/  
RUN /usr/local/bin/uv pip install -r /opt/netbox/plugin_requirements.txt  
  
# These lines are only required if your plugin has its own static files.  
COPY configuration/configuration.py /etc/netbox/config/configuration.py  
COPY configuration/plugins.py /etc/netbox/config/plugins.py  
RUN DEBUG="true" SECRET_KEY="SecretKEY" \  
  /opt/netbox/venv/bin/python /opt/netbox/netbox/manage.py collectstatic --no-input
```

Dla podmiany SECRET_KEY możemy wykorzystać komendę:

```
openssl rand -base64 48
```

Przykład:

```
openssl rand -base64 48  
4Ip0/V9YQE+3VKw0mf2eCFsPoENW1Mj0iVr3sxdKFZOhDDj0KkcZjfHGGrPEYWeio
```

Lista pluginów które posiadają dodatkowe pliki

Tworzymy plugins_requirements.txt

```
netbox-secrets  
netbox-floorplan-plugin  
netbox-reorder-rack  
netbox_lifecycle
```

Konfiguracja dodatkowych pluginów by aplikacja spróbowała je uruchomić

Edytujemy `./configuration/plugins.py`

```
# Add your plugins and plugin settings here.
# Of course uncomment this file out.

# To learn how to build images with your required plugins
# See https://github.com/netbox-community/netbox-docker/wiki/Using-Netbox-Plugins

# PLUGINS = ["netbox_bgp"]

# PLUGINS_CONFIG = {
#   "netbox_bgp": {
#     ADD YOUR SETTINGS HERE
#   }
# }

PLUGINS = ["netbox_secrets", "netbox_floorplan", "netbox_reorder_rack", "netbox_lifecycle"]
```