Projekt – Homelab

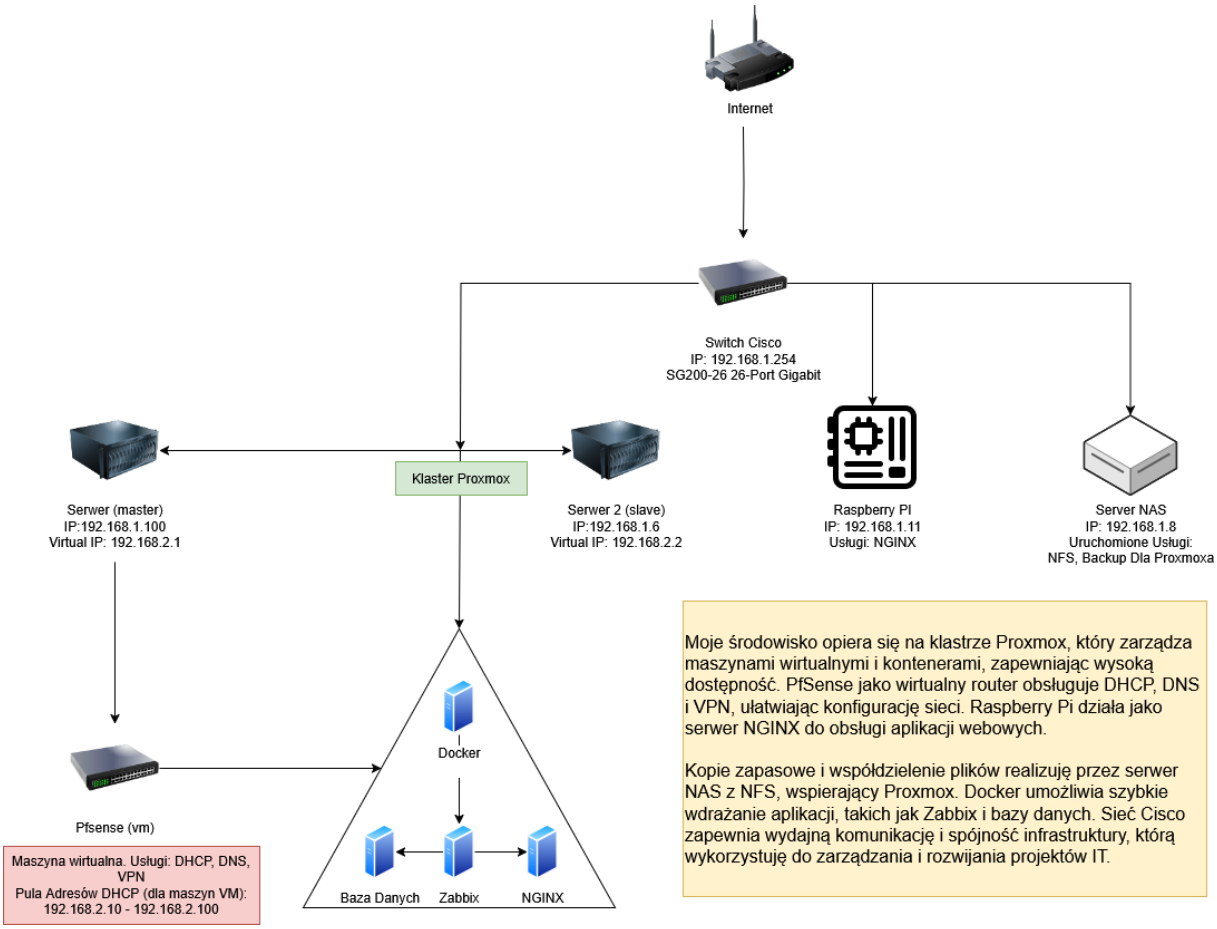
Spis treści

Warszawa, 9 luty 2025

Ver. II

Cel projektu:

Wdrożenie serwerowni do utrzymania aplikacji z użyciem klastra HA.



Moje środowisko opiera się na klastrze Proxmox, który zarządza maszynami wirtualnymi i kontenerami, zapewniając wysoką dostępność. PfSense jako wirtualny router obsługuje DHCP, DNS i VPN, ułatwiając konfigurację sieci. Raspberry Pi działa jako serwer NGINX do obsługi aplikacji webowych.

Kopie zapasowe i współdzielenie plików realizuję przez serwer NAS z NFS, wspierający Proxmox. Docker umożliwia szybkie wdrażanie aplikacji, takich jak Zabbix i bazy danych. Sieć Cisco zapewnia wydajną komunikację i spójność infrastruktury, którą wykorzystuję do zarządzania i rozwijania projektów IT.

Wprowadzenie

Cel dokumentacji

- Opis aktualnej infrastruktury IT, jej elementów oraz zastosowanych technologii.
- Zdefiniowanie sposobu komunikacji pomiędzy poszczególnymi komponentami sieci.
- Udokumentowanie procesów wdrożeń i utrzymania środowiska (CI/CD, kopie zapasowe, zarządzanie konfiguracją).

Zakres

- Opis fizycznej topologii (sprzęt) i logicznej topologii (VLAN, podsieci, usługi).
- Omówienie technologii wirtualizacji (Proxmox, Docker, Kubernetes) i usług sieciowych (DNS, DHCP, VPN, Firewall).
- Opis usług wspierających rozwój oprogramowania i automatyzację (Jenkins, Ansible, GitHub Actions).
- Informacje dotyczące bezpieczeństwa (AD, SSL, RAID, firewall, segmentacja).

Architektura fizyczna:

Router

- Urządzenie brzegowe zapewniające dostęp do Internetu i routing w sieci LAN.
- Dedykowany sprzęt (router ISP) oraz wirtualna maszyna (pfSense) działająca w roli bramy domyślnej dla maszyn wirtualnych.

Switch Cisco (SG200-26)

- Model: SG200-26 26-Port Gigabit
- Adres IP zarządzania: 192.168.1.254
- Odpowiedzialny za rozdzielanie ruchu w sieci LAN i obsługę VLAN-ów.

Serwer NAS

- IP: 192.168.1.8
- Uruchomione usługi: NFS, backup dla Proxmoxa, przechowywanie plików.
- Konfiguracja RAID (zależnie od Twoich ustawień, np. RAID 5/6/10) dla zwiększenia bezpieczeństwa danych.

Serwery (Proxmox Cluster)

- **Serwer Master**: IP: 192.168.1.100, Virtual IP: 192.168.2.1
- **Serwer Slave**: IP: 192.168.1.6, Virtual IP: 192.168.2.2
- Działają w klastrze Proxmox (High Availability).

Raspberry Pi

- IP: 192.168.1.11
- Usługi: NGINX (np. reverse proxy, serwer aplikacji web, Home Assistant).

UPS

- Zapewnia zasilanie awaryjne dla krytycznych elementów sieci (serwery, NAS, router).

Architektura logiczna i VLAN-y

1. **Podsieci**
    - Główna podsieć LAN: 192.168.1.0/24
    - Podsieć dla maszyn wirtualnych (pula DHCP pfSense): 192.168.2.0/24 (np. 192.168.2.10–192.168.2.100).
    - (Ewentualnie) dodatkowe VLAN-y dla IoT, strefy DMZ, kamer IP itp. – zależnie od potrzeb.
2. **VLAN-y** (przykładowa propozycja)
    - VLAN 10: Sieć produkcyjna (serwery i urządzenia główne, 192.168.1.0/24).
    - VLAN 20: Maszyny wirtualne, strefa DMZ, testy (192.168.2.0/24).
    - VLAN 30: IoT (urządzenia Raspberry Pi, Home Assistant itp.), jeśli chcesz je odseparować.
3. **Routing między VLAN-ami**
    - Realizowany przez pfSense (zainstalowane jako VM na Proxmox) lub przez główny router.
    - Możliwe reguły firewall ograniczające dostęp z VLAN-u IoT do sieci produkcyjnej.

Usługi sieciowe

1. **pfSense (VM)**
    - Usługi:
        - **DHCP** – rozdaje adresy w 192.168.2.0/24.

        - **DNS** – (BIND, AD Guard lub DNS Resolver pfSense) z możliwością split-DNS.
        - **VPN** – konfiguracja połączeń zdalnych (np. OpenVPN/IPsec/WireGuard).
        - **Firewall** – główna zapora ruchu na granicy VLAN-ów i Internetu.
2. **Windows Server** (jeśli występuje)
    - Rola w infrastrukturze: kontroler domeny Active Directory, DNS, DHCP (jeżeli nie pfSense).
    - Zarządzanie użytkownikami i grupami AD, integracja z usługami, Single Sign-On.
3. **Linux**
    - Dystrybucje serwerowe (Debian/Ubuntu/CentOS) z rolami:
        - **DNS (BIND)** – alternatywnie do pfSense/Windows Server.
        - **Monitoring** (Zabbix, Nagios) – w Twoim przypadku Zabbix w Dockerze.
        - **Kali Linux** – testy penetracyjne i audyty bezpieczeństwa (opcjonalnie w VM).
4. **AD Guard**
    - Filtrowanie reklam i złośliwych domen w sieci lokalnej.
    - Może być zainstalowane na Raspberry Pi lub jako kontener Docker.

Wirtualizacja i kontenery

1. **Proxmox**
    - Klastrowanie (HA) – serwer Master (192.168.1.100) i Slave (192.168.1.6).
    - Przykładowe VM: pfSense, Windows Server, Linux (bazy danych, Jenkins, GitLab/GitHub Actions runner itp.).
    - Kontenery LXC – szybsza wirtualizacja lekkich usług.
2. **Docker**
    - Uruchomiony na klastrze Proxmox (lub dedykowany węzeł Docker).
    - Kontenery:
        - **Zabbix** – monitoring.
        - **NGINX** – front-end/proxy dla aplikacji webowych.
        - **Baza danych** (MySQL/PostgreSQL) – przechowywanie danych aplikacji i Zabbixa.
3. **Kubernetes (k8s)**

- Opcjonalnie wykorzystywany do orkiestracji kontenerów.
- Może działać na klastrze Proxmox jako węzeł z kilkoma workerami.
- Ułatwia skalowanie i zarządzanie usługami kontenerowymi.

Usługi DevOps i CI/CD

1. **Jenkins**
   - Zainstalowany w VM lub kontenerze Docker.
   - Używany do automatyzacji procesów build/test/deploy aplikacji.
2. **Ansible**
   - Zarządzanie konfiguracją serwerów i kontenerów.
   - Playbooki do szybkiego wdrażania usług, konfigurowania systemów i patchowania.
3. **GitHub Actions**
   - Alternatywa lub uzupełnienie dla Jenkins.
   - CI/CD w chmurze GitHub, integracja z repozytoriami kodu.
   - Może współpracować z infrastrukturą on-premise (self-hosted runners).

Aplikacje webowe i bazy danych

1. **LAMP/LEMP stack**
   - **NGINX/Apache**: hosting stron i aplikacji (Raspberry Pi, kontenery w Dockerze).
   - **MySQL/PostgreSQL**: bazy danych dla Zabbixa, aplikacji webowych.
   - **PHP/Python**: środowisko uruchomieniowe dla aplikacji.
2. **NGINX na Raspberry Pi**
   - Może pełnić rolę reverse proxy lub serwera statycznych treści.
   - Możesz dodać certyfikaty SSL (np. Let's Encrypt).

Bezpieczeństwo i kopie zapasowe

1. **Firewall**
   - PfSense (policy-based routing, NAT, VPN, blokady ruchu przychodzącego).

- Dodatkowe reguły na Switchu Cisco (ACL) lub Routerze (jeśli wspiera).

2. **SSL/TLS**
   - Certyfikaty SSL wystawiane przez Let's Encrypt (lub własne CA, zależnie od potrzeb).
   - Wykorzystanie protokołów HTTPS i zabezpieczenie panelu administracyjnego (Proxmox, pfSense).
3. **Active Directory**
   - Zarządzanie użytkownikami, politykami grup (GPO), uwierzytelnianiem do stacji roboczych, serwerów, usług.
   - Integracja AD z Jenkins, Docker Registry itp. (jeśli wymaga loginu domenowego).
4. **Kopie zapasowe (NAS, NFS)**
   - Serwer NAS: 192.168.1.8, usługi NFS jako storage dla VM Proxmoxa.
   - Regularne snapshoty maszyn wirtualnych w klastrze Proxmox.
   - Kopie zapasowe plików i baz danych na NAS (lub w chmurze).
5. **RAID**
   - Konfiguracja RAID (w NAS lub w serwerach) zapewniająca redundancję dysków.

Monitoring i diagnostyka

1. **Zabbix**
   - Kontener Docker (komponenty: Zabbix Server, Zabbix Frontend, Agent).
   - Monitorowanie:
     - Hosty fizyczne (Serwery Proxmox, NAS, Raspberry Pi).
     - Usługi sieciowe (ping, HTTP, SNMP).
     - Wskaźniki wydajności (CPU, RAM, obciążenie sieci).
2. **Logi systemowe**
   - Centralizacja logów (np. w Elastic Stack lub syslog na dedykowanej VM).
   - Analiza logów pod kątem bezpieczeństwa i błędów.
3. **Testy penetracyjne (Kali Linux)**
   - Analiza bezpieczeństwa sieci, usług, aplikacji.
   - Audyty i raporty poprawek.

Zarządzanie infrastrukturą

1. **Ansible**
    - Playbooki do automatyzacji konfiguracji serwerów Proxmox, Docker, pfSense.
    - Możliwa integracja z Jenkins (uruchamianie playbooków po zatwierdzeniu zmian w repozytorium).
2. **CI/CD**
    - Pipeline'y w Jenkins lub GitHub Actions do wdrażania aplikacji na środowisko testowe/produkcyjne.
    - Integracja z Docker (budowa obrazów), Kubernetes (deployment), serwerem NAS (przechowywanie artefaktów).
3. **Git**
    - Repozytoria kodu i konfiguracji (GitHub/GitLab lub serwer lokalny Git).
    - Kontrola wersji skryptów Ansible, definicji Jenkins pipeline, Dockerfile itp.

Rozwój i integracje AI

1. **AI**
    - Możliwe wykorzystanie modeli AI/ML do analizy logów (np. anomalii w ruchu sieciowym) lub wsparcia projektów.
    - Kontenery z frameworkami AI (PyTorch, TensorFlow) – przydatne w środowiskach testowych.
2. **Integracja z Home Assistant**
    - Inteligentna automatyka domowa, uczenie maszynowe w celu optymalizacji zużycia energii, planowania zadań.