Projekt: Rozwiązanie Serwer-Klient z Centralnym Zarządzaniem

Spis treści

---

1. Wstęp teoretyczny

1.1 Architektura

W tym projekcie zostało przedstawienie rozwiązania klient - serwer. Serwery centralizują zarządzanie zasobami, bezpieczeństwem i usługami, podczas gdy klienci korzystają z tych zasobów w sposób kontrolowany i bezpieczny.

Infrastruktura IT wymaga integracji następujących elementów:

- **Zarządzanie tożsamością i dostępem** - centralne uwierzytelnianie i autoryzacja
- **Usługi sieciowe** - DHCP, DNS, routing
- **Bezpieczeństwo** - firewall, VPN, szyfrowanie
- **Udostępnianie zasobów** - pliki, drukarki, aplikacje web
- **Backup i archiwizacja** - ochrona danych przed utratą
- **Narzędzia deweloperskie** - systemy kontroli wersji, CI/CD

1.2 Wybrane technologie

Projekt wykorzystuje następujące technologie:

- **Ubuntu 22.04 LTS** - stabilny system operacyjny z długoterminowym wsparciem
- **FreeIPA** - rozwiązanie zarządzania tożsamością kompatybilne z Active Directory
- **Samba** - udostępnianie plików w sieci mieszanej Windows/Linux
- **OpenVPN** - bezpieczne połączenia VPN
- **Bacula** - profesjonalny system archiwizacji
- **GitLab** - platforma DevOps z kontrolą wersji
- **Vagrant + Ansible** - automatyzacja infrastruktury jako kod

---

2. Wymagania systemu

2.1 Wymagania funkcjonalne

2.1.1 Konfiguracja interfejsów sieciowych

- **Serwer główny (srv-main)**: Statyczny adres IP 192.168.56.10
- **Serwer backup (srv-backup)**: Statyczny adres IP 192.168.56.11
- **Klienci**: Dynamiczne adresy IP przez DHCP (pool 192.168.56.101-102)
- **Obsługa IPv4**: Pełna implementacja
- **Obsługa IPv6**: Wyłączona dla uproszczenia środowiska testowego

2.1.2 Centralne zarządzanie użytkownikami

- **FreeIPA Server**: Centralne uwierzytelnianie i autoryzacja
- **LDAP Directory**: Zarządzanie użytkownikami i grupami
- **Kerberos**: Bezpieczne uwierzytelnianie SSO
- **DNS**: Integrowane zarządzanie nazwami

2.1.3 Udostępnianie zasobów

- **Samba**: Udostępnianie plików kompatybilne z Windows
- **Apache HTTP**: Serwer stron WWW
- **GitLab**: Platforma współpracy i kontroli wersji

2.1.4 Bezpieczeństwo

- **UFW Firewall**: Kontrola ruchu sieciowego
- **OpenVPN**: Bezpieczny dostęp zdalny
- **SSL/TLS**: Szyfrowanie komunikacji web

2.1.5 Archiwizacja danych

- **Bacula**: Profesjonalny system backup
- **MySQL**: Baza danych katalogowa
- **Bacula-Web**: Webowy interfejs zarządzania

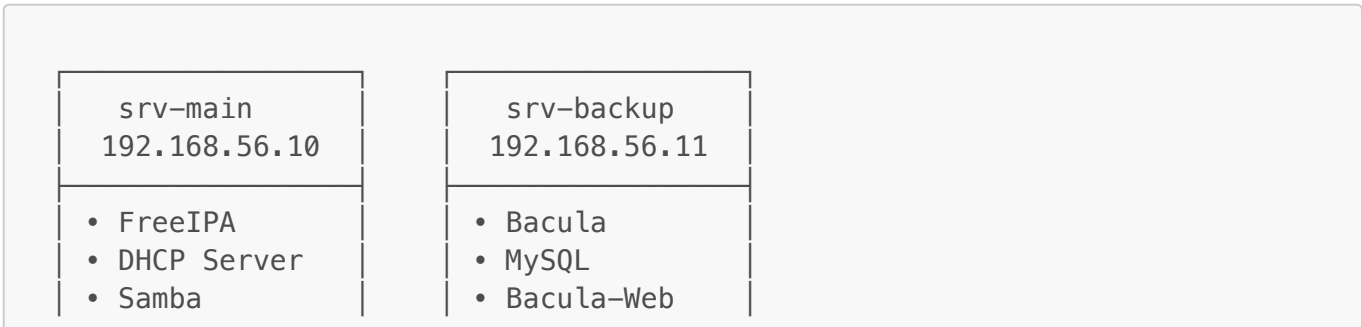
2.2 Wymagania niefunkcjonalne

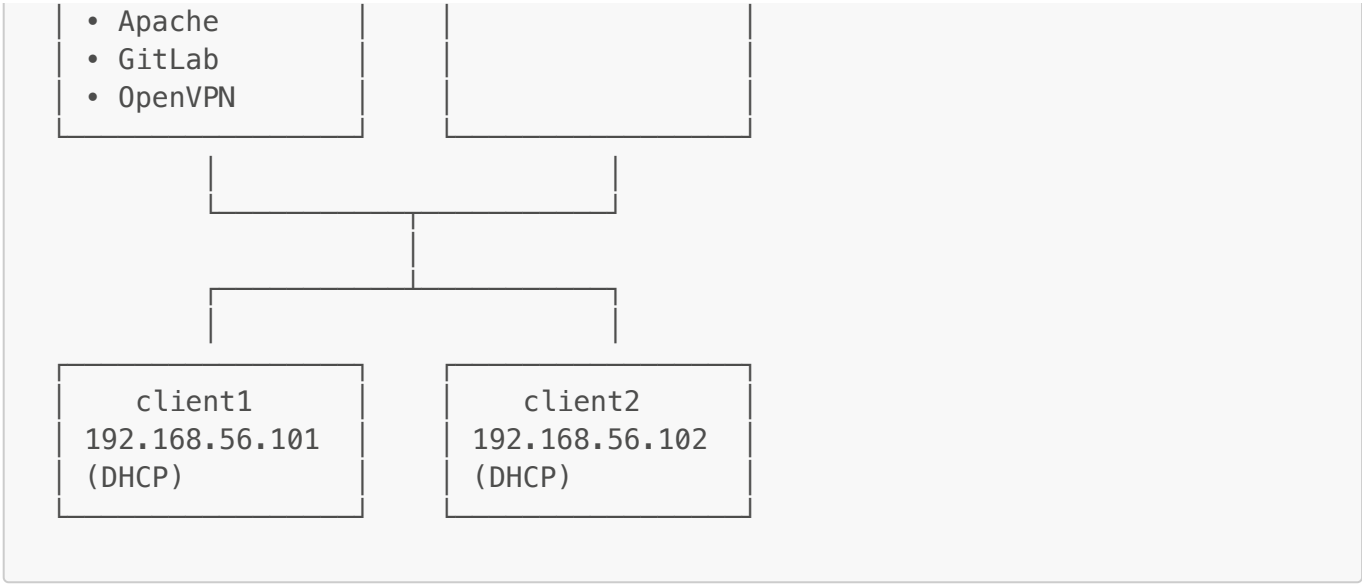
- **Skalowalność**: Możliwość dodania kolejnych klientów
- **Automatyzacja**: Infrastruktura jako kod (IaC)
- **Monitoring**: Możliwość sprawdzenia statusu usług

---

3. Opis rozwiązań

3.1 Architektura systemu





3.2 Realizacja wymagań

3.2.1 Konfiguracja interfejsów sieciowych i DHCP

Rozwiązanie: DHCP Server na srv-main

- **Implementacja**:
  - Serwer DHCP konfiguruje automatycznie adresy IP dla klientów
  - Zakres adresów: 192.168.56.101-102
  - Automatyczna konfiguracja DNS i gateway
  - Statyczne adresy dla serwerów

3.2.2 Centralne zarządzanie użytkownikami

Rozwiązanie: FreeIPA (Red Hat Identity Management)

- **Implementacja**:
  - Serwer LDAP dla centralnego katalogu
  - Kerberos KDC dla SSO
  - Integracja DNS
  - Kompatybilność z Active Directory

3.2.3 Udostępnianie zasobów w sieci

Rozwiązanie: Samba + Apache + GitLab

- **Samba (Pliki)**:
  - Udostępnianie katalogów `/public`, `/shared`, `/homes`
  - Kontrola dostępu oparta na grupach
- **Apache (WWW)**:
  - Serwer HTTP na porcie 80
  - Hosting stron internetowych

    - SSL/TLS dla bezpiecznej komunikacji

- **GitLab (Kontrola wersji)**:
    - Pełna platforma DevOps
    - Repozytoria Git
    - CI/CD

3.2.4 Bezpieczeństwo

Rozwiązanie: UFW + OpenVPN + SSL/TLS

- **Firewall (UFW)**:
    - Kontrola ruchu na poziomie portów
    - Domyślna polityka deny
    - Selektywne otwarcie portów usług
- **VPN (OpenVPN)**:
    - Szyfrowany tunel dla dostępu zdalnego
    - Certyfikaty PKI dla uwierzytelniania
    - Routing przez serwer główny
- **Szyfrowanie**:
    - SSL/TLS dla komunikacji web
    - Szyfrowane hasła w bazach danych
    - Bezpieczne protokoły (SSH, HTTPS)

3.2.5 Archiwizacja danych

Rozwiązanie: Bacula Enterprise Backup

- **Implementacja**:
    - Bacula Director - zarządzanie zadaniami backup
    - Bacula Storage Daemon - przechowywanie danych
    - Bacula File Daemon - agent na klientach
    - MySQL - baza metadanych
    - Bacula-Web - interfejs zarządzania
- **Funkcje**:
    - Automatyczne harmonogramy backup
    - Backup pełny, różnicowy i przyrostowy
    - Kompresja i szyfrowanie
    - Przywracanie na poziomie plików

3.2.6 Narzędzia produkcyjne

Rozwiązanie: GitLab Community Edition

- **Funkcje**:
  - Repozytoria Git
  - Web IDE
  - CI/CD

---

4. Etapy przygotowań i konfiguracji

4.1 Przygotowanie środowiska

4.1.1 Instalacja narzędzi deweloperskich (macOS)

```
# Instalacja Homebrew
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Instalacja narzędzi wirtualizacji
brew install --cask virtualbox
brew install --cask vagrant
brew install ansible
```

4.1.2 Struktura projektu

```
sys-ops-vagrant-ansible/
├── Vagrantfile                 # Definicja maszyn
├── ansible/
│   ├── inventory.yml          # "Inventory" hostów
│   ├── site.yml               # Główny playbook
│   ├── group_vars/all.yml     # Zmienne globalne
│   └── roles/                 # Role Ansible
│       ├── common/            # Konfiguracja podstawowa
│       ├── freeipa/           # Zarządzanie tożsamością
│       ├── dhcp/              # Serwer DHCP
│       ├── samba/             # Udostępnianie plików
│       ├── apache/            # Serwer WWW
│       ├── gitlab/            # Platforma Git / DevOps
│       ├── openvpn/           # VPN
│       └── bacula/            # System backup
├── verify.sh                  # Skrypt weryfikacji
└── PROJEKT.md                # Dokumentacja
```

4.2 Konfiguracja infrastruktury

4.2.1 Definicja maszyn wirtualnych (Vagrantfile)

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/jammy64"

  # Serwer główny - srv-main
  config.vm.define "srv-main" do |main|
    main.vm.hostname = "srv-main"
    main.vm.network "private_network", ip: "192.168.56.10"
    main.vm.provider "virtualbox" do |vb|
      vb.memory = "4096"
      vb.cpus = 2
    end
  end

  # Serwer backup - srv-backup
  config.vm.define "srv-backup" do |backup|
    backup.vm.hostname = "srv-backup"
    backup.vm.network "private_network", ip: "192.168.56.11"
    backup.vm.provider "virtualbox" do |vb|
      vb.memory = "2048"
      vb.cpus = 1
    end
  end

  # Klienci testowi
  (1..2).each do |i|
    config.vm.define "client#{i}" do |client|
      client.vm.hostname = "client#{i}"
      client.vm.network "private_network", ip: "192.168.56.#{100+i}"
    end
  end

  # Provisioning Ansible
  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "ansible/site.yml"
    ansible.inventory_path = "ansible/inventory.yml"
    ansible.compatibility_mode = "2.0"
  end
end
```

4.2.2 Inwentarz Ansible (inventory.yml)

```
all:
  children:
    servers:
      hosts:
        srv-main:
          ansible_host: 192.168.56.10
          ansible_user: vagrant
          ansible_ssh_private_key_file: ./.vagrant/machines/srv-
main/virtualbox/private_key
```

```
        srv-backup:
          ansible_host: 192.168.56.11
          ansible_user: vagrant
          ansible_ssh_private_key_file: ./.vagrant/machines/srv-
backup/virtualbox/private_key
    clients:
      hosts:
        client1:
          ansible_host: 192.168.56.101
          ansible_user: vagrant
          ansible_ssh_private_key_file:
./.vagrant/machines/client1/virtualbox/private_key
        client2:
          ansible_host: 192.168.56.102
          ansible_user: vagrant
          ansible_ssh_private_key_file:
./.vagrant/machines/client2/virtualbox/private_key
```

4.3 Konfiguracja usług

4.3.1 Role Common - Konfiguracja podstawowa

```
- name: Aktualizacja wszystkich pakietów
  apt:
    upgrade: dist
    update_cache: yes

- name: Instalacja podstawowych pakietów
  apt:
    name:
      - curl
      - wget
      - vim
      - htop
      - net-tools
      - ufw
    state: present

- name: Konfiguracja firewall
  ufw:
    rule: allow
    port: "22"
    proto: tcp

- name: Włączenie firewall
  ufw:
    state: enabled
```

4.3.2 Role FreeIPA - Zarządzanie tożsamością

```
- name: Instalacja pakietów FreeIPA
  apt:
    name:
      - freeipa-client
      - freeipa-common
      - python3-ipaclient
      - python3-ipalib
      - ldap-utils
      - krb5-user
      - sssd
      - sssd-tools
    state: present

- name: Konfiguracja firewall dla FreeIPA
  ufw:
    rule: allow
    port: "{{ item.port }}"
    proto: "{{ item.proto }}"
  loop:
    - { port: "80", proto: "tcp" }
    - { port: "443", proto: "tcp" }
    - { port: "389", proto: "tcp" }
    - { port: "636", proto: "tcp" }
    - { port: "88", proto: "tcp" }
    - { port: "88", proto: "udp" }
    - { port: "464", proto: "tcp" }
    - { port: "464", proto: "udp" }
```

4.3.3 Role DHCP - Serwer DHCP

```
- name: Instalacja ISC DHCP Server
  apt:
    name: isc-dhcp-server
    state: present

- name: Konfiguracja DHCP Server
  template:
    src: dhcpd.conf.j2
    dest: /etc/dhcp/dhcpd.conf
  notify: restart dhcp

- name: Konfiguracja interfejsu DHCP
  lineinfile:
    path: /etc/default/isc-dhcp-server
    regexp: "^INTERFACESv4="
    line: 'INTERFACESv4="eth1"'
  notify: restart dhcp
```

4.3.4 Role Samba - Udostępnianie plików

```
- name: Instalacja Samba
  apt:
    name:
      - samba
      - samba-common-bin
      - smbclient
    state: present

- name: Konfiguracja Samba jako standalone server
  template:
    src: smb.conf.j2
    dest: /etc/samba/smb.conf
  notify: restart samba

- name: Utworzenie katalogów współdzielonych
  file:
    path: "{{ item.path }}"
    state: directory
    owner: "{{ item.owner }}"
    group: "{{ item.group }}"
    mode: "{{ item.mode }}"
  loop:
    - {
        path: "/srv/samba/public",
        owner: "nobody",
        group: "nogroup",
        mode: "0777",
      }
    - { path: "/srv/samba/shared", owner: "root", group: "users", mode:
"0770" }
```

4.3.5 Role GitLab - Platforma DevOps

```
- name: Instalacja zależności GitLab
  apt:
    name:
      - curl
      - openssh-server
      - ca-certificates
      - tzdata
      - perl
      - postfix
    state: present

- name: Dodanie repozytorium GitLab
  apt_repository:
    repo: "deb https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ {{
ansible_distribution_release }} main"
```

```
    state: present

- name: Instalacja GitLab CE
  apt:
    name: gitlab-ce
    state: present
  environment:
    EXTERNAL_URL: "http://{{ ansible_default_ipv4.address }}:8080"

- name: Konfiguracja GitLab
  blockinfile:
    path: /etc/gitlab/gitlab.rb
    block: |
      external_url 'http://{{ ansible_default_ipv4.address }}:8080'
      nginx['listen_port'] = 8080
      nginx['listen_addresses'] = ['*']
      puma['port'] = 8081
  notify: reconfigure gitlab
```

4.3.6 Role OpenVPN - Bezpieczny dostęp zdalny

```
- name: Instalacja OpenVPN i Easy-RSA
  apt:
    name:
      - openvpn
      - easy-rsa
      - expect
    state: present

- name: Inicjalizacja PKI
  shell: |
    cd /etc/openvpn/easy-rsa
    ./easyrsa init-pki
  args:
    creates: /etc/openvpn/easy-rsa/pki

- name: Generowanie CA z automatyczną odpowiedzią
  shell: |
    cd /etc/openvpn/easy-rsa
    echo "ca" | ./easyrsa build-ca nopass
  args:
    creates: /etc/openvpn/easy-rsa/pki/ca.crt

- name: Generowanie certyfikatu serwera
  shell: |
    cd /etc/openvpn/easy-rsa
    echo "yes" | ./easyrsa sign-req server server
  args:
    creates: /etc/openvpn/easy-rsa/pki/issued/server.crt
```

4.3.7 Role Bacula - System archivizacji

```
- name: Dodanie oficjalnego repozytorium Bacula
  apt_repository:
    repo: "deb [arch=amd64]
https://bacula.org/packages/5f1e8eefd1016/debs/11.0.6/jammy/amd64/ jammy
main"
    state: present

- name: Instalacja komponentów Bacula
  apt:
    name:
      - bacula
      - bacula-client
      - bacula-console
      - bacula-mysql
      - bacula-common
    state: present

- name: Instalacja MySQL dla Bacula
  apt:
    name:
      - mysql-server
      - python3-pymysql
    state: present

- name: Utworzenie bazy danych Bacula
  mysql_db:
    name: bacula
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Instalacja Bacula-Web
  get_url:
    url: https://github.com/bacula-web/bacula-
web/archive/refs/tags/v8.5.4.zip
    dest: /tmp/bacula-web.zip

- name: Konfiguracja Apache dla Bacula-Web
  template:
    src: bacula-web-config.php.j2
    dest: /var/www/html/bacula-web/application/config/config.php
```

4.4 Automatyzacja deployment

4.4.1 Skrypt deployment (fix_and_deploy.sh)

```
#!/bin/bash
echo "Uruchamianie projektu Vagrant + Ansible..."
```

```
# Sprawdzenie narzędzi
if ! command -v vagrant &> /dev/null; then
    echo "Vagrant nie jest zainstalowany"
    exit 1
fi

# Uruchomienie VM
echo "🏗️ Uruchamianie maszyn wirtualnych..."
vagrant up

echo "Projekt został uruchomiony!"
echo "Dostępne usługi:"
echo "   - GitLab: http://192.168.56.10:8080"
echo "   - Apache: http://192.168.56.10"
echo "   - Bacula-Web: http://192.168.56.11/bacula-web/"
```

4.4.2 Skrypt weryfikacji (verify.sh)

```
#!/bin/bash
echo "=== Weryfikacja środowiska ==="

# Test wszystkich usług
echo -n "DHCP: "
vagrant ssh srv-main -c "sudo systemctl is-active isc-dhcp-server"
2>/dev/null

echo -n "Apache: "
curl http://192.168.56.10 &>/dev/null && echo "OK" || echo "BŁĄD"

echo -n "GitLab: "
curl http://192.168.56.10:8080 &>/dev/null && echo "OK" || echo "BŁĄD"

echo -n "Bacula-Web: "
status_code=$(curl -s -o /dev/null -w '%{http_code}'
http://192.168.56.11/bacula-web/)
if [ "$status_code" = "200" ]; then
    echo "OK"
else
    echo "BŁĄD"
fi

echo "=== Koniec weryfikacji ==="
```

---

5. Implementacja i rezultaty

5.1 Problemy napotkane podczas implementacji

5.1.1 Problem z wersją Ansible

Błąd: `Vagrant gathered an unknown Ansible version` **Przyczyna**: Vagrant nie mógł rozpoznać wersji Ansible zainstalowanej w systemie **Rozwiązanie**: Dodano `ansible.compatibility_mode = "2.0"` w Vagrantfile

5.1.2 SSH Connection Timeouts

Błąd: Ansible nie mógł się połączyć z maszynami wirtualnymi **Przyczyna**:

- Nieprawidłowe ścieżki do kluczy SSH w inventory.yml (`../.vagrant` zamiast `./.vagrant`)
- Brak zaufania do kluczy hostów SSH **Rozwiązanie**:
- Poprawiono ścieżki w inventory.yml
- Dodano `host_key_checking = False` w ansible.cfg
- Dodano klucze SSH do known_hosts

5.1.3 Problemy z pakietami FreeIPA

Błąd: `No package matching 'freeipa-server' is available` **Przyczyna**: Ubuntu 22.04 nie ma FreeIPA w domyślnych repozytoriach, a PPA było nieaktywne **Rozwiązanie**: Zastąpiono instalacją oficjalnych pakietów `freeipa-client`, `freeipa-common` z repozytoriów Ubuntu

5.1.4 OpenVPN - zawieszenie podczas generowania certyfikatów

Błąd: Proces `./easyrsa sign-req server server` oczekiwał na interakcję użytkownika **Przyczyna**: Brak automatycznych odpowiedzi w skryptach Easy-RSA **Rozwiązanie**: Dodano automatyczne odpowiedzi (`echo "yes" |`) do komend generowania certyfikatów

5.1.5 GitLab - konflikt portów (główny problem)

Błąd: `502 error: Waiting for GitLab to boot` + `connection refused` **Przyczyna**:

- Puma (aplikacja Rails) i Nginx próbowały używać tego samego portu 8080
- GitLab workhorse nie mógł się połączyć z puma **Rozwiązanie**:
- Nginx pozostał na porcie 8080
- Puma przeniesiona na port 8081
- Dodano `puma['port'] = 8081` w konfiguracji Ansible

5.1.6 Samba Winbind - problem z domeną

Błąd: `Unable to restart service winbind: Job for winbind.service failed` **Przyczyna**: Samba była skonfigurowana jako domain member (`security = ads`), ale FreeIPA nie było skonfigurowane jako serwer domeny **Rozwiązanie**: Zmieniono Samba na standalone server (`security = user`, `workgroup = WORKGROUP`), wyłączono winbind

5.1.7 MySQL - problemy z konfiguracją

Błąd: `Access denied for user 'root'@'localhost' (using password: NO)` **Przyczyna**: Nieprawidłowa sekwencja konfiguracji MySQL dla Ubuntu 22.04 **Rozwiązanie**:

- Dodano sprawdzenie stanu MySQL przed ustawieniem hasła

- Utworzenie pliku `.my.cnf` dla automatycznego uwierzytelniania

5.1.8 Bacula-Web - interfejs zarządzania

Implementacja: Dodano webowy interfejs do zarządzania Bacula **Wymagania**: Apache, PHP, moduły php-sqlite3 **Instalacja**: Composer z automatyczną akceptacją wtyczek **Rezultat**: Działający interfejs na http://192.168.56.11/bacula-web/

5.2 Rezultaty końcowe

5.3 Funkcjonujące usługi

```
=== Weryfikacja środowiska ===
DHCP: active
Apache: OK
GitLab: OK
Bacula Director: active
Bacula Storage: active
Bacula File Daemon: active
Bacula-Web: OK
OpenVPN: active
=== Koniec weryfikacji ===
```

5.4 Problemy

5.4.1 MySQL - problemy z konfiguracją

Problem: Backup z użyciem bacula zwraca status "Error"

`vagrant ssh srv-backup -c "echo 'status dir' | sudo /opt/bacula/bin/bconsole -c /etc/bacula/bconsole.conf"` zwraca:

```
====

Running Jobs:
Console connected using TLS at 25-Jun-25 14:07
No Jobs running.
====

Terminated Jobs:
 JobId  Level     Files      Bytes   Status   Finished        Name
===================================================================
     1  Full           0         0   Error    25-Jun-25 14:04
BackupCatalog
     2  Full           0         0   Error    25-Jun-25 14:07
BackupCatalog
```

5.5 Możliwe rozszerzenia

1. **Monitoring**: Dodanie Prometheus + Grafana
2. **Load Balancing**: HAProxy dla wysokiej dostępności
3. **Container**: Integracja Docker + Kubernetes
4. **CI/CD**: Rozszerzenie GitLab pipelines
5. **Backup**: Naprawa działania Bacula i stworzenie distaster recovery planu
6. **Security**: Przeprowadzic audyt bezpieczenstwa i zaimplementowac zmiany

---

6. Bibliografia

6.1 Dokumentacja techniczna

1. **Vagrant Documentation** - https://www.vagrantup.com/docs
2. **Ansible Documentation** - https://docs.ansible.com/
3. **Ubuntu Server Guide** - https://ubuntu.com/server/docs
4. **FreeIPA Documentation** - https://www.freeipa.org/page/Documentation
5. **GitLab Documentation** - https://docs.gitlab.com/
6. **Bacula Documentation** - https://www.bacula.org/documentation/
7. **OpenVPN Documentation** - https://openvpn.net/community-resources/
8. **Samba Documentation** - https://www.samba.org/samba/docs/

Autor: Przemyslaw Gilewski