

Pojekt Home-Lab

Spis treści

1. Cel projektu	2
2. Architektura Sieciowa	3
2.1 Szczegółowy opis architektury	4
3. Główne komponenty laboratorium	5
4. Etapy realizacji projektu	6
4.1. Instalacja oraz konfiguracja hypervisora (Proxmox)	6
4.2. Instalacja oraz konfiguracja zapory sieciowej pfSense	9
4.2.1. Tworzenie maszyny wirtualnej dla pfSense oraz instalacja systemu	9
4.2.2. Konfiguracja maszyny wirtualnej pfSense	10
4.2.3. Szczegółowy opis konfiguracji zapory	12
4.3. Instalacja serwera Ubuntu, Dockera, Portainera	13
4.3.1. Tworzenie maszyny wirtualnej dla Ubuntu oraz instalacja systemu	13
4.3.2. Instalacja Dockera na Ubuntu Server	13
4.3.3. Instalacja Portainera (Ubuntu + Docker)	14
4.3.4. Konfiguracja sieci dla kontenerów Docker	15
4.4. Budowa serwera Wazuh	17
4.4.1. Tworzenie maszyny wirtualnej oraz instalacja systemu bazowego (Ubuntu)	17
4.4.1. Instalacja Wazuh	18
4.4.2. Instalacja agentów Wazuh	20

Autor: Kamil Iskra
Rzeszów, 30.06.2025r.
Ver. 1.0

1. Cel projektu

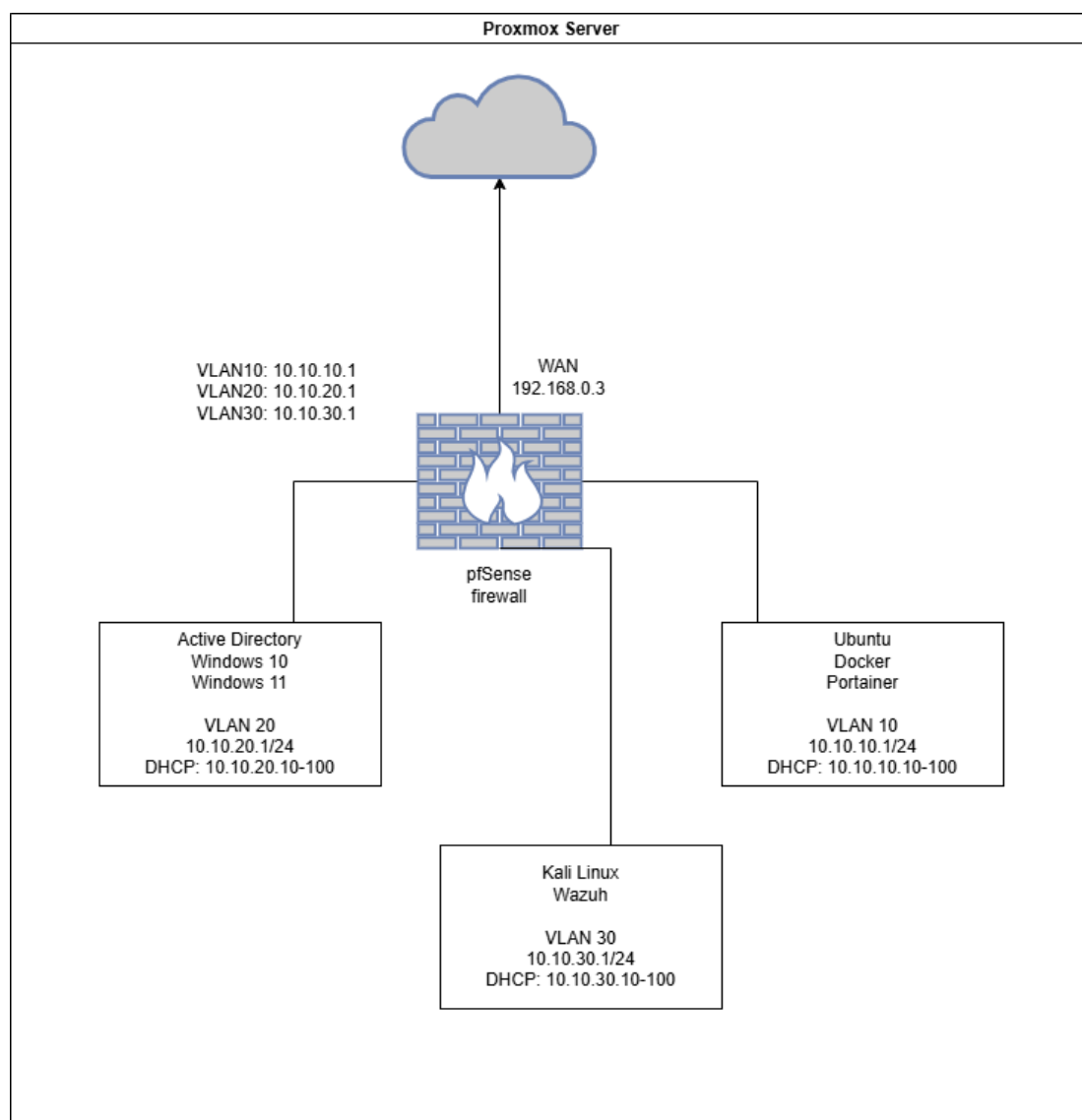
Projekt „Home-Lab” to kompleksowy projekt mający na celu praktyczne zdobycie umiejętności i naukę nowych technologii z zakresu cyberbezpieczeństwa, administracji sieciami / systemami, automatyzacji zadań oraz sztucznej inteligencji. Jest to przykład praktycznego wykorzystania różnorodnych narzędzi i konfiguracji, umożliwiających m.in. zarządzanie infrastrukturą, skanowanie podatności, monitorowanie sieci, wykrywanie zagrożeń oraz reagowanie na incydenty.

2. Architektura Sieciowa

Całe środowisko laboratoryjne jest zbudowane na platformie wirtualizacyjnej Proxmox. Jego centralnym punktem jest firewall pfSense, przy pomocy którego sieć jest podzielona na kilka podsieci. Segmentacja odbywa się za pomocą VLAN-ów (Virtual Local Area Network), co pozwala na ich izolację:

- **VLAN 10 (10.10.10.0/24):** podsieć dla maszyn Linux i kontenerów (Docker).
- **VLAN 20 (10.10.20.0/24):** podsieć z maszynami Windows.
- **VLAN 30 (10.10.30.0/24):** podsieć dla narzędzi bezpieczeństwa i monitoringu.

Tak zaplanowana architektura umożliwi łatwą rozbudowę w przyszłości np. poprzez utworzenie dodatkowych VLAN-ów.



Rys. 1 – Schemat sieci.

2.1 Szczegółowy opis architektury

- Serwer **Proxmox** będzie hostował wszystkie maszyny wirtualne laboratorium.
- Interfejs WAN zapory **pfSense** połączony z siecią domową (statyczny adres IP: 192.168.0.3/24)
- Interfejs LAN zapory **pfSense** (vbr1 w Proxmox) będzie trunkiem dla wszystkich wewnętrznych VLANów laboratorium. Został stworzony jako mostek (Linux Bridge) w Proxmox.
- Segmentacja VLAN:
 - **VLAN 10** (Docker / Kontenery): 10.10.10.0/24
 - Adres IP: 10.10.10.1
 - Zakres DHCP: 10.10.10.10-100
 - **VLAN 20** (Środowisko Windows): 10.10.20.0/24
 - Adres IP: 10.10.20.1
 - Zakres DHCP: 10.10.20.10-100
 - **VLAN 30** (Narzędzia bezpieczeństwa): 10.10.30.0/24
 - Adres IP: 10.10.30.1
 - Zakres DHCP: 10.10.30.10-100

3. Główne komponenty laboratorium

- **Zapora sieciowa (Firewall):**
 - pfSense: Główna zapora sieciowa, zarządzająca ruchem i segmentacją sieci.
- **Docker / Kontenery – VLAN 10:**
 - Ubuntu Server: Host dla kontenerów Docker.
 - Docker: Platforma konteneryzacji.
 - Portainer: Narzędzie do zarządzania kontenerami.
- **Środowisko Windows (Windows Server, Active Directory) – VLAN 20:**
 - Windows Server 2022: Kontroler domeny (AD, Group Policy, DHCP, DNS).
 - Windows 10: Maszyna kliencka.
 - Windows 11: Maszyna kliencka.
- **Narzędzia bezpieczeństwa – VLAN 30:**
 - Kali Linux: Maszyna do ataków i testów penetracyjnych.
 - Wazuh: Rozwiązanie SIEM/XDR.
 - Nessus: Skaner podatności.

4. Etapy realizacji projektu

4.1. Instalacja oraz konfiguracja hypervisora (Proxmox)

Kluczowym elementem każdego laboratorium jest wybór odpowiedniego systemu, który zostanie zainstalowany na fizycznym sprzęcie. W tym wypadku wybór padł na platformę Proxmox Virtual Environment. Istotną zaletą tego rozwiązania jest to, że opiera się na zmodyfikowanym jądrze Debiana oraz jest rozwiązaniem open source z aktywną społecznością. System został zainstalowany na sprzęcie o specyfikacji:

Model: Lenovo PC ThinkCentre M920x Tiny USFF

Procesor: Intel® Core™ i7-9700 @ 3.00GHz (8 rdzeni)

Pamięć operacyjna: 64 GB RAM DDR4 3200MHz

Dysk: 1 x Samsung PM9B1 NVMe 512 GB, 2 x WD SN580 2TB NVMe

Karta dźwiękowa: Zintegrowana Realtek® ALC233VB2 High Definition (HD) Audio

Karta sieciowa: Zintegrowana Intel® I219-V Gigabit Ethernet 10/100/1000 Mbit/s

Karta graficzna: Zintegrowana Intel® UHD Graphics 630

Chipset: Intel® Q370

Porty rozszerzeń:

- 2 x SODIMM

- 1 x SATA

- 1 x M.2 PCIe 2230 (dla WLAN)

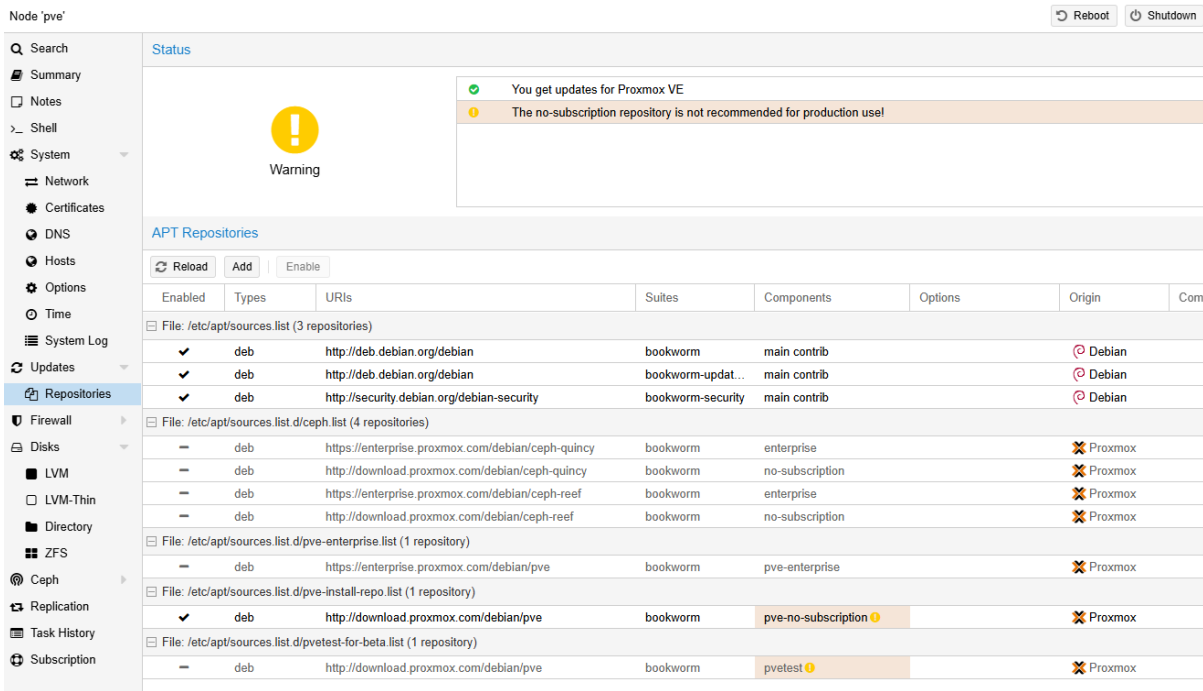
- 1 x M.2 PCIe 2280

- 1 x M.2 PCIe 2280 / 2242



Rys. 2 – Lenovo PC ThinkCentre M920x Tiny USFF.











Proces instalacji nie różni się znacząco od typowej instalacji systemu Linux. Uwagi natomiast wymaga konfiguracja Proxmox po zakończeniu instalacji. Pierwszą rzeczą jest konfiguracja repozytorium pakietów i aktualizacja systemu.



Rys. 3 – Konfiguracja repozytorium pakietów w Proxmox.

Dyski zostały wykorzystane następująco:

- Dysk Samsung PM9B1 NVMe 512 GB został przeznaczony w całości na system Proxmox oraz na magazynowanie obrazów instalacyjnych maszyn wirtualnych.
- Dwa dyski WD SN580 2TB NVMe zostały przeznaczone pod wirtualne maszyny, dyski zostały połączone w RAID1 (Mirroring) w celu zapewnienia bezpieczeństwa przed utratą danych w przypadku awarii jednego dysku. Zastosowany został system plików ZFS.

Reload	Show S.M.A.R.T. values	Initialize Disk with GPT	Wipe Disk				
Device	Type	Usage	Size	GPT	Model	Serial	S.M.A.R.T.
 /dev/nvme0n1	nvme	partitions	512.11 GB	Yes	PM9B1 NVMe Samsung 512GB	S6MZNFMW702050	PASSED
 /dev/nvme0n...	partition	BIOS boot	1.03 MB	Yes			
 /dev/nvme0n...	partition	EFI	1.07 GB	Yes			
 /dev/nvme0n...	partition	ZFS	511.04 GB	Yes			
 /dev/nvme1n1	nvme	partitions	2.00 TB	Yes	WD Blue SN580 2TB	24401W804826	PASSED
 /dev/nvme1n...	partition	ZFS	2.00 TB	Yes			
 /dev/nvme1n...	partition	ZFS reserved	8.39 MB	Yes			
 /dev/nvme2n1	nvme	partitions	2.00 TB	Yes	WD Blue SN580 2TB	24401W804804	PASSED
 /dev/nvme2n...	partition	ZFS	2.00 TB	Yes			
 /dev/nvme2n...	partition	ZFS reserved	8.39 MB	Yes			

Rys. 4 – Konfiguracja dysków.

Status: tank0

Reload

Health

ONLINE

Errors

No known data errors

Devices

Name	Health	READ	WRITE	CKSUM
<div> <div></div> <div>tank0</div> </div>	<div> <div></div> <div>ONLINE</div> </div>	0	0	0
<div> <div></div> <div>mirror-0</div> </div>	<div> <div></div> <div>ONLINE</div> </div>	0	0	0
<div> <div></div> <div>/dev/disk/by-id/nvme-WD_Blue_SN580_2TB_24401W804826-part1</div> </div>	<div> <div></div> <div>ONLINE</div> </div>	0	0	0
<div> <div></div> <div>/dev/disk/by-id/nvme-WD_Blue_SN580_2TB_24401W804804-part1</div> </div>	<div> <div></div> <div>ONLINE</div> </div>	0	0	0

Rys. 4 – Konfiguracja dysków – ZFS.

W celu robienia backupu maszyn wirtualnych podmontowany został zewnętrzny zasób sieciowy (exos-backup) przez protokół SMB/CIFS. Zasób dostępny z lokalnej sieci LAN.

Edit: SMB/CIFS

General

Backup Retention

ID:

exos-backup

Nodes:

All (No restrictions)

Server:

192.168.0.100

Enable:

☒

Username:

backup

Content:

Backup

Password:

Domain:

Share:

proxmox

Subdirectory:

Preallocation:

Default

Help

Advanced ☒

OK

Rys. 5 – Konfiguracja backupu.

4.2. Instalacja oraz konfiguracja zapory sieciowej pfSense

4.2.1. Tworzenie maszyny wirtualnej dla pfSense oraz instalacja systemu

ID VM: 200

Nazwa: pfSense

OS: Użycie pobranego obrazu ISO pfSense (pfSense-CE-2.7.2-RELEASE-amd64.iso)

Dysk: 50 GB (vm-drives).

CPU: 4 rdzenie.

RAM: 8 GB.

Interfejsy sieciowe:


net0: podłączony do vmbr0 (WAN).













net1: podłączony do vmbr1 (LAN).

Virtual Machine 200 (pfSense) on node 'pve' No Tags

Summary	Add	Remove	Edit	Disk Action	Revert
Console	Memory	8.00 GiB			
Hardware	Processors	4 (1 sockets, 4 cores) [host]			
Cloud-Init	BIOS	Default (SeaBIOS)			
Options	Display	Default			
Task History	Machine	Default (i440fx)			
Monitor	SCSI Controller	VirtIO SCSI single			
Backup	Hard Disk (scsi0)	vm-drives:200/vm-200-disk-0.qcow2,iothread=1,size=50G			
Replication	Network Device (net0)	virtio=BC:24:11:93:E0:C5,bridge=vmbr0,firewall=1			
Snapshots	Network Device (net1)	virtio=BC:24:11:2B:AD:9A,bridge=vmbr1,firewall=1			
Firewall					
Permissions					

Rys. 6 – Konfiguracja maszyny wirtualnej pfSense.

Virtual Machine 200 (pfSense) on node 'pve' No Tags 

 Summary
  Console
  Hardware
  Cloud-Init
  Options
  Task History
  Monitor
  Backup
  Replication
  Snapshots
  Firewall
  Permissions

Edit Revert

Name	pfSense
Start at boot	Yes
Start/Shutdown order	order=any
OS Type	Linux 6.x - 2.6 Kernel
Boot Order	scsi0
Use tablet for pointer	Yes
Hotplug	Disk, Network, USB
ACPI support	Yes
KVM hardware virtualization	Yes
Freeze CPU at startup	No
Use local time for RTC	Default (Enabled for Windows)
RTC start date	now
SMBIOS settings (type1)	uuid=4f318fb3-3d2f-48ed-b837-b9a481576d0f
QEMU Guest Agent	Enabled
Protection	No
Spice Enhancements	none
VM State storage	Automatic
AMD SEV	Default (Disabled)

Rys. 7 – Konfiguracja maszyny wirtualnej pfSense (2).

4.2.2. Konfiguracja maszyny wirtualnej pfSense

Po zainstalowaniu systemu pierwszym krokiem jaki należy zrobić jest konfiguracja interfejsów sieciowych. Tak jak to już wcześniej zostało skonfigurowane maszyna posiada dwa interfejsy sieciowe: WAN oraz LAN. W pfSense są one oznaczone odpowiednio *vtnet0* oraz *vtnet1*.

```

QEMU Guest - Netgate Device ID: 2e8ce3397f538cef95b6

*** Welcome to pfSense 2.8.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> vtnet0      -> v4: 192.168.0.3/24
LAN (lan)      -> vtnet1      -> v4: 10.10.1.1/24
VLAN10 (opt1)  -> vtnet1.10 -> v4: 10.10.10.1/24
VLAN20 (opt2)  -> vtnet1.20 -> v4: 10.10.20.1/24
VLAN30 (opt3)  -> vtnet1.30 -> v4: 10.10.30.1/24

0) Logout / Disconnect SSH
1) Assign Interfaces
2) Set interface(s) IP address
3) Reset admin account and password
4) Reset to factory defaults
5) Reboot system
6) Halt system
7) Ping host
8) Shell
9) pfTop
10) Filter Logs
11) Restart GUI
12) PHP shell + pfSense tools
13) Update from console
14) Enable Secure Shell (sshd)
15) Restore recent configuration
16) Restart PHP-FPM

```

Rys. 7 – Konfiguracja interfejsów sieciowych w pfSense.

Po skonfigurowaniu interfejsów sieciowych istnieje możliwość zarządzania zaporą z poziomu panelu WWW.

System Information

Name	pfSense.home.arpa
User	admin@192.168.0.226 (Local Database)
System	QEMU Guest Netgate Device ID: 2e8ce3397f538cef95b6
BIOS	Vendor: SeaBIOS Version: rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org Release Date: Tue Apr 1 2014
Version	2.7.2-RELEASE (amd64) built on Wed Dec 6 21:10:00 CET 2023 FreeBSD 14.0-CURRENT The system is on the latest version. Version information updated at Tue Jun 10 10:41:57 CEST 2025
CPU Type	QEMU Virtual CPU version 2.5+ 4 CPUs: 1 package(s) x 4 core(s) AES-NI CPU Crypto: Yes (Inactive) QAT Crypto: No
Hardware crypto	Inactive
Kernel PTI	Enabled

Interfaces

Interface	Status	Speed	MAC	IP Address
WAN	↑	10Gbase-T <full-duplex>		192.168.0.3
LAN	↑	10Gbase-T <full-duplex>		10.10.1.1
VLAN10	↑	10Gbase-T <full-duplex>		10.10.10.1
VLAN20	↑	10Gbase-T <full-duplex>		10.10.20.1
VLAN30	↑	10Gbase-T <full-duplex>		10.10.30.1













Interface Statistics

	WAN	LAN	VLAN10	VLAN20	VLAN30
Packets In	1267	0	0	0	0
Packets Out	1559	5	4	4	5
Bytes In	431 KiB	0 B	0 B	0 B	0 B
Bytes Out	907 KiB	416 B	344 B	344 B	416 B
Errors In	0	0	0	0	0
Errors Out	0	0	0	0	0
Collisions	0	0	0	0	0

Rys. 8 – Panel zarządzania pfSense.

Na powyższych zrzutach ekranu widoczne są już utworzone i skonfigurowane VLANy. Każdy z nich posiada połączenie z siecią zewnętrzną Internet oraz zapewnioną komunikację między

sobą. Natomiast domyślnie zapora blokuje ruch z zewnątrz. Aby możliwy był dostęp do VLAN-ów z zewnątrz i tym samym do hostowanych w nich usługach konieczne jest zezwolenie na ruch przez dodanie odpowiednich reguł.

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓	8/4.12 MiB	IPv4 *	192.168.0.0/24	*	*	*	*	none		    
<div>  Add  Add  Delete  Toggle  Copy  Save  Separator </div>											

Rys. 9 – Reguła pfSense zezwalająca na ruch z zewnątrz.

4.2.3. Szczegółowy opis konfiguracji zapory

1. Konfiguracja VLANów w pfSense (przez interfejs webowy):

- Przejście do Interfaces -> Assignments -> VLANs.
- Dodanie VLANów:
 - VLAN 10: Parent interface vtnet1 (LAN), VLAN tag 10.
 - VLAN 20: Parent interface vtnet1 (LAN), VLAN tag 20.
 - VLAN 30: Parent interface vtnet1 (LAN), VLAN tag 30.
- Przypisanie interfejsów do VLANów (Interfaces -> Assignments):
 - Dodanie OPT1 dla VLAN 10 na vtnet1.
 - Dodanie OPT2 dla VLAN 20 na vtnet1.
 - Dodanie OPT3 dla VLAN 30 na vtnet1.
- Konfiguracja interfejsów dla VLANów:
 - OPT1 (VLAN10): Włączenie, zmiana nazwy na VLAN10, adres IPv4 statyczny 10.10.10.1/24.
 - OPT2 (VLAN20): Włączenie, zmiana nazwy na VLAN20, adres IPv4 statyczny 10.10.20.1/24.
 - OPT3 (VLAN30): Włączenie, zmiana nazwy na VLAN30, adres IPv4 statyczny 10.10.30.1/24.

2. Konfiguracja reguł zapory dla VLANów w pfSense:

- Przejście do Firewall -> Rules.
- Dla każdego nowo utworzonego interfejsu VLAN (VLAN10, VLAN20, VLAN30):
 - Skopiowanie domyślnej reguły "Allow LAN to any" z interfejsu LAN.
 - Zmiana źródła (Source) z "LAN subnets" na odpowiedni "[NazwaVLAN] subnets".
- Dodanie reguły na interfejsie WAN zezwalającej na ruch z zewnątrz.

3. Konfiguracja serwera DHCP dla VLANów w pfSense:

- Przejście do Services -> DHCP Server.
- Dla każdego interfejsu VLAN (VLAN10, VLAN20, VLAN30):

- Włączenie serwera DHCP.
- Ustawienie zakresu adresów (np. dla VLAN10: od 10.10.10.10 do 10.10.10.100).
- Dodanie serwerów DNS: 10.10.X.1 (gdzie X to numer VLAN) oraz 1.0.0.2, 1.1.1.1, 8.8.8.8.

4.3. Instalacja serwera Ubuntu, Dockera, Portainera

4.3.1. Tworzenie maszyny wirtualnej dla Ubuntu oraz instalacja systemu

ID VM: 202

Nazwa: docker

OS:Ubuntu Server 22.04. (Live).

Dysk: 250 GB (vm-drives).

CPU: 6 rdzeni.

RAM: 16 GB.

Interfejs sieciowy: podłączony do vmbr1 (LAN) z tagiem VLAN 10.

Instalacja z ustawieniami domyślnymi + dodatkowo instalacja usługi serwera ssh.

Adres IP maszyny przydzielony przez DHCP (10.10.10.10).

4.3.2. Instalacja Dockera na Ubuntu Server

Instalacja zgodnie z dokumentacją – <https://docs.docker.com/engine/install/ubuntu/>

```
#Run the following command to uninstall all conflicting packages:
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker
containerd runc; do sudo apt-get remove $pkg; done

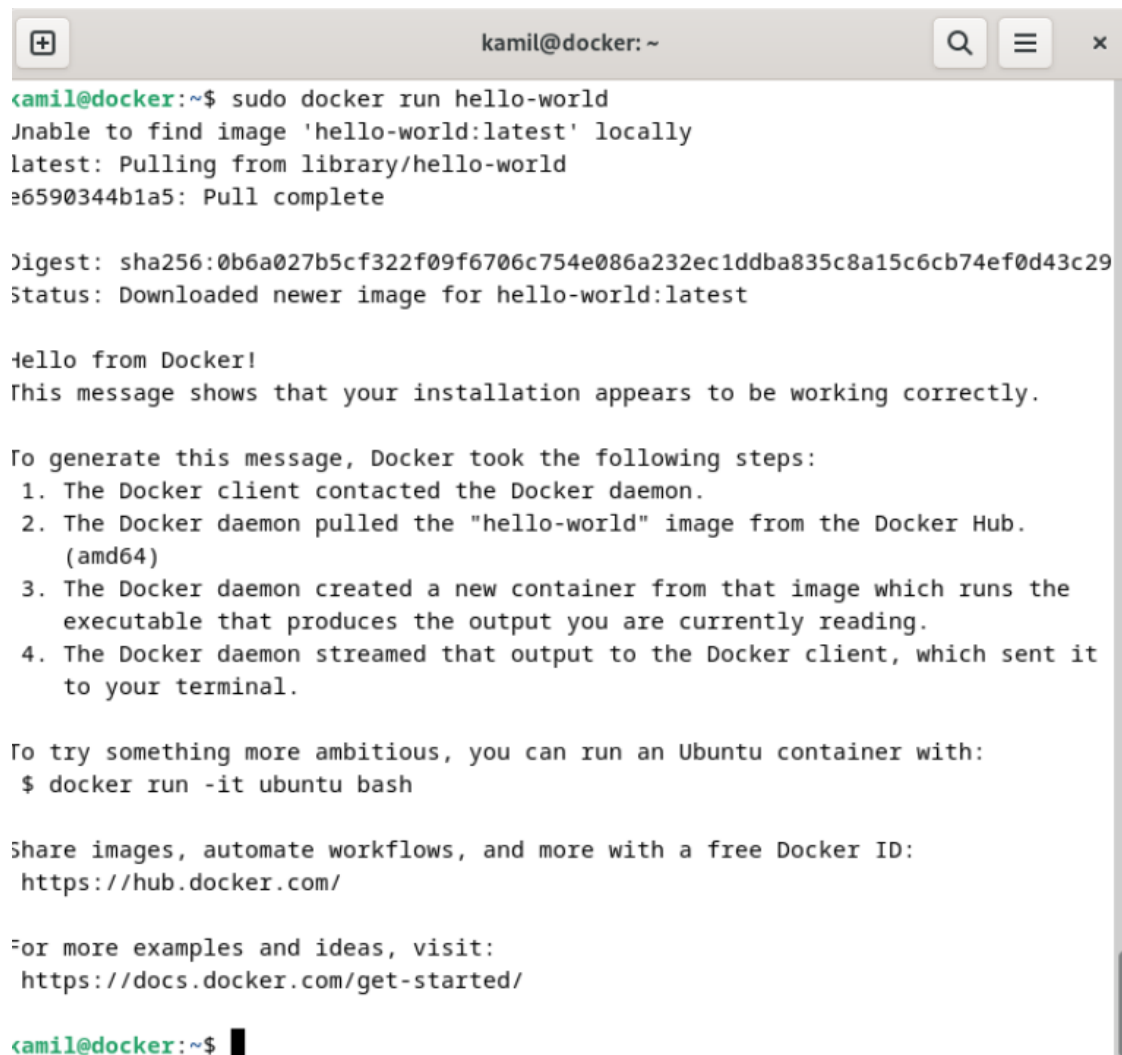
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

#1. Set up Docker's apt repository.
# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update

#2. Install the Docker packages.
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

#3. Verify that the installation is successful by running the hello-world image:
sudo docker run hello-world
```

A screenshot of a terminal window titled 'kamil@docker: ~'. The terminal shows the command 'sudo docker run hello-world' being executed. The output indicates that the 'hello-world:latest' image was pulled from the Docker Hub and the container executed successfully, displaying a 'Hello from Docker!' message. The terminal also lists the steps Docker took to generate this message and provides links for more information.

```
kamil@docker:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete

Digest: sha256:0b6a027b5cf322f09f6706c754e086a232ec1ddba835c8a15c6cb74ef0d43c29
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

kamil@docker:~$
```

Rys. 10 – Weryfikacja instalacji i działania Dockera.

4.3.3. Instalacja Portainera (Ubuntu + Docker)

Tak jak w przypadku Dockera, instalacja Portainera zgodnie z oficjalną dokumentacją – <https://docs.portainer.io/start/install-ce/server/docker/linux>

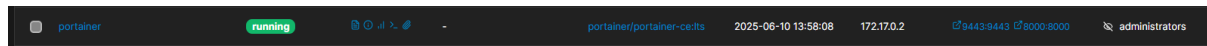
```
#First, create the volume that Portainer Server will use to store its database:
```

```

sudo docker volume create portainer_data

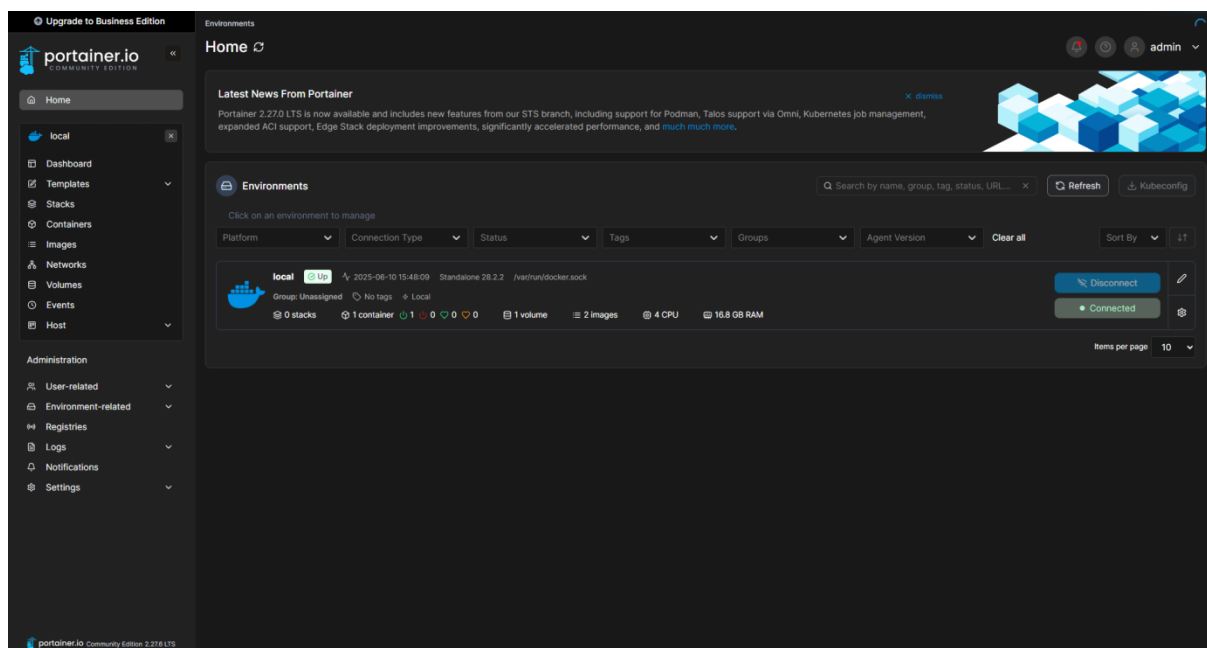
#Then, download and install the Portainer Server container:
sudo docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest

```



Rys. 11 – Uruchomiony kontener - Portainer.

Dostęp do interfejsu webowego Portainera dostępny w sieci z przeglądarki pod adresem **10.10.10.10:9443** (domyślny port portainera).



Rys. 12 – Panel Portainera.

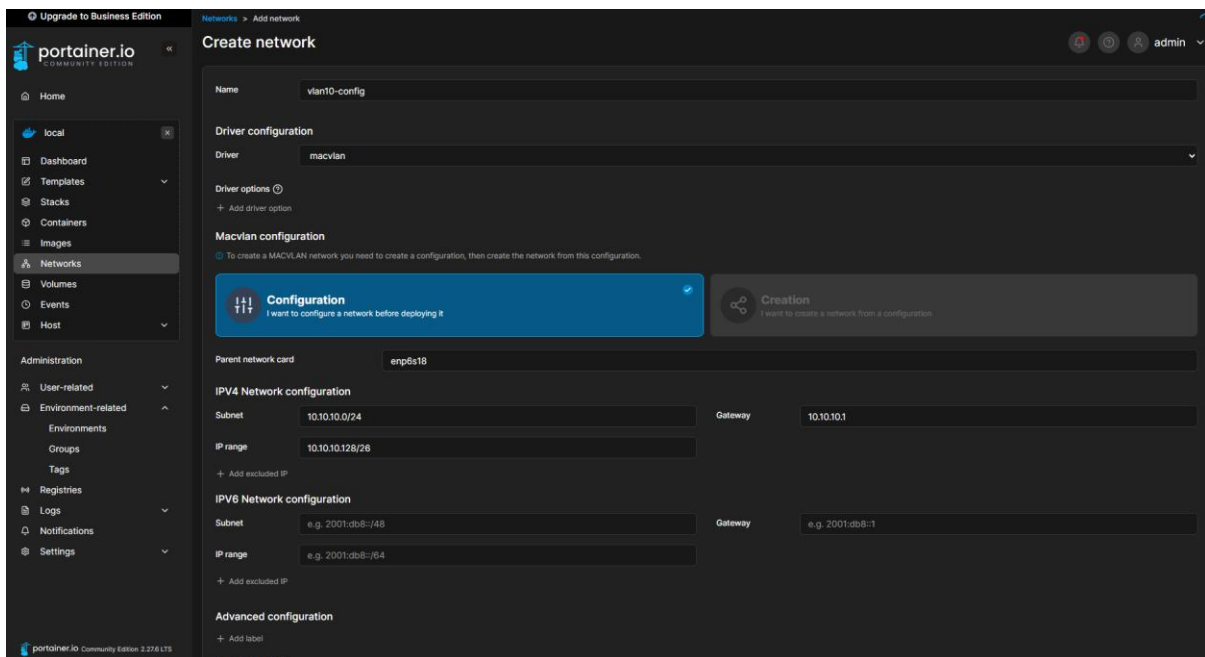
4.3.4. Konfiguracja sieci dla kontenerów Docker

MacVLAN umożliwia każdemu kontenerowi Docker posiadanie własnego, unikalnego adresu MAC i adresu IP w sieci hosta, co sprawia, że wyglądają one jak indywidualne urządzenia w sieci LAN. Jest to idealne rozwiązanie do skanowania podatności na poziomie sieci.

Konfiguracja sprowadza się do dwóch kroków:

1. Utworzenia konfiguracji:
 - Networks → Add network
 - Name: vlan10-config (nazwa konfiguracji)
 - Driver: macvlan

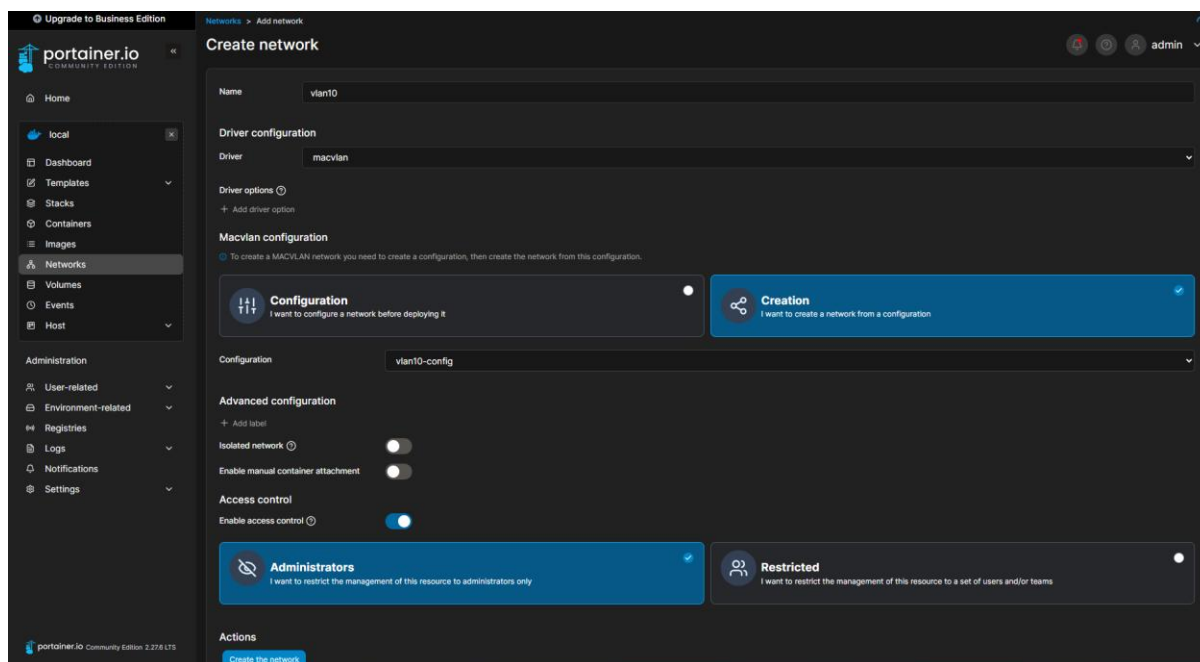
- Macvlan configuration: Configuration: „I want to configure a network before deploying it”
 - Parent network card (Nadrzędna karta sieciowa): nazwa interfejsu hosta enp6s18
- IPv4 Network configuration:
- Subnet: 10.10.10.0/24 (Cała podsieć VLAN 10)
 - Gateway: 10.10.10.1 (brama pfSense dla VLAN 10)
 - IP range: (Zakres adresów IP wewnątrz podsieci 10.10.10.0/24, który nie jest zarezerwowany dla serwera DHCP pfSense dla hostów VM)
10.10.10.128/26 (zapewnia to adresy IP od .129 do .190 dla kontenerów – 62 hosty)
 - Create the Network



Rys. 13 – Portainer – vlan10-config.

2. Wdrożenie konfiguracji:

- Networks → Add network
- Name: vlan10 (nazwa wdrożenia)
- Driver: macvlan
- Macvlan configuration: „Creation: I want to create a network from a configuration”
- Configuration: vlan10-config (wybór sieci konfiguracyjnej)
- Create the network



Rys. 14 – Portainer – vlan10.

4.4. Budowa serwera Wazuh

Wazuh to darmowa platforma SIEM (Security Information and Event Management) oraz XDR (Extended Detection and Response) typu open-source, która pomaga organizacjom wykrywać zagrożenia, monitorować bezpieczeństwo i zachować zgodność.

4.4.1. Tworzenie maszyny wirtualnej oraz instalacja systemu bazowego (Ubuntu)

Wazuh został zainstalowany na bazowym systemie Ubuntu Server 24.04 w taki sam sposób jak w rozdziale 4.3.

ID VM: 205

Nazwa: wazuh

OS:Ubuntu Server 24.04. (Live).

Dysk: 200 GB (vm-drives).

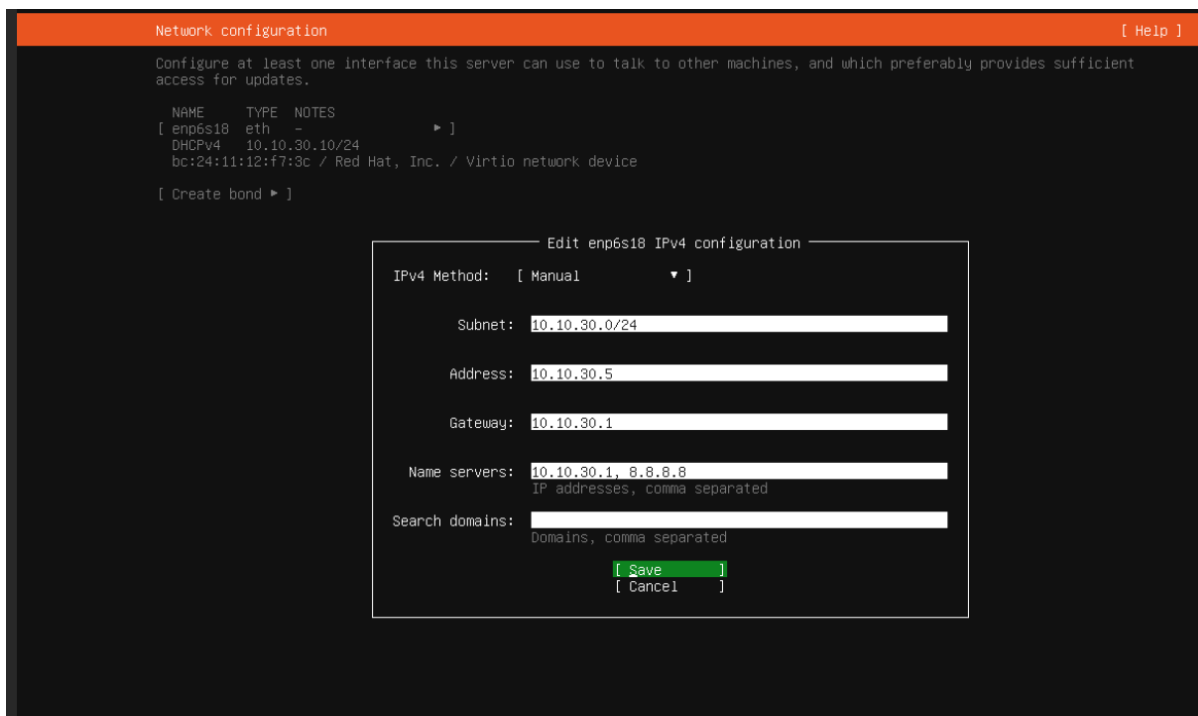
CPU: 8 rdzeni.

RAM: 8 GB.

Interfejs sieciowy: podłączony do vmbr1 (LAN) z tagiem VLAN 30.

Instalacja z ustawieniami domyślnymi + dodatkowo instalacja usługi serwera ssh.

Adres IP maszyny ustawiony statycznie na 10.10.30.5.



Rys. 15 – Instalacja Ubuntu – konfiguracja sieci.

4.4.1. Instalacja Wazuh

Instalacja Wazuh na Ubuntu jest bardzo prosta i sprowadza się do uruchomienia skryptu instalacyjnego **wazuh-install.sh**.

Sposób instalacji jest dokładnie przedstawiony w oficjalnej dokumentacji:

<https://documentation.wazuh.com/current/quickstart.html>

i polega na wykonaniu jednego polecenia:

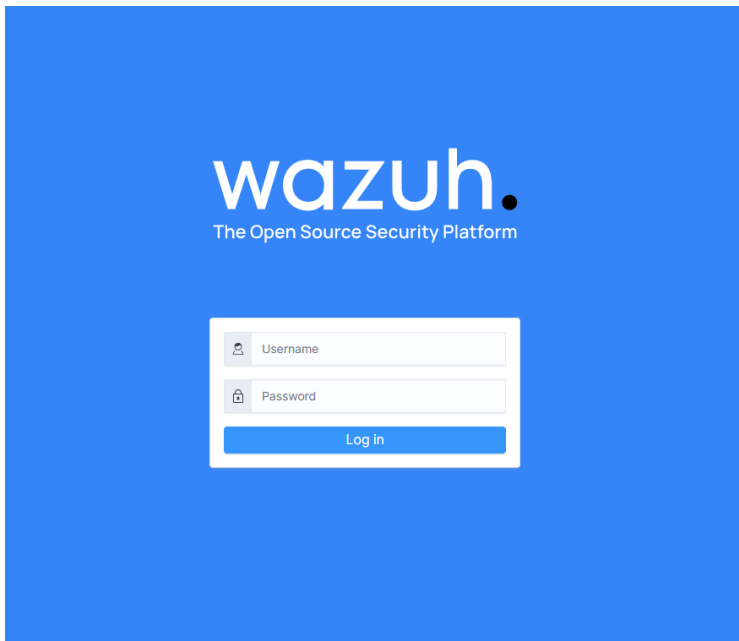
```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
```

To polecenie pobiera skrypt instalacyjny Wazuh i uruchamia go z flagą **-a**, która instaluje wszystkie komponenty (Manager, Indexer, Dashboard) na jednym serwerze.

Ważne na tym etapie jest zachowanie hasła do panelu webowego przedstawionego przez instalator.

```
kamil@wazuh: ~  
01/07/2025 09:19:49 INFO: Wazuh indexer installation finished.  
01/07/2025 09:19:49 INFO: Wazuh indexer post-install configuration finished.  
01/07/2025 09:19:49 INFO: Starting service wazuh-indexer.  
01/07/2025 09:19:57 INFO: wazuh-indexer service started.  
01/07/2025 09:19:57 INFO: Initializing Wazuh indexer cluster security settings.  
01/07/2025 09:20:08 INFO: Wazuh indexer cluster initialized.  
01/07/2025 09:20:08 INFO: --- Wazuh server ---  
01/07/2025 09:20:08 INFO: Starting the Wazuh manager installation.  
01/07/2025 09:22:35 INFO: Wazuh manager installation finished.  
01/07/2025 09:22:35 INFO: Starting service wazuh-manager.  
01/07/2025 09:22:50 INFO: wazuh-manager service started.  
01/07/2025 09:22:50 INFO: Starting Filebeat installation.  
01/07/2025 09:23:08 INFO: Filebeat installation finished.  
01/07/2025 09:23:09 INFO: Filebeat post-install configuration finished.  
01/07/2025 09:23:09 INFO: Starting service filebeat.  
01/07/2025 09:23:09 INFO: filebeat service started.  
01/07/2025 09:23:09 INFO: --- Wazuh dashboard ---  
01/07/2025 09:23:09 INFO: Starting Wazuh dashboard installation.  
01/07/2025 09:25:35 INFO: Wazuh dashboard installation finished.  
01/07/2025 09:25:35 INFO: Wazuh dashboard post-install configuration finished.  
01/07/2025 09:25:35 INFO: Starting service wazuh-dashboard.  
01/07/2025 09:25:35 INFO: wazuh-dashboard service started.  
01/07/2025 09:25:51 INFO: Initializing Wazuh dashboard web application.  
01/07/2025 09:25:53 INFO: Wazuh dashboard web application initialized.  
01/07/2025 09:25:53 INFO: --- Summary ---  
01/07/2025 09:25:53 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443  
User: admin  
Password:   
01/07/2025 09:25:53 INFO: Installation finished.  
kamil@wazuh:~$
```

Rys. 16 – Instalacja Wazuh.



Rys. 17 – Okno logowania do panelu Wazuh.

4.4.2. Instalacja agentów Wazuh

Agenci Wazuh zostaną zainstalowani na maszynach: pfSense, Docker oraz Kali Linux. Umożliwi to zbieranie logów z tych maszyn przez serwer Wazuh.

Instalacja agenta na Kali Linux

Przełączenie się na użytkownika Root

```
sudo -i
```

Dodanie repozytorium Wazuh Agent (zgodnie z dokumentacją:

<https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-linux.html> -> sekcja "Deploy a Wazuh agent", zakładka APT):

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --dearmor | tee
/usr/share/keyrings/wazuh.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" |
tee -a /etc/apt/sources.list.d/wazuh.list
apt-get update
```

Instalacja agenta Wazuh, wskazując adres IP serwera Wazuh:

```
WAZUH_MANAGER='10.10.30.5' apt-get install wazuh-agent
```

Uruchomienie usługi agenta Wazuh:

```
systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent
```