

# Projekt sieci i systemów korporacyjnej firmy telemarketingowej

Patryk Kaniewski, Dominik Gandziarek, Jakub Caban

2021-05-19

# Wymagania Funkcjonalne

## Ogólne

- ▶ Dostępność sieci w każdym pomieszczeniu firmy
  - ▶ sieć kablowa dla każdego stanowiska
  - ▶ sieć bezprzewodowa na sali konferencyjnej
  - ▶ sieć bezprzewodowa dla gości (konferencja)
- ▶ Wysoka dostępność systemów
- ▶ Zapewnienie redundantnej komunikacji między gałęziami sieci
- ▶ Możliwość zarządzania usługami dostarczonymi za pomocą technologii docker
  - ▶ uruchomienie app1 + db1
- ▶ Możliwość zarządzania usługami dostarczonymi za pomocą virtualizacji
  - ▶ uruchomienie app2 + db2

# Wymagania Funkcjonalne (kont.)

## Monitoring sieci

- ▶ Administrator ma dostęp do statystyk użytkowania systemów
- ▶ Administrator dostaje automatyczne powiadomienia przy dużych zmianach w pracy

## Autoryzacja na poziomie sieci

- ▶ użytkownik siadając do jakiegokolwiek biurka loguje się do sieci
- ▶ ograniczenie dostępu do części sieci nieuprawnionym użytkownikom (zarząd, pracownik, administrator, gość konferencyjny)

## Wymagania Funkcjonalne (kont.)

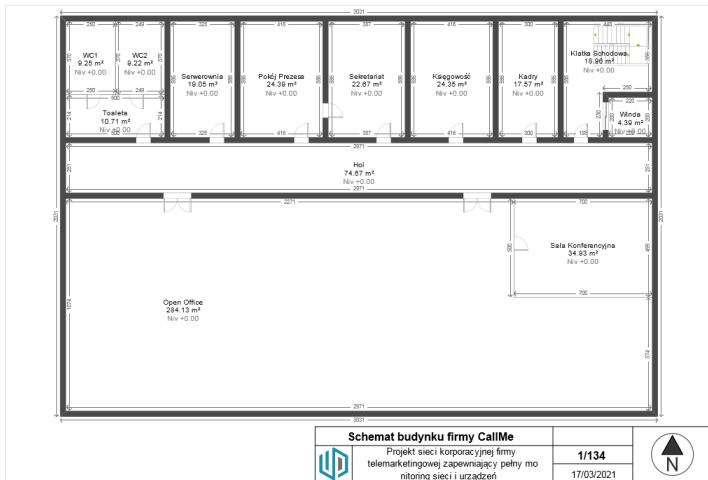
### DHCP

- ▶ usługa musi być redundantna
- ▶ jeśli jeden serwer DHCP ulegnie awarii, drugi serwer zastpczy przejmując jego zadania
- ▶ Adresy IP muszą być rozdzielane z określonej puli adresów.
- ▶ Serwer zastępczy (slave) nasłuchuje na odpowiednim interfejsie sieciowym, jeśli nie otrzymuje odpowiedzi
- ▶ od serwera podstawowego (master) zaczyna rozdawać hostom adresy.

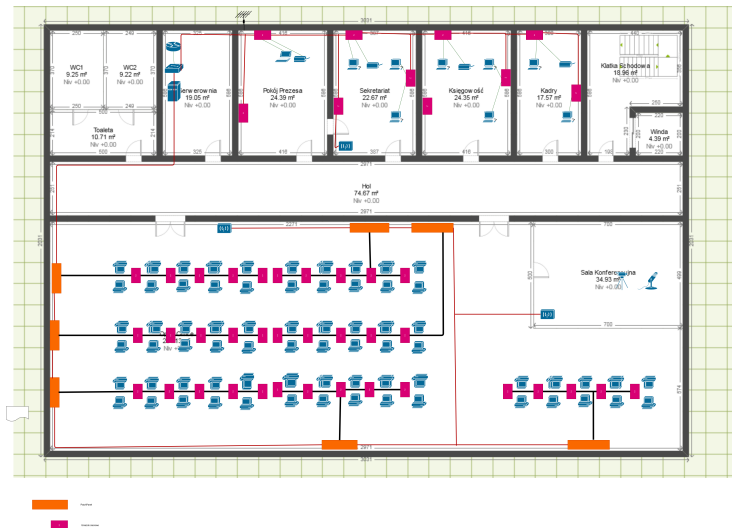
### DNS

- ▶ usługa musi umożliwiać translację nazw domenowych na adresy IP jak i odwrotnie (strefy wyszukiwania wstecz i do przodu);

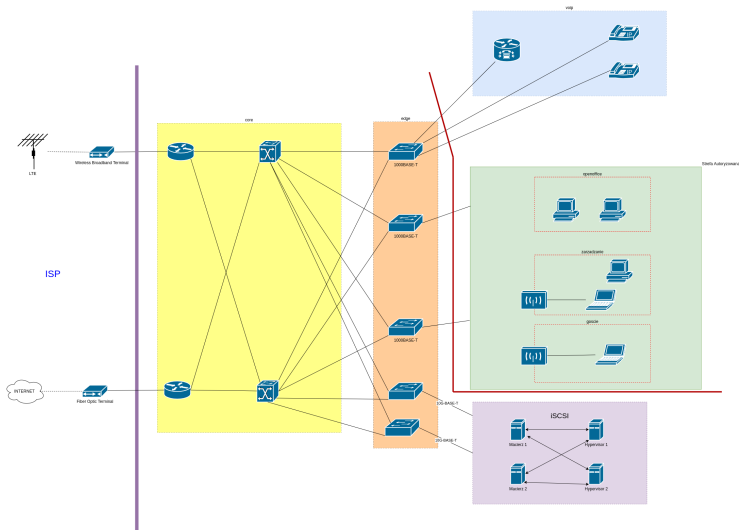
# Plan bundynku



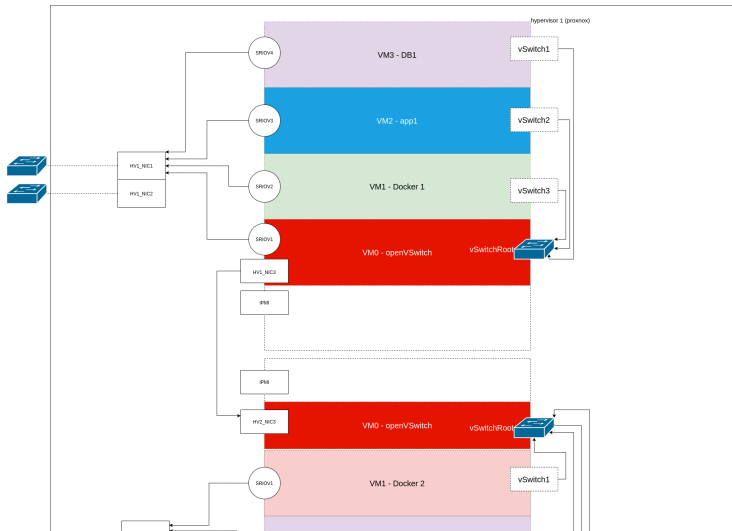
# Plan fizyczny sieci



# Plan logiczny sieci



# Architektura serwerowa

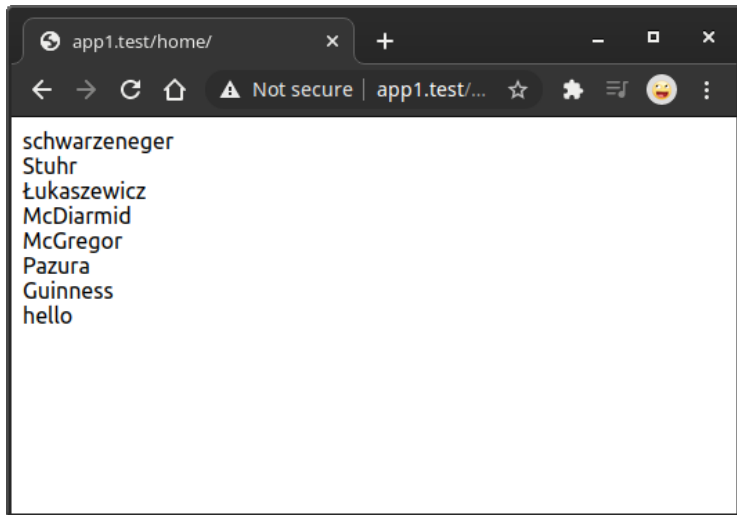




# App1

```
<?php
$host = '192.168.200.132';
$db = 'lab3'; $user = 'admin'; $pass = 'pwsz';
$port = '3306'; $charset = 'utf8';
$options = [
    \PDO::ATTR_ERRMODE            => \PDO::ERRMODE_EXCEPTION,
    \PDO::ATTR_DEFAULT_FETCH_MODE => \PDO::FETCH_ASSOC,
    \PDO::ATTR_EMULATE_PREPARES  => false,
];
$dsn = "mysql:host=$host;dbname=$db;charset=$charset;port=$port";
try {
    $pdo = new \PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}
$stmt = $pdo->query("SELECT * FROM aktorzy");
while ($row = $stmt->fetch()) {
    echo $row['nazwisko']."<br />\n";
}
?>
```

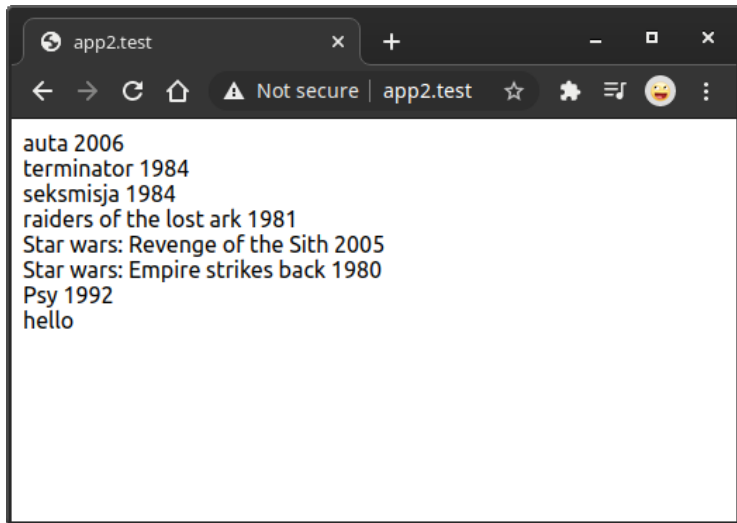
# App1 + db1



# App2

```
<?php
$host = '192.168.200.145';
$db = 'lab3'; $user = 'admin'; $pass = 'pwsz';
$port = '3306'; $charset = 'utf8';
$options = [
    \PDO::ATTR_ERRMODE            => \PDO::ERRMODE_EXCEPTION,
    \PDO::ATTR_DEFAULT_FETCH_MODE => \PDO::FETCH_ASSOC,
    \PDO::ATTR_EMULATE_PREPARES   => false,
];
$dsn = "mysql:host=$host;dbname=$db;charset=$charset;port=$port";
try {
    $pdo = new \PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}
$stmt = $pdo->query("SELECT * FROM filmy");
while ($row = $stmt->fetch()) {
    echo $row['tytul']." ".$row['rok']."<br />\n";
}
?>
```

## App2 + db2



# Zabbix

Zabbix jest rozwiązanie open-source (GPLv2) do monitorowania dużej ilości komponentów sieci komputerowej w tym:

- ▶ sieci
- ▶ urządzeń sieciowych
- ▶ stacji roboczych
- ▶ serwerów
- ▶ usług

# Zabbix

The screenshot displays the Zabbix web interface in a browser window. The top navigation bar includes the Zabbix logo and a sidebar menu with options like Monitoring, Inventory, Reports, Configuration, Administration, Support, Show, Help, User settings, and Sign out. The main content area is titled 'Global view' and contains several panels:

- System information:** A table showing various system metrics such as 'Zabbix server is running' (Yes), 'Number of hosts' (3), 'Number of templates' (180), 'Number of items' (127), 'Number of triggers' (56), 'Number of users' (3), and 'Required server performance' (1.52).
- Host availability:** A summary table showing the status of hosts (Available, Not available, Unreachable, Total) and a visual representation of the status (1 Critical, 0 High, 1 Average, 0 Warning, 0 Information, 0 Not classified).
- Problems:** A table listing recent problems, including 'Zabbix server' and 'Zabbix agent' issues, with columns for Time, Recovery time, Status, Info, Host, Problem, Severity, Duration, Ack, Actions, and Tags.
- Favorite graphs:** A section showing links to favorite graphs, such as 'Zabbix server CPU use time' and 'Zabbix server interface output: Bits received'.

# Powiadomienia

```
thisconnect@zabbix-server:/usr/lib/zabbix/alertscripts$ cat zabbix-ssmt  
#!/bin/bash  
  
echo "sending mail"  
echo "$3" | /usr/bin/mail -s "$2" $1  
  
exit 0  
thisconnect@zabbix-server:/usr/lib/zabbix/alertscripts$
```

## Powiadomienia (kont.)

Media type

Message templates 1

Options

\* Name

email-script

Type

Script

\* Script name

zabbix-ssmtp

Script parameters

Parameter	Action
{ALERT.SENDTO}	<a href="#">Remove</a>
{ALERT.SUBJECT}	<a href="#">Remove</a>
{ALERT.MESSAGE}	<a href="#">Remove</a>
<a href="#">Add</a>	

Description

zabbix alert using smtp

Enabled

☒

Update

Clone

Delete

Cancel





# 802.1x

# DHCP

# DNS