DEPLOYMENT_GUIDE_ISARD

IsardVDI & Proxmox Copy-Paste Instalatzeko Gida

Bertsioa: 3.2 (Instalazio Gida Osoa) **Aurrebaldintzak:** 1. **6 BM sortuta** IsardVDI/Proxmox-en **Debian 12** instalatuta. 2. **SSH Sarbidea** BM guztietara (Erabiltzailea: debian edo root). 3. **Interneteko Sarbidea** Gateway BM-aren WAN interfazearekin.

🛑 0. FASEA: BM Sortzearen Espezifikazioak (Eskuzko Urratsa)

Sortu BM hauek zure hiperbisorean.

BM Izena	vCPU	RAM	HDD	Sarea 1 (WAN)	Sarea 2 (LAN)
ZG-Gateway	1	1GB	10GB	Bridge (Internet)	Internal (Isolated)
ZG-App	2	4GB	20GB	-	Internal (Isolated)
ZG-Data	2	4GB	20GB	-	Internal (Isolated)
ZG-SecOps	4	8GB	40GB	-	Internal (Isolated)
ZG-OT	1	2GB	10GB	-	Internal (Isolated)
ZG-Client	2	4GB	20GB	-	Internal (Isolated)

🌍 1. FASEA: ZG-GATEWAY IMPLEMENTAZIOA

Hasi saioa ZG-Gateway-n SSH bidez.

1.1. Identifikatu Sareko Interfazeak (KRITIKOA)

Debian 12 KVM-n interfazeak ens18, ens19, etab. izenez daude, EZ eth0.

Exekutatu komando hau zure interfazeak ikusteko:

```
ip -br link
```

- IP helbidea duena (DHCP-tik) zure **WAN** da.
- “DOWN” dagoena edo IPrik gabea zure **LAN** da.

👇 KOPIATU ETA ITSASTU BLOKE HAU (Izenak eguneratu behar badira) 👇

```
# EZARRI ZURE INTERFAZE IZENAK HEMEN
WAN_IF="ens18"   # Ordeztu zure WAN interfazearekin
```

```
LAN_IF="ens19"   # Ordeztu zure LAN interfazearekin

# Bihurtu root
sudo -i

# Instalatu oinarrizko tresnak
apt update && apt install -y vim curl wget nftables isc-dhcp-server

# Konfiguratu Interfazeak
cp /etc/network/interfaces /etc/network/interfaces.bak
cat <<EOF > /etc/network/interfaces
source /etc/network/interfaces.d/*
auto lo
iface lo inet loopback
allow-hotplug $WAN_IF
iface $WAN_IF inet dhcp
allow-hotplug $LAN_IF
iface $LAN_IF inet static
    address 192.168.1.1
    netmask 255.255.0.0
    up ip addr add 192.168.10.1/24 dev $LAN_IF
    up ip addr add 192.168.20.1/24 dev $LAN_IF
    up ip addr add 192.168.50.1/24 dev $LAN_IF
    up ip addr add 192.168.200.1/24 dev $LAN_IF
EOF

# Gaitu IP Birbideraketa
echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-routing.conf
sysctl -p /etc/sysctl.d/99-routing.conf

# Berrabiarazi Sarea
systemctl restart networking

# Konfiguratu DHCP
sed -i "s/INTERFACESv4=""/INTERFACESv4=\"$LAN_IF\"/" /etc/default/isc-dhcp-server
cat <<EOF > /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
authoritative;
subnet 192.168.10.0 netmask 255.255.255.0 {
  range 192.168.10.100 192.168.10.200;
  option routers 192.168.10.1;
  option domain-name-servers 8.8.8.8;
}
subnet 192.168.20.0 netmask 255.255.255.0 {
  range 192.168.20.100 192.168.20.200;
  option routers 192.168.20.1;
  option domain-name-servers 8.8.8.8;
}
subnet 192.168.50.0 netmask 255.255.255.0 {
  range 192.168.50.100 192.168.50.200;
  option routers 192.168.50.1;
}
subnet 192.168.200.0 netmask 255.255.255.0 {
  range 192.168.200.100 192.168.200.200;
  option routers 192.168.200.1;
}
EOF
systemctl restart isc-dhcp-server

# Konfiguratu NFTables
cat <<EOF > /etc/nftables.conf
#!/usr/sbin/nft -f
flush ruleset
table ip filter {
    chain input {
        type filter hook input priority 0; policy drop;
        iifname "lo" accept
```

```
        ct state established,related accept
        tcp dport 22 accept
        ip protocol icmp accept
    }
    chain forward {
        type filter hook forward priority 0; policy drop;
        ct state established,related accept
        ip saddr 192.168.10.0/24 ip daddr 192.168.20.0/24 tcp dport { 80, 443 } accept
        ip saddr 192.168.20.0/24 ip daddr 192.168.20.0/24 tcp dport { 5432, 6379 } accept
        ip saddr 192.168.200.0/24 accept
        iifname "$LAN_IF" oifname "$WAN_IF" accept
    }
    chain output {
        type filter hook output priority 0; policy accept;
    }
}
table ip nat {
    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
        oifname "$WAN_IF" masquerade
    }
}
EOF
nft -f /etc/nftables.conf
systemctl enable nftables
```

💾 2. FASEA: ZG-DATA IMPLEMENTAZIOA

Hasi saioa ZG-Data-n.

```
sudo -i
apt update && apt install -y curl
curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh
mkdir -p /opt/zabala-data && cd /opt/zabala-data
cat <<EOF > docker-compose.yml
version: '3.8'
services:
  postgres:
    image: postgres:16-alpine
    container_name: zabala-postgres
    restart: always
    environment:
      POSTGRES_USER: zabala_user
      POSTGRES_PASSWORD: secure_password_123
      POSTGRES_DB: zabala_db
    volumes:
      - ./pgdata:/var/lib/postgresql/data
    ports:
      - "5432:5432"
  redis:
    image: redis:7-alpine
    container_name: zabala-redis
    restart: always
    command: redis-server --requirepass secure_redis_123
    ports:
      - "6379:6379"
EOF
docker compose up -d
```

💻 3. FASEA: ZG-APP IMPLEMENTAZIOA

Hasi saioa ZG-App-n.

```
sudo -i
apt update && apt install -y curl git
curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh
mkdir -p /opt/zabala-app && cd /opt/zabala-app
cat <<EOF > docker-compose.yml
version: '3.8'
services:
  api:
    image: php:8.4-apache
    container_name: zabala-api
    restart: always
    environment:
      DB_HOST: 192.168.20.20
      DB_USER: zabala_user
      DB_PASS: secure_password_123
      DB_NAME: zabala_db
      REDIS_HOST: 192.168.20.20
      REDIS_PASS: secure_redis_123
    ports:
      - "80:80"
    volumes:
      - ./src:/var/www/html
EOF
# Klonatu edo kopiatu src fitxategiak hemen.
docker compose up -d
```

🛡️ 4. FASEA: ZG-SECOPS IMPLEMENTAZIOA

Hasi saioa ZG-SecOps-n.

```
sudo -i
apt update && apt install -y curl git
curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh
sysctl -w vm.max_map_count=262144
echo "vm.max_map_count=262144" >> /etc/sysctl.conf

#Wazuh
mkdir -p /opt/wazuh && cd /opt/wazuh
git clone https://github.com/wazuh/wazuh-docker.git .
docker compose -f generate-indexer-certs.yml run --rm generator
docker compose up -d

# Honeypots
mkdir -p /opt/honeypots && cd /opt/honeypots
cat <<EOF > docker-compose.yml
version: '3'
services:
  conpot:
    image: honeycomb/conpot:latest
    container_name: honey-conpot
    ports:
      - "5020:502"
      - "1610:161"
  dionaea:
    image: dinotools/dionaea:latest
    container_name: honey-dionaea
    ports:
      - "21:21"
      - "445:445"
```

```
    cap_add:
      - NET_ADMIN
EOF
docker compose up -d
```

🏭 5. FASEA: ZG-OT IMPLEMENTAZIOA

Hasi saioa ZG-OT-n.

```
sudo -i
apt update && apt install -y curl
curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh
mkdir -p /opt/openplc && cd /opt/openplc
cat <<EOF > docker-compose.yml
version: '3.8'
services:
  openplc:
    image: thiagoralves/openplc:v3
    container_name: ot-plc
    ports:
      - "8080:8080"
      - "502:502"
    privileged: true
EOF
docker compose up -d
```

🗄️ 7. FASEA: DATU-BASEAREN MIGRAZIOAK

Hasi saioa ZG-App-n.

```
cd /opt/zabala-app
docker compose exec -T api apt update && docker compose exec -T api apt install -y postgresql-client
# Exekutatu SQL fitxategiak migrations karpetatik
docker compose exec -T api /bin/bash -c \
  'for f in /var/www/html/migrations/*.sql; do
    echo "Aplikatzen $f..."
    export PGPASSWORD=$DB_PASS
    psql -h $DB_HOST -U $DB_USER -d $DB_NAME -f "$f"
  done
'
```

Instalazioa Osatuta.