Final Project

Alejandro Pérez Bueno

Jun 09, 2024

Table of Contents

OS Setup and user creation 2

Install OS into the VM 2

User creation 3

Services stack 4

Server postgres 4

Web Server Deno 6

Reverse Proxy 6

Annexes 7

OS Setup and user creation

Install OS into the VM

First and foremost, we need the following:

1. A Linux Server ISO image. I downloaded mine from here: https://ubuntu.com/download/server.
2. A Virtualization Program. I typically use `virt-manager`, a well-known QEMU/KVM client for Linux.

You then set up the VM specifying:

- RAM usage
- CPU cores
- Virtual disk space
- Specify other hardware (input devices, GPU acceleration, USB redireccion)

See Figure 1.

Then we must run the ISO installer to setup:

- Language setup
- Network
- Base packages
- User creation

See Figure 2.

ℹ Note

The live installer has some policy that prevents uppercase characters in the hostname. Thus, I had to set up the correct hostname *after* the initial installation. See Figure 3.

Evidence

> ⚠ Warning
>
> For ease of use, from this point on I will ssh into the guest machine so I can more easily manage the VM from the host. To set it up, I did the folloing:
>
> ```
> sudo apt update                                                  ①
> sudo apt install ssh                                             ②
> sudo nano /etc/ssh/ssh_config                                    ③
> sudo ufw allow 22                                                ④
> sudo systemctl enable --now ssh                                  ⑤
> ```
>
> ① Update repositories
> ② Install `ssh`
> ③ Open port 22 to allow ssh on that port
> ④ Add firewall rule to allow ssh to go through it
> ⑤ Start ssh service and set it to start on boot
>
> Then, all I had to do is:
>
> 1. Find guest IP (run `ip a` from guest)
> 2. Access it from the host. See Figure 4

- Host IP: *192.168.1.206*
- Guest IP: *192.168.122.175*

See Figure 5 to verify that I can access the guest Ubuntu Server from my host machine.

User creation

The simplest way to create four users and set their passwords to automatically expire every year is with a small shell script:

Listing 0.1 `user_creation.sh`

```
#!/bin/bash

for i in {1..4}                                                        ①
do
  username="user$(printf '%02d' $i)"                                   ②
  sudo useradd $username                                               ③
  echo "User $username created"

  sudo chage --maxdays 365 $username                                   ④
  echo "Password for $username set to expire every year!"
done                                                                   ①
```

① Loop over four users to create and modify their password policy

② Dynamically change username to `user` + `[01..04]`
③ Create user
④ Set password to reset after 365 days (every year) for the previously created user

See Figure 6a.

Evidence

Listing 0.2 `user_info.sh`

```
#!/bin/bash

for i in {1..4}
do
  username="user$(printf '%02d' $i)"
  sudo chage -l $username
done
```

See Figure 6b.

Services stack

Server postgres

Install *docker* and *docker-compose*

Following the official documentation:

```
# 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

# 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 \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

# Install docker
```

```
sudo apt-get install docker-ce docker-ce-cli \
  containerd.io docker-buildx-plugin docker-compose-plugin

## Run a test container
sudo docker run hello-world
```

Configuring a service stack with docker compose

```
git clone https://github.com/jestebangr/prac20232-orig.git
```

My resulting docker compose file:

Listing 0.3 `docker-compose.yaml`

```
version: '3.9'

services:
  db:
    image: postgres:16.2
    container_name: dbhost
    ports:                                                        ①
      - "5432:5432"
    environment:                                                  ②
      POSTGRES_DB: ${POSTGRES_DB}
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes:                                                      ③
      - postgres_data:/var/lib/postgresql/data
      - ./dataset/init.sql:/docker-entrypoint-initdb.d/init.sql   ④

volumes:                                                          ⑤
  postgres_data:
```

① Add a ports configuration to expose Postgres default port (5432) to the host

② Use environment variables to configure the database without hardcoding sensitive information

③ Define a volume to ensure data persistence

④ To load the init.sql file automatically, use the docker-entrypoint-initdb.d directory which is automatically executed during container startup

⑤ Define the named volume

ℹ Note

It is necessary to set the credentials related to postgress before running the docker container in an env file:

Listing 0.4 `.env`

```
POSTGRES_DB="uoc2023"
POSTGRES_USER="aperez-b"
POSTGRES_PASSWORD="1234"
```

Evidence

```
cat docker-compose.yml
```

See Figure 7a.

```
sudo docker ps
```

See Figure 7b.

```
sudo netstat -a | grep postgresql
```

See Figure 7c.

```
nmap -p- --open -n 192.168.122.175
```

See Figure 7d.

```
psql -h localhost -p 5432 -U aperez-b -d uoc2023
```

See Figure 7e.

Web Server Deno

Evidence

```
cat docker-compose.yml
```

See Figure 8a.

```
cat Dockerfile
```

See Figure 8b.

```
sudo docker ps
```

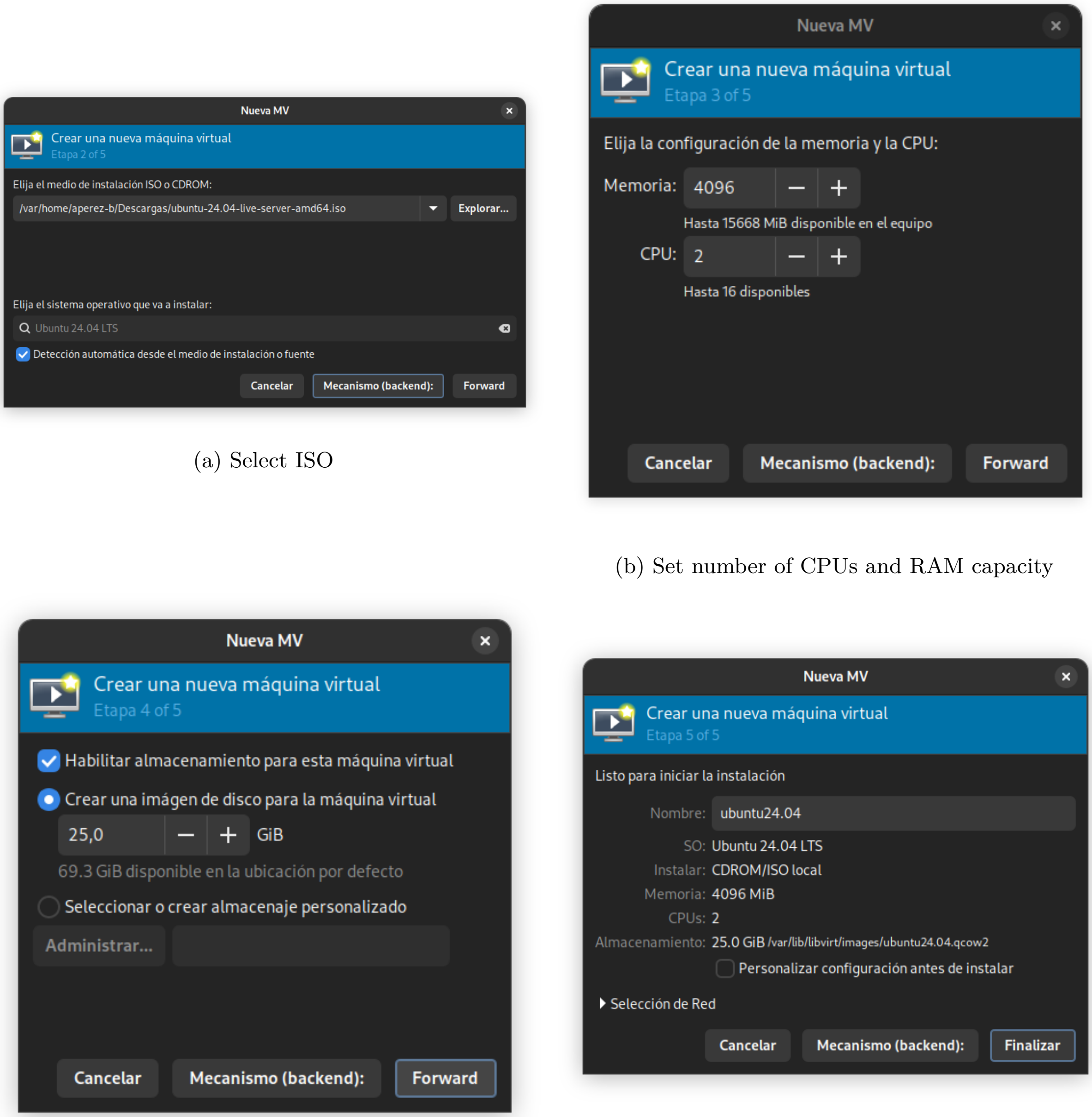
See Figure 8c.

- Webhost connection: see Figure 8d.

Reverse Proxy

See Figure 9 and Figure 10.

Annexes



Etapa 2 of 5

Elija el medio de instalación ISO o CDROM:

/var/home/aperez-b/Descargas/ubuntu-24.04-live-server-amd64.iso

Elija el sistema operativo que va a instalar:

Ubuntu 24.04 LTS

☑ Detección automática desde el medio de instalación o fuente

(a) Select ISO

Etapa 3 of 5

Elija la configuración de la memoria y la CPU:

Memoria: 4096

Hasta 15668 MiB disponible en el equipo

CPU: 2

Hasta 16 disponibles

(b) Set number of CPUs and RAM capacity

Etapa 4 of 5

☑ Habilitar almacenamiento para esta máquina virtual

● Crear una imágen de disco para la máquina virtual

25,0 GiB

69.3 GiB disponible en la ubicación por defecto

○ Seleccionar o crear almacenaje personalizado

(c) Create virtual disk

Etapa 5 of 5

Listo para iniciar la instalación

Nombre: ubuntu24.04

SO: Ubuntu 24.04 LTS

Instalar: CDROM/ISO local

Memoria: 4096 MiB

CPUs: 2

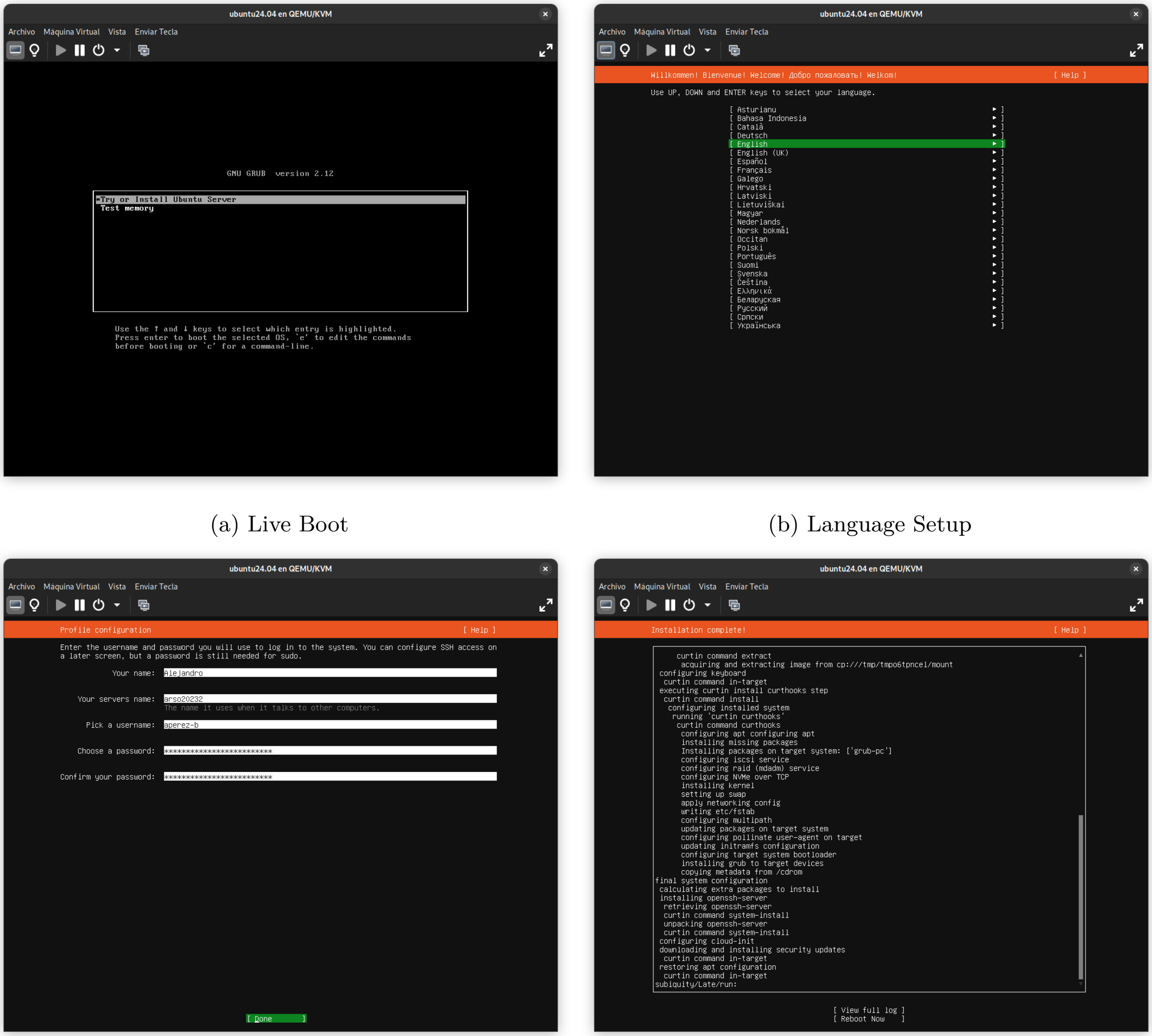
Almacenamiento: 25.0 GiB /var/lib/libvirt/images/ubuntu24.04.qcow2

□ Personalizar configuración antes de instalar

▸ Selección de Red

(d) Finish installation

Figure 1: Ubuntu Installation



```
GNU GRUB  version 2.12

*Try or Install Ubuntu Server
 Test memory

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
```

(a) Live Boot

```
Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom!        [ Help ]

Use UP, DOWN and ENTER keys to select your language.

[ Asturianu                                  ▸ ]
[ Bahasa Indonesia                           ▸ ]
[ Català                                     ▸ ]
[ Deutsch                                    ▸ ]
[ English                                    ▸ ]
[ English (UK)                               ▸ ]
[ Español                                    ▸ ]
[ Français                                   ▸ ]
[ Galego                                     ▸ ]
[ Hrvatski                                   ▸ ]
[ Latviski                                   ▸ ]
[ Lietuviškai                                ▸ ]
[ Magyar                                     ▸ ]
[ Nederlands                                 ▸ ]
[ Norsk bokmål                               ▸ ]
[ Occitan                                    ▸ ]
[ Polski                                     ▸ ]
[ Português                                  ▸ ]
[ Suomi                                      ▸ ]
[ Svenska                                    ▸ ]
[ Čeština                                    ▸ ]
[ Ελληνικά                                   ▸ ]
[ Беларуская                                 ▸ ]
[ Русский                                    ▸ ]
[ Српски                                     ▸ ]
[ Українська                                 ▸ ]
```

(b) Language Setup

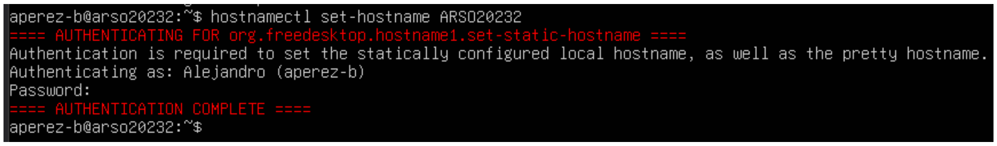
```
Profile configuration                                            [ Help ]

Enter the username and password you will use to log in to the system. You can configure SSH access on
a later screen, but a password is still needed for sudo.

            Your name: Alejandro

    Your servers name: arso20232
                       The name it uses when it talks to other computers.

     Pick a username: aperez-b

    Choose a password: *************************

Confirm your password: *************************

                        [ Done       ]
```

(c) User setup with UOC information

```
Installation complete!                                           [ Help ]

     curtin command extract
      acquiring and extracting image from cp:///tmp/tmpo6tpncel/mount
 configuring keyboard
  curtin command in-target
 executing curtin install curthooks step
  curtin command install
   configuring installed system
    running 'curtin curthooks'
     curtin command curthooks
      configuring apt configuring apt
      installing missing packages
      Installing packages on target system: ['grub-pc']
      configuring iscsi service
      configuring raid (mdadm) service
      configuring NVMe over TCP
      installing kernel
      setting up swap
      apply networking config
      writing etc/fstab
      configuring multipath
      updating packages on target system
      configuring pollinate user-agent on target
      updating initramfs configuration
      configuring target system bootloader
      installing grub to target devices
      copying metadata from /cdrom
final system configuration
 calculating extra packages to install
 installing openssh-server
  retrieving openssh-server
  curtin command system-install
  unpacking openssh-server
  curtin command system-install
 configuring cloud-init
 downloading and installing security updates
  curtin command in-target
 restoring apt configuration
  curtin command in-target
subiquity/Late/run:

                        [ View full log ]
                        [ Reboot Now    ]
```

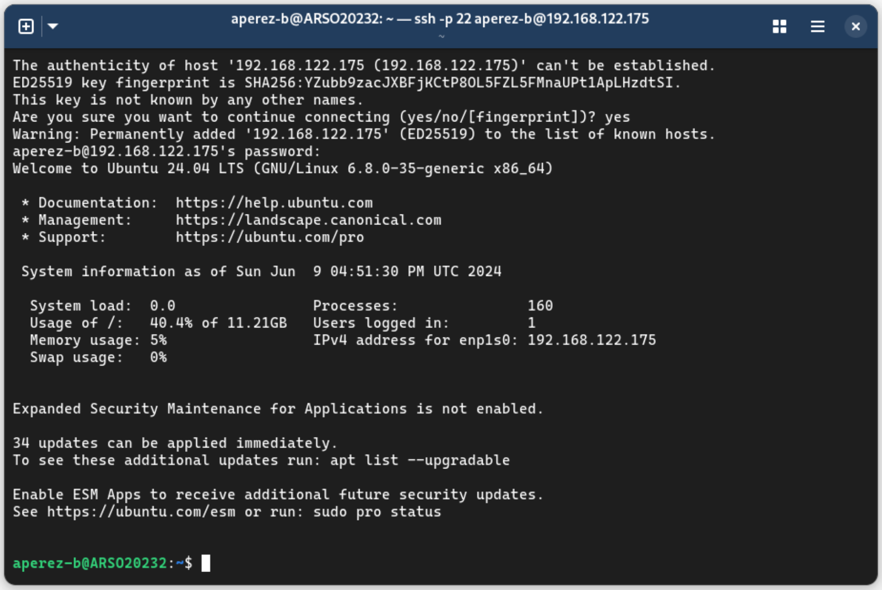
(d) Install complete

Figure 2: Live Boot configuration



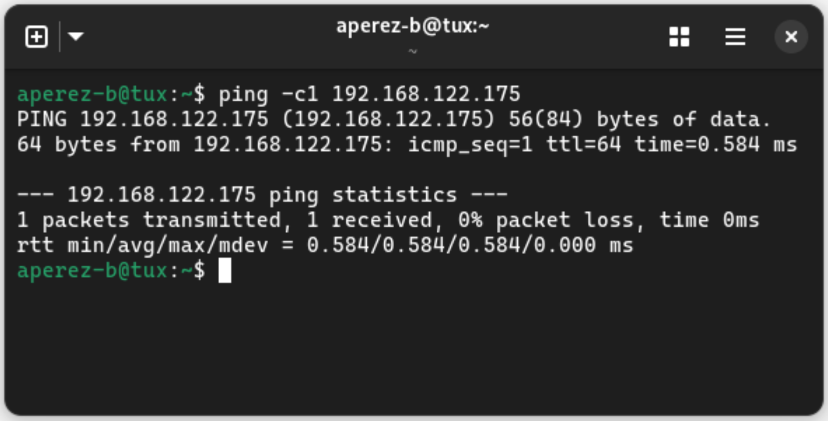
```
aperez-b@arso20232:~$ hostnamectl set-hostname ARSO20232
==== AUTHENTICATING FOR org.freedesktop.hostname1.set-static-hostname ====
Authentication is required to set the statically configured local hostname, as well as the pretty hostname.
Authenticating as: Alejandro (aperez-b)
Password:
==== AUTHENTICATION COMPLETE ====
aperez-b@arso20232:~$
```

Figure 3: Set hostname to ARSO20232 in the guest



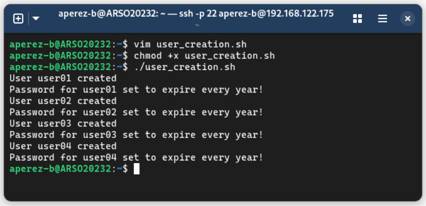
```
The authenticity of host '192.168.122.175 (192.168.122.175)' can't be established.
ED25519 key fingerprint is SHA256:YZubb9zacJXBFjKCtP8OL5FZL5FMnaUPt1ApLHzdtSI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.122.175' (ED25519) to the list of known hosts.
aperez-b@192.168.122.175's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Jun  9 04:51:30 PM UTC 2024

  System load:  0.0               Processes:               160
  Usage of /:   40.4% of 11.21GB   Users logged in:         1
  Memory usage: 5%                IPv4 address for enp1s0: 192.168.122.175
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

34 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


aperez-b@ARSO20232:~$ 
```

Figure 4: ssh into guest machine



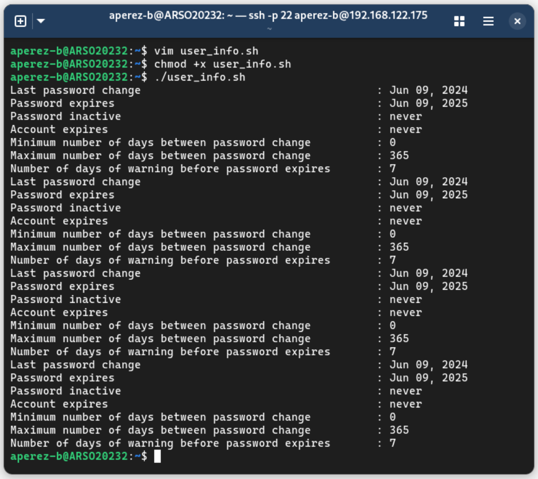
```
aperez-b@tux:~$ ping -c1 192.168.122.175
PING 192.168.122.175 (192.168.122.175) 56(84) bytes of data.
64 bytes from 192.168.122.175: icmp_seq=1 ttl=64 time=0.584 ms

--- 192.168.122.175 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.584/0.584/0.584/0.000 ms
aperez-b@tux:~$ 
```

Figure 5: Ping guest machine



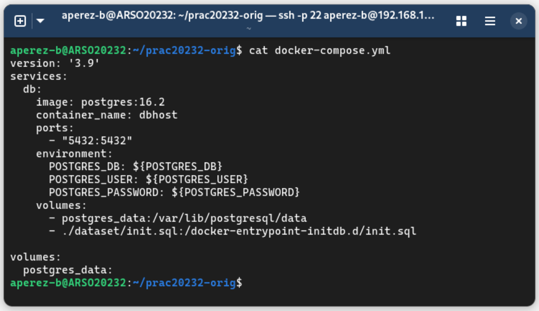
```
aperez-b@ARSO20232:~$ vim user_creation.sh
aperez-b@ARSO20232:~$ chmod +x user_creation.sh
aperez-b@ARSO20232:~$ ./user_creation.sh
User user01 created
Password for user01 set to expire every year!
User user02 created
Password for user02 set to expire every year!
User user03 created
Password for user03 set to expire every year!
User user04 created
Password for user04 set to expire every year!
aperez-b@ARSO20232:~$ 
```

(a) User creation and password policy script



```
aperez-b@ARSO20232:~$ vim user_info.sh
aperez-b@ARSO20232:~$ chmod +x user_info.sh
aperez-b@ARSO20232:~$ ./user_info.sh
Last password change                                    : Jun 09, 2024
Password expires                                        : Jun 09, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 365
Number of days of warning before password expires       : 7
Last password change                                    : Jun 09, 2024
Password expires                                        : Jun 09, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 365
Number of days of warning before password expires       : 7
Last password change                                    : Jun 09, 2024
Password expires                                        : Jun 09, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 365
Number of days of warning before password expires       : 7
Last password change                                    : Jun 09, 2024
Password expires                                        : Jun 09, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 365
Number of days of warning before password expires       : 7
aperez-b@ARSO20232:~$ 
```

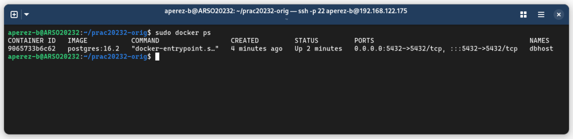
(b) sudo chage -l userXX



aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.1...

```
aperez-b@ARSO20232:~/prac20232-orig$ cat docker-compose.yml
version: '3.9'
services:
  db:
    image: postgres:16.2
    container_name: dbhost
    ports:
      - "5432:5432"
    environment:
      POSTGRES_DB: ${POSTGRES_DB}
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./dataset/init.sql:/docker-entrypoint-initdb.d/init.sql

volumes:
  postgres_data:
aperez-b@ARSO20232:~/prac20232-orig$
```

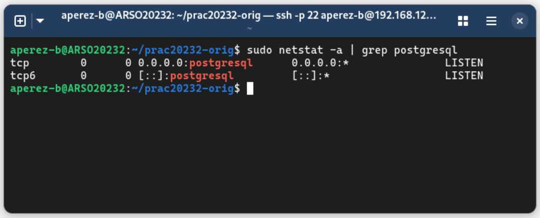
(a) `cat docker-compose.yaml`



aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ sudo docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS                                       NAMES
9065733b6c62   postgres:16.2   "docker-entrypoint.s…"   4 minutes ago   Up 2 minutes   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   dbhost
aperez-b@ARSO20232:~/prac20232-orig$ 
```

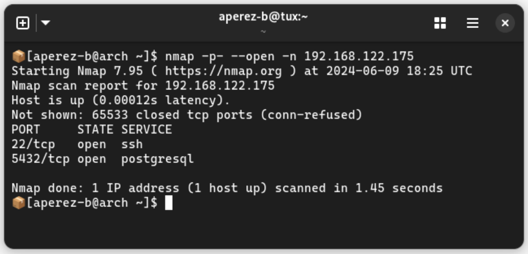
(b) `sudo docker ps`



aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.12...

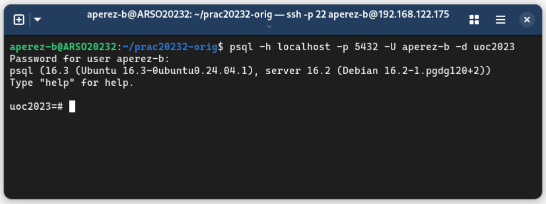
```
aperez-b@ARSO20232:~/prac20232-orig$ sudo netstat -a | grep postgresql
tcp        0      0 0.0.0.0:postgresql      0.0.0.0:*               LISTEN
tcp6       0      0 [::]:postgresql         [::]:*                  LISTEN
aperez-b@ARSO20232:~/prac20232-orig$ 
```

(c) `sudo netstat -a | grep postgresql`



```
📦[aperez-b@arch ~]$ nmap -p- --open -n 192.168.122.175
Starting Nmap 7.95 ( https://nmap.org ) at 2024-06-09 18:25 UTC
Nmap scan report for 192.168.122.175
Host is up (0.00012s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 1.45 seconds
📦[aperez-b@arch ~]$ 
```

(d) `nmap -p- --open -n 192.168.122.175`

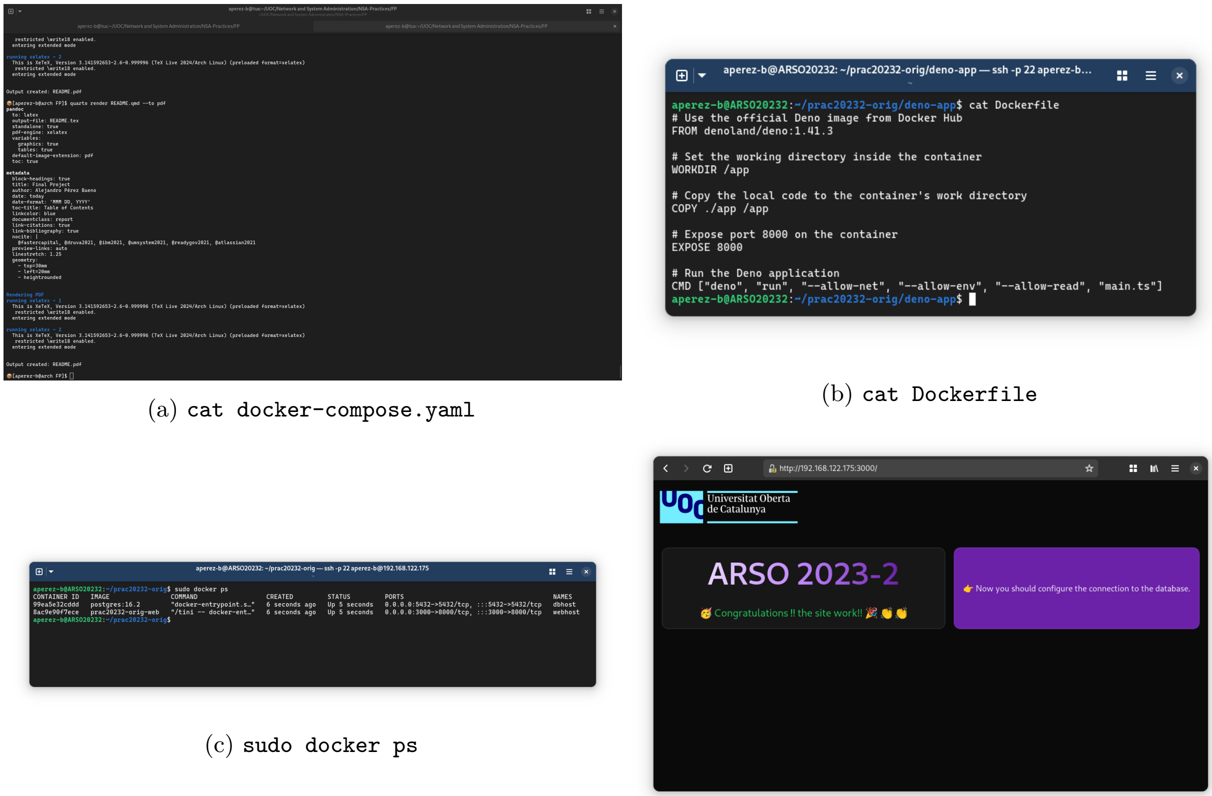


aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ psql -h localhost -p 5432 -U aperez-b -d uoc2023
Password for user aperez-b:
psql (16.3 (Ubuntu 16.3-0ubuntu0.24.04.1), server 16.2 (Debian 16.2-1.pgdg120+2))
Type "help" for help.

uoc2023=# 
```

(e) `psql -h localhost -p 5432 -U aperez-b -d uoc2023`

Figure 7: Evidences for `postgres` server


aperez-b@tux:~/UOC/Network and System Administration/NSA-Practices/FP
~/UOC/Network and System Administration/NSA-Practices/FP
aperez-b@tux:~/UOC/Network and System Administration/NSA-Practices/FP
aperez-b@tux:~/UOC/Network and System Administration/NSA-Practices/FP

```
   restricted \write18 enabled.
  entering extended mode

running xelatex - 2
  This is XeTeX, Version 3.141592653-2.6-0.999996 (TeX Live 2024/Arch Linux) (preloaded format=xelatex)
   restricted \write18 enabled.
  entering extended mode


Output created: README.pdf

📦[aperez-b@arch FP]$ quarto render README.qmd --to pdf
pandoc
  to: latex
  output-file: README.tex
  standalone: true
  pdf-engine: xelatex
  variables:
    graphics: true
    tables: true
  default-image-extension: pdf
  toc: true

metadata
  block-headings: true
  title: Final Project
  author: Alejandro Pérez Bueno
  date: today
  date-format: 'MMM DD, YYYY'
  toc-title: Table of Contents
  linkcolor: blue
  documentclass: report
  link-citations: true
  link-bibliography: true
  nocite: |
    @fastercapital, @druva2021, @ibm2021, @umsystem2021, @readygov2021, @atlassian2021
  preview-links: auto
  linestretch: 1.25
  geometry:
    - top=30mm
    - left=20mm
    - heightrounded


Rendering PDF
running xelatex - 1
  This is XeTeX, Version 3.141592653-2.6-0.999996 (TeX Live 2024/Arch Linux) (preloaded format=xelatex)
   restricted \write18 enabled.
  entering extended mode

running xelatex - 2
  This is XeTeX, Version 3.141592653-2.6-0.999996 (TeX Live 2024/Arch Linux) (preloaded format=xelatex)
   restricted \write18 enabled.
  entering extended mode


Output created: README.pdf

📦[aperez-b@arch FP]$ 
```

(a) `cat docker-compose.yaml`

aperez-b@ARSO20232: ~/prac20232-orig/deno-app — ssh -p 22 aperez-b...

```
aperez-b@ARSO20232:~/prac20232-orig/deno-app$ cat Dockerfile
# Use the official Deno image from Docker Hub
FROM denoland/deno:1.41.3

# Set the working directory inside the container
WORKDIR /app

# Copy the local code to the container's work directory
COPY ./app /app

# Expose port 8000 on the container
EXPOSE 8000

# Run the Deno application
CMD ["deno", "run", "--allow-net", "--allow-env", "--allow-read", "main.ts"]
aperez-b@ARSO20232:~/prac20232-orig/deno-app$ 
```

(b) `cat Dockerfile`

aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ sudo docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED         STATUS         PORTS                                         NAMES
99ea5e32cddd   postgres:16.2       "docker-entrypoint.s…"   6 seconds ago   Up 5 seconds   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   dbhost
8ac9e90f7ece   prac20232-orig-web  "/tini -- docker-ent…"   6 seconds ago   Up 5 seconds   0.0.0.0:3000->8000/tcp, :::3000->8000/tcp   webhost
aperez-b@ARSO20232:~/prac20232-orig$
```

(c) `sudo docker ps`

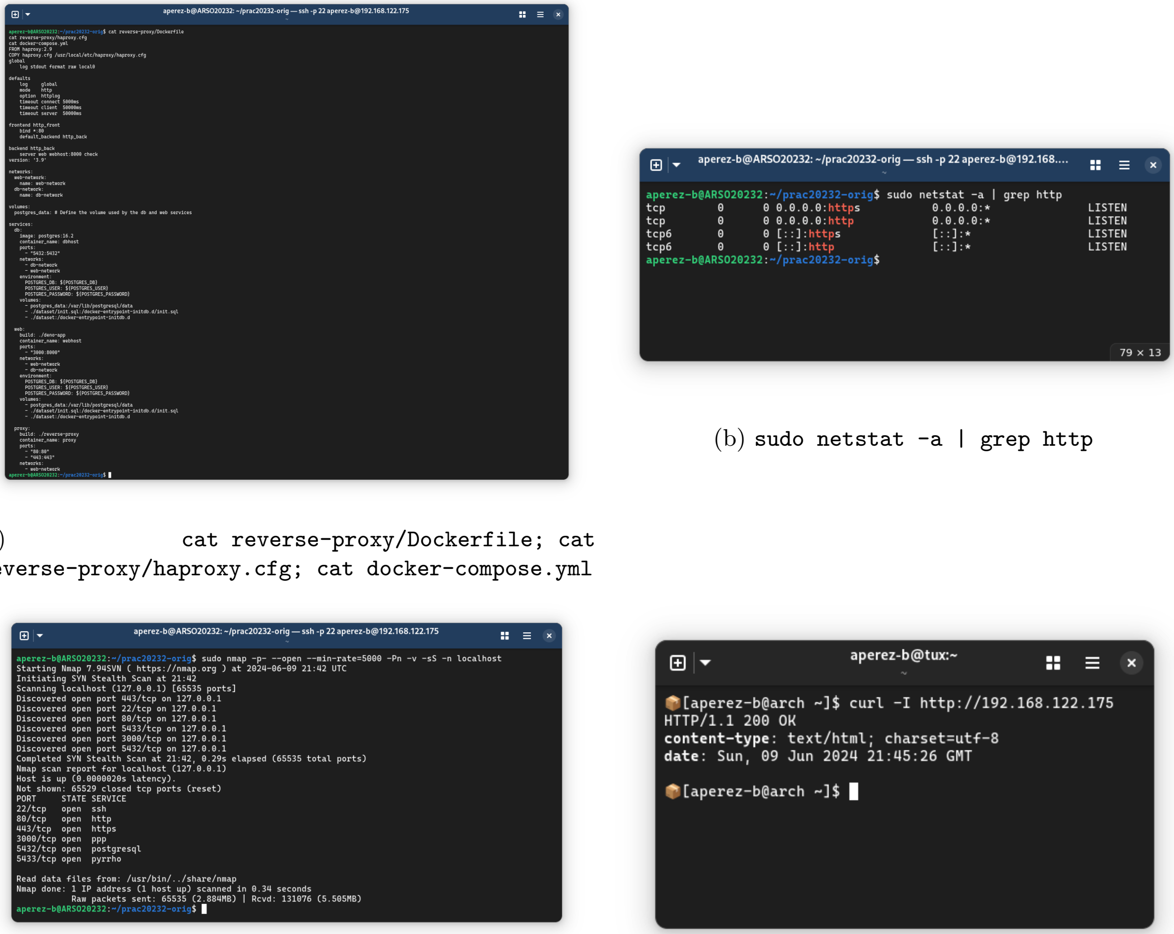
ARSO 2023-2

🥳 Congratulations !! the site work!! 🎉 👏 👏

👉 Now you should configure the connection to the database.

(d) Deno Web

Figure 8: Evidences for `postgres` + `Deno` server



aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ cat reverse-proxy/Dockerfile
cat reverse-proxy/haproxy.cfg
cat docker-compose.yml
FROM haproxy:2.9
COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg
global
    log stdout format raw local0

defaults
    log     global
    mode    http
    option  httplog
    timeout connect 5000ms
    timeout client  50000ms
    timeout server  50000ms

frontend http_front
    bind *:80
    default_backend http_back

backend http_back
    server web webhost:8000 check
version: '3.9'

networks:
  web-network:
    name: web-network
  db-network:
    name: db-network

volumes:
  postgres_data: # Define the volume used by the db and web services

services:
  db:
    image: postgres:16.2
    container_name: dbhost
    ports:
      - "5432:5432"
    networks:
      - db-network
      - web-network
    environment:
      POSTGRES_DB: ${POSTGRES_DB}
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./dataset/init.sql:/docker-entrypoint-initdb.d/init.sql
      - ./dataset:/docker-entrypoint-initdb.d

  web:
    build: ./deno-app
    container_name: webhost
    ports:
      - "3000:8000"
    networks:
      - web-network
      - db-network
    environment:
      POSTGRES_DB: ${POSTGRES_DB}
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./dataset/init.sql:/docker-entrypoint-initdb.d/init.sql
      - ./dataset:/docker-entrypoint-initdb.d

  proxy:
    build: ./reverse-proxy
    container_name: proxy
    ports:
      - "80:80"
      - "443:443"
    networks:
      - web-network
aperez-b@ARSO20232:~/prac20232-orig$ 
```

(a) `cat reverse-proxy/Dockerfile; cat reverse-proxy/haproxy.cfg; cat docker-compose.yml`

aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168....
79 × 13

```
aperez-b@ARSO20232:~/prac20232-orig$ sudo netstat -a | grep http
tcp        0      0 0.0.0.0:https           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:http            0.0.0.0:*               LISTEN
tcp6       0      0 [::]:https              [::]:*                  LISTEN
tcp6       0      0 [::]:http               [::]:*                  LISTEN
aperez-b@ARSO20232:~/prac20232-orig$
```

(b) `sudo netstat -a | grep http`

aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

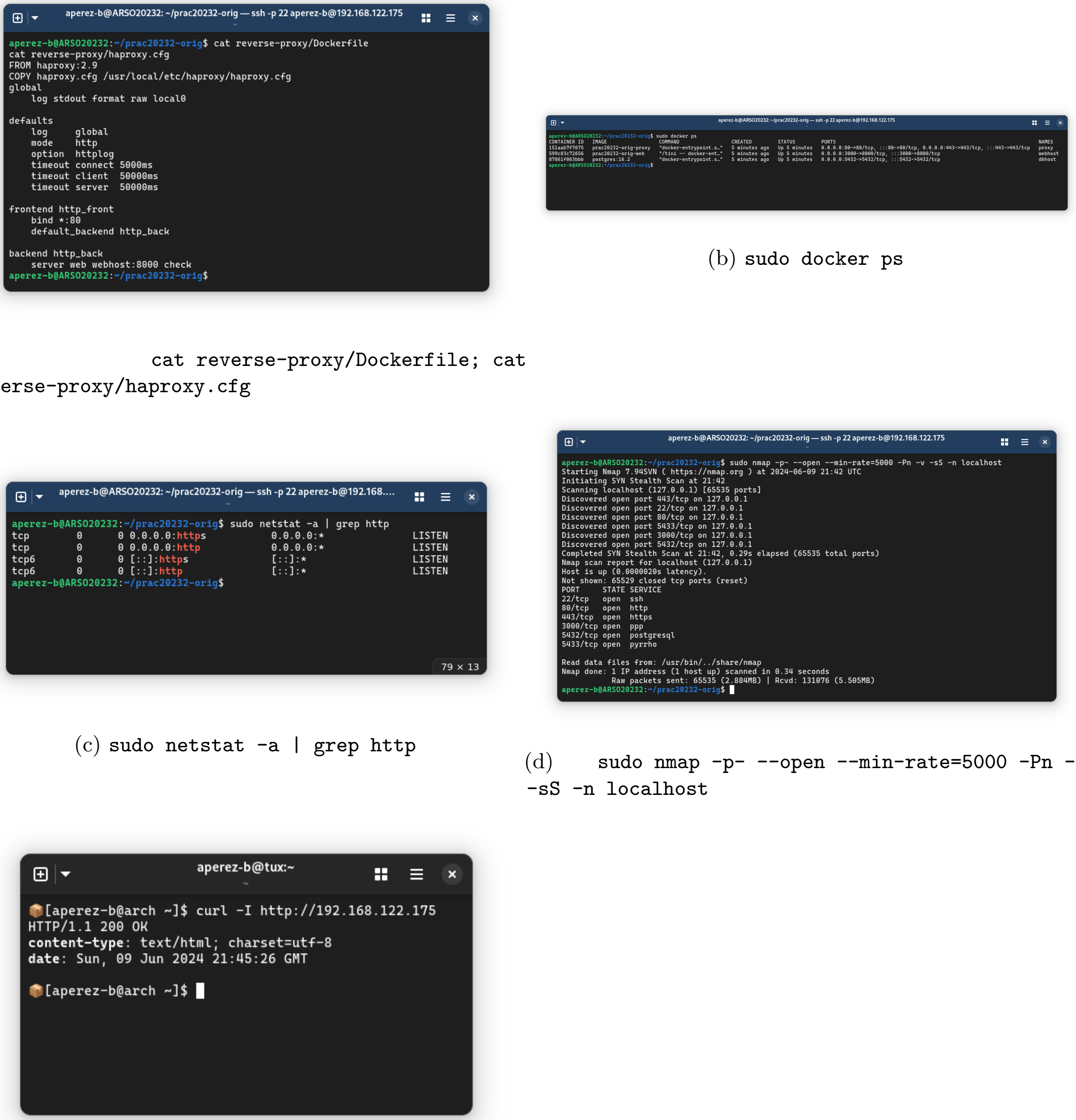
```
aperez-b@ARSO20232:~/prac20232-orig$ sudo nmap -p- --open --min-rate=5000 -Pn -v -sS -n localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-09 21:42 UTC
Initiating SYN Stealth Scan at 21:42
Scanning localhost (127.0.0.1) [65535 ports]
Discovered open port 443/tcp on 127.0.0.1
Discovered open port 22/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 5433/tcp on 127.0.0.1
Discovered open port 3000/tcp on 127.0.0.1
Discovered open port 5432/tcp on 127.0.0.1
Completed SYN Stealth Scan at 21:42, 0.29s elapsed (65535 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Not shown: 65529 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3000/tcp open  ppp
5432/tcp open  postgresql
5433/tcp open  pyrrho

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
           Raw packets sent: 65535 (2.884MB) | Rcvd: 131076 (5.505MB)
aperez-b@ARSO20232:~/prac20232-orig$ 
```

(c) `sudo nmap -p- --open --min-rate=5000 -Pn -v -sS -n localhost`

```
📦[aperez-b@arch ~]$ curl -I http://192.168.122.175
HTTP/1.1 200 OK
content-type: text/html; charset=utf-8
date: Sun, 09 Jun 2024 21:45:26 GMT

📦[aperez-b@arch ~]$ 
```

(d) `curl -I http://192.168.122.175`

Figure 9: Evidences for `reverse-proxy` HTTP


aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ cat reverse-proxy/Dockerfile
cat reverse-proxy/haproxy.cfg
FROM haproxy:2.9
COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg
global
    log stdout format raw local0

defaults
    log     global
    mode    http
    option  httplog
    timeout connect 5000ms
    timeout client  50000ms
    timeout server  50000ms

frontend http_front
    bind *:80
    default_backend http_back

backend http_back
    server web webhost:8000 check
aperez-b@ARSO20232:~/prac20232-orig$
```

aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ sudo docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED         STATUS         PORTS                                                                     NAMES
151aab7f7675   prac20232-orig-proxy   "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp   proxy
599c83c72656   prac20232-orig-web     "/tini -- docker-ent…"   5 minutes ago   Up 5 minutes   0.0.0.0:3000->8000/tcp, :::3000->8000/tcp                                 webhost
87861f063bbb   postgres:16.2          "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp                                 dbhost
aperez-b@ARSO20232:~/prac20232-orig$
```

(b) `sudo docker ps`

(a) `cat reverse-proxy/Dockerfile; cat reverse-proxy/haproxy.cfg`

aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168....

```
aperez-b@ARSO20232:~/prac20232-orig$ sudo netstat -a | grep http
tcp        0      0 0.0.0.0:https           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:http            0.0.0.0:*               LISTEN
tcp6       0      0 [::]:https              [::]:*                  LISTEN
tcp6       0      0 [::]:http               [::]:*                  LISTEN
aperez-b@ARSO20232:~/prac20232-orig$
```

aperez-b@ARSO20232: ~/prac20232-orig — ssh -p 22 aperez-b@192.168.122.175

```
aperez-b@ARSO20232:~/prac20232-orig$ sudo nmap -p- --open --min-rate=5000 -Pn -v -sS -n localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-09 21:42 UTC
Initiating SYN Stealth Scan at 21:42
Scanning localhost (127.0.0.1) [65535 ports]
Discovered open port 443/tcp on 127.0.0.1
Discovered open port 22/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 5433/tcp on 127.0.0.1
Discovered open port 3000/tcp on 127.0.0.1
Discovered open port 5432/tcp on 127.0.0.1
Completed SYN Stealth Scan at 21:42, 0.29s elapsed (65535 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Not shown: 65529 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3000/tcp open  ppp
5432/tcp open  postgresql
5433/tcp open  pyrrho

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
           Raw packets sent: 65535 (2.884MB) | Rcvd: 131076 (5.505MB)
aperez-b@ARSO20232:~/prac20232-orig$
```

(c) `sudo netstat -a | grep http`

(d) `sudo nmap -p- --open --min-rate=5000 -Pn -v -sS -n localhost`

```
[aperez-b@arch ~]$ curl -I http://192.168.122.175
HTTP/1.1 200 OK
content-type: text/html; charset=utf-8
date: Sun, 09 Jun 2024 21:45:26 GMT

[aperez-b@arch ~]$
```

(e) `curl -I http://192.168.122.175`

Figure 10: Evidences for `reverse-proxy` HTTPS