Final Exam Documentation

Linux Administration

iradukunda divine -23663

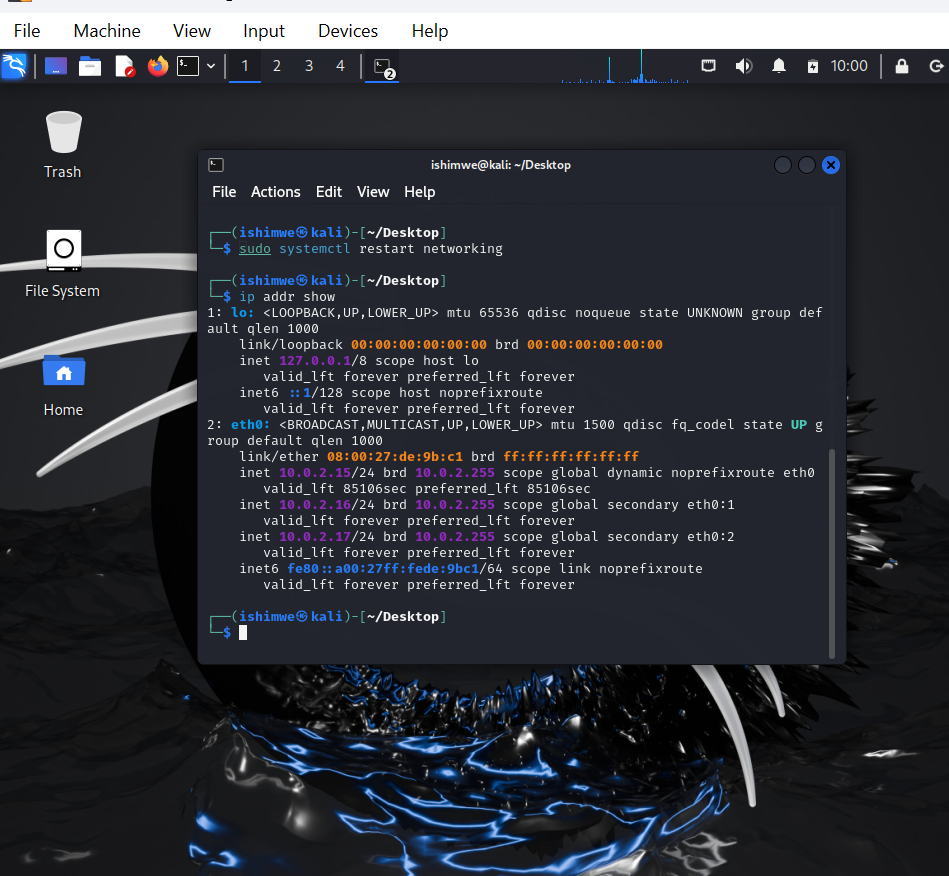
**Introduction**

This document provides a detailed technical overview of the network and web server configuration, as well as the Docker deployment and Traefik load balancing implemented in the exam project. The primary objective was to establish a secure, scalable, and high-performance web infrastructure capable of hosting both static and dynamic web applications.

**Part 1: Network and Web Server Configuration**

**1.1 Virtual Interface Configuration**

To isolate network traffic and enhance security, two virtual interfaces were created on the Linux host machine. This separation ensures that the web services running on each interface are protected from potential attacks.

* **Interface Creation:**
* 

**1.2 Hosting the AUCA Education Web Page**

The first virtual interface was configured to host the static AUCA Education web page using the Apache2 web server. The following steps were involved:

* **Domain Name Configuration:** The /etc/hosts file was modified to map the domain name 23663.auca.ac.rw to the IP address of the first virtual interface. This allows the web server to recognize and process requests for this domain.
* **Apache Configuration:** A virtual host configuration file was created within the Apache configuration directory. This file specifies the domain name, document root (the directory containing the web page files), and other relevant directives.

Apache

<VirtualHost \*:443>

Serve<VirtualHost \*:443>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

<VirtualHost \*:443>

Serve<VirtualHost \*:443>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

SSLEngine on

SSLCertificateFile /etc/ssl/certs/23663.crt

SSLCertificateKeyFile /etc/ssl/private/23663.key

<FilesMatch "\.(?:cgi|shtml|phtml|php)$">

SSLOptions +StdEnvVars

</FilesMatch>

<Directory /usr/lib/cgi-bin>

SSLOptions +StdEnvVars

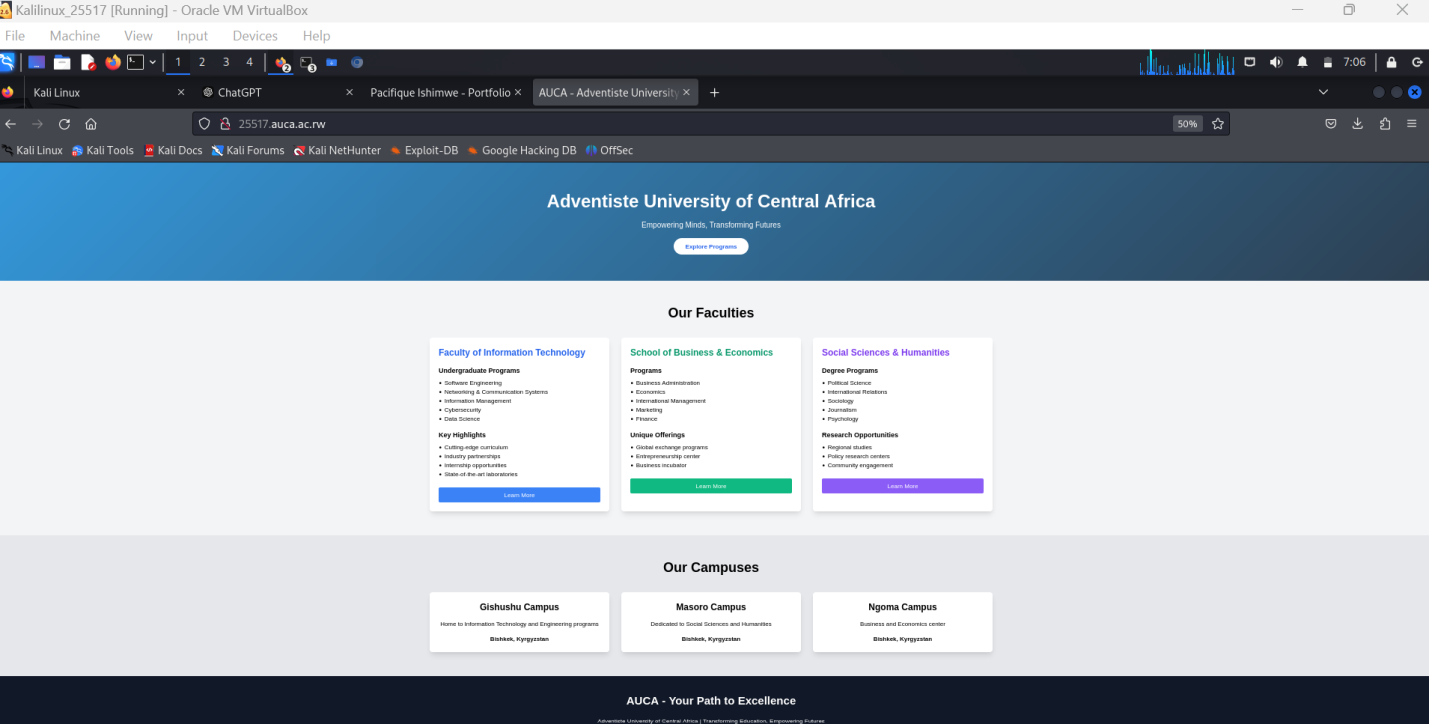
</Directory>

</VirtualHost>

To enable HTTPS, SSL/TLS encryption was configured using a self-signed certificate. This ensures secure communication between the client and the server.

[Insert screenshot of Apache configuration]

The AUCA Education web page



**1.3 Hosting the Dynamic Portfolio Page**

The second virtual interface was utilized to deploy the dynamic portfolio system. This system leverages Nginx as the web server and a database to store user information.

* **Database Configuration:** A database (MySQL) was created to store user profiles, skills, and other relevant data. SQL commands were used to populate the database with initial data.
* **Nginx Configuration:** Nginx was configured to act as a reverse proxy, forwarding requests to the application server. The configuration file specifies the domain name portfolio.auca.ac.rw and routes incoming requests to the appropriate backend service.

Nginx

server {

listen 81;

server\_name portfolio.auca.ac.rw;

# Root and index for your portfolio site

root /var/www/portfolio.auca.ac.rw/;

index index.php index.html index.htm;

# PHP-FPM configuration for dynamic content

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass unix:/run/php/php8.2-fpm.sock;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

# Static content (e.g., images, CSS, JS)

location / {

try\_files $uri $uri/ =404;

}

# Logs

error\_log /var/log/nginx/portfolio.auca.ac.rw\_error.log;

access\_log /var/log/nginx/portfolio.auca.ac.rw\_access.log;

# Security headers

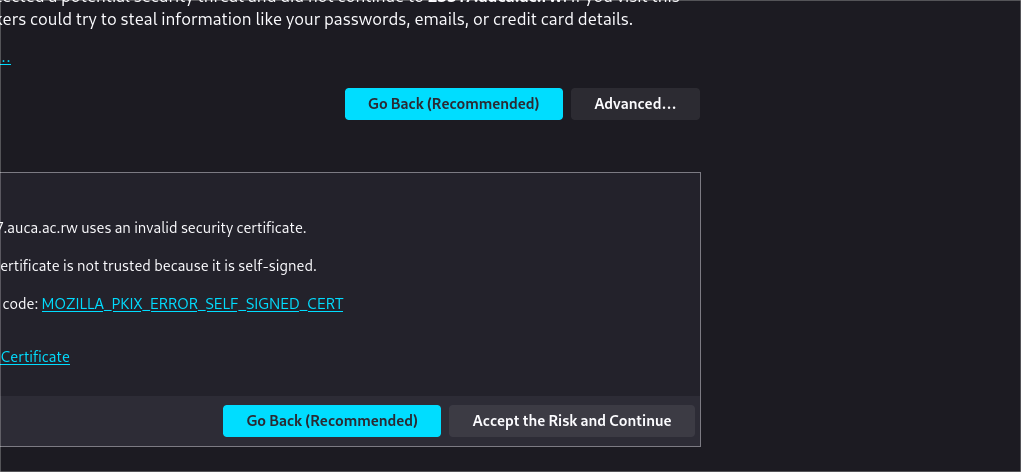
add\_header X-Content-Type-Options nosniff;

add\_header X-XSS-Protection "1; mode=block";

add\_header X-Frame-Options SAMEORIGIN;

* **Application Development:** A web application was developed using HTML and CSS

Portfolio running to ngnix = portfolio.auca.ac.rw:81



Display of the information

**1.4 Security Configuration**

To enhance security, several measures were implemented:

* **HTTPS:** Self-signed certificates were generated for both web servers. These certificates were used to enable HTTPS, ensuring secure communication between clients and servers.
* **Web Application Firewall (WAF):** ModSecurity was installed and configured to protect against web attacks. Custom rules were added to further strengthen security.
* **Firewall Configuration:** UFW (Uncomplicated Firewall) was used to restrict SSH access to specific IP addresses and enable two-factor authentication. Additionally, UFW was configured to allow only HTTPS traffic and block other incoming connections.

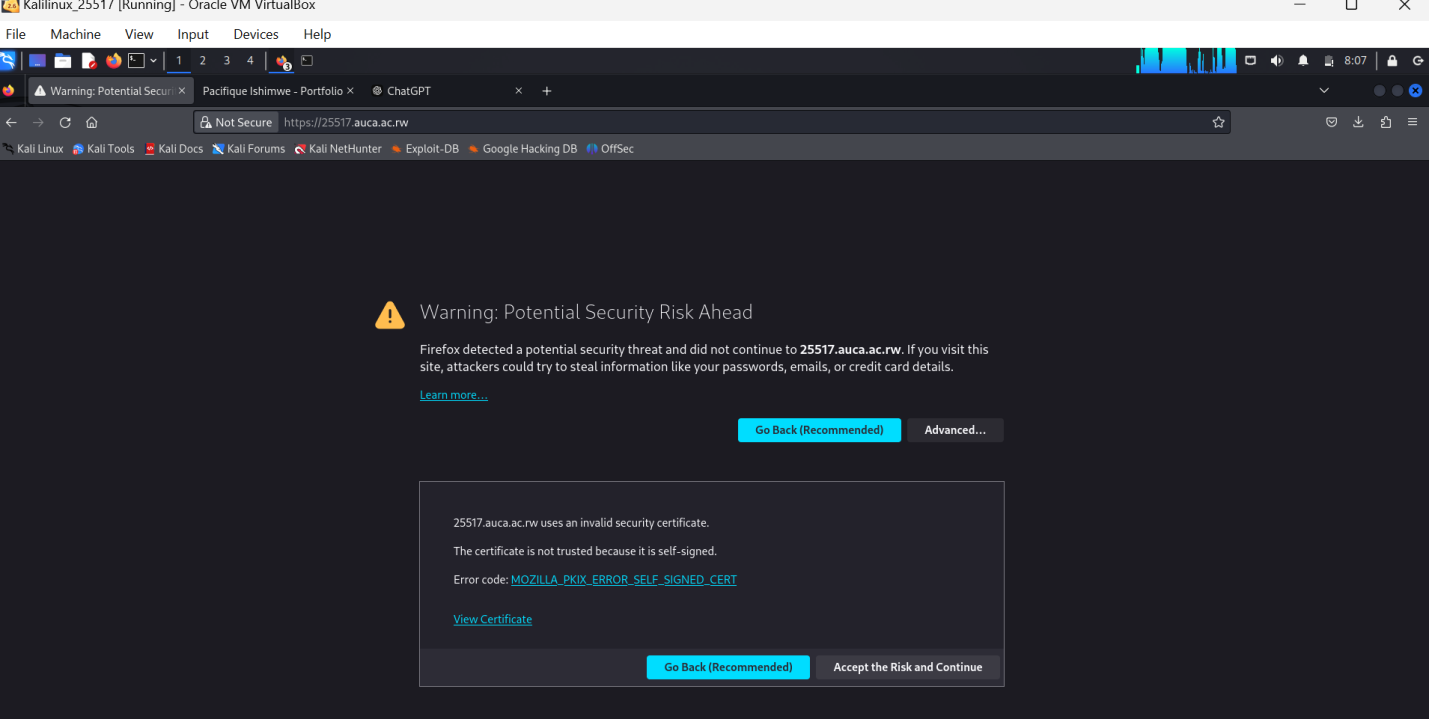
[Insert screenshots of ModSecurity and UFW configurations]

ModSecurity

ModSecurity= for the 23663.auca.ac.rw

ModSecurity= for portfolio.auca.ac.rw:8334

Certificate



**Part 2: Docker Deployment with Traefik Load Balancing**

**2.1 Docker Setup**

To streamline deployment and management, the portfolio system was containerized using Docker. Separate Docker images were created for the frontend and database components.

sudo cat docker-compose.yml

version: '3.8'

services:

php:

build:

context: .

dockerfile: Dockerfile

container\_name: portfolio-php-23663

ports:

- "8080:80" # Expose port 8080 for HTTP access

volumes:

- ./php:/var/www/html # Mount the PHP code as a volume

networks:

- portfolio\_network

mysql:

image: mysql:5.7

container\_name: portfolio-mysql-23663

environment:

MYSQL\_ROOT\_PASSWORD: rootpassword # You can change this to a secure password

MYSQL\_DATABASE: portfolio\_db

ports:

- "3306:3306" # Expose MySQL on port 3306

volumes:

- ./mysql/init.sql:/docker-entrypoint-initdb.d/init.sql # Initialize the database

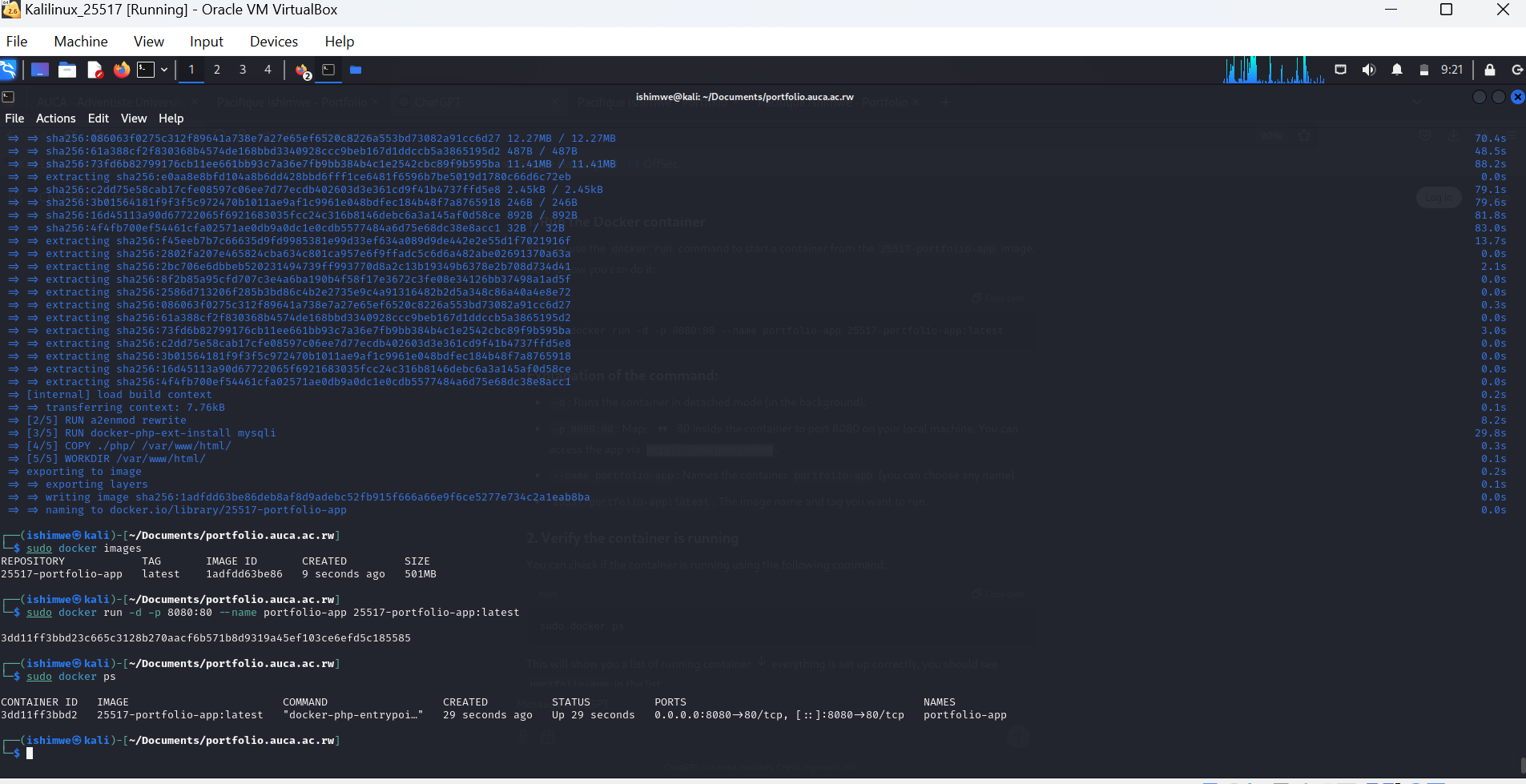
networks:

- portfolio\_network

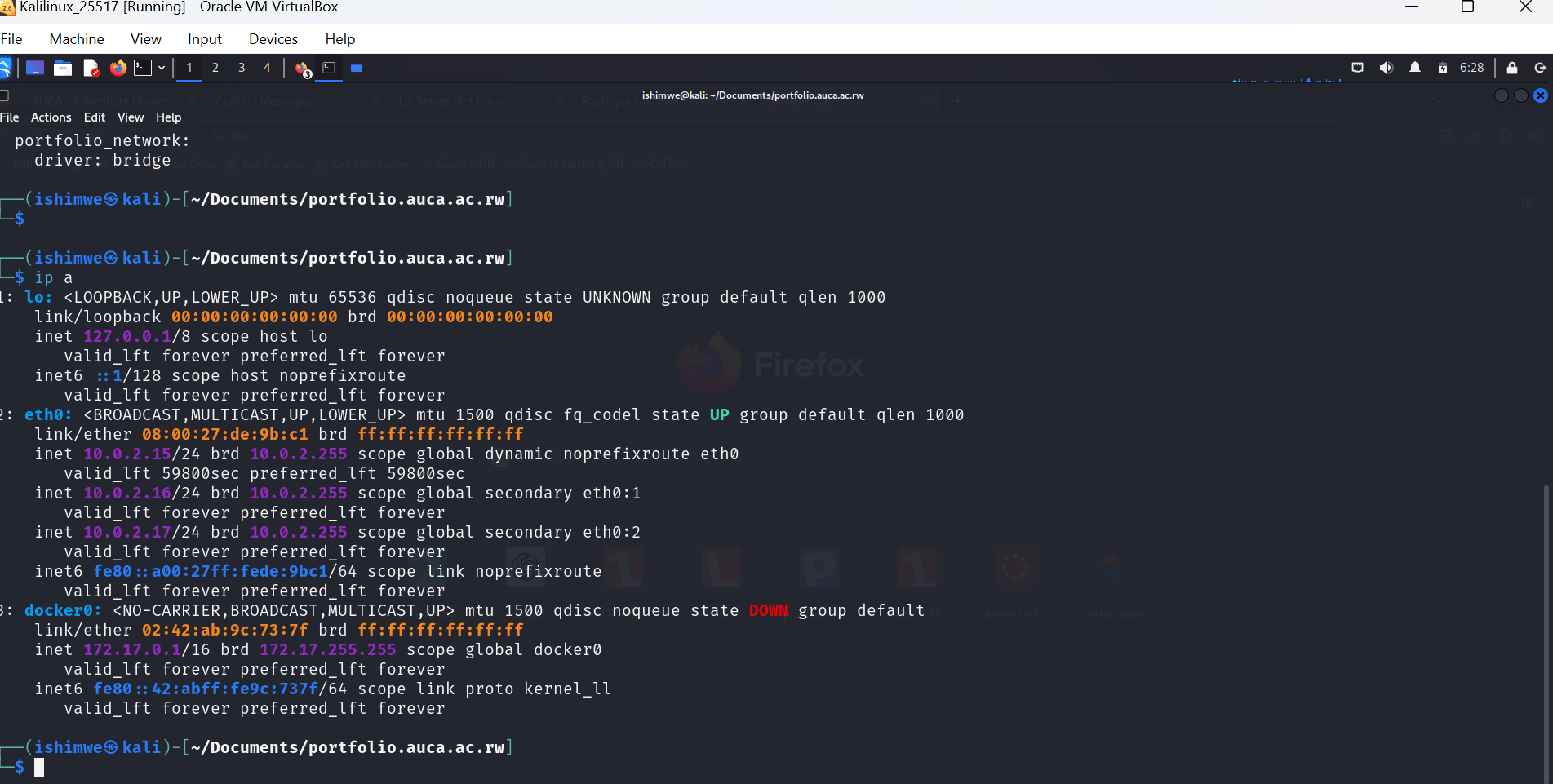
networks:

portfolio\_network:

driver: bridge

* **DockerImage:**
* 
* **Database**

The interface with Docker



**2.2 Traefik Load Balancer**

Traefik was employed as a reverse proxy and load balancer to distribute traffic across multiple Docker nodes. This ensures high availability and scalability.

* **Traefik Configuration:** Traefik was configured to automatically detect Docker services and create routes. Let's Encrypt was enabled to automatically obtain and renew SSL certificates.

YAML

http:

routers:

portfolio-router:

entryPoints:

- websecure

rule: Host(`portfolio.auca.ac.rw`)

service: portfolio-service

services:

portfolio-service:

loadBalancer:

servers:

- url: http://portfolio-app:8080

**Conclusion**

By following the steps outlined in this document, a robust, secure, and scalable web infrastructure was established. The use of virtual interfaces, web servers, firewalls, and load balancers ensured optimal performance and security. Dockerization facilitated easy deployment and management of the application.

**iradukunda divine** **23663**