

Project Report on

**Multi-Server Security Architecture using DMZSubmitted by**

**Komal Vede (230944223049)**

Under the guidance of

**Mr. Gajanan Taur**

**In partial fulfillment of the award of Industrial Training and Internship in DevOps and Cyber Security**



**Sunbeam Institute of Information Technology,**

**Pune (Maharashtra)**

**PG-DITISS -2023**

# DECLARATION

I hereby declare that this written submission represents my own ideas in my own words. Wherever I have included ideas or words from others, I have appropriately cited and referenced the original sources. I further declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated or falsified any idea, data, fact or source in this submission.

I understand that any violation of the above may result in disciplinary action by the Institute and may also lead to legal or penal action from the original sources if proper permission has not been obtained or proper citation has not been made.

**Komal Vede (230944223049)**

Place: Pune

Date: 7th October 2025

# CERTIFICATE

This is to certify that the project report entitled **“Multi-Server Security Architecture using DMZ”**, submitted by **Komal Vede (230944223049)** is the bonafide work completed under our supervision and guidance in partial fulfillment of the requirements for the award of **Industrial Training and Internship in DevOps and Cyber Security** at Sunbeam Institute of Information Technology, Pune.

Place: Pune

Date: 7th October 2025

**Mr. Gajanan Taur Mr. Vishal Salunkhe**

Guide Course Coordinator

**Mr. Nitin Kudale**

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

# APPROVAL CERTIFICATE

This is to certify that the Project-I report entitled **“Multi-Server Security Architecture using DMZ”** by **Komal Vede (230944223049)** is approved for **Industrial Training and Internship in DevOps and Cyber Security** at Sunbeam Institute of Information Technology, Pune.

Place: Pune

Date:

Examiner:

**(Signature)**

**(Name)**

# CONTENTS

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO** |
| **Declaration** |  |
| **Certificate** |  |
| **Approval Certificate** |  |
| **Abstract** | i |
| **1.INTRODUCION** | 1 |
| 1.1 Applications | 1 |
| 1.2 Organization and Project Plan | 3 |
| **2. LITERATURE SURVEY** | 4 |
| Paper 1 | 4 |
| Paper 2 | 4 |
| Paper 3 | 5 |
| **3. SYSTEM DEVELOPMENT AND DESIGN** | 6 |
| 3.1 Proposed System | 6 |
| 3.2 Flow Chart | 7 |
| 3.3 Technology used | 8 |
| 3.3.1 AWS EC2 | 8 |
| 3.3.2 Git | 9 |
| 3.3.3 Docker | 10 |
| 3.3.4 Jenkins | 11 |
| 3.3.5 Snort | 12 |
| 3.3.6 Nagios | 13 |
| **4. PROJECT OUTPUT** | 14 |
| **5. CONCLUSION** | 18 |
| 5.1 Conclusion | 18 |
| 5.2 Future Scope | 18 |
| **REFERENCES** | 19 |

# ABSTRACT

In the modern digital era, organizations face an increasing number of cyber threats, making it essential to design and implement secure infrastructures that can withstand attacks, ensure confidentiality, integrity, and availability of services, and maintain operational continuity. This project demonstrates the deployment and security hardening of a multi-tier infrastructure in a controlled lab environment using the custom domain **sunbeam.local**.

The implementation begins with setting up a **database server** and a **web server** hosting a Flask-based application, followed by enabling **secure HTTPS access** with self-signed SSL certificates. A **mail server** is configured with Postfix and Dovecot to provide domain-based email services, and **Roundcube webmail** is deployed for user-friendly access.

To ensure operational reliability, **Nagios XI** is integrated for monitoring critical services such as HTTP, HTTPS, SMTP, IMAP, POP3, CPU, and memory usage. Alerts are routed through the local mail server to notify administrators of service outages. For proactive security, **Snort IDS/IPS** is deployed on the web server with custom rules to detect SYN floods, Ping floods, and Ping of Death attacks.

System hardening measures include the installation of **Fail2ban** to prevent brute-force SSH attacks, configuring **iptables** on the database server to restrict access only from the web server, and setting up a **pfSense firewall** for layered protection.

Overall, this project integrates **DevOps practices with security mechanisms**, demonstrating a complete lifecycle of deployment, monitoring, and defense. The outcome is a resilient, secure, and monitored environment that reflects industry-grade best practices in system administration, DevOps, and cybersecurity.

# INTRODUCTION

Organizations today depend on web-based applications, email communication, and databases for their day-to-day operations. However, deploying these services without security controls makes them highly vulnerable to cyber-attacks. To overcome this, a secure infrastructure must be designed where services such as **web server, database server, and mail server** are not only deployed but also **secured, monitored, and protected against intrusions**.

This project demonstrates the **end-to-end deployment of a secure enterprise-like environment** under the domain **sunbeam.local**, covering service setup, monitoring, intrusion detection, and hardening.

This project is implemented in multiple phases. Each phase builds upon the previous one to create a fully secured infrastructure.

### ****Database and Web Server Setup****

* Two machines are configured:
  1. Database Server
  2. Web Server
* The application from GitHub repository KartvyaPath-Project is cloned and deployed.
* Necessary changes are applied to ensure proper communication between web server and database server.

### ****Secure Web Access (SSL/TLS Certificates)****

* The Flask-based application is made accessible via https://todo.sunbeam.local.
* SSL/TLS certificates are generated and configured to ensure encrypted communication.
* Application configurations are updated to redirect insecure HTTP traffic to HTTPS.

### ****Mail Server Setup****

* A third machine is configured as a **Mail Server** with the domain **sunbeam.local**.
* SMTP, IMAP, and POP3 services are enabled and tested.
* Two user accounts are created to verify email sending and receiving functionalities.

### ****Webmail Configuration****

* A webmail service is installed and configured on the mail server.
* Webmail is accessed through http://mail.sunbeam.local providing browser-based email access.

### ****Monitoring with Nagios XI****

* Nagios XI is deployed and integrated with both the web server and mail server.
* Monitoring is configured for:
  + Web Server: HTTP, HTTPS, Apache2, CPU, Memory
  + Mail Server: SMTP, IMAP, POP3, CPU, Memory
* Nagios XI is further configured to send alerts to the internal mail server (sunbeam.local).

### ****Intrusion Detection with Snort****

* Snort IDS/IPS is installed on the web server.
* Custom rules are written to detect:
  1. **SYN Flood Attack**
  2. **Ping Flood Attack**
  3. **Ping of Death**
* Alerts are generated when these attacks are simulated in the environment.

### ****System Hardening and Firewall****

* **Fail2ban** is installed on all servers to protect SSH access from brute-force attempts.
* **iptables** rules are configured on the database server to allow access only from the web server.
* **pfSense Firewall** is deployed as a network-level firewall to provide perimeter protection.

The main objectives of this project are:

* To deploy a secure **multi-server environment** consisting of web, database, and mail servers.
* To ensure **secure communication** using SSL/TLS.
* To provide **email communication** with webmail support.
* To implement **real-time monitoring** using Nagios XI.
* To deploy **Snort IDS/IPS** for detecting common DoS attacks.
* To perform **system hardening** with Fail2ban, iptables, and pfSense firewall.

## ****Scope of the Project:****

This project simulates an **enterprise-level secure infrastructure** within a lab environment. It ensures that essential IT services (web, database, and email) are deployed in a secure manner, monitored for performance, and protected against cyber threats. The project also provides a hands-on understanding of **monitoring tools, IDS/IPS, and security hardening techniques**.

## 1.2 Project Plan

**Table: Activities Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **ACTIVITY** | **WEEK** | | | |
| **1** | **2** | **3** | **4** |
| 1 | Project Selection and Requirement Understanding |  |  |  |  |
| 2 | Database Server and Web Server Setup (KartvyaPath App Deployment) |  |  |  |  |
| 3 | Configure HTTPS with SSL/TLS Certificates (todo.sunbeam.local) |  |  |  |  |
| 4 | Mail Server Setup with Postfix & Dovecot (sunbeam.local) |  |  |  |  |
| 5 | Webmail Configuration with Roundcube (mail.sunbeam.local) |  |  |  |  |
| 6 | Monitoring Setup with Nagios XI (Web & Mail Servers) |  |  |  |  |
| 7 | Intrusion Detection using Snort IDS/IPS (Custom Rules: SYN flood, Ping flood, Ping of Death) |  |  |  |  |
| 8 | System Hardening – Fail2ban, iptables rules on DB Server |  |  |  |  |
| 9 | pfSense Firewall Deployment and Policy Configuration |  |  |  |  |
| 10 | Testing, Troubleshooting, and Report Preparation |  |  |  |  |
| 11 | Final Presentation & Submission |  |  |  |  |

# 2. LITERATURE SURVEY

### Paper 1 – ****Snort IDS and its Applications****

**Author(s):** Martin Roesch, et al.  
**Published Year:** 1999  
**Summary:**  
Snort is an open-source Network Intrusion Detection System (NIDS) used for real-time traffic analysis and packet logging. It supports signature-based detection, protocol analysis, and anomaly detection capabilities.

**Applications in Project:**

* Monitoring network traffic within the Sunbeam.local environment.
* Detecting attacks like SYN Flood, Ping Flood, and Ping of Death.
* Generating alerts using custom rules.

### Paper 2 – ****Nagios XI Monitoring in IT Infrastructure****

**Author(s):** Ethan Galstad, et al.  
**Published Year:** 2010  
**Summary:**  
Nagios XI is an enterprise-grade IT infrastructure monitoring tool that monitors servers, switches, applications, services, and network devices. It provides alerts, notifications, SLA reporting, and dashboards.

**Applications in Project:**

* Monitoring uptime and health of Web Server, Database Server, and Mail Server.
* Setting up centralized monitoring in the Sunbeam.local environment.
* Tracking system status using alerts and dashboards.

### Paper 3 – ****Fail2ban & System Hardening Approaches****

**Author(s):** Cyril Jaquier, et al.  
**Published Year:** 2012  
**Summary:**  
Fail2ban is an intrusion prevention framework that detects failed login attempts and blocks suspicious IPs. System hardening involves configuring firewalls, disabling unnecessary services, and setting up secure authentication.

**Applications in Project:**

* Blocking brute-force attacks on the Database Server.
* Configuring iptables rules to prevent unauthorized access.
* Securing the Sunbeam.local environment.

# SYSTEM DEVELOPMENT AND DESIGN

### 3.1 Proposed System

The proposed system for the **Sunbeam.local Security Project** is designed to provide a **secure, monitored, and resilient IT infrastructure** within a lab environment. It integrates multiple components to ensure data security, network monitoring, and system hardening.

**Key Components:**

1. **Database Server (MySQL)** – Stores application and user data securely.
2. **Web Server (Flask + Apache2)** – Hosts the web application over HTTPS.
3. **Mail Server (Postfix + Dovecot + Roundcube)** – Provides internal email services and secure webmail access.
4. **SSL/TLS Certificates** – Enables encrypted communication for web and mail servers.
5. **Nagios XI** – Centralized monitoring system for servers, services, and network devices.
6. **Snort IDS/IPS** – Real-time intrusion detection and prevention system for network traffic.
7. **Fail2ban & iptables** – Protect servers against brute-force attacks and unauthorized access.
8. **pfSense Firewall** – Manages network traffic policies and provides perimeter security.

This architecture ensures that all servers are secure, monitored, and can detect and respond to potential threats.

### 3.2 Flowchart / Architecture Diagram

**Description of Architecture Diagram:**

1. **User Layer:** End users access the web application via HTTPS and internal email via Roundcube.
2. **Web Layer:** Flask application hosted on Apache2 handles requests securely.
3. **Database Layer:** MySQL stores application data, protected by Fail2ban and iptables.
4. **Mail Layer:** Postfix/Dovecot handles mail, with secure SSL/TLS connections.
5. **Monitoring Layer:** Nagios XI continuously monitors the web, mail, and DB servers, generating alerts when issues occur.
6. **Security Layer:** Snort IDS/IPS monitors network traffic for malicious activities.
7. **Firewall Layer:** pfSense manages network access, applies security policies, and integrates with IDS/IPS and Fail2ban.

### 3.3 Technology Used

| **Component** | **Technology / Tool** | **Purpose** |
| --- | --- | --- |
| Web Server | Flask + Apache2 | Hosts the web application securely over HTTPS |
| Database Server | MySQL | Stores application and user data |
| Mail Server | Postfix + Dovecot + Roundcube | Internal mail and webmail service with secure access |
| SSL/TLS Certificates | OpenSSL / Let’s Encrypt | Enables HTTPS and encrypted mail connections |
| Monitoring | Nagios XI | Centralized monitoring, alerting, and dashboards |
| Intrusion Detection | Snort IDS/IPS | Detects network attacks (SYN Flood, Ping Flood, etc.) |
| System Hardening | Fail2ban + iptables | Blocks unauthorized access and brute-force attempts |
| Firewall | pfSense | Manages network traffic policies and perimeter security |

## 3.2 Network Architecture OverviewC:\Users\Admin\OneDrive\Desktop\KartvyaPath-DITISS.jpg

**Figure: Network Architecture**

**3.3. Technology used**

### 3.3.4 Flask

Flask is a lightweight Python web framework used for building web applications. It provides routing, templating, and API support while being flexible and easy to extend.

**Key features of Flask:**

* **Lightweight and Modular:** Minimal core, allowing developers to add only the components they need.
* **Routing:** Provides simple URL routing to map requests to Python functions.
* **Template Engine (Jinja2):** Supports dynamic HTML rendering.
* **Extensibility:** Easy integration with databases, authentication modules, and other Python libraries.
* **Development Server:** Built-in server for testing and development.

**Purpose:** Hosts the KartvyaPath web application securely and efficiently, serving dynamic content over HTTPS.

### 3.3.5 Apache2

Apache2 is a widely used web server software that handles HTTP/HTTPS requests and serves web applications.

**Key features of Apache2:**

* **Virtual Hosts:** Supports multiple websites on the same server with separate configurations.
* **SSL/TLS Support:** Enables HTTPS for secure communication.
* **mod\_wsgi:** Allows integration with Python applications like Flask.
* **Logging:** Provides detailed access and error logs for monitoring.
* **Security Modules:** Supports modules like mod\_security to enhance server security.

**Purpose:** Serves the Flask application, handles HTTPS traffic, and ensures secure web access.

### 3.3.6 MySQL

MySQL is a relational database management system (RDBMS) used to store structured data for applications.

**Key features of MySQL:**

* **Relational Data Model:** Organizes data into tables with relationships between them.
* **SQL Queries:** Supports complex queries to retrieve, insert, update, and delete data.
* **User Management:** Provides authentication, access control, and permissions.
* **Replication & Backup:** Supports replication and backup strategies for data availability and security.
* **Performance Optimization:** Indexing, caching, and query optimization improve performance.

**Purpose:** Centralized storage for application data, with secure access controls to protect sensitive information.

### 3.3.7 Postfix & Dovecot

**Postfix** is an open-source SMTP server for sending and receiving emails.  
**Dovecot** is an IMAP/POP3 server for accessing emails securely.

**Key features:**

* **Email Handling:** Supports sending, receiving, and storing emails.
* **SSL/TLS Encryption:** Secures email transmission.
* **User Authentication:** Supports secure login for multiple users.
* **Integration:** Works with Roundcube for web-based email access.

**Purpose:** Provides a secure internal email system for Sunbeam.local, including webmail access.

### 3.3.8 Roundcube

Roundcube is a web-based email client that allows users to access their mailbox via a browser.

**Key features:**

* **Webmail Interface:** User-friendly browser-based email access.
* **IMAP Support:** Synchronizes emails with the mail server.
* **Plugins & Customization:** Allows customization for UI and functionality.
* **Security:** Supports SSL/TLS for encrypted connections.

**Purpose:** Enables convenient and secure webmail access for internal users.

### 3.3.9 Nagios XI

Nagios XI is a comprehensive IT infrastructure monitoring tool.

**Key features:**

* **Server & Service Monitoring:** Tracks uptime, performance, and availability.
* **Alerting System:** Sends notifications via email/SMS on failures or warnings.
* **Dashboards & Reports:** Provides visualization of system health and SLA metrics.
* **Extensibility:** Supports plugins for custom monitoring needs.

**Purpose:** Ensures continuous monitoring of web, database, and mail servers, detecting issues proactively.

### 3.3.10 Snort IDS/IPS

Snort is an open-source intrusion detection and prevention system.

**Key features:**

* **Signature-Based Detection:** Detects known attacks using predefined rules.
* **Protocol Analysis:** Inspects traffic for anomalies.
* **Custom Rules:** Allows defining rules for specific threats like SYN Flood or Ping of Death.
* **Alerts & Logging:** Generates alerts and logs for suspicious activity.

**Purpose:** Protects the network by detecting and preventing malicious traffic in real-time.

### 3.3.11 Fail2ban & iptables

**Fail2ban:** Monitors logs for failed login attempts and blocks suspicious IPs automatically.  
**iptables:** Linux firewall utility to filter and control network traffic.

**Key features:**

* **Automated Threat Mitigation:** Blocks brute-force attacks.
* **Custom Rules:** Allows specific rules for ports, IPs, and protocols.
* **Integration:** Works with Snort and pfSense for layered security.

**Purpose:** Strengthens server security by preventing unauthorized access and attacks.

### 3.3.12 pfSense Firewall

pfSense is an open-source firewall and router platform.

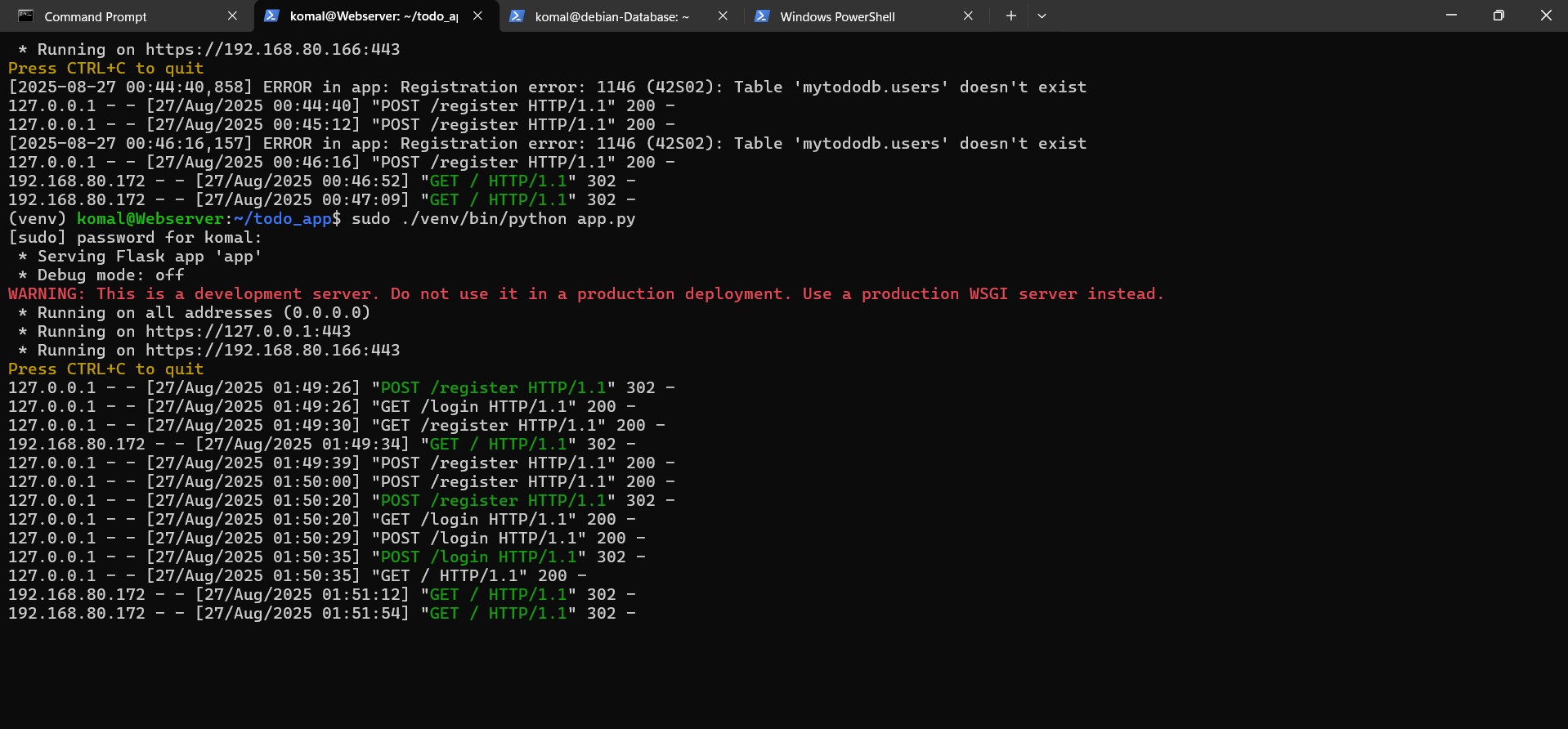
**Key features:**

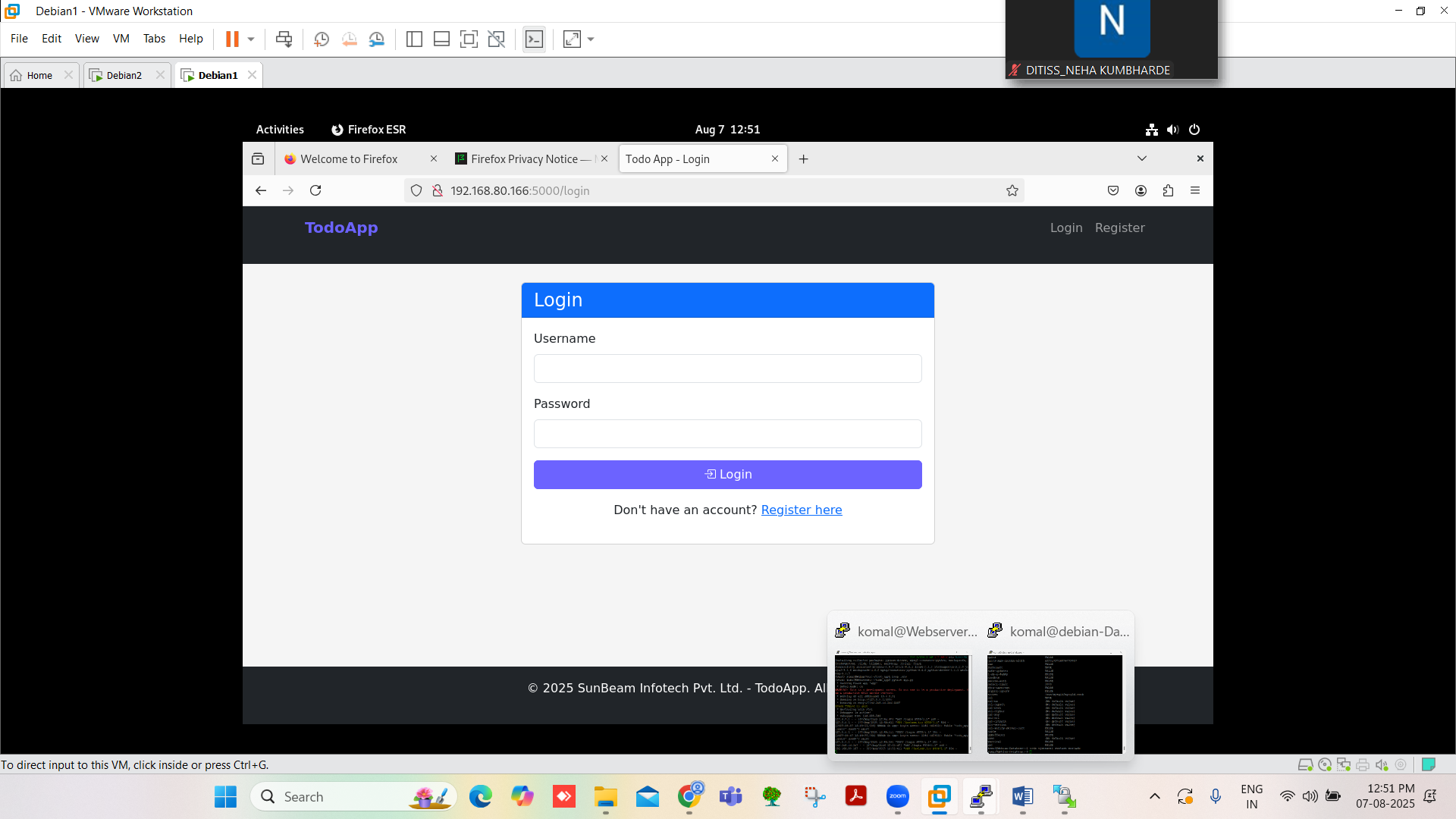
* **Traffic Filtering:** Controls network traffic based on IP, port, and protocol.
* **NAT & VPN Support:** Provides network address translation and secure VPN connections.
* **High Configurability:** VLANs, firewall rules, and policy management.
* **Integration:** Works with IDS/IPS and Fail2ban for enhanced protection.

**Purpose:** Provides perimeter security for the Sunbeam.local environment and manages network access policies.

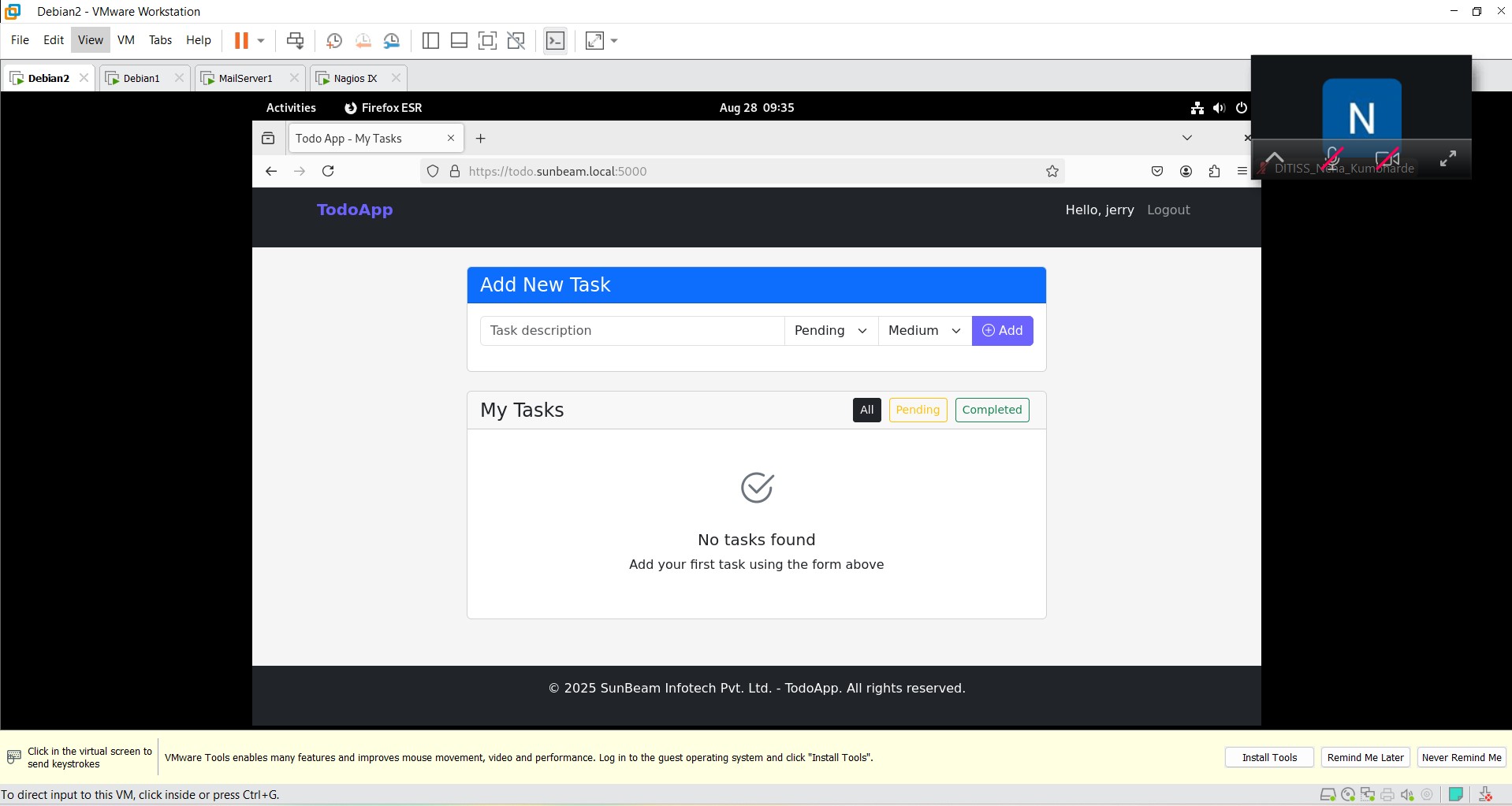
# 4. Project Output

## Todo App:

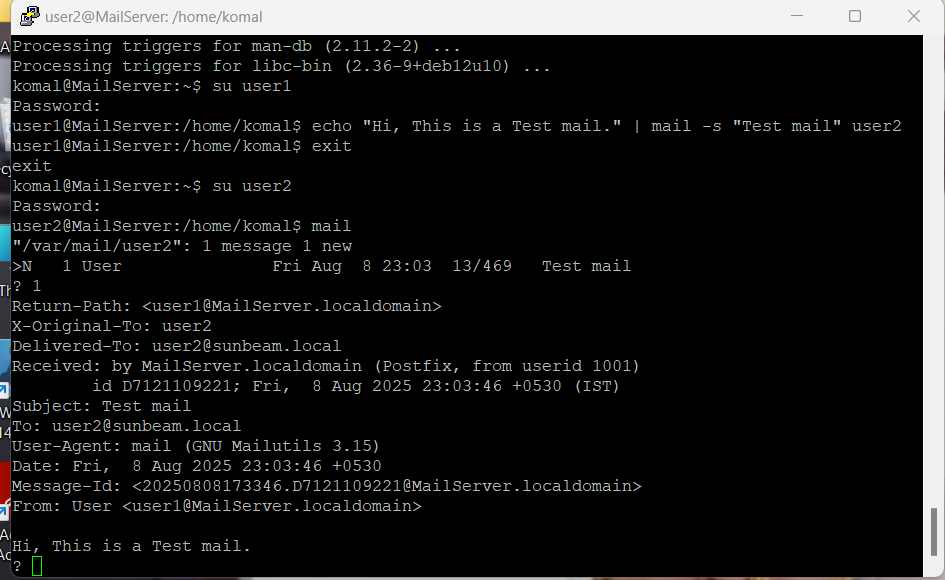


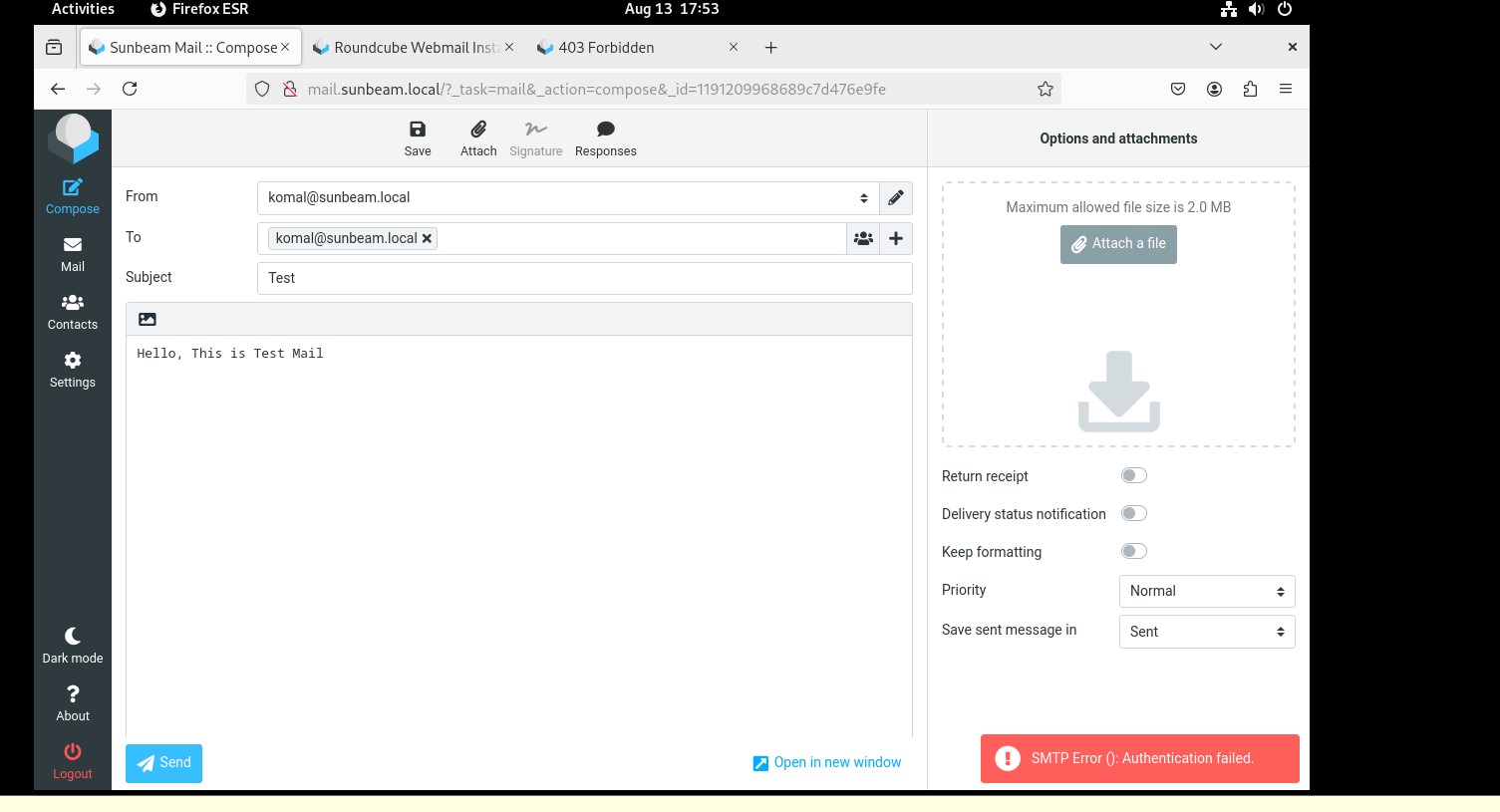


## HTTPS and SSL

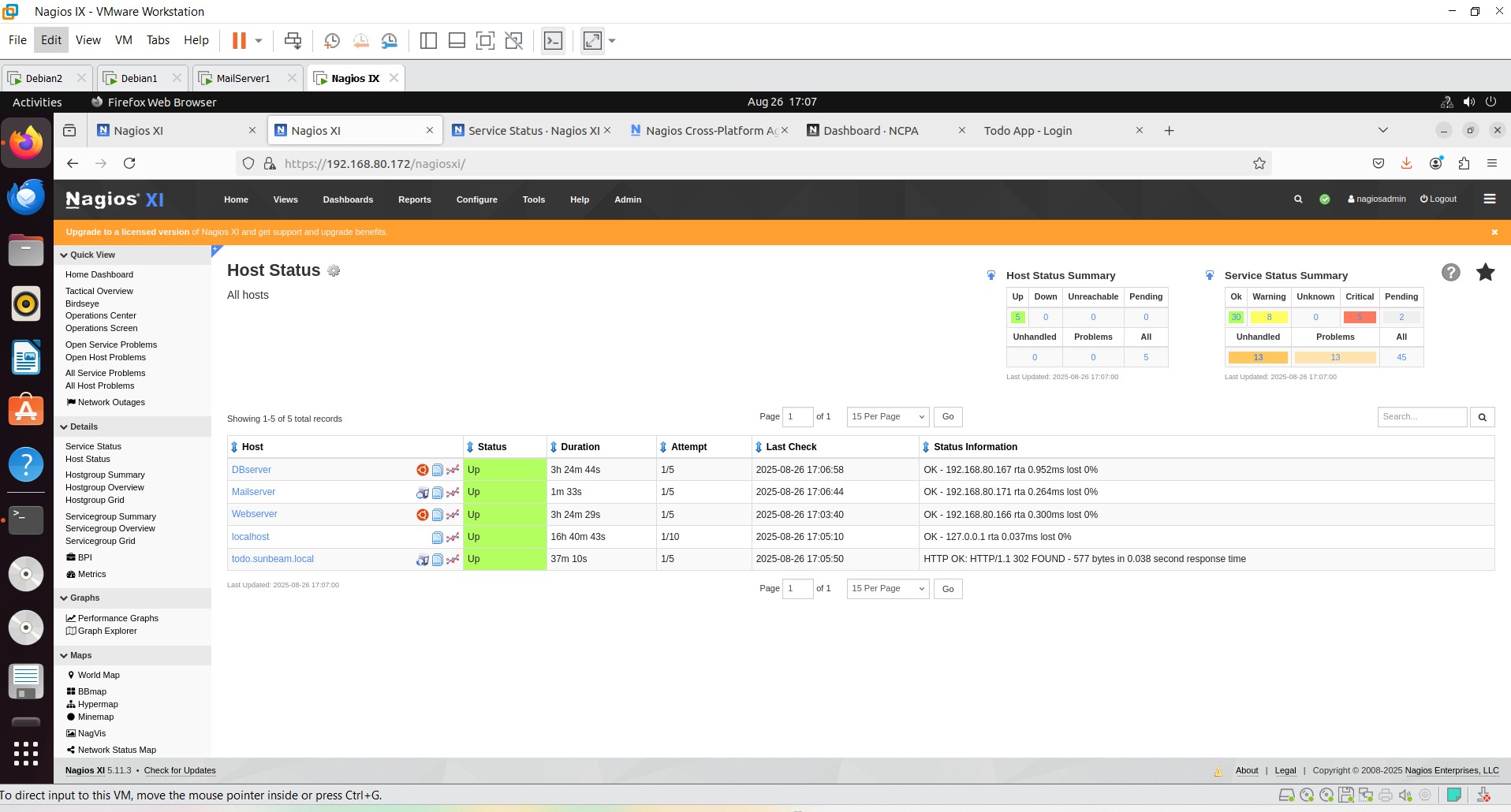


## Mail Server and Roundcube

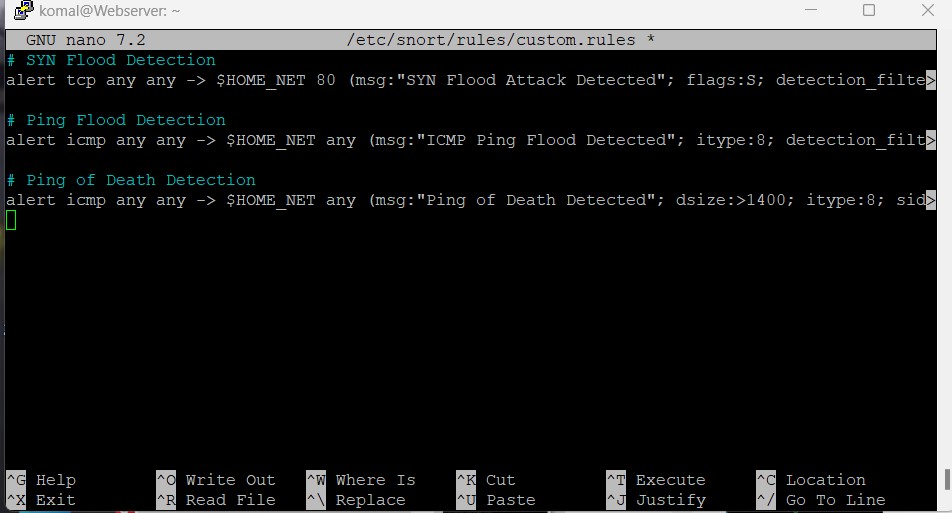


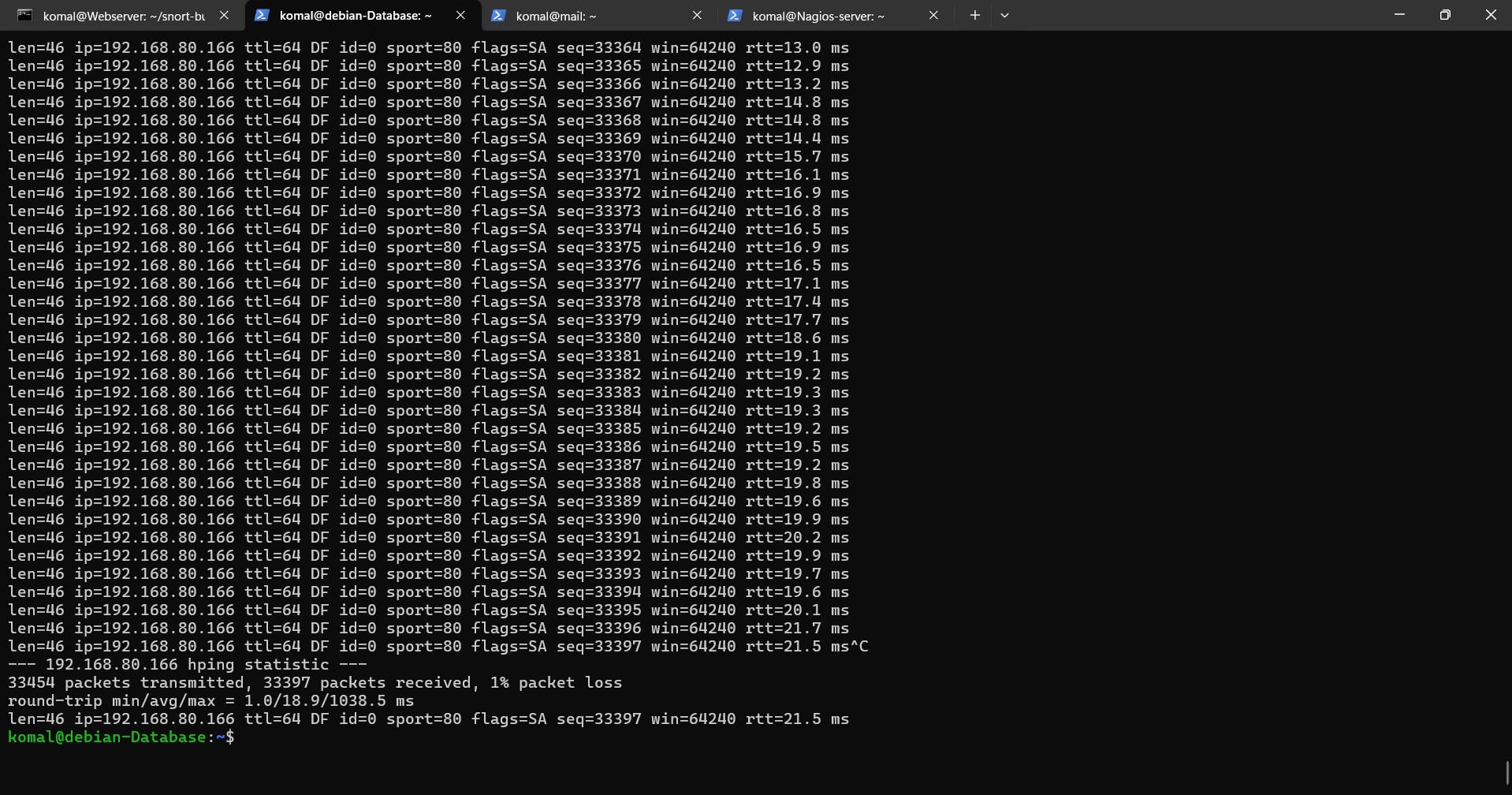


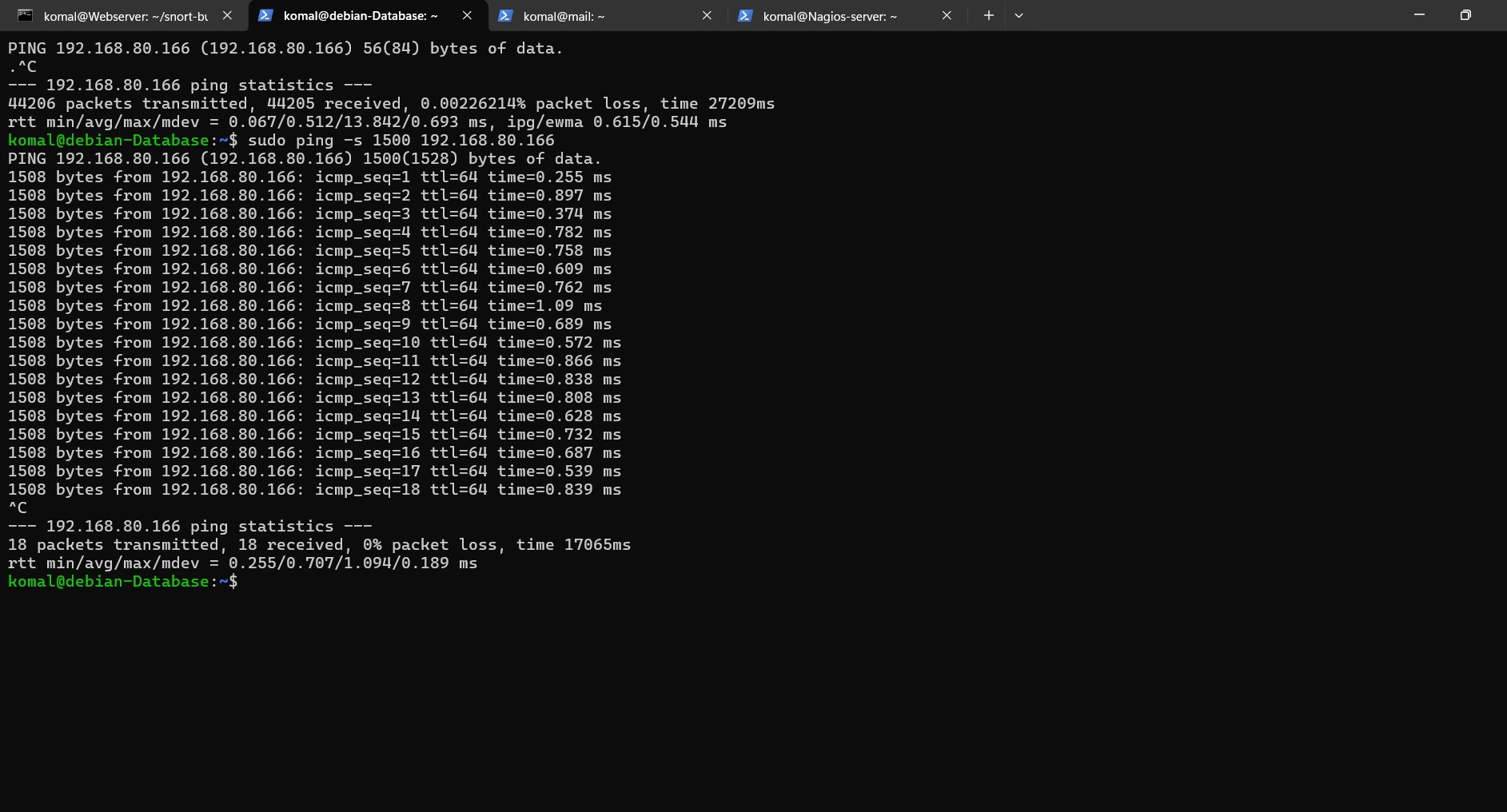
* 1. **Nagios**

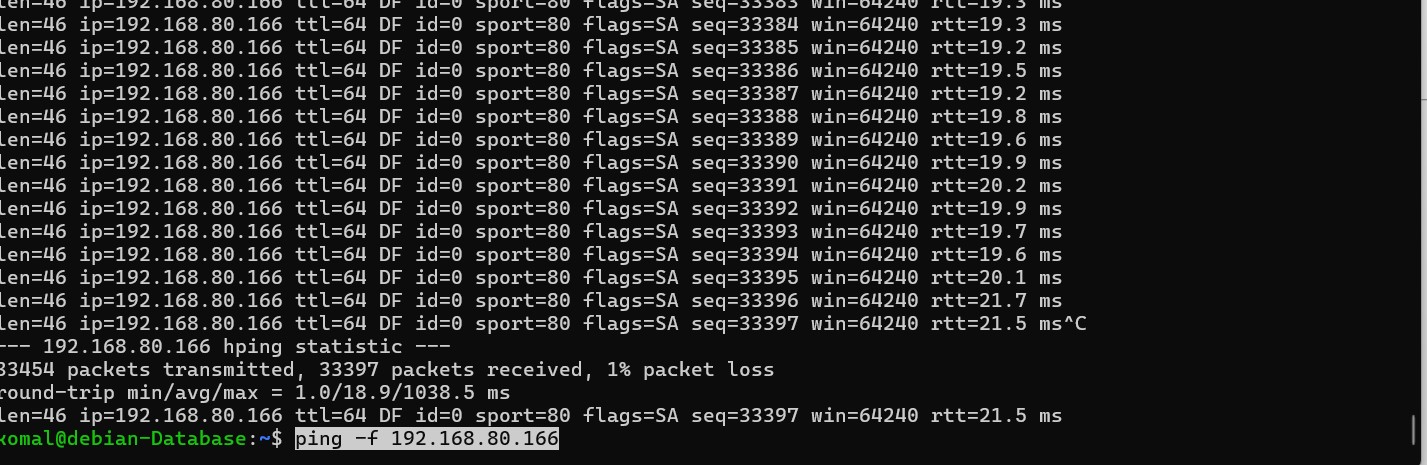


* 1. **Snort**

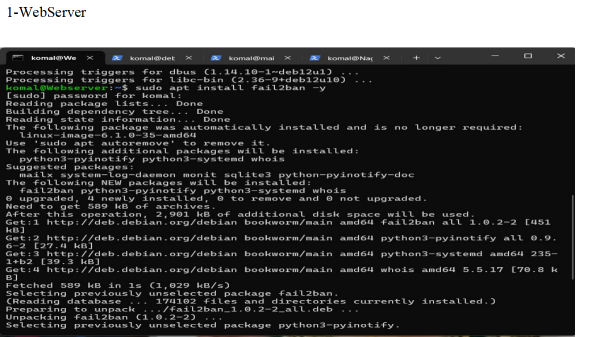
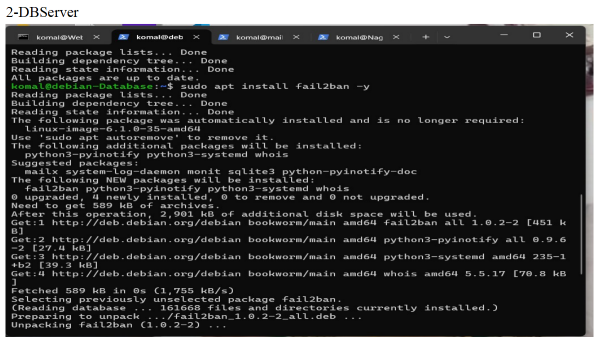
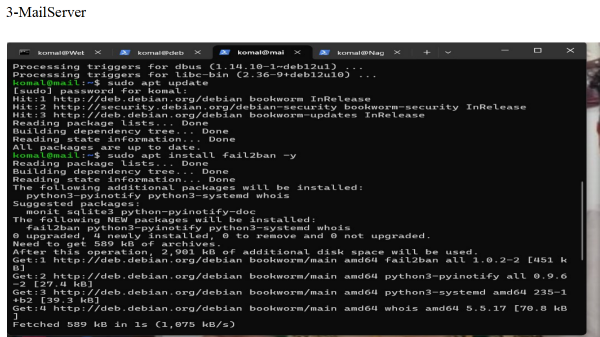
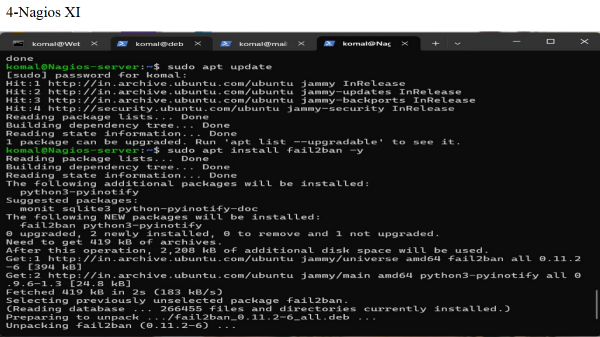




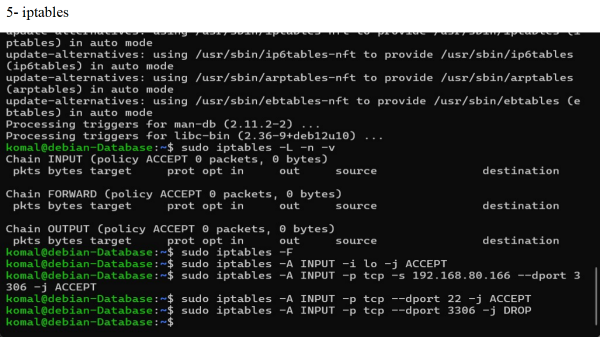




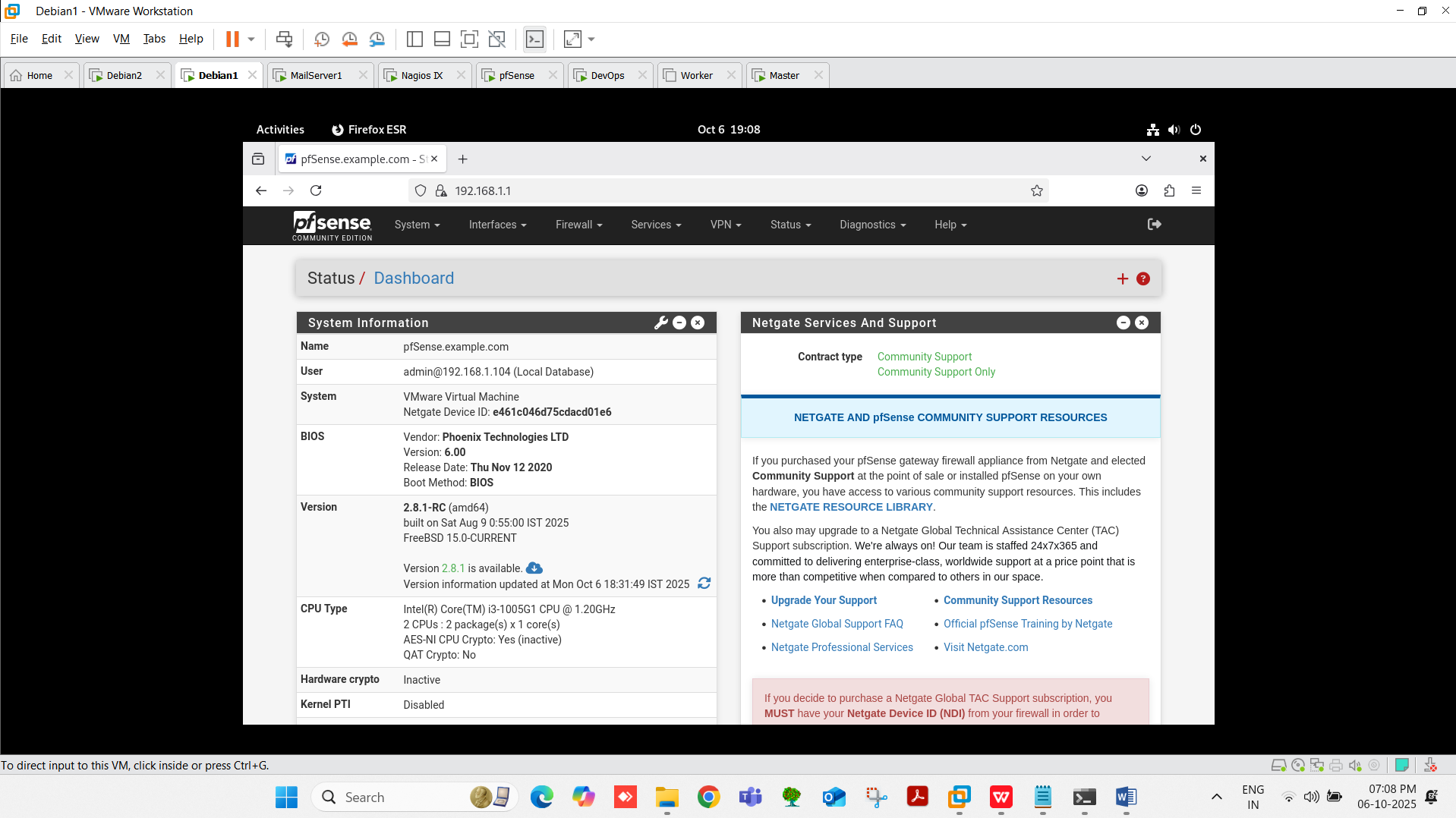
* 1. **Fail2ban**

**** **** 

**4.6 iptables**



**4.7 pfSense**



# 5. CONCLUSION

## 5.1 Conclusion

Hence, we have successfully deployed a highly available and secure web server environment on Amazon Web Services (AWS). And ensured the reliability, performance, and security of the web application while maintaining efficient development and operational processes.

## 5.2 Future Scope

Docker is majorly considered as a best solution for service availability. It can be attached to implement continuous integration and continuous development i.e., CICD. It can be used in development when the software is getting develop this tool can used there to continuous security evaluation of software so that developers can program it more securely. As we have developed this tool for small scale. In future It can be used for large scale.

# REFERENCES

**Paper 1:** - A Qualitative Study of DevOps Usage in Practice

Author: Floris Erich, C. Amrit & M. Daneva

**Paper 2:** - Devops, A New Approach To Cloud Development & Testing

Author: Dhaya Sindhu Battina

**Paper 3:** - Review paper on Snort and reviewing its applications in different fields

Author: Harpreet Sandhu, Manpreet Kaur.