LIA1_distance_task

Building and Deploying a Linux-Based Platform for Network Services & Monitoring

Student: Muhammad Adnan
Handledare : Daniel Thyselius
Företag: Mindful Stack AB

Kurs : LIA1
Kurs Kod: NÄTD24LIN_LIA140)
Klass : NÄTD24LIN (Nätverkstekniker 2024 - Linköping)

GitHub Repository: https://github.com/muhad308/Lia-Linux-Project

Summary:

This report outlines the design, implementation, configuration, and documentation of a virtual server environment leveraging Ubuntu Server LTS. The project's goal was to deploy essential network services: DHCP, DNS (BIND9), Centralized Syslog, and Zabbix Monitoring, for a hypothetical medium-sized company (approx. 150 users). All configurations were performed via the command-line interface (CLI). The setup involved a Windows 11 host with VirtualBox, hosting a server VM (liaserver, 192.168.56.10) and a client VM (liaserver2, initially static 192.168.56.20, later DHCP-assigned). Key aspects included server hardening, automation scripting for backups and service monitoring, and continuous documentation within a GitHub repository.

Table of Contents

1.Introduction

This LIA1 project focuses on designing, deploying, and securing a Linux-based environment for managing network services and monitoring using tools like SSH, UFW, Fail2Ban, BIND (DNS), DHCP, and Zabbix. I worked with two virtual machines on VirtualBox – one as an Ubuntu 24.04 LTS Server (lia-server) and the other as a client (liaserver2), and all configuration and versioning were maintained in a GitHub repository.

1.1 Background & Problem Statement

The purpose of the LIA project was to gain hands-on experience managing Linux servers, configuring network services, applying security measures, and setting up monitoring solutions in a lab environment. The target platform is a small-scale company infrastructure running on Linux using open-source tools.

Key learning outcomes include:

- Linux server deployment
- Network configuration and IP management
- Secure remote access (SSH)
- Hardening using firewall and Fail2Ban
- DHCP & DNS server setup
- Log collection & backup automation
- Zabbix monitoring installation and alerting
- Using Git and GitHub for configuration management

1.2 Purpose and Research Questions

The main goal of this project was to gain practical experience in installing, configuring, troubleshooting, and documenting real-world network services on a Linux platform

Key questions addressed include:

- How to design and deploy a virtualized Ubuntu Server environment for critical network services?
- What are the best practices for hardening a Linux server for security?
- How to configure DHCP, DNS, Syslog, and Zabbix for a corporate network?
- How can we monitor system and network health effectively in real-time?
- What automation strategies (Bash/Python scripting) can streamline administrative tasks?
- How to maintain comprehensive documentation in a Git-based environment?

1.3 Methodology and Sources

I used Ubuntu Server LTS as the base operating system in a VirtualBox virtualized environment. All services were configured using terminal commands (no GUI tools). Configuration files and logs were version-controlled via Git and pushed to GitHub. Key references include the official documentation for Ubuntu, BIND9, Zabbix, rsyslog, and tutorials by DigitalOcean, Linuxize, and Zabbix.com.

2. Background

This section provides a brief overview of the core network services implemented, explaining their role and significance.

2.1 DHCP (Dynamic Host Configuration Protocol)

DHCP automates the distribution of network configuration parameters (IP addresses, DNS server info, gateway) to devices. It eliminates manual IP configuration, reducing errors and administrative overhead, crucial for scalability in a company with many users.

2.2 DNS (Domain Name System) with BIND9

DNS translates human-readable domain names into numerical IP addresses. BIND9 is a widely used DNS software for authoritative name serving and caching. In a corporate setting, internal DNS management is essential for resolving local hostnames and services, while forwarders handle external queries.

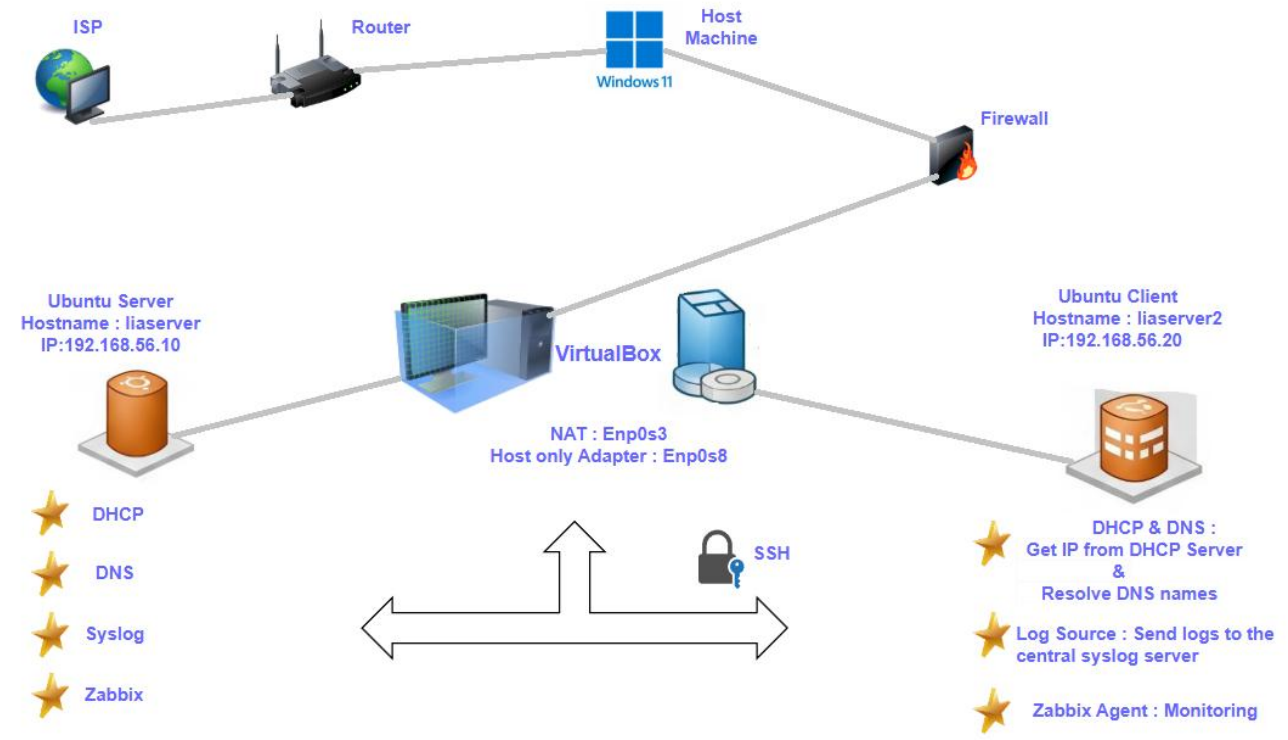
2.3 Syslog Server (rsyslog + Logrotate)

Syslog is a standard for centralized log message collection from network devices and servers. rsyslog is a powerful daemon for this, offering advanced filtering. Logrotate manages log file sizes by automating compression, rotation, and removal, preventing excessive disk space consumption and simplifying auditing and troubleshooting.

2.4 Monitoring (Zabbix Server + Agent)

Zabbix is an open-source monitoring tool for network parameters, server health, and application performance. It provides real-time monitoring, data visualization, and flexible alerting. Implementing Zabbix allows proactive identification and resolution of issues, ensuring high availability and optimal performance of network services.

3. Network Toplogy



4. Tools and Environment

Component	Details
Host Machine	Windows 11
Virtualization	Oracle VirtualBox
Linux Distribution	Ubuntu Server 24.04 LTS
Git & GitHub	Version control & backup
SSH	Secure remote access
Netplan	Static IP configuration
UFW	Firewall
Fail2Ban	Brute-force protection
DHCP/DNS	isc-dhcp-server & bind9
Zabbix	Monitoring platform
Automation	Bash scripts
Documentation	Markdown, GitHub repo
GitHub repository:	https://github.com/muhad308/Lia-Linux-Project

5. Project Execution (Weekly Tasks)

This section details the implementation and configuration of the virtual server environment and the four core network services, integrated with the project's weekly progression.

Week 1: Installation & Network Setup

- Installed VirtualBox on Windows 11.
- Installed Ubuntu Server (Hostnames: liaserver and liaserver2).
- Configured VirtualBox network:
    - Server: Host-only Adapter 192.168.56.10
    - Client: Initially DHCP, later static 192.168.56.20
    - enp0s3 (NAT) for Internet access
    - enp0s8 (Host-only) for internal network communication
- Verified network connectivity between server and client.
- Configured Netplan for static IPs on both machines.

Week 2: SSH Setup & Hardening

- Enabled SSH server.
- Configured SSH key-based authentication.
- Disabled password authentication (PasswordAuthentication no).
- Disabled root SSH login.

- Hardened SSH using proper permissions.
- Tested secure remote login between server and client.
- Git repository initialized.

Week 3: Security Hardening

- Installed and configured UFW firewall.
- UFW is configured to allow only necessary traffic (SSH, DHCP, DNS, Zabbix) and block all others by default. Logging is enabled for auditing.
- Allowed necessary ports (22, 53, 67, 80, 443, 10050).
- Installed Fail2Ban:
    - Fail2Ban protects against brute-force attacks by monitoring SSH login attempts and banning IPs with multiple failures. Configuration is managed in jail.local and logs are reviewed regularly.
    - SSH brute-force protection
    - Configured jail.local
- Created backups of configuration files in Git repository.
- Documented hardening in hardening/ and checklists/.

Week 4: DHCP & DNS Configuration

- Installed isc-dhcp-server on server.
- The DHCP server on liaserver automatically assigns IP addresses to client machines, reducing manual errors. Configuration is managed via dhcpd.conf and tested to ensure reliable IP assignment.
- Configured /etc/dhcp/dhcpd.conf for DHCP scope 192.168.56.20-192.168.56.100.
- Installed bind9 DNS server.
- BIND9 is configured to resolve internal hostnames to IP addresses, making network navigation easier. The DNS server is set up with appropriate zone files and tested for correct resolution.
- Configured zones:
    - Forward: lia.local
    - Reverse: 56.168.192.in-addr.arpa
- Verified name resolution from client using dig.
- Backed up DNS & DHCP config to GitHub.

Syslog (rsyslog)

- Syslog centralizes logs from both server and client, aiding troubleshooting and auditing. Logs are rotated using logrotate to conserve disk space.
- Configured /etc/rsyslog.d/remote.conf to listen and receive logs
- Client logs stored at /var/log/remote/<liaserver2>
- Logrotate applied to rotate logs daily and archive weekly
- Syslog centralizes logs from both server and client, aiding troubleshooting and auditing. Logs are rotated using logrotate to conserve disk space.

Week 5: Monitoring with Zabbix

- Installed MariaDB, Apache, PHP, and Zabbix server on liaserver.
- Configured Zabbix database.
- Installed Zabbix agent on client (liaserver2).
- Added host to Zabbix Web UI.
- Configured email alert system using Gmail SMTP.
- Triggered and tested alerts (stopping agent to simulate failure).
- Exported Zabbix templates for future import.

6. DNS Redundancy Plan

Redundancy means having a backup DNS server that can respond if the primary fails

In this Project only one VM is used, DNS redundancy is not implemented, but this plan would be followed in production:

- Set up a second Ubuntu VM
- Configure it as a BIND slave
- Use round-robin DNS or resolv.conf on clients to list both servers

Client DNS Redundancy

Clients can be configured with both DNS IPs:

- nameserver 192.168.56.10
- nameserver 192.168.56.20

If the first fails, the second is used automatically.

Benefits

- Avoid downtime due to DNS failure
- Balanced query load
- Automatic failover without user action

7. Logs and Errors Faced

- **SSH Issues**: Host key verification failed due to incorrect key ownership → resolved by resetting permissions to 600
- **DNS Error**: BIND9 not starting – missing ; in named.conf
- **Zabbix Issue**: Web UI showed “Zabbix Server is not running” – resolved by checking MariaDB password in zabbix_server.conf
- **Git Errors**: Push rejected due to remote changes → resolved using git pull --rebase
- **Email Alert Error**: Gmail blocking SMTP → solved by enabling app password for Gmail

All logs are stored in:

- /lia-linux-project/logs/
- /var/log/remote/lia-server & /lia-client/

8. Logbook & Documentation

- logbook.md tracks all activities
- checklist.md updated weekly.
- All configuration files saved in folders: netplan/, dns/, dhcp/, fail2ban/, zabbix/
- Screenshots stored in screenshots/
- Scripts and automation saved in scripts/

GitHub Repo: https://github.com/muhad308/Lia-Linux-Project

9. Learning Outcomes

- Comprehensive experience with Linux-based network services (DHCP, DNS, Syslog, Zabbix)
- Implementation of hardening measures (SSH, UFW, Fail2ban, syslog).
- Development of Bash and Python automation scripts.
- Understood centralized monitoring with Zabbix.
- Practiced Git version control and documentation.

10. Future Considerations and Improvements

- **TLS-Encrypted Syslog (RFC 5425):** Implement TLS encryption for syslog communication to secure log data in transit (for VG requirement).
- **Redundancy with Secondary DNS:** Deploy an additional VM as a secondary DNS server for high availability (alternative VG requirement).
- **Advanced Monitoring:** Further customize Zabbix with more specific items, complex triggers, and integration with external alerting systems.
- **Infrastructure as Code:** Explore tools like Ansible for automated, consistent deployment and management of server configurations.

11. References

- https://ubuntu.com/server/docs
- https://www.zabbix.com/documentation/current/en/
- https://linuxize.com/
- https://www.digitalocean.com/community/tutorials