Secure Headless Database Infrastructure Using VirtualBox

A virtualized Linux setup with SSH-only access and centralized remote control

PROJECT SUMMARY

This project demonstrates the setup of a secure, headless MySQL database hosted on a Linux Desktop VM within Oracle VirtualBox. The system is designed to prevent any direct local access, relying solely on a Linux Server VM to manage the database remotely via SSH key authentication. Security measures include disabling GUI and TTY logins, firewall rule enforcement, restricted sudo access, and SSH hardening. The result is a locked-down environment ideal for secure data handling and administrative isolation in virtualized lab environments.

Tebogo Matseding

Project

1st time logging into Desktop

Sudo Apt Update

MySQL Installation

MySQL Status

Create Client Databases

Creating Table Format

Data Entries of Client Information

Data Entries of Client Information

Network Configurations

Installing UFW

Assigning port 22 to my Server

Installing OpenSSH-Server

SSH Security Configurations

PermitRootLogin no

PasswordAuthentication no

AllowUser Admin@192.168.1.68

Verifying Changes Didn’t Break SSH

Verifying Root Login is Denied

Disable GUI (sudo systemctl mask gdm3)

Verifying GUI is disabled

Verifying Server IP Address

Installing OpenSSH-Client

Generating SSH-Key (w Passphase)

Granting Server Sudo Privileges (Admin:192.168.1.69:)

Login Into Database Server

Masking Services (sudo systemctl mask getty@tyy1.services through to 6)

Verifying Masking Services

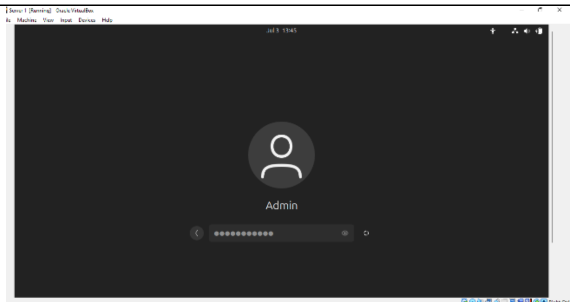
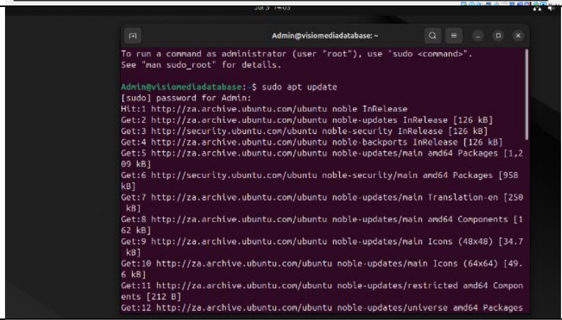
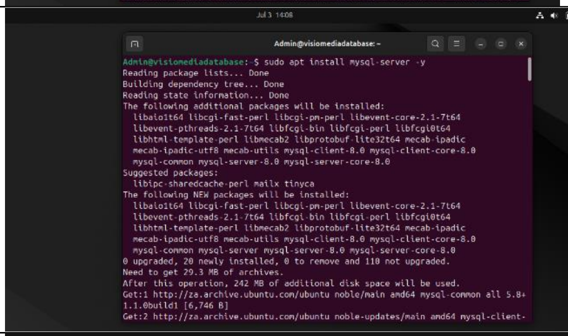
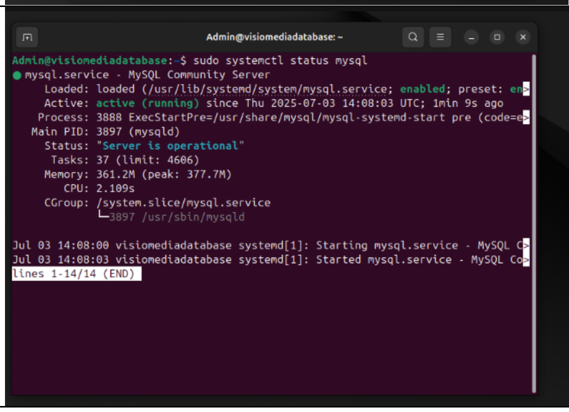
Project Architecture Illustration Summary

Additional Security Risks to Address

START

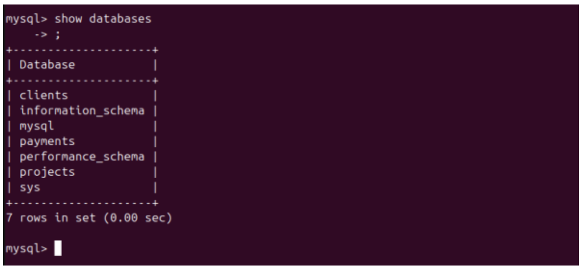
<table>
<tr>
<td>The purpose of this project is not only to demonstrate hands-on experiences but to be used as a guide for those looking to complete a similar project (I will be linking a video of how to install the software we will be using to complete our task/project)</td>
<td>

- Installing Oracle VirtualBox on Windows
- Installing Oracle VirtualBox on Mac
- Installing Linux Desktop on Oracle VirtualBox
- Installing Ubuntu Server on Oracle VirtualBox

</td>
</tr>
<tr>
<td>1st time logging into Desktop
The initial login to the Ubuntu Desktop VM was completed successfully using the configured user account.</td>
<td>
</td>
</tr>
<tr>
<td>Sudo Apt Update
After logging into the Desktop VM, the system was updated using sudo apt update to ensure all packages and security dependencies were up to date.</td>
<td>
</td>
</tr>
<tr>
<td>MySQL Installation
MySQL Server was installed on the Desktop VM to manage client data in a secure and structured relational format.</td>
<td>
</td>
</tr>
<tr>
<td>MySQL Status
The status of the MySQL service was checked to confirm it was active and running properly.</td>
<td>
</td>
</tr>
</table>

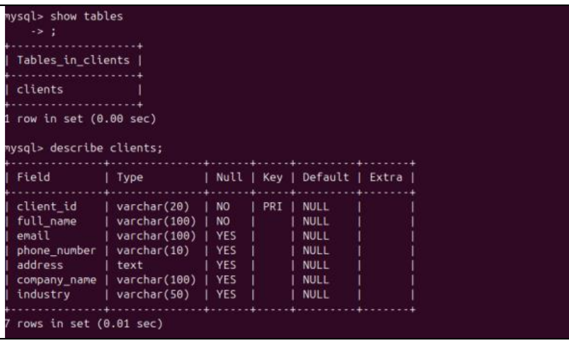
Create Client Databases

Databases were created to store Visiomedia client information using basic SQL commands from the terminal.



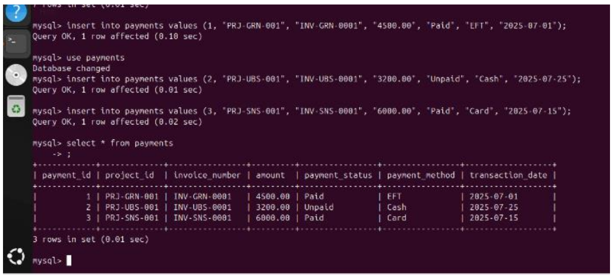
Creating Table Format

Sample client data was added to the database to simulate a real-world scenario with multiple entries.



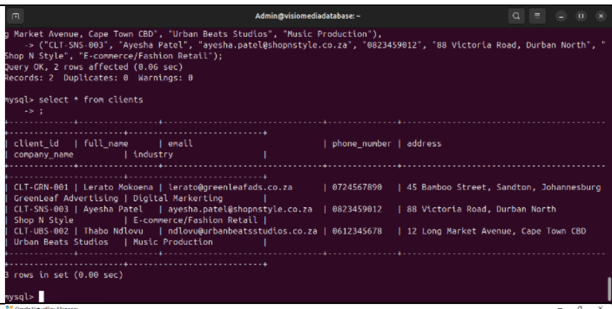
Data Entries of Client Information

Sample data was inserted into the database to simulate real Visiomedia client records.



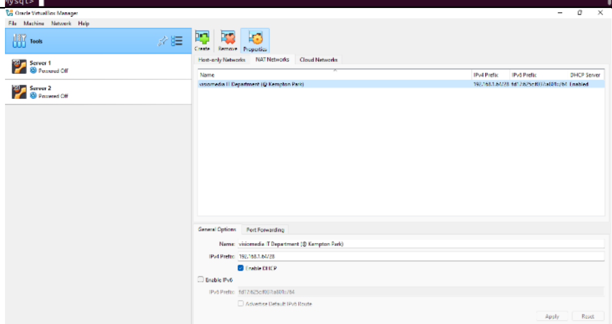
Data Entries of Client Information

Another sample



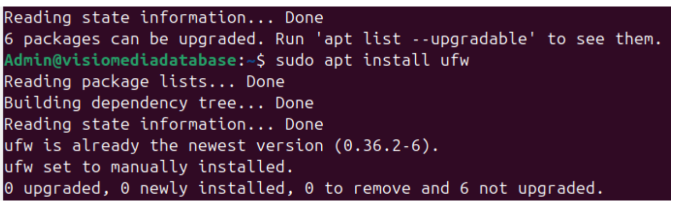
Network Configurations

Both VMs were assigned static IP addresses within a NAT network to ensure consistent communication.



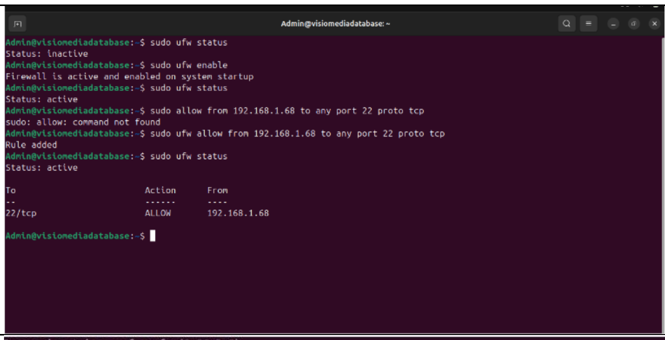
Installing UFW

The Uncomplicated Firewall (UFW) was installed on the Desktop VM to restrict incoming connections.



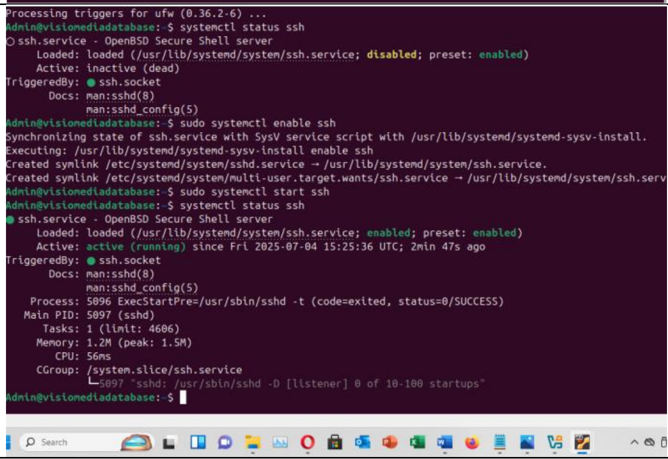
Assigning port 22 to my Server

Port 22 was allowed through UFW to permit SSH access only from the Server VM.



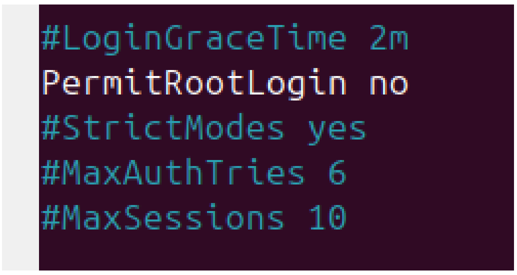
Installing OpenSSH-Server

The OpenSSH server was installed and configured to accept secure, key-based SSH connections.



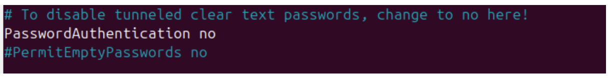
SSH Security Configurations

This directive in the SSH configuration (/etc/ssh/sshd_config) prevents the root user from logging in directly over SSH. This is a key security measure that reduces the risk of brute-force attacks against the root account. Instead, administrative actions must be performed using a standard user with sudo privileges.



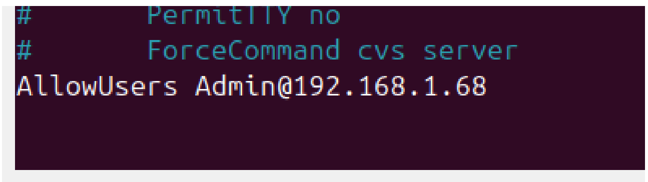
PasswordAuthentication no

This disables password-based login entirely, enforcing SSH key-based authentication. It prevents attackers from attempting to guess or brute-force user passwords over SSH. Only users with a valid SSH private key (and matching public key on the server) can access the system.



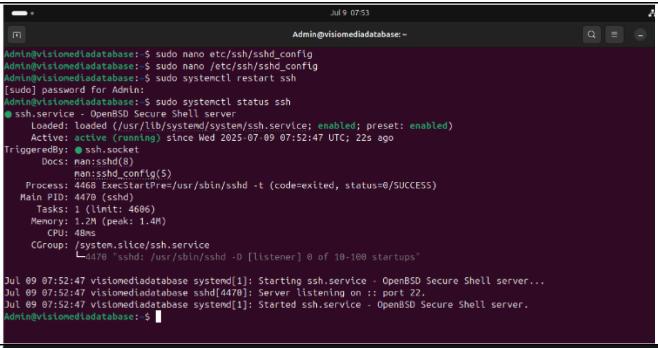
AllowUsers admin@192.168.1.68

This line strictly limits who can SSH into the machine. It specifies that only the admin user from IP address 192.168.1.68 (the Server VM) is allowed to initiate an SSH session. All other users or source IPs will be denied access, even if they have valid keys or credentials.



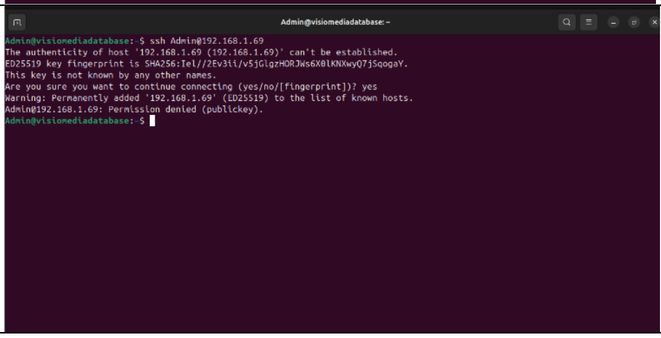
Verifying Changes Didn’t Break SSH

A connection test was performed to confirm SSH access still worked after security settings were applied.



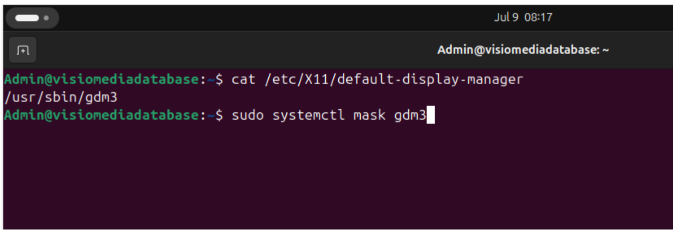
Verifying Root Login is Denied

An SSH attempt as root was denied, confirming that root login was successfully disabled.



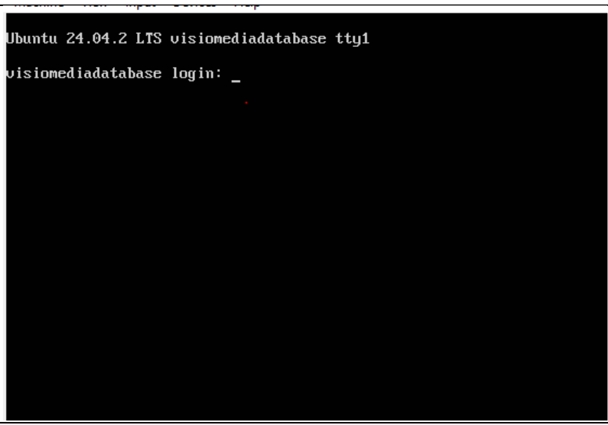
Disable GUI

The graphical login manager was masked, disabling all GUI access on the Desktop VM.



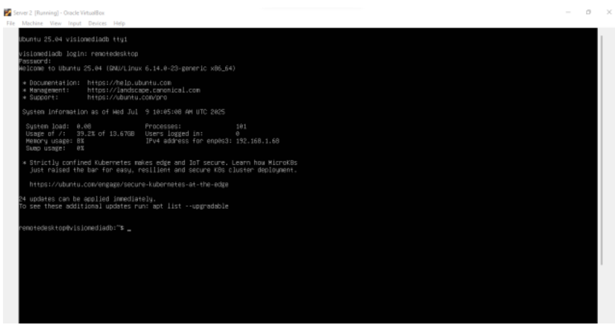
Verifying GUI is disabled

Upon reboot, the Desktop system showed no GUI, confirming the machine was operating in headless mode.



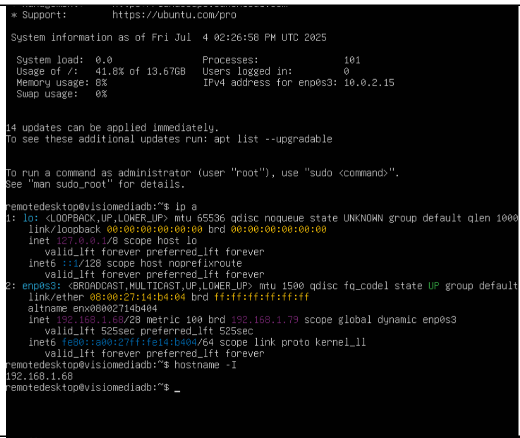
Login into Server

The Server VM was accessed using its local user account in VirtualBox. This system serves as the centralized administrator, responsible for managing the Desktop VM. Once logged in, all secure configurations and SSH key operations were carried out from this machine.



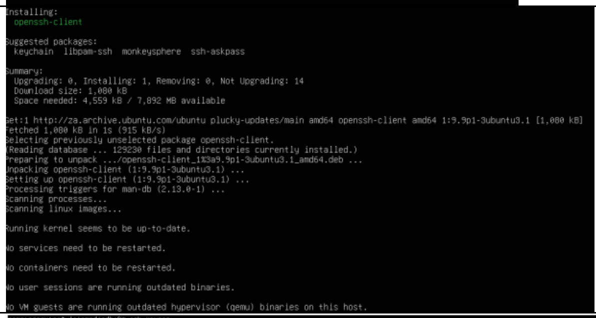
Verifying Server IP Address

The IP address of the Server VM was verified using ip a and hostname -I, confirming correct network placement.



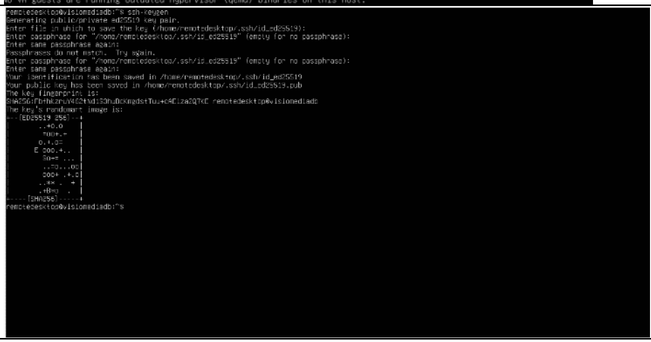
Installing OpenSSH-Client

The OpenSSH client was installed on the Server VM to enable key generation and outbound SSH connections.



Generating SSH-Key (w Passphase)

An SSH key pair was generated on the Server with a secure passphrase for enhanced security.

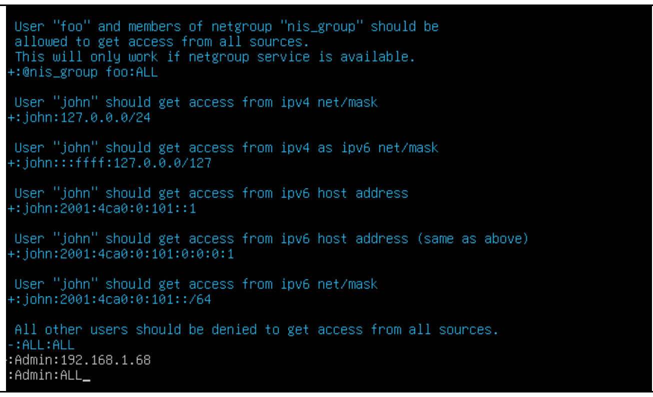


Copying SSH key to Database Server

After generating the SSH key pair on the Server VM, the public key was copied to the Desktop VM using the ssh-copy-id command. This placed the public key in the ~/.ssh/authorized_keys file on the Desktop, enabling secure, passwordless login from the Server.

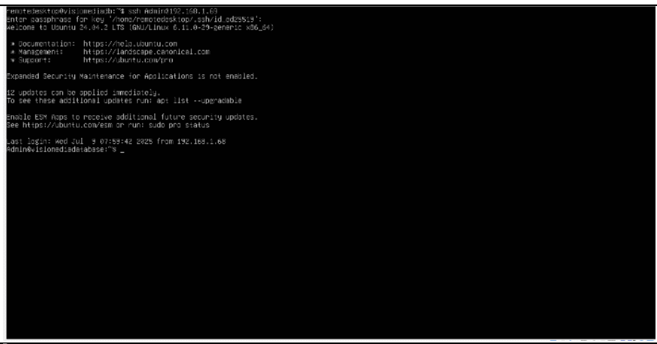
Granting Server Sudo Privileges

Access rules were defined using access.conf to allow only the Server to use sudo on the Desktop.



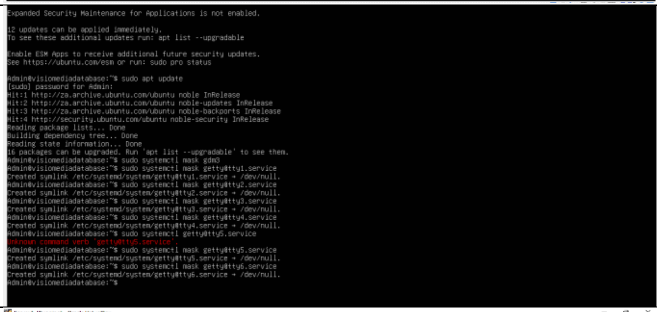
Login Into Database Server

SSH access was tested from the Server into the Desktop VM using the generated SSH key.



Masking Services

TTY services were masked to prevent logins via Ctrl+Alt+F1 through F6 on the Desktop.

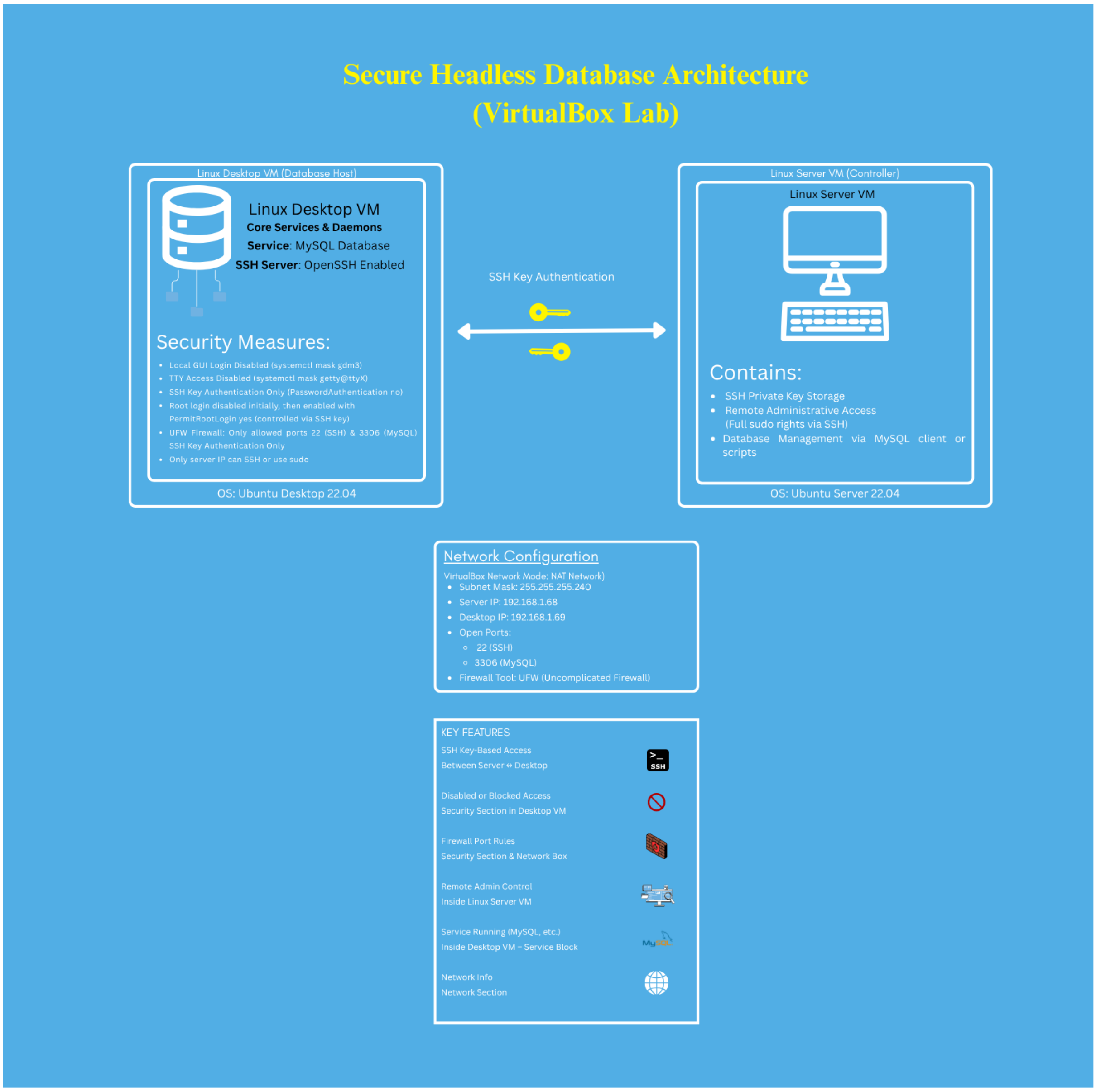


Verifying Masking Services

A reboot confirmed that all masked services were disabled and local access was completely blocked.

Project Architecture Illustration Summary

The illustration below represents the secure, headless database infrastructure designed and implemented during this project. It shows the two virtual machines configured in Oracle VirtualBox: a Linux Desktop VM hosting the MySQL database and a Linux Server VM that acts as the sole administrative controller. The diagram highlights the SSH key-based authentication, disabled local access, firewall configuration, and PAM restrictions that enforce remote-only management. This setup simulates a real-world centralized management model with minimal attack surface and strict access control.



Additional Security Risks to Address

Lack of Encrypted Database Communication

MySQL currently accepts connections on port 3306 without SSL. Enabling SSL/TLS encryption would protect sensitive client data in transit, even within a virtualized internal network.

No Intrusion Detection or Logging Mechanism

The system lacks tools to monitor for unauthorized login attempts, file changes, or privilege escalation. Adding auditd or fail2ban would provide early warning of suspicious behavior.

No Automatic Backup or Recovery Plan

The database currently lacks scheduled backups or recovery testing. Implementing a backup script or replication system would prevent data loss in case of system failure or corruption.