



## **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 Passphrase)

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

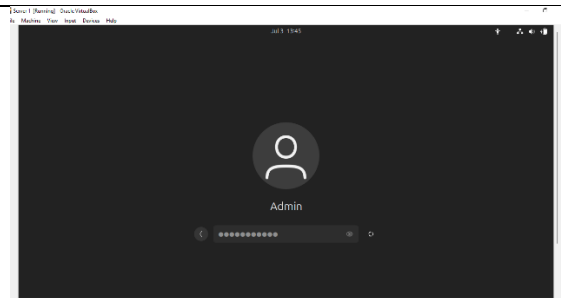
# START

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)

- [Installing Oracle VirtualBox on Windows](#)
- [Installing Oracle VirtualBox on Mac](#)
- [Installing Linux Desktop on Oracle VirtualBox](#)
- [Installing Ubuntu Server on Oracle VirtualBox](#)

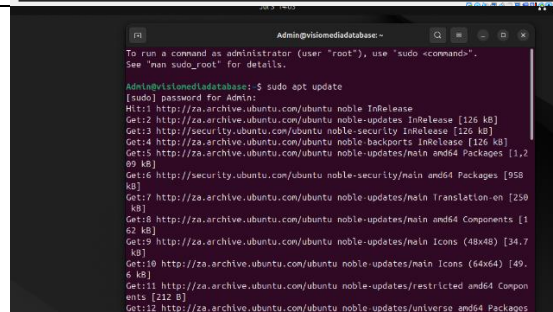
## 1<sup>st</sup> time logging into Desktop

The initial login to the Ubuntu Desktop VM was completed successfully using the configured user account.



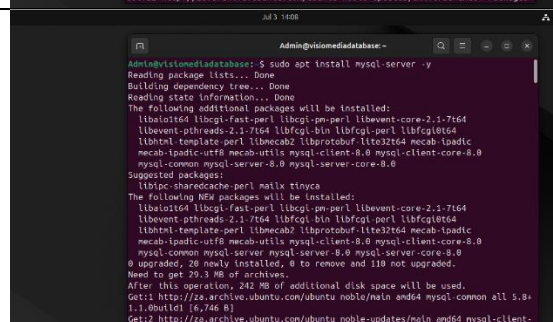
## Sudo Apt Update

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.



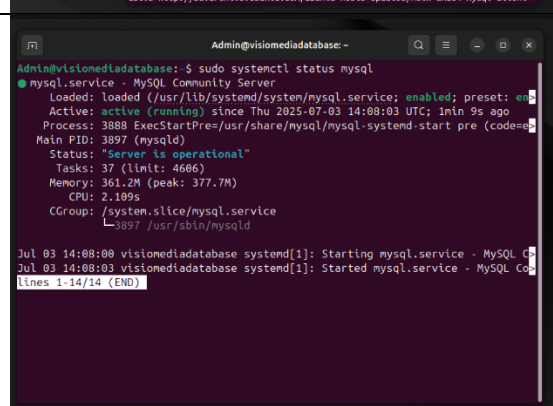
## MySQL Installation

MySQL Server was installed on the Desktop VM to manage client data in a secure and structured relational format.



## MySQL Status

The status of the MySQL service was checked to confirm it was active and running properly.



## Create Client Databases

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

```
mysql> show databases
-> ;
+-----+
| Database |
+-----+
| clients  |
| information_schema |
| mysql   |
| payments |
| performance_schema |
| projects |
| sys     |
+-----+
7 rows in set (0.00 sec)

mysql>
```

## Creating Table Format

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

```
mysql> show tables
-> ;
+-----+
| Tables_in_clients |
+-----+
| clients            |
+-----+
1 row in set (0.00 sec)

mysql> describe clients;
+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+
| client_id  | varchar(20) | NO   | PRI | NULL    |       |
| full_name  | varchar(100) | NO   |     | NULL    |       |
| email      | varchar(100) | YES  |     | NULL    |       |
| phone_number | varchar(10) | YES  |     | NULL    |       |
| address    | text       | YES  |     | NULL    |       |
| company_name | varchar(100) | YES  |     | NULL    |       |
| industry   | varchar(50) | YES  |     | NULL    |       |
+-----+
7 rows in set (0.01 sec)
```

## Data Entries of Client Information

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

```
mysql> insert into payments values (1, "PRJ-GRN-001", "INV-GRN-0001", "4500.00", "Paid", "EFT", "2025-07-01");
Query OK, 1 row affected (0.10 sec)

mysql> use payments
Database changed
mysql> insert into payments values (2, "PRJ-UBS-001", "INV-UBS-0001", "3200.00", "Unpaid", "Cash", "2025-07-25");
Query OK, 1 row affected (0.01 sec)

mysql> insert into payments values (3, "PRJ-SNS-001", "INV-SNS-0001", "6000.00", "Paid", "Card", "2025-07-15");
Query OK, 1 row affected (0.02 sec)

mysql> select * from payments
-> ;
+-----+
| payment_id | project_id | invoice_number | amount | payment_status | payment_method | transaction_date |
+-----+
| 1          | PRJ-GRN-001 | INV-GRN-0001   | 4500.00 | Paid           | EFT            | 2025-07-01      |
| 2          | PRJ-UBS-001 | INV-UBS-0001   | 3200.00 | Unpaid         | Cash           | 2025-07-25      |
| 3          | PRJ-SNS-001 | INV-SNS-0001   | 6000.00 | Paid           | Card           | 2025-07-15      |
+-----+
3 rows in set (0.01 sec)

mysql>
```

## Data Entries of Client Information

Another sample

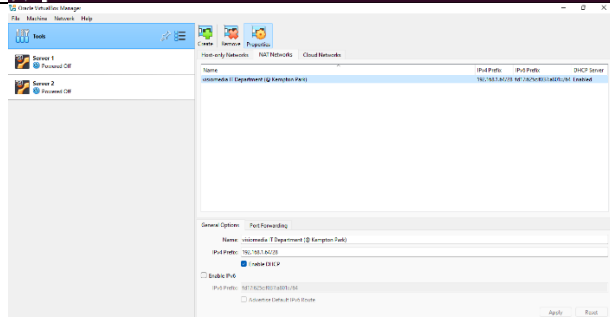
```
Admin@visiomedicadatabase:~$ mysql> insert into clients values ("CLT-SNS-003", "Ayesha Patel", "ayesha.patel@shopstyle.co.za", "8823459012", "88 Victoria Road, Durban North", "Shop N Style", "E-commerce/Fashion Retail");
Query OK, 2 rows affected (0.06 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select * from clients
-> ;
+-----+
| client_id | full_name | email | phone_number | address |
+-----+
| CLT-GRN-001 | Lerato Mokuane | lerato@greenleafads.co.za | 0724567890 | 45 Bamboo Street, Sandton, Johannesburg |
| CLT-SNS-003 | Ayesha Patel | ayesha.patel@shopstyle.co.za | 8823459012 | 88 Victoria Road, Durban North |
| CLT-UBS-002 | Ithabo Ndlovu | ithabo@urbanbeatsstudios.co.za | 0612345678 | 12 Long Market Avenue, Cape Town CBD |
| Urban Beats Studios | Music Production |
+-----+
3 rows in set (0.00 sec)

mysql>
```

## 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.

```
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
Admin@visiomedicadatabase:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
```

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

```
Admin@visionedlatabase:~$ sudo ufw status
Status: inactive
Admin@visionedlatabase:~$ sudo ufw enable
Firewall is active and enabled on system startup
Admin@visionedlatabase:~$ sudo ufw status
Status: active
Admin@visionedlatabase:~$ sudo allow from 192.168.1.68 to any port 22 proto tcp
sudo: allow: command not found
Admin@visionedlatabase:~$ sudo ufw allow from 192.168.1.68 to any port 22 proto tcp
Rule added
Admin@visionedlatabase:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW 192.168.1.68
Admin@visionedlatabase:~$
```

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

```
Processing triggers for ufw (0.36.2-6) ...
Admin@visionedlatabase:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: inactive (dead)
     TriggeredBy: ● ssh.socket
    Docs: man:ssh(8)
          man:ssh_config(5)
Admin@visionedlatabase:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/ssh.service → /usr/lib/systemd/system/ssh.service.
Admin@visionedlatabase:~$ sudo systemctl start ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-07-04 15:25:36 UTC; 2min 47s ago
     TriggeredBy: ● ssh.socket
    Docs: man:ssh(8)
          man:ssh_config(5)
   Process: 5896 ExecStartPre=/usr/sbin/ssh -t (code=exited, status=0/SUCCESS)
   Main PID: 5897 (sshd)
     Tasks: 1 (limit: 4686)
    Memory: 1.2M (peak: 1.5M)
       CPU: 56ms
    CGroup: /system.slice/ssh.service
            └─5897 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
Admin@visionedlatabase:~$
```

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.

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

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.

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

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.

```
# PermitTTY no
# ForceCommand cvs server
AllowUsers Admin@192.168.1.68
```

Verifying Changes Didn't Break SSH  
A connection test was performed to confirm SSH access still worked after security settings were applied.

```
Admin@visionmediadatabase:~$ sudo nano /etc/ssh/sshd_config
Admin@visionmediadatabase:~$ sudo nano /etc/ssh/sshd_config
Admin@visionmediadatabase:~$ sudo systemctl restart ssh
[sudo] password for Admin:
Admin@visionmediadatabase:~$ sudo systemctl status ssh
● ssh.service - OpenSSH Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-07-09 07:52:47 UTC; 22s ago
   TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
         man:sshd_config(5)
   Process: 4468 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 4470 (sshd)
   Tasks: 1 (limit: 4096)
   Memory: 1.2M (peak: 1.4M)
   CPU: 48ms
   CGroup: /system.slice/ssh.service
           └─/usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jul 09 07:52:47 visionmediadatabase systemd[1]: Starting ssh.service - OpenSSH Secure Shell server...
Jul 09 07:52:47 visionmediadatabase sshd[4470]: Server listening on :: port 22.
Jul 09 07:52:47 visionmediadatabase systemd[1]: Started ssh.service - OpenSSH Secure Shell server.
Admin@visionmediadatabase:~$
```

Verifying Root Login is Denied  
An SSH attempt as root was denied, confirming that root login was successfully disabled.

```
Admin@visionmediadatabase:~$ ssh Admin@192.168.1.69
The authenticity of host '192.168.1.69 (192.168.1.69)' can't be established.
ED25519 key fingerprint is SHA256:1e1/2Ev31/v5JG1xH0N2a6s8d10Nkey075pogay.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.69' (ED25519) to the list of known hosts.
Admin@192.168.1.69: Permission denied (publickey).
Admin@visionmediadatabase:~$
```

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

```
Admin@visionmediadatabase:~$ cat /etc/X11/default-display-manager
/usr/sbin/gdm3
Admin@visionmediadatabase:~$ sudo systemctl mask gdm3
```

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

```
Ubuntu 24.04.2 LTS visionmediadatabase tty1
visionmediadatabase login: _
```

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.

```
Ubuntu 24.04 visionmediadb tty1
visionmediadb login: root@localhost
Welcome to Ubuntu 24.04 (GNU/Linux 6.14.0-20-generic x86_64)

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

System information as of Wed Jul 9 12:45:06 AM UTC 2025

System load: 0.00          Processes: 201
Usage of /: 19.1% of 13.4TB    Memory usage: 15
Swap usage: 0%              Disk space for /snapshots: 192.0GB-1.6B

 * Inotify: Confirmed that the kernel is not using inotify to watch the /etc directory.
   Just raised the bar for easy, reliable and secure file cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Updates can be installed immediately.
   To see these additional updates run: apt list --upgradable

root@visionmediadb:~$
```

## Verifying Server IP Address

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

```
+ Support: https://ubuntu.com/pro

System Information as of Fri Jul 4 02:26:58 PM UTC 2025

System load: 0.0          Processes: 101
Usage of /: 41.8% of 13.67GB    Users logged in: 0
Memory usage: 51%          IPv4 address for enp0s3: 10.0.2.15
Swap usage: 0%

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

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

remotedesktop@visionedlab:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 08:00:27:14:04:04 brd ff:ff:ff:ff:ff:ff
    altname enx000027140404
    inet 192.168.1.56/28 metric 100 brd 192.168.1.79 scope global dynamic enp0s3
        valid_lft 525sec preferred_lft 525sec
    inet6 fe80::a00:27:ff:fe14:404/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
remotedesktop@visionedlab:~$ hostname -I
192.168.1.56
remotedesktop@visionedlab:~$ _
```

## Installing OpenSSH-Client

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

```
Installing:
  openssh-client

Suggested packages:
  lftpchain libopenssh monkeysphere ssh-askpass

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 14
  Download size: 1,400 kB
  Space needed: 4,559 kB / 7,892 MB available

Get:1 http://qa.archive.ubuntu.com/ubuntu plucky-updates/main amd64 openssh-client amd64 1:9.9p1-Subuntud.1 [1,400 kB]
Get:2 http://qa.archive.ubuntu.com/ubuntu plucky-updates/main amd64 openssh-client amd64 1:9.9p1-Subuntud.1 [1,400 kB]
Fetched 1,400 kB in 1s (915 kB/s)
Selecting previously unselected package openssh-client.
Reading database ... 12529 files and directories currently installed.
Preparing to unpack .../openssh-client_1:9.9p1-Subuntud.1.amd64.deb ...
Unpacking openssh-client (1:9.9p1-Subuntud.1) ...
Setting up openssh-client (1:9.9p1-Subuntud.1) ...
Processing triggers for man-db (2.13.0-1) ...
Scanning processes...
Scanning linux images...

gpming kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

## Generating SSH-Key (w Passphrase)

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

```
remotedesktop@visionedlab:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/remotedesktop/.ssh/id_ed25519):
Enter passphrase (or leave empty for no passphrase) (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/remotedesktop/.ssh/id_ed25519.
Your public key has been saved in /home/remotedesktop/.ssh/id_ed25519.pub.
The key fingerprint is: 192:168:1:56:192:168:1:56:192:168:1:56:192:168:1:56
The key's randomart image is:
[+] .+..+
     |..+..+
     |..+..+
     |..+..+
     |..+..+
     |..+..+
     |..+..+
     |..+..+
     |..+..+
     |..+..+
remotedesktop@visionedlab:~$
```

## 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.

```
remotedesktop@visionedlab:~$ ssh-copy-id admin@192.168.1.59
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/remotedesktop/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: Target for key(s): 192.168.1.59
/usr/bin/ssh-copy-id: INFO: Warning: Permanently added '192.168.1.59' (ED25519) to the list of known hosts.
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new key
admin@192.168.1.59's password:

Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'admin@192.168.1.59'"
We add to new users that only the key(s) you wanted were added.

remotedesktop@visionedlab:~$
```

## Granting Server Sudo Privileges

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

```
User "foo" and members of netgroup "nis_group" should be
allowed to get access from all sources.
This will only work if netgroup service is available.
+@nis_group foo:ALL

User "john" should get access from ipv4 net/mask
+john:127.0.0.0/24

User "john" should get access from ipv4 as ipv6 net/mask
+john::ffff:127.0.0.0/127

User "john" should get access from ipv6 host address
+john:2001:4ca0:0:101::1

User "john" should get access from ipv6 host address (same as above)
+john:2001:4ca0:0:101:0:0:0:1

User "john" should get access from ipv6 net/mask
+john:2001:4ca0:0:101::/64

All other users should be denied to get access from all sources.
-:ALL:ALL
:Admin:192.168.1.58
:Admin:ALL_
```



## Login Into Database Server

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

[illegible]

## Masking Services

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

[illegible]

## Verifying Masking Services

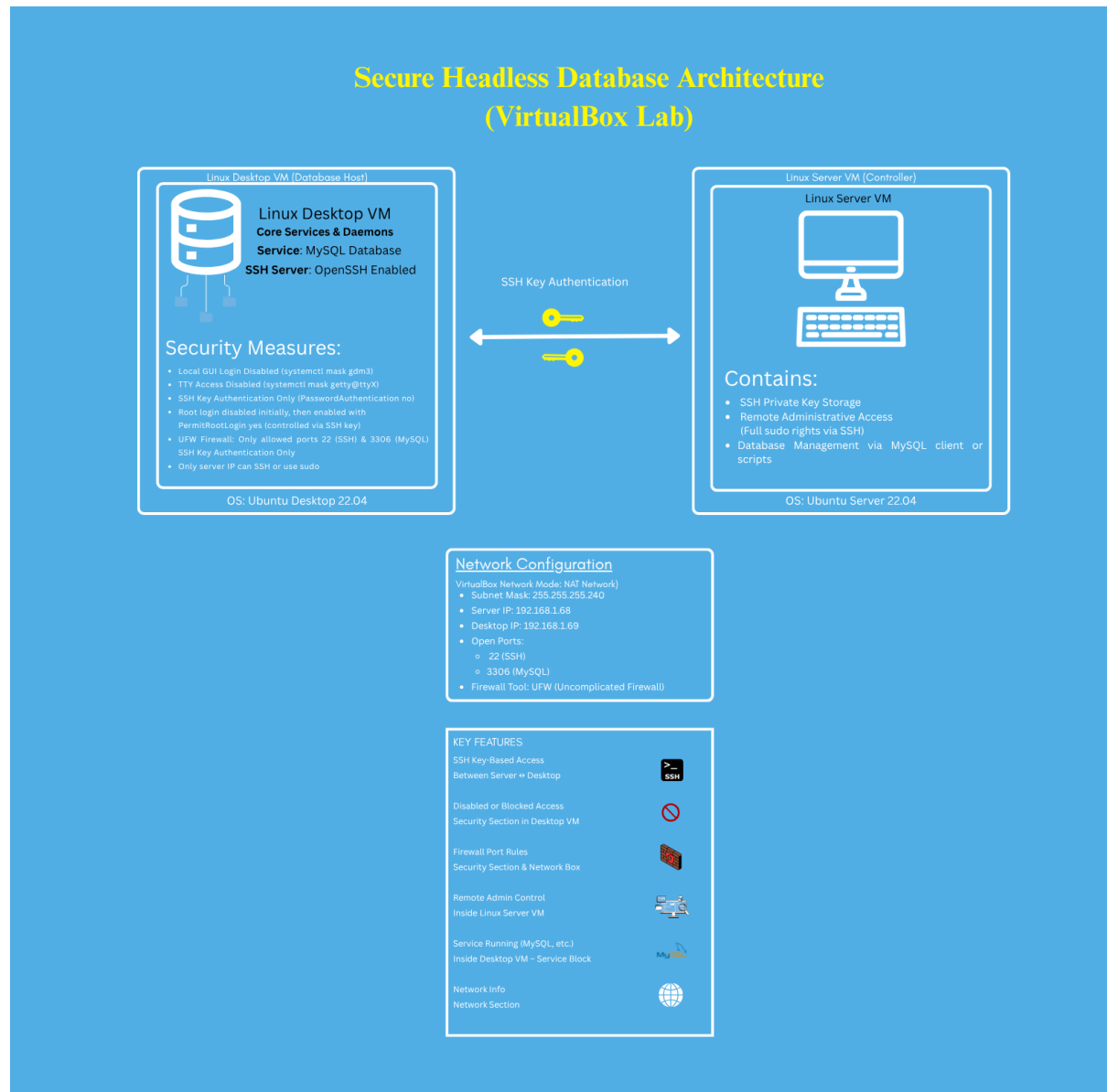
A reboot confirmed that all masked services were disabled and local access was completely blocked.





# 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.