

PROJECT REPORT
Data Center Security
(INCS – 775 – Section)
PRIYANSHU TOMAR
1316941
Michael Colef

CONTENTS

ABSTRACT	3
REQUIREMENTS	4
INTRODUCTION	5
PROJECT NETWORK DIAGRAM	9
SETTING IN VIRTUAL NETWORK	10
INSTALLATION, CONFIGURATION AND TESTING OF FIREWALL	11
INSTALLATION, CONFIGURATION AND TESTING OF DMZ (Webserver)	16
INSTALLATION, CONFIGURATION AND TESTING OF DNS (Server)	19
INSTALLATION, CONFIGURATION AND TESTING OF MYSQL Server	26
CONCLUSION	30

Abstract

This infrastructure design provides a secure and reliable network environment for the small business by using a firewall to filter traffic between the DMZ and the internal network. The firewall is set up to allow the DMZ webserver to be visible to both the Internet and the local network, while the DNS and MySQL/WorkDB servers are only exposed to the internal network. This helps to prevent unauthorized Internet access to these systems.

The DNS server is set up to resolve IP addresses/hostnames on the internal network, allowing internal network devices to connect with one another using domain names rather than IP addresses. This simplifies network management and may enhance network dependability.

The firewall is also set to prevent any communication from the Internet to the internal network. This helps to prevent illegal network access and lowers the danger of external threats.

Overall, by utilizing a firewall to filter traffic between the DMZ and the internal network and setting the DNS server to resolve IP addresses/hostnames on the inside network, this architecture design delivers a safe and dependable network environment for the small business.

Requirements

Hardware:

- A host machine with sufficient hardware resources (e.g. processor, memory, storage) to run VirtualBox and the virtual machines for the infrastructure.
- Network hardware, such as switches and network interface cards, to connect the virtual machines to the appropriate network segments (e.g. DMZ, internal network).

Software:

- A copy of VirtualBox software, such as VirtualBox Workstation.
- A copy of Opnsense software to install on the virtual machine designated for the firewall.
- Operating system installation media for the virtual machines (e.g. ISO files for Windows or Linux).

INTRODUCTION

Info about the VirtualBox environment used in this project.

VirtualBox stands out as a versatile and accessible virtualization software, particularly notable for its open-source, cross-platform capabilities. Developed and maintained by Oracle Corporation, it enables users to run multiple operating systems on a single physical machine through virtual machines (VMs). This feature-rich platform supports a wide range of guest operating systems, including Windows, Linux distributions, Solaris, and macOS (when used on Apple hardware). A key advantage of VirtualBox is its cost-effectiveness; being open source, it's available for free, making it an attractive option for those with budget constraints or for educational purposes. While it may not match the advanced performance features of VMware Workstation, such as 3D graphics support and high-resolution display compatibility, VirtualBox is often lauded for its simplicity and user-friendly interface, making it accessible to both novices and experienced users alike.

In comparison to VMware Workstation, which is a commercial product with robust integration with VMware vSphere for enterprise environments, VirtualBox offers a more straightforward approach without direct integration with specific enterprise virtualization management solutions. This difference positions VirtualBox as a popular choice for personal use, testing, development, and educational settings, where its ease of use and flexibility are highly valued. It provides the same core benefits of virtualization technology, such as secure and isolated environments for VMs, efficient resource allocation, and simplified backup and recovery processes. Users can allocate different

resources to each VM based on specific requirements, ensuring a tailored and efficient virtual environment. VirtualBox's ability to isolate vulnerabilities within individual VMs enhances overall system security, and its backup and recovery features simplify data management. Additionally, its role in efficient IT resource management, with quick provisioning of new resources and easy workload migration, adds to its appeal. Overall, VirtualBox is an excellent choice for users seeking a free, open-source virtualization solution that balances functionality, ease of use, and broad operating system support.

Virtualization with VirtualBox products and services has a number of advantages.

1. Cost-Effectiveness:

- As an open-source solution, VirtualBox is free to use, which is a significant advantage for individuals, small businesses, or educational institutions operating on limited budgets.

2. Cross-Platform Compatibility:

- VirtualBox runs on various host operating systems, including Windows, macOS, Linux, and Solaris, making it highly versatile for different user environments.

3. Ease of Use:

- Known for its user-friendly interface, VirtualBox is suitable for those who may not have extensive technical expertise in virtualization.

4. Good Performance for Standard Needs:

- Offers sufficient performance for most standard applications, making it suitable for development, testing, and personal use.

5. Flexibility in Guest OS Support:

- Supports a broad range of guest operating systems, which is beneficial for users who need to run different OS environments.

6. Snapshot and Cloning Features:

- VirtualBox provides easy-to-use snapshot and cloning features, allowing users to save the state of a VM and clone VMs for quick duplication.

Potential Downsides of VirtualBox

1. Performance Limitations:

- While adequate for many tasks, VirtualBox may not match the high-end performance capabilities of VMware, particularly for resource-intensive applications.

2. Limited Enterprise Features:

- Lacks some of the advanced enterprise-focused features found in Virtualbox, such as extensive integration with enterprise management tools.

3. Community-Based Support:

- Being open-source, support mainly comes from community forums and documentation, which might not be sufficient for enterprise-level requirements.

4. Less Optimal for Large-Scale Deployments:

- While capable, VirtualBox may not be the ideal solution for large-scale, enterprise-grade virtualization deployments.

5. No Direct Integration with Major Cloud Providers:

- Unlike VMware, VirtualBox does not have direct integration with major cloud providers for hybrid cloud environments.

Information about the Linux distributions used.

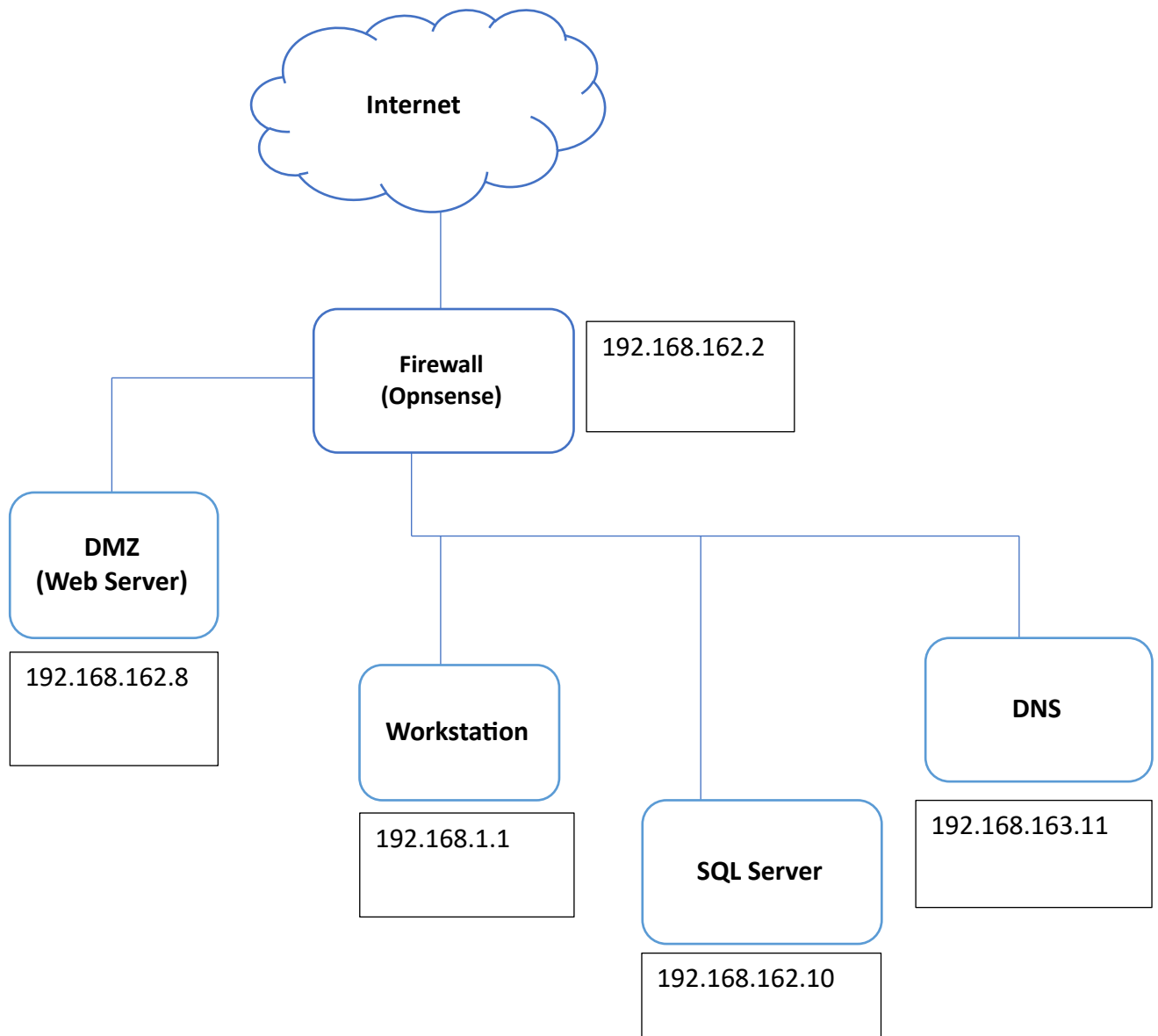
Canonical Ltd. created Ubuntu, a popular and extensively used Linux distribution. It is based on the Debian Linux distribution and comes in desktop and server flavors. It is well-known for its ease of use, regular releases, and extensive software package selection. Ubuntu is a prominent Linux distribution for servers and desktop computers that serves as the foundation for many other distributions. It is also the most popular cloud computing operating system, with Amazon Web Services offering a version of Ubuntu as one of its official images.

There are numerous Linux distributions to pick from, and which one is best suited for your project will be determined by your particular needs. The first thing you should consider is your computer skills. Your options will be more limited if you have never used Linux or the command line. Another obvious consideration is if you intend to use your Linux OS to execute apps on a desktop or as a server.

Some server distributions, which include a number of pre-bundled services, are suitable for specific applications. CentOS is a great example because it includes a lot of what you need to run a whole server right out of the box. You can even begin with a desktop installation and then add Linux operating system components as needed. If so, consider Debian or Ubuntu. So, I'm using Ubuntu 18.04 for Workstation, DNS, DMZ (Web Server), and SQL Server for this project.

Project Installation, Configuration and Testing

This document describes how I set up my Project infrastructure. Use it as a guide to determine what you should configure. My configuration might not work with your VMs, so you'll have to modify it for them.



Setting done in Virtual Network Editor before beginning of project

My first step was to install Virtualbox workstation pro. My second step was to create a virtual machine. I have created a four Ubuntu machine with 1GB RAM and 20GB hard disk. My third step was to install the necessary software and tools. I have installed tools that are necessary for development. My fourth step was to configure the virtual machine. I have configured the virtual machine to use the host only network adapters.

The screenshot shows the Virtual Network Editor interface. At the top, there are three tabs: 'Host-only Networks', 'NAT Networks', and 'Cloud Networks'. The 'Host-only Networks' tab is selected, displaying a table with the following data:

Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
VirtualBox Host-Only Ethernet ...	169.254.134.45/16		Enabled
VirtualBox Host-Only Ethernet ...	192.168.100.1/24		Enabled
VirtualBox Host-Only Ethernet ...	192.168.162.1/24		Enabled

Below the table, there is a section for configuring the network adapter. It has two tabs: 'Adapter' and 'DHCP Server'. The 'Adapter' tab is selected, showing the following configuration options:

- ☒ Configure Adapter Automatically
- ☐ Configure Adapter Manually
- IPv4 Address: 192.168.162.1
- IPv4 Network Mask: 255.255.255.0
- IPv6 Address: fe80::f8c9:585c:25f6:7003
- IPv6 Prefix Length: 64

At the bottom right, there are 'Apply' and 'Reset' buttons.

Once the VMs were created, you would need to configure the networking for each VM. This would involve setting up the IP addresses, subnet masks, default gateways, and DNS servers. You would also need to configure the firewall (Pfsense) to allow traffic between the internal network and the DMZ. Additionally, you would need to configure the DNS server to resolve hostnames for the VMs in the internal network and the DMZ.

Firewall (OPNsense)

Internet adapter (WAN): This adapter is connected to the internet and is responsible for providing internet access to the network.

OPNsense Console Setup

Initial Configuration: When OPNsense boots up for the first time, it presents a console-based setup wizard. This text-based interface allows you to assign network interfaces to their respective roles within your network infrastructure.

For instance, you designate **em0** as your WAN interface, which connects to the broader internet, and **em1** and **em2** for your LAN and DMZ, respectively. This initial setup is critical for establishing the foundational network pathways on which your firewall and other network security measures will operate.

```
You can now access the web GUI by opening
the following URL in your web browser:

https://192.168.162.2

*** OPNsense.localdomain: OPNsense 23.7 ***

LAN (em1)      -> v4: 192.168.162.2/24
WAN (em0)      -> v4/DHCP4: 192.168.1.168/24

HTTPS: SHA256 DC 36 8E 91 75 25 16 F1 54 65 27 BB 90 64 22 98
        0F 9A 92 BB 44 FA F0 0E 28 F1 F2 B0 08 C4 0E 75
SSH:   SHA256 4TvcEPkP6b0EqtguYiZzgYfIVV4Bc4md1LQ2d0Yyr/4 (ECDSA)
SSH:   SHA256 ow+McpD7UmGTok6rDb710LTerhRDWP7nUrpGD9ZbjLI (ED25519)
SSH:   SHA256 aQ2BQ5xQ80bCbZqTa8yyxat6tvLBb73HyT4TrKYKbuM (RSA)

0) Logout                      7) Ping host
1) Assign interfaces           8) Shell
2) Set interface IP address    9) pfTop
3) Reset the root password     10) Firewall log
4) Reset to factory defaults   11) Reload all services
5) Power off system            12) Update from console
6) Reboot system               13) Restore a backup

Enter an option: █
```

Assign Interfaces:

- From the console menu, choose the option to 'Assign interfaces' (usually option 1).
- Assign your WAN, LAN (LAN10), and OPT1 (LAN11) interfaces as required. For example, **em0** could be your WAN, **em1** your LAN (192.168.162.2/24), and **em2** your OPT1 (192.168.162.1/24 for the DMZ).

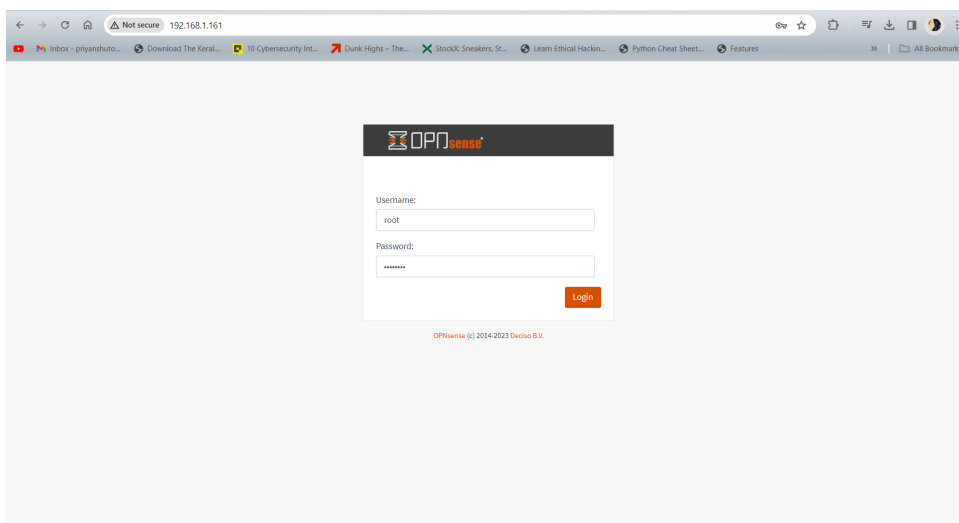
Set Interface IP Addresses:

- Next, select 'Set interface IP address' to assign IP addresses to your LAN and OPT1 interfaces.
- For LAN (em1), you might set **192.168.162.2/24**, and for OPT1 (em2), **192.168.162.1/24**, ensuring that DHCP is disabled if it's not needed.

Access the Web GUI:

- To configure the pfSense first I have assigned an interfaces by using option 1 as you can see in diagram above and then I have choose option 2 to set an interfaces ip address to 192.168.162.2/24 to LAN10 -> em1 and 192.168.162.1/24 to LAN11 -> em2 and did not allowed DHCP while setting the interfaces and enabling GUI version by typing saying yes to it during the setup of ip address. So once you are done setting up the interface and ip address go to your web browser and type ip address 192.168.163.1 to view the GUI version of the firewall as shown below.
- Once the interface assignment and IP address setup are complete, access the web GUI by typing the LAN IP address into a web browser on a computer connected to the LAN network.

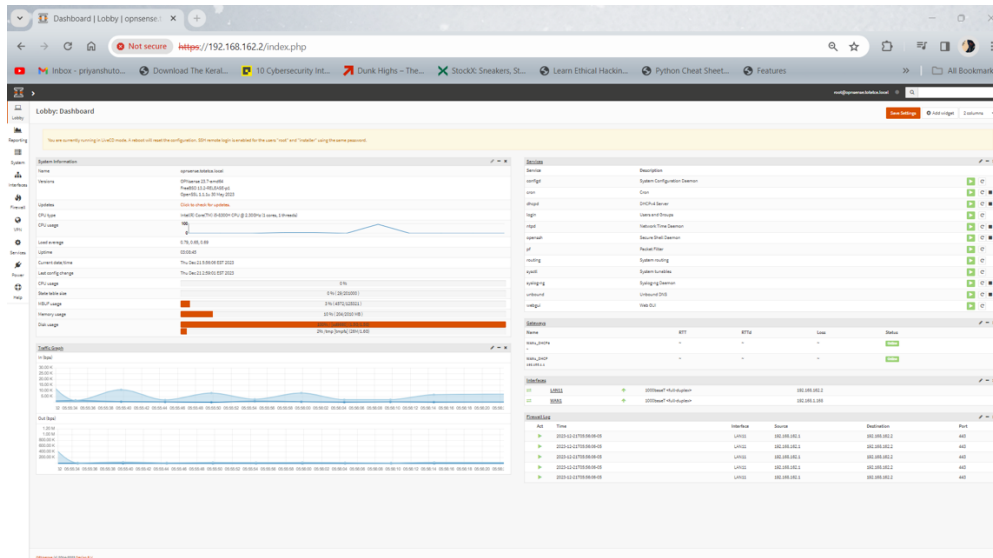
- On the login page, you will need to enter the default username "admin" and password "pfsense" (without the quotes). Once you have entered these credentials, click the "Login" button to log in to your pfsense firewall.
- If this is the first time you are logging in to your pfsense firewall, you will be prompted to set up the firewall. Follow the steps provided to set up your firewall. This may include configuring your network settings, setting up firewall rules, and configuring other settings such as VPN and traffic shaping.



Follow the steps below to configure interfaces using GUI version of firewall:

Upon accessing the OPNsense Web GUI:

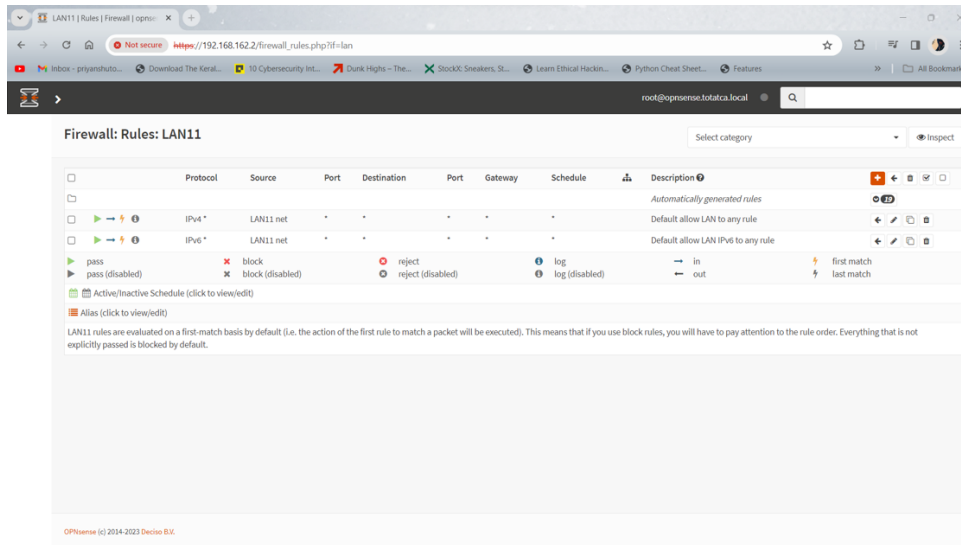
1. **Login:**
 - Use the default credentials (**admin / opnsense**) to log in.
2. **Initial Setup Wizard:**
 - If it's the first time logging in, you might be prompted to start the initial setup wizard. Follow the steps to configure the firewall's general settings.



Follow the following steps to configure the firewall rules:

Firewall Rules:

- In the OPNsense web interface, navigate to **Firewall > Rules**.
- Here you can add firewall rules for each interface by clicking on the interface name (e.g., LAN, LAN11) and using the 'Add' button.
- Define the rules with actions (Pass/Block/Reject), protocols, source, and destination addresses, and any other specific settings such as logging.



Follow the following steps to configure the port forwarding rules:

Port Forwarding:

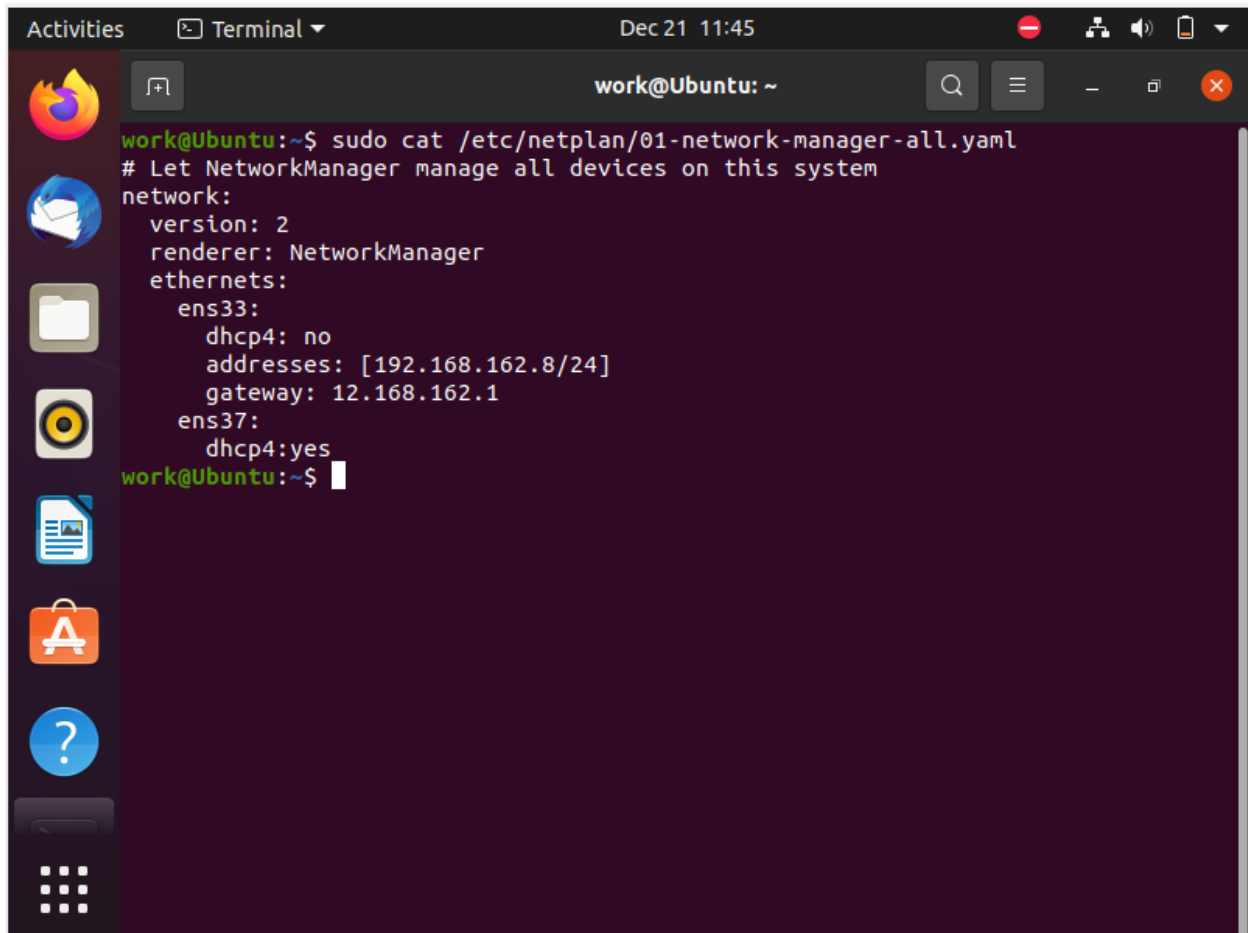
- To set up NAT port forwarding, go to Firewall > NAT > Port Forward.
- Click on the 'Add' button to create new rules for incoming connections on the WAN interface that need to be forwarded to an internal IP address.
- Specify the interface (WAN), protocol, source, destination, and the redirect target IP and port.

Testing & Verification:

- After setting up your rules and NAT port forwards, test to ensure everything is working as expected. Attempt to access the DMZ and LAN from the appropriate zones, and check connectivity from the WAN as well.
- Remember to save and apply changes after each configuration step. Once completed, you should have a functional OPNsense firewall setup with interfaces and firewall rules.

DMZ (Web Server)

Connect the NAT network adapter to the internet, either through a router or directly to a modem. This will allow the Web Server to access the internet and be accessible from the internet. Connect the network adapter to an internal network, such as a DMZ (Demilitarized Zone). This network should be separate from your main internal network and should only be used for hosting Web Server resources. Configure the Web Server to listen on the internal IP address assigned to the network adapter. This will ensure that the Web Server is only accessible from the internal network and is not directly accessible from the internet. Below diagram shows network configuration for DMZ

A screenshot of a terminal window in Ubuntu. The window title is "Terminal" and the date/time is "Dec 21 11:45". The terminal shows the command `sudo cat /etc/netplan/01-network-manager-all.yaml` being executed. The output is a YAML configuration file for NetworkManager. The configuration sets the version to 2, the renderer to NetworkManager, and defines two ethernet interfaces: ens33 and ens37. ens33 is configured with dhcp4: no, addresses: [192.168.162.8/24], and gateway: 12.168.162.1. ens37 is configured with dhcp4: yes. The terminal prompt is `work@Ubuntu:~$`.

```
work@Ubuntu:~$ sudo cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    ens33:
      dhcp4: no
      addresses: [192.168.162.8/24]
      gateway: 12.168.162.1
    ens37:
      dhcp4: yes
work@Ubuntu:~$
```


Set up firewall rules and NAT port forwarding rules to allow external devices to access the Web Server. For example, you might set up a NAT rule that forwards incoming HTTP (port 80) and HTTPS (port 443) traffic from the internet to the internal IP address of the Web Server.

Configure the Web Server to serve content to external devices. This may involve setting up virtual hosts, configuring directory permissions, and other tasks depending on the specific Web Server software you are using. Below steps are given for installing, configuring, and testing web server .

To configure Apache2 on a Debian-based system, you can follow these steps:

- `sudo apt update`
- `sudo apt install apache2`
- `sudo ufw enable`
- `sudo systemctl start apache2`
- `sudo ufw app list`
- `sudo ufw allow 'Apache'`
- `sudo ufw status`
- `sudo systemctl status apache2`
- `hostname -l`
- `curl -4 icanhazip.com`

Test your Apache2 installation by opening a web browser and going to <http://localhost> (192.168.162.8). You should see the default Apache2 page.

To configure your Apache2 server, you can edit the configuration files located in

the `/etc/apache2` directory. The main configuration file is `/etc/apache2/apache2.conf`, and you can create additional configuration files in the `/etc/apache2/conf-available` and `/etc/apache2/conf-enabled` directories.


```
Activities Terminal Dec 21 11:48
work@Ubuntu: ~
version display version information

Application profile commands:
app list list application profiles
app info PROFILE show information on PROFILE
app update PROFILE update PROFILE
app default ARG set default application policy

work@Ubuntu:~$ sudo ufw app list
Available applications:
Apache
Apache Full
Apache Secure
Bind9
CUPS
work@Ubuntu:~$ sudo ufw allow 'Apache'
Rule added
Rule added (v6)
work@Ubuntu:~$ sudo ufw status
Status: active

To Action From
--
53 ALLOW Anywhere
Apache ALLOW Anywhere
53 (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)

work@Ubuntu:~$
```



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.Load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

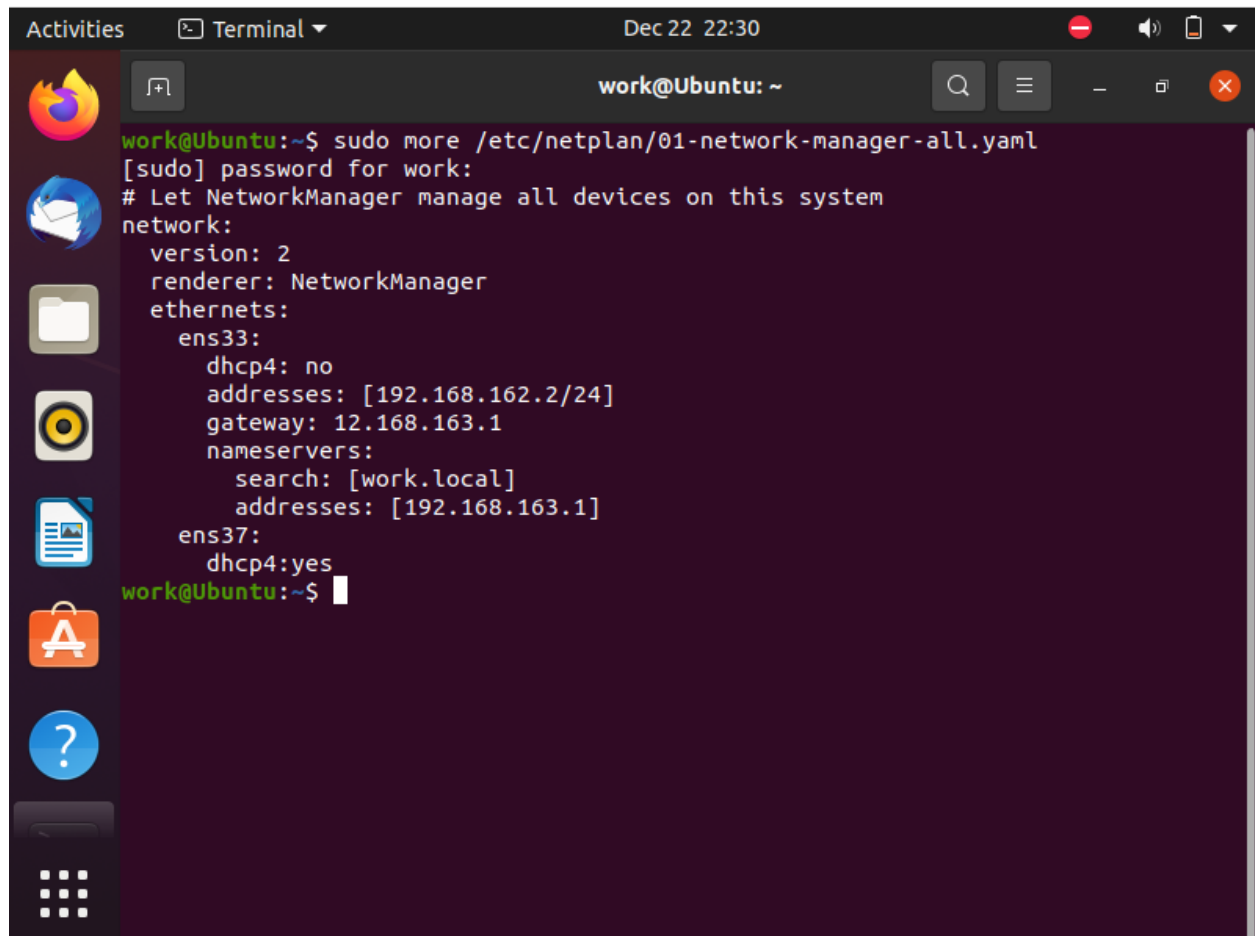
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

DNS Server

Configure the NAT adapter to connect to the internet. This will allow you to download any necessary updates or packages for the DNS server.

Configure the internal network adapter (VMnet10) with a static IP address. This will be the IP address that other devices on your network will use to access the DNS server.

Configure the DNS server to listen on the internal network adapter (VMnet10). This will allow devices on your internal network to access the DNS server. Below you can see the network configuration that i have done for DNS Server.

A screenshot of a terminal window in Ubuntu. The window title is "Terminal" and the date/time is "Dec 22 22:30". The user is "work@Ubuntu". The terminal shows the command "sudo more /etc/netplan/01-network-manager-all.yaml" being executed. The output shows the configuration for the ens33 interface, including static IP addresses, gateway, and nameservers.

```
work@Ubuntu:~$ sudo more /etc/netplan/01-network-manager-all.yaml
[sudo] password for work:
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.162.2/24]
      gateway: 12.168.163.1
      nameservers:
        search: [work.local]
        addresses: [192.168.163.1]
    ens37:
      dhcp4: yes
work@Ubuntu:~$
```

Once the network adapter is set and network configuration is done, now to configure the DNS server do the following steps shown below.

Install DNS Server

Update the package list and install any available updates, Install the DNS server software (Bind9),

- `sudo apt install -y bind9 bind9utils bind9-doc dnsutils`

Configure the DNS server to listen on the correct interface. Open the options file:

The `/etc/bind/` directory is the main configuration directory of the DNS server, and it holds configuration files and zone lookup files. Global configuration file is `/etc/bind/named.conf`. You should not use this file for your local DNS zone rather you can use `/etc/bind/named.conf.local` file.

Use the following code to open the options file!

- `sudo cat /etc/bind/named.conf.options`

Add a line to specify the IP address of the interface that the DNS server will listen on (e.g. "192.168.162.11"). The file should look something like this:

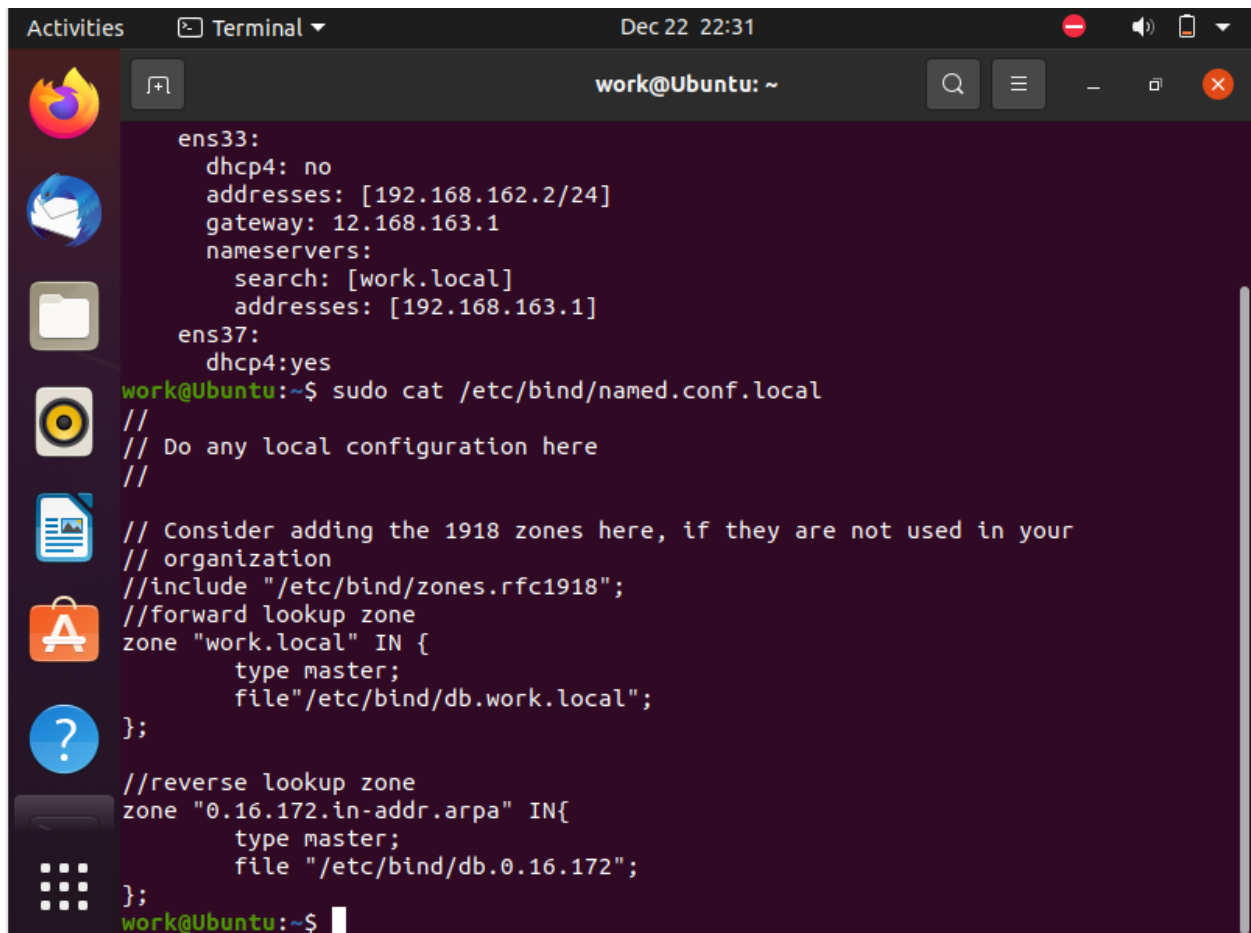
```
options {  
    directory "/var/cache/bind";
```

```
listen-on { 192.168.162.11; };
```

```
// other options };
```

Create a forward lookup zone for your domain by editing the named configuration file:

- `sudo cat /etc/bind/named.conf.local`

A terminal window titled 'Terminal' with a timestamp of 'Dec 22 22:31'. The window shows the output of a command to view DHCP configuration for network interfaces. The output shows details for 'ens33' and 'ens37', including DHCP status, IP addresses, gateway, and nameservers. Below this, the user runs 'sudo cat /etc/bind/named.conf.local', displaying the contents of the file which includes configuration for a forward lookup zone 'work.local' and a reverse lookup zone '0.16.172.in-addr.arpa'.

```
work@Ubuntu: ~  
ens33:  
  dhcp4: no  
  addresses: [192.168.162.2/24]  
  gateway: 12.168.163.1  
  nameservers:  
    search: [work.local]  
    addresses: [192.168.163.1]  
ens37:  
  dhcp4:yes  
work@Ubuntu:~$ sudo cat /etc/bind/named.conf.local  
//  
// Do any local configuration here  
//  
  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
//forward lookup zone  
zone "work.local" IN {  
    type master;  
    file "/etc/bind/db.work.local";  
};  
  
//reverse lookup zone  
zone "0.16.172.in-addr.arpa" IN{  
    type master;  
    file "/etc/bind/db.0.16.172";  
};  
work@Ubuntu:~$
```

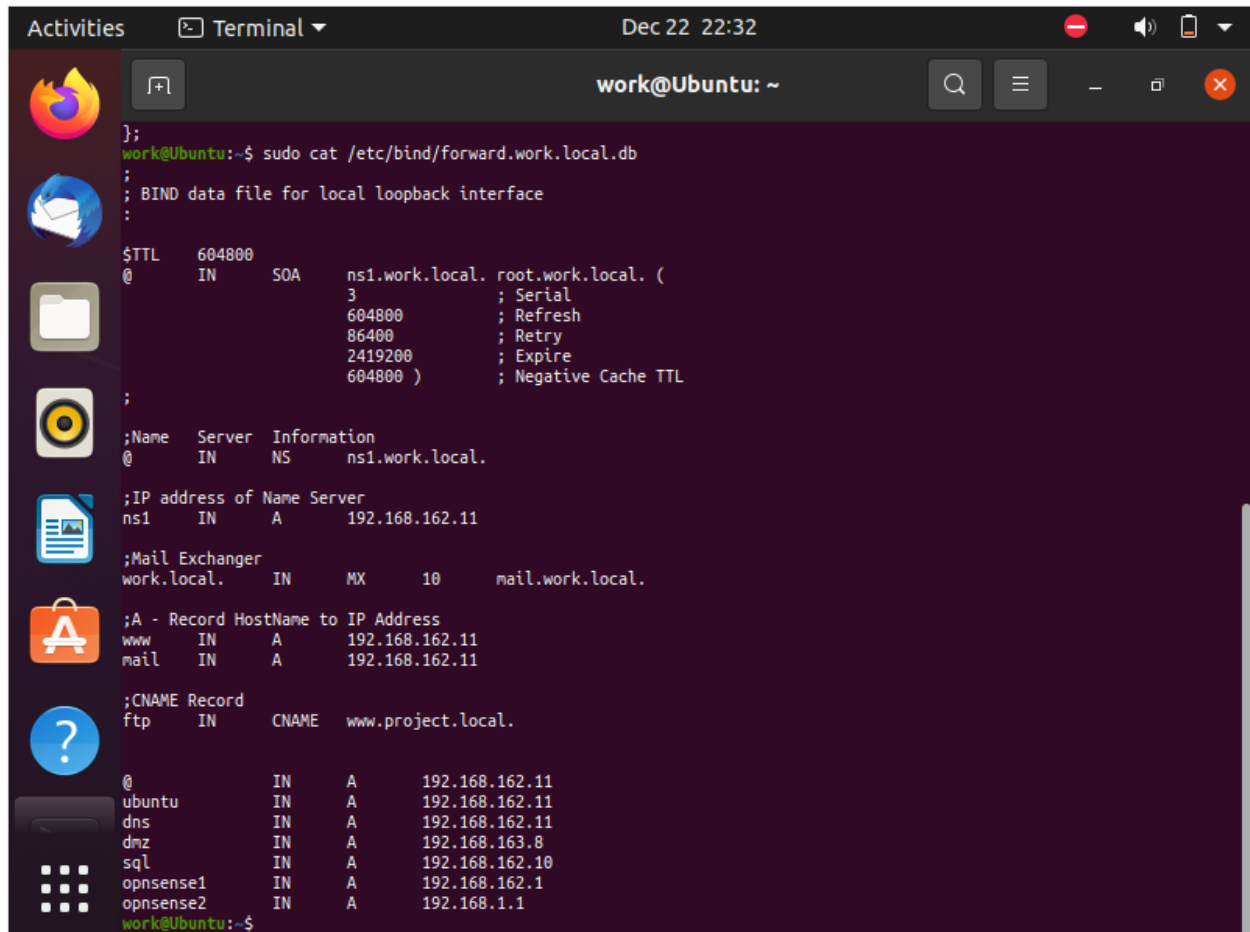
Once you create zones, you can go ahead and create zone data files that hold DNS records for the forward zone and reverse zone.

Forward Zone lookup file

Copy the sample entries to a zone file called forward.work.local.db for the forward zone under /etc/bind directory.

- `sudo cat/etc/bind/forward.work.local.db`

Use the below configuration for forward zone for reference



```
};
work@Ubuntu:~$ sudo cat /etc/bind/forward.work.local.db
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns1.work.local. root.work.local. (
                        3      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200; Expire
                        604800 )      ; Negative Cache TTL
;
;Name      Server      Information
@          IN          NS       ns1.work.local.
;IP address of Name Server
ns1        IN          A        192.168.162.11
;Mail Exchanger
work.local. IN          MX        10      mail.work.local.
;A - Record HostName to IP Address
www        IN          A        192.168.162.11
mail       IN          A        192.168.162.11
;CNAME Record
ftp        IN          CNAME     www.project.local.
@          IN          A        192.168.162.11
ubuntu    IN          A        192.168.162.11
dns       IN          A        192.168.162.11
dmz       IN          A        192.168.163.8
sql       IN          A        192.168.162.10
opnsense1 IN          A        192.168.162.1
opnsense2 IN          A        192.168.1.1
work@Ubuntu:~$
```

Reverse Zone lookup file

Copy the sample entries to the zone file called reverse.work.local.db for the reverse zone under the /etc/bind directory and create reverse pointers for the above forward zone records.

- `sudo cat /etc/bind/reverse.work.local.db`

Use the below configuration for reverse zone for reference

```
work@Ubuntu:~$ sudo cat /etc/bind/reverse.work.local.db
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      work.local. root.work.local. (
        3             ; Serial
        604800        ; Refresh
        86400         ; Retry
        2419200       ; Expire
        604800 )      ; Negative Cache TTL
;
;@         IN      NS       localhost.
;1.0.0     IN      PTR      localhost.
@         IN      AAAA     ::1
;Name Server Information
@         IN      NS       ns1.work.local.

;Reverse lookup for Name Server
11        IN      PTR      ns1.work.local.

;PTR Record IP address to HostName
11        IN      PTR      www.work.local.
11        IN      PTR      mail.work.local.

@         IN      PTR      work.local
@         IN      A        192.168.162.11
ubuntu    IN      A        192.168.162.11
dns       IN      A        192.168.162.11
dmz       IN      A        192.168.163.8
sql       IN      A        192.168.162.10
opnsense1 IN      A        192.168.1.1
opnsense2 IN      A        192.168.1.1
2         IN      PTR      opnsense1.work.local
2         IN      PTR      opnsense2.work.local
10        IN      PTR      sql.work.local
11        IN      PTR      dns.work.local
work@Ubuntu:~$
```

Check BIND Configuration Syntax

To check the syntax and named.conf* files for any errors, also you can use named-checkzone to check the syntax errors in zone files by executing the following command and to check the syntax of a specific zone file use the following commands:

- `sudo named-checkconf`
- `sudo named-checkzone work.local /etc/bind/forward.work.local.db`
- `Named-checkzone 162.168.192.in-addr.arpa`

`etc/bind/reverse.work.local.db`

Restart bind service now to apply the configured configuration by executing the following commands.

- `sudo systemctl restart bind9`
- `sudo systemctl enable bind9`

- `sudo systemctl status bind9`

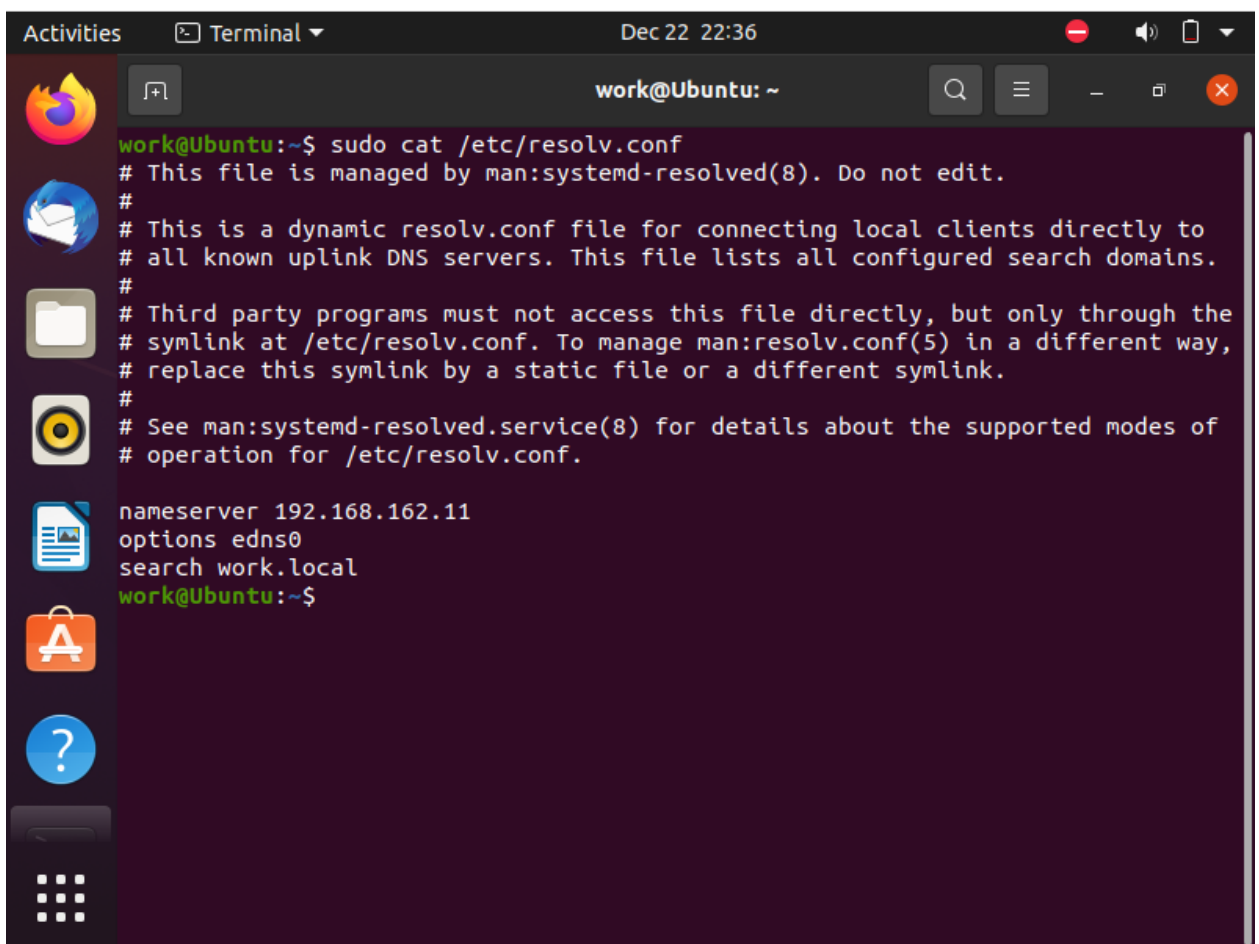
DNS Record Update

- To reload the forward and reverse zone do the following command ###
Forward Zone ###
- • `sudo rndc reload work.local` ### Reverse Zone ###
- • `sudo rndc reload 162.168.192.in-addr.arpa`

Verify DNS Server

Go to any client machine and add our new DNS server IP Address in `/etc/resolv.conf` file by following the below commands.

- • `sudo cat /etc/resolv.conf`
Make an entry like below shown in figure.



The screenshot shows a terminal window titled "work@Ubuntu: ~" with a dark background. The terminal displays the output of the command `sudo cat /etc/resolv.conf`. The output consists of several lines of comments and configuration entries. The comments explain that the file is managed by `man:systemd-resolved(8)` and is a dynamic file for connecting local clients to upstream DNS servers. The configuration entries include `nameserver 192.168.162.11`, `options eds0`, and `search work.local`. The terminal prompt `work@Ubuntu:~$` is visible at the bottom.

```
work@Ubuntu:~$ sudo cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.162.11
options eds0
search work.local
work@Ubuntu:~$
```


Use the dig command to check the forward and reverse zone.

- \$ dig www.work.local

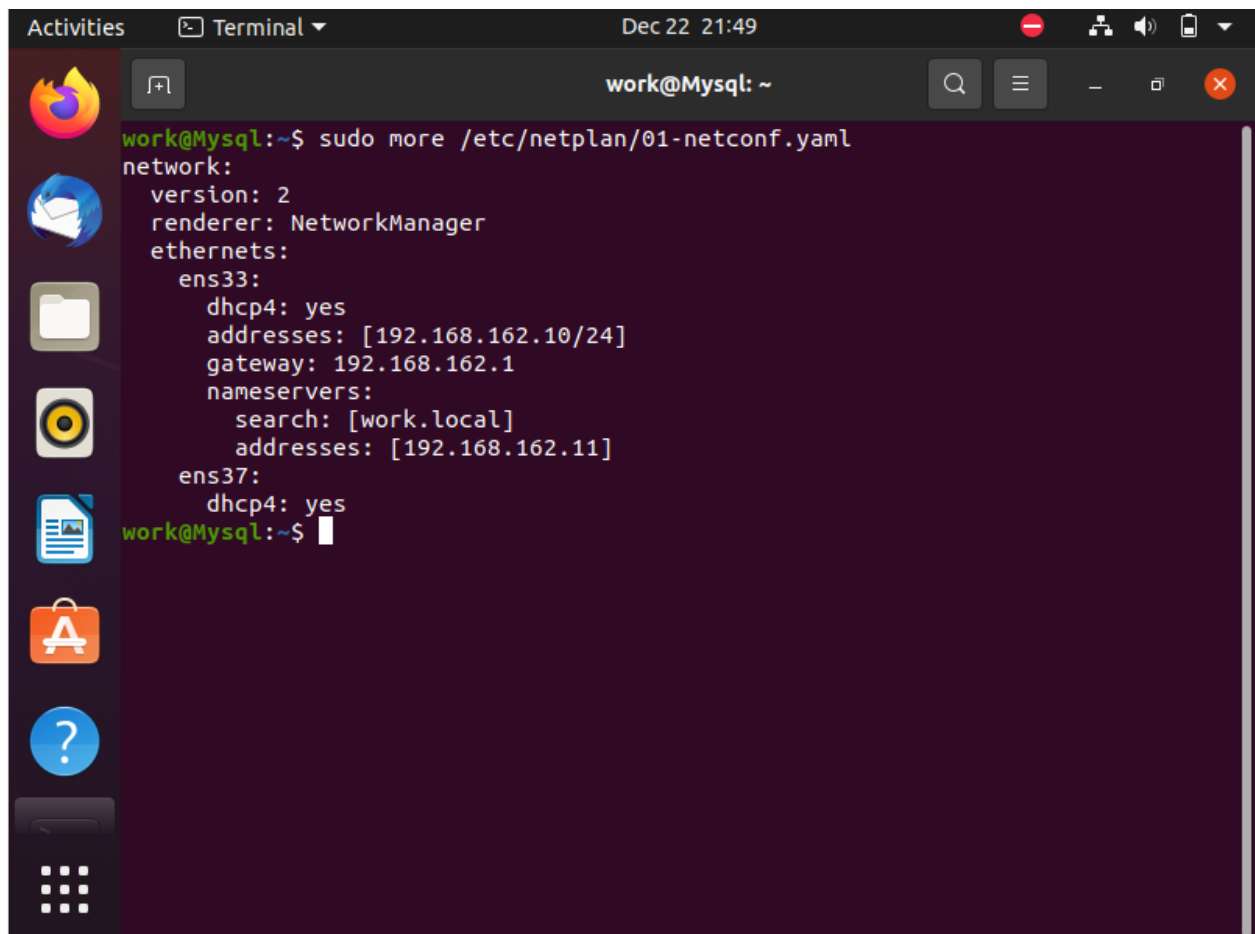
```
Activities Terminal Dec 22 23:11
work@Ubuntu: ~
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:35840
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 8661d85ed11dbaa3a187c9af63a3cde4ad6cf649e35a2b1d (good)
;; QUESTION SECTION:
www.work.local.                IN      A
;; ANSWER SECTION:
www.work.local.                IN      A
;; AUTHORITY SECTION:
www.work.local.                604800  IN      A      192.168.162.11
;; AUTHORITY SECTION:
work.local.                    604800  IN      NS      ns1.work.local.
;; ADDITIONAL SECTION:
ns1.work.local.                604800  IN      A      192.168.162.11
;; Query time: 0 msec
;; SERVER: 192.168.162.11#53(192.168.162.11)
;; WHEN: Wed Dec 21 09:37:20 EST 2023
;; MSG SIZE rcvd: 127
work@Ubuntu:~$
```

\$ dig -x 192.168.162.11

```
Activities Terminal Dec 22 23:09
work@Ubuntu: ~
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:74529
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL: 2
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: a8549a7b5da55448b4fb2f2c53b3cdfd21ba914f5c7d91b2 (good)
;; QUESTION SECTION:
11.162.168.192.in-addr-arpa.   IN      PTR
;; ANSWER SECTION:
11.162.168.192.in-addr-arpa.   604800  IN      PTR      www.work.local.
11.162.168.192.in-addr-arpa.   604800  IN      PTR      mail.work.local.
11.162.168.192.in-addr-arpa.   604800  IN      PTR      dns.work.local.162.168.
192.in-addr.arpa.
11.162.168.192.in-addr-arpa.   604800  IN      PTR      ns1.work.local.
;; AUTHORITY SECTION:
162.168.192.in-addr-arpa.      604800  IN      NS      ns1.work.local.
;; ADDITIONAL SECTION:
ns1.work.local.                604800  IN      A      192.168.162.11
;; Query time: 0 msec
;; SERVER: 192.168.162.11#53(192.168.162.11)
;; WHEN: Wed Dec 21 09:24:45 EST 2023
;; MSG SIZE rcvd: 246
work@Ubuntu:~$
```

My SQL Server

For MYSQL Server network configuration (i have use 2 network adapters where first one is used for internet which is NAT adapter which will help use in beginning during our configuration phase and the other one is internal network adapter which is VMnet10) below you can see the network configuration that i have done for MYSQL Server below.

A screenshot of a Linux terminal window. The window title is "work@Mysql: ~". The terminal shows the command "sudo more /etc/netplan/01-netconf.yaml" being executed. The output displays the network configuration for two interfaces: ens33 and ens37. The configuration for ens33 includes dhcp4: yes, addresses: [192.168.162.10/24], gateway: 192.168.162.1, and nameservers: search: [work.local], addresses: [192.168.162.11]. The configuration for ens37 includes dhcp4: yes. The terminal prompt is "work@Mysql:~\$".

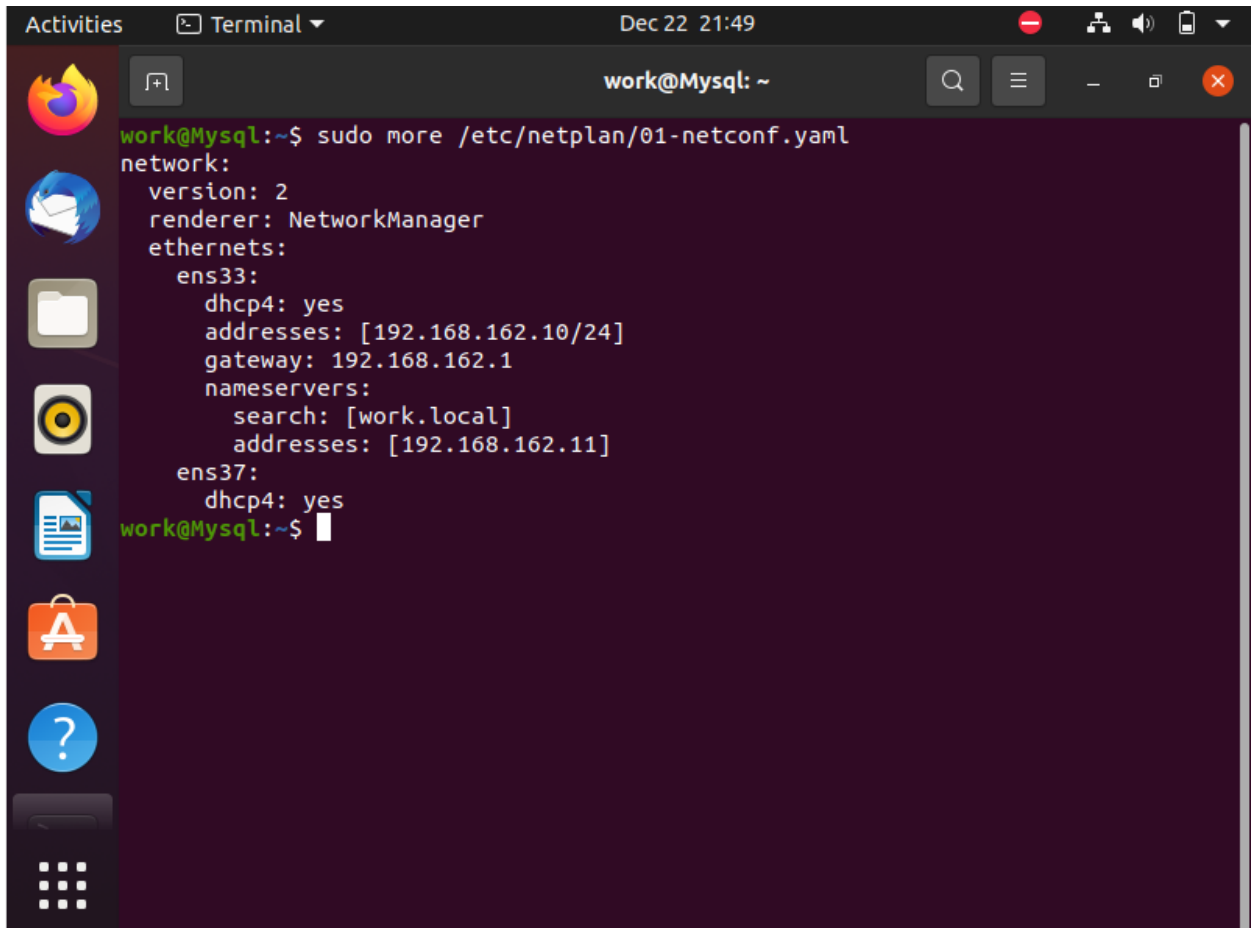
```
work@Mysql:~$ sudo more /etc/netplan/01-netconf.yaml
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: yes
      addresses: [192.168.162.10/24]
      gateway: 192.168.162.1
      nameservers:
        search: [work.local]
        addresses: [192.168.162.11]
    ens37:
      dhcp4: yes
work@Mysql:~$
```

Installing MySQL

Update the package manager's package index: • `sudo apt update`

Install MySQL Server using the package manager: • `sudo apt install mysql-server`

Ensure that the server is running using the systemctl start command: • `sudo systemctl start mysql.service`

A screenshot of a Linux terminal window. The window title is "work@MySQL: ~". The terminal shows the command `sudo more /etc/netplan/01-netconf.yaml` being executed. The output displays the contents of the netplan configuration file, which includes network version, renderer, and two ethernet interfaces (ens33 and ens37) with their respective DHCP settings, IP addresses, gateway, and DNS servers. The terminal prompt is `work@MySQL:~$`.

```
work@MySQL:~$ sudo more /etc/netplan/01-netconf.yaml
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    ens33:
      dhcp4: yes
      addresses: [192.168.162.10/24]
      gateway: 192.168.162.1
      nameservers:
        search: [work.local]
        addresses: [192.168.162.11]
    ens37:
      dhcp4: yes
work@MySQL:~$
```

After the installation is complete, run the MySQL security script to improve the security of the default MySQL installation:

- `sudo mysql_secure_installation`

The script will prompt you to enter the MySQL root password and ask you to answer a few questions to configure the security settings.

Once the security script is finished, you can start the MySQL service and enable it to start automatically on boot:

- `sudo systemctl start mysql`
- `sudo systemctl enable mysql`

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-12-21 11:36:29 EST; 1min 46s ago
   Process: 5174 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=
 Main PID: 5182 (mysqld)
    Status: "Server is operational"
   Tasks: 37 (limit: 2261)
  Memory: 365.0M
   CGroup: /system.slice/mysql.service
           └─5182 /usr/sbin/mysqld

work@Ubuntu:~$
```

To test the installation, you can log in to the MySQL shell as the root user:

- `sudo mysql`
- `> SELECT user,authentication_string,plugin,host FROM mysql.user;`

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
+-----+-----+-----+-----+
| user                | authentication_string | plugin               | host |
+-----+-----+-----+-----+
| debian-sys-maint    | $A$005$3+sb7R-hwpV80hk0bWROLO1fB1DQmfjLrQP5uGjK0vp5mCKZj.cHk3 | caching_sha2_password | localhost |
| mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| root                | *2478C0C06DEE42FD16188B99005ADCAZEC9D1E19 | mysql_native_password | localhost |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

The MySQL shell will prompt you to enter the root password. Once you are logged in, you can run SQL commands to create databases, tables, and users, and perform other tasks.

In this example, you can see that the root user does in fact authenticate using the `auth_socket` plugin. To configure the root account to authenticate with a password, run the following `ALTER USER` command. Be sure to change password

to a strong password of your choosing, and note that this command will change the root password you set.

- > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';

Then, run FLUSH PRIVILEGES which tells the server to reload the grant tables and put your new changes into effect:

- > FLUSH PRIVILEGES;

Check the authentication methods employed by each of your users again to confirm that root no longer authenticates using the auth_socket plugin:

- • > SELECT user,authentication_string,plugin,host FROM mysql.user;

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

user	authentication_string	plugin	host
debian-sys-maint	\$A\$00\$5\$3+sb7R-hwpV80hk0bmROLO1fB1DQmfJLrQP5uGjJKvpSmCkZj.chk3	caching_sha2_password	localhost
mysql.infoschema	\$A\$00\$5\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED	caching_sha2_password	localhost
mysql.session	\$A\$00\$5\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED	caching_sha2_password	localhost
mysql.sys	\$A\$00\$5\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED	caching_sha2_password	localhost
root	*2470C0C06DEE42FD1618B899005ADCA2EC9D1E19	mysql_native_password	localhost

5 rows in set (0.00 sec)

```
mysql>
```

- • >exit
- • Sudo mysql -u root -p

Now create a strong password for a new user by following these commands

- • CREATE USER 'project'@'localhost' IDENTIFIED BY 'password';
- • GRANT ALL PRIVILEGES ON *.* TO 'project'@'localhost' WITH GRANT

OPTION;

- • exit

Testing MySQL, Regardless of how you installed it, MySQL should have started running automatically. To test this, check its status by running those below commands.

- • systemctl status mysql.service
- • sudo mysqladmin -p -u root version

Now you now have a basic MySQL setup installed on your server.

Conclusion

A small business infrastructure with an internet connection, Install and configure a firewall with at least three network interfaces: one for Internet access, one for DMZ access, and one for internal network access. A web server is housed in the DMZ. Configure the router to redirect incoming Internet traffic to the DMZ and to restrict incoming DMZ traffic to the internal network. This will build a DMZ that will operate as a buffer between the Internet and the internal network, shielding it from external threats. Configure the web server to listen on the DMZ interface for incoming HTTP and HTTPS traffic and to pass requests to the internal network via the internal network interface. This allows the web server to access resources on the internal network while remaining secure from outside assaults. Configure firewall rules to allow only necessary traffic while blocking all others. The internal network includes the MySQL Server, DNS, and workstations. A firewall, which is a device or program that limits access to the internal network from external networks such as the Internet, often protects the internal network from the Internet. The firewall contributes to network security by denying or restricting access to unauthorized users and preventing unauthorized traffic from entering the network. Due to the setup of the firewall, the Web Server in the DMZ is only visible on the internal network. Only computers on the internal network have access to the workstation, DNS server, and MySQL server. Because the entire infrastructure was built with VMware, the firewall was handled by Pfsense, and DNS, SQL, the DMZ, and workstations were handled by Ubuntu. On the internal network, the DNS server resolves the IP address/hostname. In addition, the infrastructure was tested to confirm that it was properly setup and running as planned.