

Lab 5a

Step 1:

Command: cat /etc/os-release

```
[kavvisna@kavvserver ~]$ cat /etc/os-release
NAME="Rocky Linux"
VERSION="9.6 (Blue Onyx)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.6"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Rocky Linux 9.6 (Blue Onyx)"
ANSI_COLOR="0;32"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:rocky:rocky:9::baseos"
HOME_URL="https://rockylinux.org/"
VENDOR_NAME="RESF"
VENDOR_URL="https://resf.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
SUPPORT_END="2032-05-31"
ROCKY_SUPPORT_PRODUCT="Rocky-Linux-9"
ROCKY_SUPPORT_PRODUCT_VERSION="9.6"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.6"
[kavvisna@kavvserver ~]$
```

Cat is used to display the content of a file.

/etc/os-release is a system file found in Linux distributions that contains information about the operating system. It's a plain text file with key-value pairs

Command: uname -a

```
[kavvisna@kavvserver ~]$ uname -a
Linux kavvserver.bungkus.org 5.14.0-570.26.1.el9_6.x86_64 #1 SMP PREEMPT_DYNAMIC Wed Jul 16 21:00:44 UTC
2025 x86_64 x86_64 x86_64 GNU/Linux
[kavvisna@kavvserver ~]$
```

uname: Stands for Unix Name. It prints system information.

-a: Stands for all. It tells uname to display all available system information.

Step 2:

Command: Sudo dnf update -y

```
[kavvisna@kavvserver ~]$ sudo dnf update -y  
[sudo] password for kavvisna:
```

Command: Sudo dnf upgrade -y

```
[kavvisna@kavvserver ~]$ sudo dnf upgrade -y  
Last metadata expiration check: 0:02:37 ago on Sun 27 Jul 2025 10:11:12 PM.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[kavvisna@kavvserver ~]$
```

DNF = Dandified YUM (modern package manager for Fedora, RHEL, etc.).

A DNF repository is a source (local or online) that contains software packages (.rpm files).

Used by dnf command to install, update, and manage software.

Repo info stored in: /etc/yum.repos.d/*.repo

Example command: dnf install firefox, dnf repolist

Step 3:

Command: reboot

```
[kavvisna@kavvserver ~]$ reboot
```

Step 4:

Command: sudo dnf install -y bind bind-utils (to install the DNS Bind9 server)

```
Running transaction  
Preparing : 1/1  
Installing : bind-dnssec-doc-32:9.16.23-29.el9_6.noarch 1/5  
Installing : python3-ply-3.11-14.el9.0.1.noarch 2/5  
Installing : python3-bind-32:9.16.23-29.el9_6.noarch 3/5  
Installing : bind-dnssec-utils-32:9.16.23-29.el9_6.x86_64 4/5  
Running scriptlet: bind-32:9.16.23-29.el9_6.x86_64 5/5  
Installing : bind-32:9.16.23-29.el9_6.x86_64 5/5  
Running scriptlet: bind-32:9.16.23-29.el9_6.x86_64 5/5  
Verifying : python3-ply-3.11-14.el9.0.1.noarch 1/5  
Verifying : python3-bind-32:9.16.23-29.el9_6.noarch 2/5  
Verifying : bind-32:9.16.23-29.el9_6.x86_64 3/5  
Verifying : bind-dnssec-utils-32:9.16.23-29.el9_6.x86_64 4/5  
Verifying : bind-dnssec-doc-32:9.16.23-29.el9_6.noarch 5/5  
  
Installed:  
bind-32:9.16.23-29.el9_6.x86_64 bind-dnssec-doc-32:9.16.23-29.el9_6.noarch  
bind-dnssec-utils-32:9.16.23-29.el9_6.x86_64 python3-bind-32:9.16.23-29.el9_6.noarch  
python3-ply-3.11-14.el9.0.1.noarch  
  
Complete!
```

sudo dnf install -y bind bind-utils is a command used to install the BIND DNS server and DNS troubleshooting tools (dig, nslookup, etc.) automatically without asking for confirmation, using the DNF package manager with superuser privileges.

Step 5:

Command: sudo systemctl enable named –now

```
[kavvisna@kavvserver ~]$ sudo systemctl enable named --now
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[kavvisna@kavvserver ~]$
```

sudo systemctl enable named --now enables the BIND DNS service (called named) to start automatically at boot and starts it immediately, using superuser privileges.

Command: sudo systemctl start named

```
[kavvisna@kavvserver ~]$ sudo systemctl start named
[kavvisna@kavvserver ~]$
```

sudo systemctl start named is used to start the BIND DNS service (named) immediately, using superuser privileges, without enabling it to run at boot.

Step 6:

Command: sudo systemctl status named

```
[kavvisna@kavvserver ~]$ sudo systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
  Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
  Active: active (running) since Sun 2025-07-27 22:24:03 +08; 3min 35s ago
    Process: 6459 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/nam>
    Process: 6462 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCE>
   Main PID: 6463 (named)
     Tasks: 6 (limit: 74031)
    Memory: 46.2M
      CPU: 103ms
     CGroup: /system.slice/named.service
             └─6463 /usr/sbin/named -u named -c /etc/named.conf

Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './NS/IN': 2001:500:1::5>
Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './DNSKEY/IN': 2001:7fe:>
Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './NS/IN': 2001:7fe::53#>
Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './DNSKEY/IN': 2001:500:>
Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './NS/IN': 2001:500:2d::>
Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './DNSKEY/IN': 2001:503:>
Jul 27 22:24:03 kavvserver.bungkus.org named[6463]: network unreachable resolving './NS/IN': 2001:503:c27:>
Jul 27 22:24:04 kavvserver.bungkus.org named[6463]: managed-keys-zone: Initializing automatic trust anchor>
Jul 27 22:24:04 kavvserver.bungkus.org named[6463]: managed-keys-zone: Initializing automatic trust anchor>
Jul 27 22:24:04 kavvserver.bungkus.org named[6463]: resolver priming query complete
lines 1-22/22 (END)
```

sudo systemctl status named checks and displays the status of the BIND DNS service (named), showing whether it is active, inactive, or failed, along with logs and process details.

Step 7:

Command: ll /etc/named.conf (ll is same as ls -l)

```
[kavvisna@kavvserver ~]$ ll /etc/named.conf
-rw-r-----. 1 root named 1722 Jun 25 10:45 /etc/named.conf
[kavvisna@kavvserver ~]$
```

ll /etc/named.conf lists the detailed information (like permissions, owner, size, and date modified) of the main configuration file for the BIND DNS server, located at /etc/named.conf.

Step 8:

Command: sudo cp -p /etc/named.conf /etc/named.conf.bak

(-p maintain the same permission and ownership of the original file)

```
[kavvisna@kavvserver ~]$ sudo cp -p /etc/named.conf /etc/named.conf.bak
[sudo] password for kavvisna:
[kavvisna@kavvserver ~]$
```

sudo cp -p /etc/named.conf /etc/named.conf.bak makes a backup copy of the named.conf file and saves it as named.conf.bak in the same directory, preserving the original file's permissions and timestamps, using superuser privileges.

Command: ls -l /etc (to list the content of the etc directory)

```
[kavvisna@kavvserver ~]$ ls -l /etc
total 1340
drwxr-xr-x. 3 root root      28 Jul  3 11:20 accountsservice
-rw-r--r--. 1 root root      16 Jul  3 11:38 adjtime
-rw-r--r--. 1 root root    1529 Jun 23 2020 aliases
drwxr-xr-x. 3 root root      65 Jul  3 12:01 alsa
drwxr-xr-x. 2 root root    4096 Apr 26 08:04 alternatives
-rw-r--r--. 1 root root      541 May  2 17:15 anacrontab
```

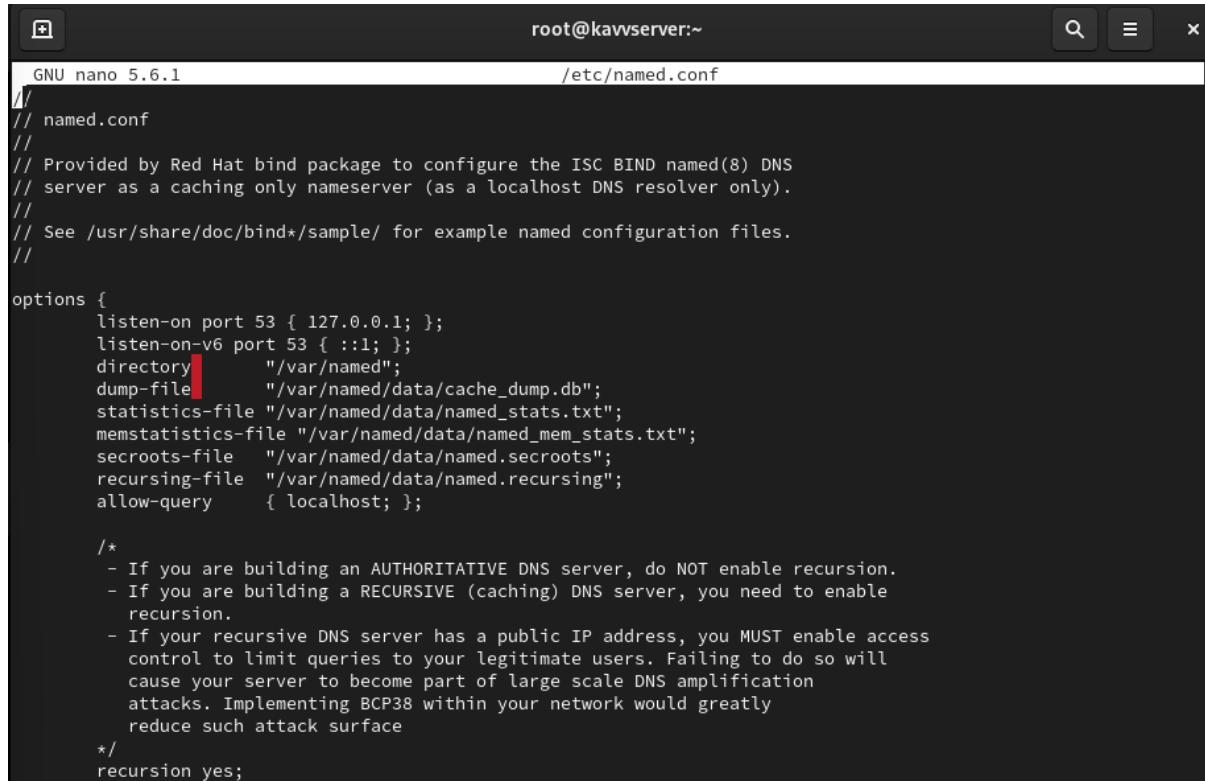
Step 9:

Command: su –

```
[kavvisna@kavvserver ~]$ su -  
Password:  
[root@kavvserver ~]#
```

Successfully logged in to root users.

Command: nano /etc/named.conf



```
GNU nano 5.6.1                                     root@kavvserver:~  
// named.conf  
//  
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS  
// server as a caching only nameserver (as a localhost DNS resolver only).  
//  
// See /usr/share/doc/bind*/sample/ for example named configuration files.  
  
options {  
    listen-on port 53 { 127.0.0.1; };  
    listen-on-v6 port 53 { ::1; };  
    directory    "/var/named";  
    dump-file    "/var/named/data/cache_dump.db";  
    statistics-file "/var/named/data/named_stats.txt";  
    memstatistics-file "/var/named/data/named_mem_stats.txt";  
    secroots-file  "/var/named/data/named.secroots";  
    recursing-file  "/var/named/data/named.reCURsing";  
    allow-query     { localhost; };  
  
    /*  
     * If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.  
     * If you are building a RECURSIVE (caching) DNS server, you need to enable  
     * recursion.  
     * If your recursive DNS server has a public IP address, you MUST enable access  
     * control to limit queries to your legitimate users. Failing to do so will  
     * cause your server to become part of large scale DNS amplification  
     * attacks. Implementing BCP38 within your network would greatly  
     * reduce such attack surface  
    */  
    recursion yes;
```

Step 10: add any for listen-on port 53 and allow-query

```
// named.conf  
//  
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS  
// server as a caching only nameserver (as a localhost DNS resolver only).  
//  
// See /usr/share/doc/bind*/sample/ for example named configuration files.  
  
options {  
    listen-on port 53 { 127.0.0.1;any; };  
    listen-on-v6 port 53 { ::1; };  
    directory    "/var/named";  
    dump-file    "/var/named/data/cache_dump.db";  
    statistics-file "/var/named/data/named_stats.txt";  
    memstatistics-file "/var/named/data/named_mem_stats.txt";  
    secroots-file  "/var/named/data/named.secroots";  
    recursing-file  "/var/named/data/named.reCURsing";  
    allow-query     { localhost;any; };
```

Step 11: Scroll down till the end of the: zone “ . ” IN {

```
''  
zone "." IN {  
    type hint;  
    file "named.ca";  
};
```

Step 12:

```
;           type "named.ca" ,  
};  
  
zone "bungkus.org" IN {  
    type master;  
    file "fwd.bungkus.org.db";  
    allow-update { none; };  
};  
  
zone "200.168.192.in-addr.arpa" IN {  
    type master;  
    file "rvs.200.168.192.db";  
    allow update { none; };  
};
```

Forward lookup zone Name: Zone “bungkus.org”

- This is the domain of your server and client hostname

Reverse lookup zone Name: Zone “200.168.192-addr.arpa.org”

- This is the reverse of the initial three octets (numbers) of our network address, for example, 192.168.200.5; the initial three will be reversed to 200.168.192.

Next, we need to identify each zone with its relevant configuration file, which we need to create later.

- Same as the zone name, we choose the best practice as below:

Forward lookup zone configuration file: “fwd.bungkus.org.db”

- Remember, bungkus.org is the domain name used in this lab; your domain will be different.

Reverse lookup zone Name: “rvs.200.168. 192.db”

- This is the reverse of the initial three octets (numbers) of our network address, for example, 192.168.200.5; the initial three will be reversed to 200.168.192.

Step 13:

Command: named-checkconf (to check for any syntax error, we should not get any message prompt.)

```
[root@kavvserver ~]# nano /etc/named.conf
[root@kavvserver ~]# named-checkconf
[root@kavvserver ~]# 
```

Step 14:

Command: cd /var/named (change directory)

```
[root@kavvserver ~]# cd /var/named/
[root@kavvserver named]# ls -l
total 16
drwxrwx---. 2 named named   23 Jul 27 22:24 data
drwxrwx---. 2 named named   60 Jul 27 22:24 dynamic
-rw-r-----. 1 root  named 2112 Jun 25 10:45 named.ca
-rw-r-----. 1 root  named  152 Jun 25 10:45 named.empty
-rw-r-----. 1 root  named  152 Jun 25 10:45 named.localhost
-rw-r-----. 1 root  named  168 Jun 25 10:45 named.loopback
drwxrwx---. 2 named named    6 Jun 25 10:45 slaves
[root@kavvserver named]# 
```

Step 15: copied and paste (forward lookup)

			fwd.bungkus.org.db	Modified
GNU nano 5.6.1				
\$ORIGIN example.com.				
\$TTL 86400				
@ IN SOA	dns1.example.com.	hostmaster.example.com.	(2001062501 ; serial 21600 ; refresh after 6 hours 3600 ; retry after 1 hour 604800 ; expire after 1 week 86400) ; minimum TTL of 1 day	
IN NS	dns1.example.com.			
IN NS	dns2.example.com.			
IN MX	10 mail.example.com.			
IN MX	20 mail2.example.com.			
dns1 IN A	10.0.1.1			
dns2 IN A	10.0.1.2			
server1 IN A	10.0.1.5			
server2 IN A	10.0.1.6			
ftp IN A	10.0.1.3			
IN A	10.0.1.4			
mail IN CNAME	server1			
mail2 IN CNAME	server2			

Step 16:

```
GNU nano 5.6.1                                     fwd.bungkus.org.db
$TTL 86400
@       IN      SOA    kavvserver.bungkus.org.  root.bungkus.org. (
                      2001062501 ; serial
                      21600    ; refresh after 6 hours
                      3600     ; retry after 1 hour
                     604800   ; expire after 1 week
                     86400 )  ; minimum TTL of 1 day

;NS records for name servers , name server information
@       IN      NS     kavvserver.bungkus.org.

;A records and IP address for name server
kavvserver   IN      A      192.168.200.4

;A record for IP address to hostname
Shaclient    IN      A      192.168.200.5
```

Explanation of forward lookup zone configuration file Key Elements:

- \$TTL: Sets the default time to live for records in the zone.
- @: Represents the origin of the zone (the domain name itself).
- SOA: Defines the Start of Authority record, containing essential information about the zone.
- NS: Specifies the authoritative nameservers for the zone.
- MX: Lists mail exchangers for email delivery.
- A: Maps hostnames to IPv4 addresses.
- CNAME: Creates aliases for hostnames.

Step 17:

```
cd /var/named
```

```
ls -l
```

```
[root@kavvserver named]# cd /var/named
[root@kavvserver named]# ls -l
total 20
drwxrwx---. 2 named named   23 Jul 27 22:24 data
drwxrwx---. 2 named named   60 Jul 27 22:24 dynamic
-rw-r--r--. 1 root  root  573 Jul 27 23:25 fwd.bungkus.org.db
-rw-r-----. 1 root  named 2112 Jun 25 10:45 named.ca
-rw-r-----. 1 root  named  152 Jun 25 10:45 named.empty
-rw-r-----. 1 root  named  152 Jun 25 10:45 named.localhost
-rw-r-----. 1 root  named  168 Jun 25 10:45 named.loopback
drwxrwx---. 2 named named    6 Jun 25 10:45 slaves
[root@kavvserver named]#
```

Our forward lookup zone file created.

Step 18: Reverse lookup zone

Copy and pasted the template

```
root@kavvserver:/var/named
GNU nano 5.6.1                                     rvs.200.168.192.db
ORIGIN 1.0.10.in-addr.arpa.
TTL 86400
    IN      SOA    dns1.example.com.    hostmaster.example.com. (
                    2001062501 ; serial
                    21600     ; refresh after 6 hours
                    3600      ; retry after 1 hour
                    604800    ; expire after 1 week
                    86400 )   ; minimum TTL of 1 day

    IN      NS     dns1.example.com.
    IN      NS     dns2.example.com.

0    IN      PTR    alice.example.com.
1    IN      PTR    betty.example.com.
2    IN      PTR    charlie.example.com.
3    IN      PTR    doug.example.com.
4    IN      PTR    ernest.example.com.
5    IN      PTR    fanny.example.com.
```

Step 19:

```
root@kavvserver:/var/named
GNU nano 5.6.1                                     rvs.200.168.192.db
$TTL 86400
@      IN      SOA    kavvserver.bungkus.org.    root.bungkus.org. (
                    2001062501 ; serial
                    21600     ; refresh after 6 hours
                    3600      ; retry after 1 hour
                    604800    ; expire after 1 week
                    86400 )   ; minimum TTL of 1 day

;Name server information
@      IN      NS     kavvserver.bungkus.org.

;Reverse lookup for name server and IP address to Hostname
4    IN      PTR    kavvserver.bungkus.org.
5    IN      PTR    Shaclient.bungkus.org.
```

Explanation of Key Elements:

- \$TTL: Sets the default time to live for records in the zone.

- @: Represents the origin of the zone (the reversed IP network address).
- SOA: Defines the Start of Authority record, containing essential information about the zone.
- NS: Specifies the authoritative nameservers for the zone.
- PTR: Maps IP addresses to hostnames.

Key Points:

- Reverse lookup zone files are named after the reversed IP network address. For example, for the 192.168.200.0/24 network, the zone file would be named 200.168.192.in-addr.arpa.
- The PTR records list IP addresses in reverse order, followed by the corresponding hostnames.

Step 20:

```
cd /var/named
```

```
ls -l
```

```
[root@kavvserver named]# cd /var/named/
[root@kavvserver named]# ls -l
total 24
drwxrwx---. 2 named named   23 Jul 27 22:24 data
drwxrwx---. 2 named named   60 Jul 27 22:24 dynamic
-rw-r--r--. 1 root  root  573 Jul 27 23:25 fwd.bungkus.org.db
-rw-r-----. 1 root  named 2112 Jun 25 10:45 named.ca
-rw-r-----. 1 root  named 152 Jun 25 10:45 named.empty
-rw-r-----. 1 root  named 152 Jun 25 10:45 named.localhost
-rw-r-----. 1 root  named 168 Jun 25 10:45 named.loopback
-rw-r--r--. 1 root  root  550 Jul 27 23:37 rvs.200.168.192.db
drwxrwx---. 2 named named    6 Jun 25 10:45 slaves
[root@kavvserver named]#
```

We now have both Lookup zones.

Check your Forward and Reverse Lookup zones for errors:

USE THIS COMMAND TO CHECK WHICH ONE OF THE FILES YOU CONFIGURED WRONG

o named-checkzone fwd.bungkus.org.db /var/named/fwd.bungkus.org.db

```
[root@kavvserver named]# named-checkzone fwd.bungkus.org.db /var/named/fwd.bungkus.org.db
zone fwd.bungkus.org.db/IN: loaded serial 2001062501
OK
[root@kavvserver named]#
```

o named-checkzone rvs.200.168.192.db /var/named/rvs.200.168.192.db

```
[root@kavvserver named]# named-checkzone rvs.200.168.192.db /var/named/rvs.200.168.192.db
zone rvs.200.168.192.db/IN: loaded serial 2001062501
OK
[root@kavvserver named]#
```

Both are successful.

Step 21: We restart and check the status of the DNS service

Command: systemctl restart named

```
[root@kavvserver named]# systemctl restart named
[root@kavvserver named]#
```

Command: systemctl status named

```
[root@kavvserver named]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
  Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
  Active: active (running) since Sun 2025-07-27 23:45:06 +08; 41s ago
    Process: 7770 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/named-chek>
    Process: 7772 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 7773 (named)
     Tasks: 6 (limit: 74031)
    Memory: 44.8M
       CPU: 83ms
      CGroup: /system.slice/named.service
              └─7773 /usr/sbin/named -u named -c /etc/named.conf

Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './NS/IN': 2001:500:2d::d#53
Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './DNSKEY/IN': 2001:503:ba3e::>
Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './NS/IN': 2001:503:ba3e::2:30>
Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './DNSKEY/IN': 2001:dc3::35#53
Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './NS/IN': 2001:dc3::35#53
Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './DNSKEY/IN': 2001:500:9f::42#53
Jul 27 23:45:06 kavvserver.bungkus.org named[7773]: network unreachable resolving './NS/IN': 2001:500:9f::42#53
Jul 27 23:45:07 kavvserver.bungkus.org named[7773]: managed-keys-zone: Key 20326 for zone . is now trusted (acce>
Jul 27 23:45:07 kavvserver.bungkus.org named[7773]: managed-keys-zone: Key 38696 for zone . is now trusted (acce>
Jul 27 23:45:07 kavvserver.bungkus.org named[7773]: resolver priming query complete
```

It is active and it is enabled.

Step 23: Then we need to allow the firewall to pass port 53 through UDP and TCP

Command: Sudo firewall-cmd --permanent --add-port=53/tcp

```
[root@kavvserver named]# firewall-cmd --permanent --add-port=53/tcp
success
```

firewall-cmd --permanent --add-port=53/tcp is used to permanently open TCP port 53 (used for DNS services) in the firewall, allowing external DNS requests over TCP.

- ◆ After using this command, you must reload the firewall with firewall-cmd --reload to apply the change.

Command: Sudo firewall-cmd --permanent --add-port=53/udp

```
[root@kavvserver named]# firewall-cmd --permanent --add-port=53/udp
success
[root@kavvserver named]#
```

firewall-cmd --permanent --add-port=53/udp is used to permanently allow UDP traffic on port 53, which is essential for DNS queries, through the firewall.

- ◆ Don't forget to run firewall-cmd --reload to apply the changes.

Command: Sudo firewall-cmd --reload

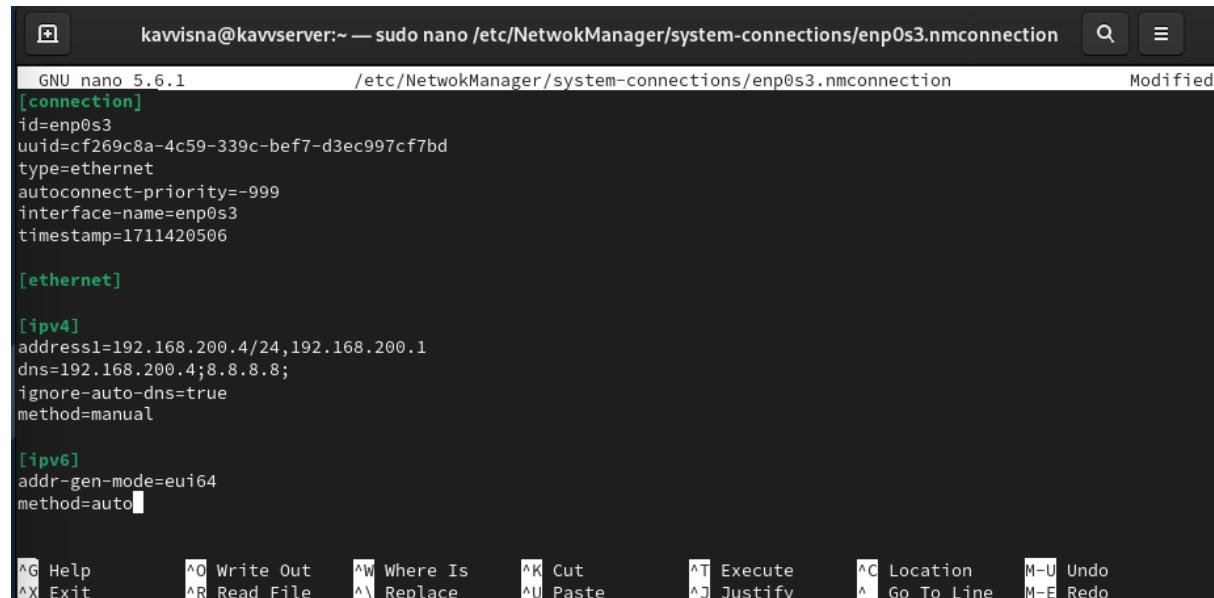
```
success
[root@kavvserver named]# firewall-cmd --reload
success
[root@kavvserver named]#
```

firewall-cmd --reload is used to apply and activate any changes made to the firewall configuration, such as newly added ports or services, without restarting the system.

Step 24: SWAPPING DNS IP ADDRESSES, ROCKY IP ADDRESS IS SET AS FIRST

To proceed with editing

```
sudo nano /etc/NetworkManager/system-connections/enp0s3.nmconnection
```



```
GNU nano 5.6.1 /etc/NetworkManager/system-connections/enp0s3.nmconnection Modified
[connection]
id=enp0s3
uuid=cf269c8a-4c59-339c-bef7-d3ec997cf7bd
type=ethernet
autoconnect-priority=-999
interface-name=enp0s3
timestamp=1711420506

[ethernet]

[ipv4]
address=192.168.200.4/24,192.168.200.1
dns=192.168.200.4;8.8.8.8;
ignore-auto-dns=true
method=manual

[ipv6]
addr-gen-mode=eui64
method=auto

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^ Go To Line M-E Redo
```

We add in all necessary details including addinf rocky dns server as second then followed by google dns server as first (corrected one)

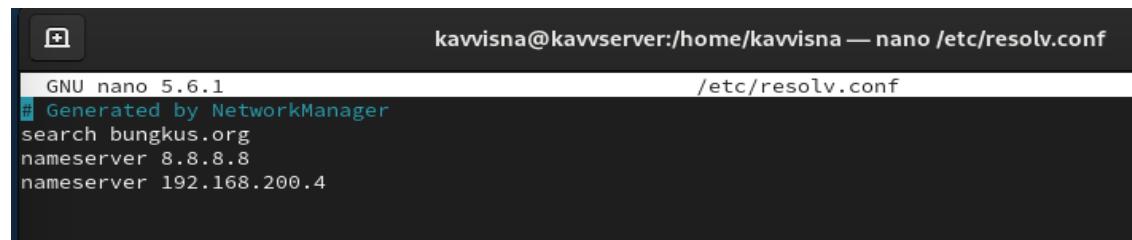
Command: sudo systemctl restart NetworkManager

```
[root@kavvserver kavvisna]# sudo systemctl restart NetworkManager
[root@kavvserver kavvisna]#
```

Step 25: Additional configuration: Check our DNS IP address in resolve.conf file and make sure our DNS server IP (Rocky Server) is the first in the line

Command: sudo nano /etc/resolv.conf

```
[root@kavvserver kavvisna]# nano /etc/resolv.conf
[root@kavvserver kavvisna]#
```



```
GNU nano 5.6.1 /etc/resolv.conf
# Generated by NetworkManager
search bungkus.org
nameserver 8.8.8.8
nameserver 192.168.200.4
```

Step 26: Restart our DNS server again

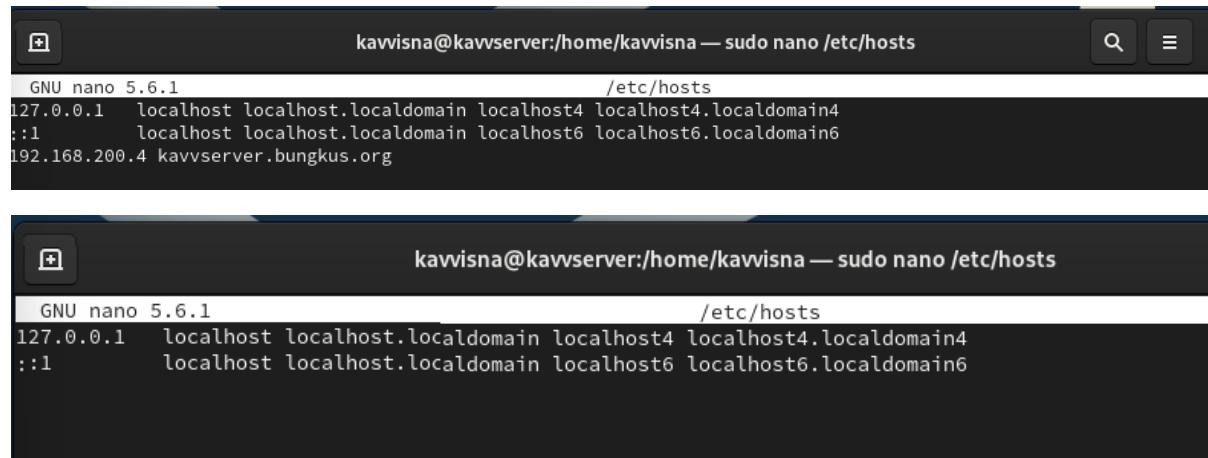
Command: sudo systemctl restart named

```
[root@kavvserver kavvisna]# sudo systemctl restart named
[root@kavvserver kavvisna]# █
```

Step 27: Hosts Configuration File

remove our IP and FQDN from the hosts file to make sure our DNS is working

Command: sudo nano /etc/hosts



```
GNU nano 5.6.1 /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.200.4 kavvserver.bungkus.org
```

```
GNU nano 5.6.1 /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

NSLOOKUP Method: Testing Our DNS Server and Client

Command: nslookup shaserver.bungkus.org

```
[root@kavvserver kavvisna]# nslookup kavvserver.bungkus.org
Server:      192.168.200.4
Address:     192.168.200.4#53

Name:   kavvserver.bungkus.org
Address: 192.168.200.4

[root@kavvserver kavvisna]# █
```

Command: nslookup shaclient.bungkus.org

```
[root@kavvserver kavvisna]# nslookup Shaclient.bungkus.org
Server:      192.168.200.4
Address:     192.168.200.4#53

Name:   Shaclient.bungkus.org
Address: 192.168.200.5

[root@kavvserver kavvisna]# █
```

Ping Method: Testing Our DNS Server and Client

Command: ping shaserver.bungkus.org (rocky DNS server)

```
[kavvisna@kavvserver ~]$ ping kavvserver.bungkus.org
PING kavvserver.bungkus.org (192.168.200.4) 56(84) bytes of data.
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=1 ttl=64 time=0.242 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=2 ttl=64 time=0.105 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=3 ttl=64 time=0.155 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=4 ttl=64 time=0.087 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=5 ttl=64 time=0.183 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=6 ttl=64 time=0.170 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=7 ttl=64 time=0.156 ms
^C
--- kavvserver.bungkus.org ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6138ms
rtt min/avg/max/mdev = 0.087/0.156/0.242/0.047 ms
[kavvisna@kavvserver ~]$
```

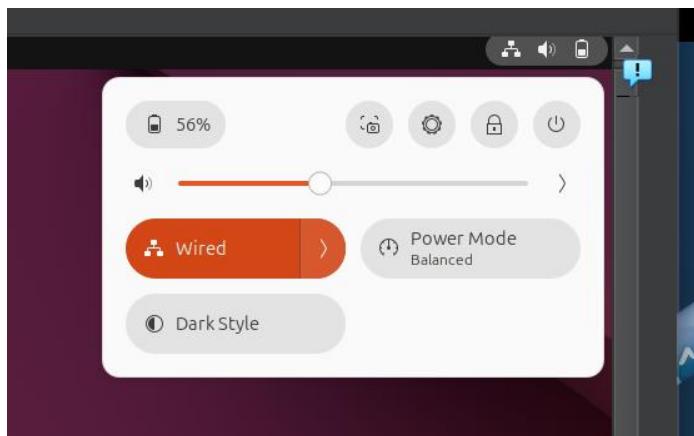
Command: ping shaclient.bungkus.org (ping ubuntu client)

```
[kavvisna@kavvserver ~]$ ping shaclient.bungkus.org
PING shaclient.bungkus.org (192.168.200.5) 56(84) bytes of data.
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=2 ttl=64 time=3.82 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=3 ttl=64 time=2.90 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=4 ttl=64 time=1.83 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=5 ttl=64 time=3.00 ms
^C
--- shaclient.bungkus.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 1.831/3.011/3.822/0.678 ms
[kavvisna@kavvserver ~]$
```

Ubuntu Version 24 Linux: DNS Configuration Setting

GUI Method: Network (Internet Connection) Setting

Step 1:

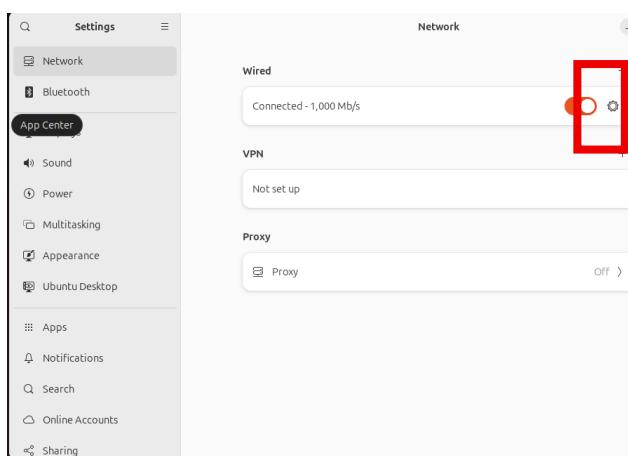


Click at wired



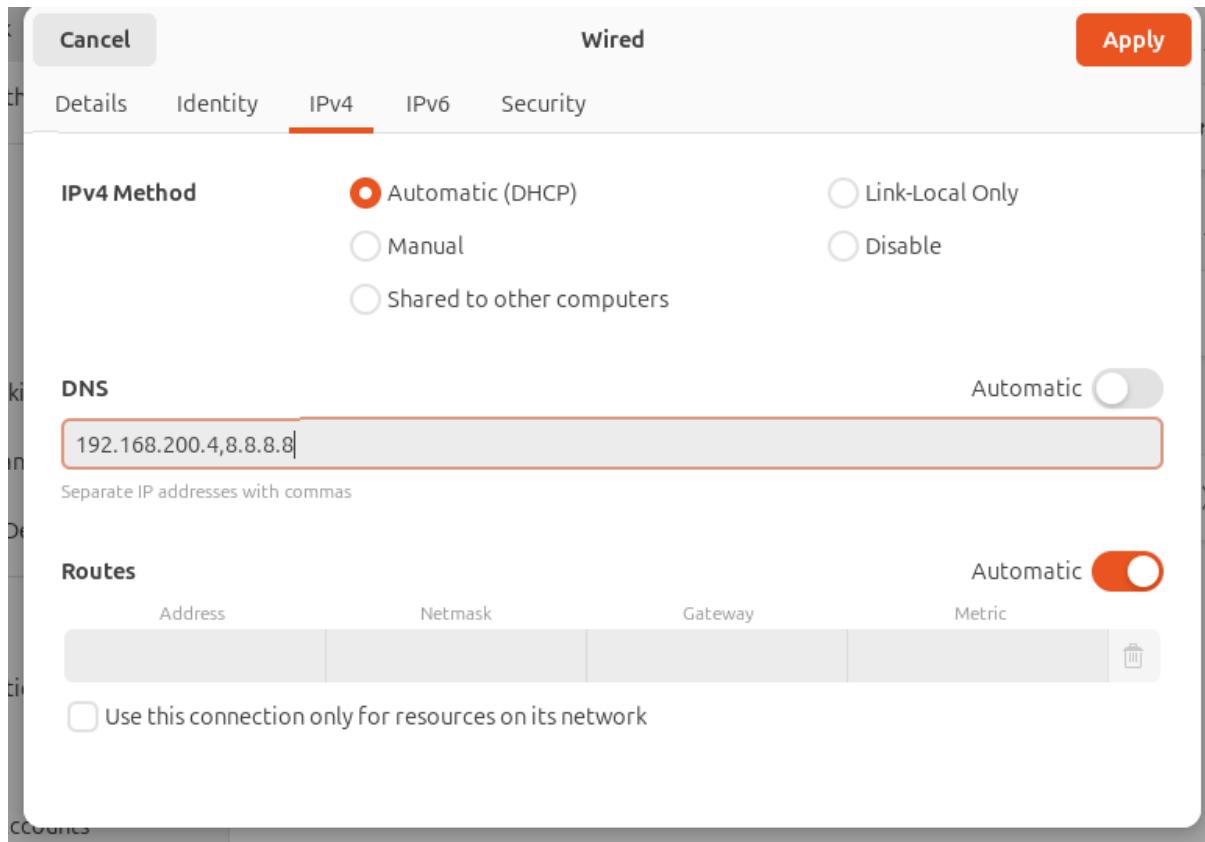
Click wired settings

Step 2:

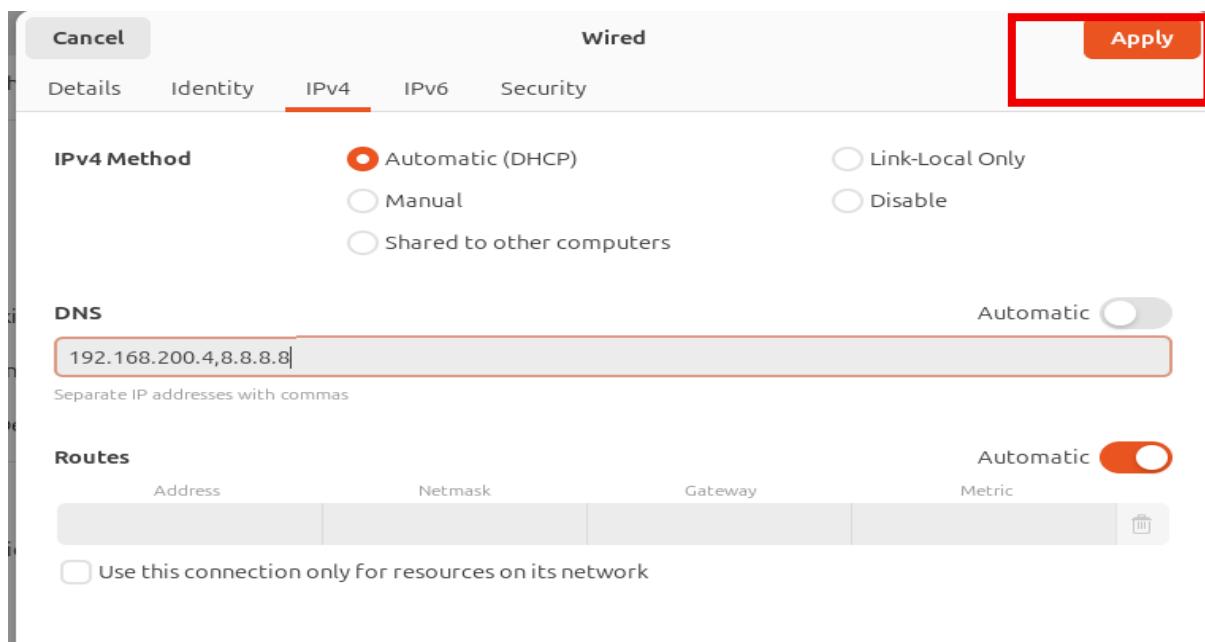


Click on the setting

Step 3: SWAPPING DNS IP ADDRESSES, ROCKY IP ADDRESS IS SET AS FIRST



Step 4:



After changes click apply

Permanent Setup of DNS IP address on Ubuntu

Step 1: enable the services permanently so it will be up and running even after reboot, and start it now

Command: sudo systemctl enable systemd-resolved

```
vboxuser@kavvclient:~$ sudo systemctl enable systemd-resolved
vboxuser@kavvclient:~$ sudo systemctl start systemd-resolved
vboxuser@kavvclient:~$ █
```

the systemd-resolved service, which is responsible for handling network name resolution (such as DNS).

Step 2: check the system-resolved and to make sure is active

Command: sudo systemctl status systemd-resolved

```
vboxuser@kavvclient:~$ sudo systemctl status systemd-resolved
● systemd-resolved.service - Network Name Resolution
  Loaded: loaded (/usr/lib/systemd/system/systemd-resolved.service; enabled; presen>
  Active: active (running) since Mon 2025-07-28 06:05:03 UTC; 28min ago
    Docs: man:systemd-resolved.service(8)
          man:org.freedesktop.resolve1(5)
          https://www.freedesktop.org/wiki/Software/systemd/writing-network-configuration/>
          https://www.freedesktop.org/wiki/Software/systemd/writing-resolver-client/>
  Terminal
  Main PID: 649 (systemd-resolve)
    Status: "Processing requests..."
      Tasks: 1 (limit: 9205)
     Memory: 7.7M (peak: 8.2M)
        CPU: 1.087s
       CGroup: /system.slice/systemd-resolved.service
                 └─649 /usr/lib/systemd/systemd-resolved

Jul 28 06:05:02 kavvclient.bungkus.org systemd-resolved[649]: Negative trust anchors:>
Jul 28 06:05:03 kavvclient.bungkus.org systemd-resolved[649]: Using system hostname '>
Jul 28 06:05:03 kavvclient.bungkus.org systemd[1]: Started systemd-resolved.service ->
Jul 28 06:05:13 kavvclient.bungkus.org systemd-resolved[649]: enp0s3: Bus client set >
```

In Ubuntu Linux version 24 (or any recent version), the command sudo systemctl status systemresolved is used to check the status of the systemd-resolved service, which is responsible for handling network name resolution (such as DNS).

Running this command will provide information about whether the service is active, any recent logs related to it, and if there are any issues or errors. You need to use sudo to execute it with administrative privileges because managing system services typically requires elevated permissions

Step 3: We will add our Rocky DNS IP address and Google DNS IP address to the resolved configuration file so is added permanently even after rebooting the system

Command: sudo nano /etc/systemd/resolved.conf

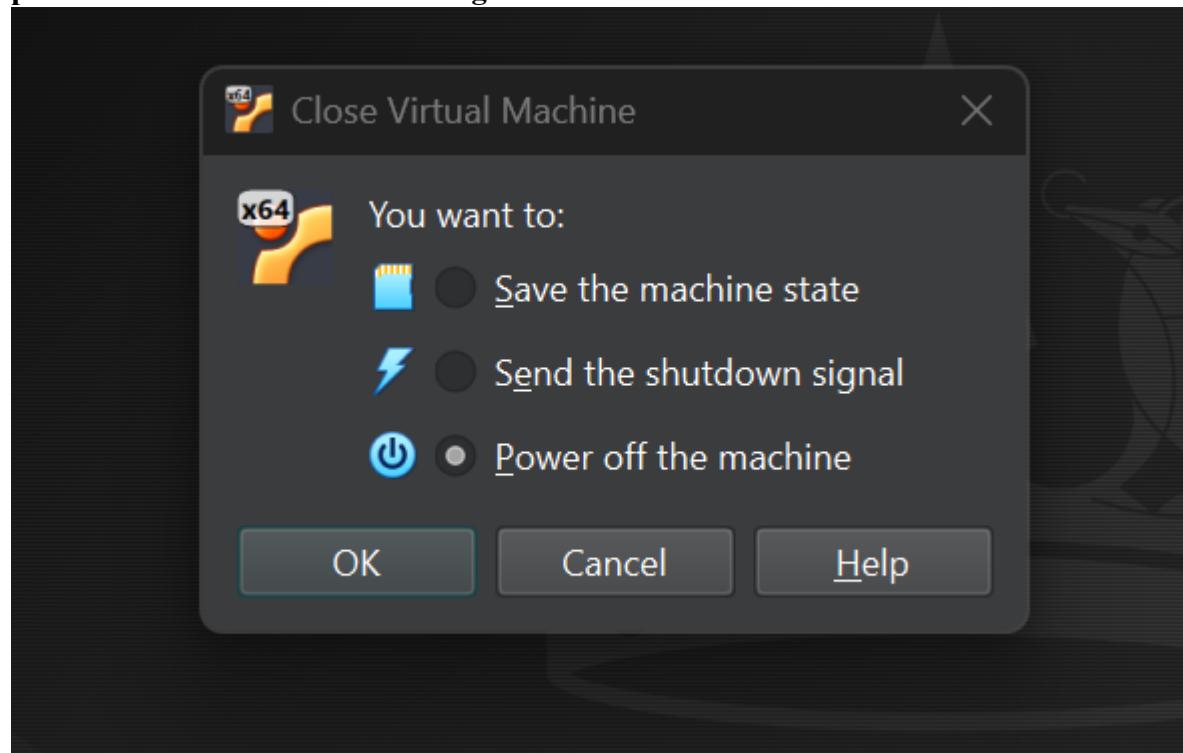
```
vboxuser@kavvclient:~$ sudo nano /etc/systemd/resolved.conf
[sudo] password for vboxuser:
File: /etc/systemd/resolved.conf                                     [New]

GNU nano 7.2          /etc/systemd/resolved.conf *

[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com 2606:4700:4>
# Google:      8.8.8.8#dns.google 8.8.4.4#dns.google 2001:4860:4860::8888#dns.go>
# Quad9:       9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net 2620:fe::fe#d>
DNS=192.168.200.4 8.8.8.8
#FallbackDNS=
```

Step 4: power off the machine and start again

power off the machine and start again



TESTING OUR DNS SERVER and Client

Command: nmcli

```
vboxuser@kavvclient:~$ nmcli
enp0s3: connected to netplan-enp0s3
    "Intel 82540EM"
    ethernet (e1000), 08:00:27:A3:A9:78, hw, mtu 1500
    ip4 default
    inet4 192.168.200.5/24
    route4 192.168.200.0/24 metric 100
    route4 default via 192.168.200.1 metric 100

lo: connected (externally) to lo
    "lo"
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
    inet4 127.0.0.1/8
    inet6 ::1/128

DNS configuration:
    servers: 192.168.200.4 8.8.8.8
    interface: enp0s3

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.
vboxuser@kavvclient:~$
```

Run nmcli command and make sure that the DNS server is set to our Rocky IP address and Google DNS IP

Our rocky Ip address should be first the followed by google DNS server IP

NSLOOKUP Method: Testing Our DNS Server and Client

One method to check our DNS Rocky server is working is to use NSLOOKUP

Command: nslookup shaserver.bungkus.org

```
vboxuser@kavvclient:~$ nslookup kavvserver.bungkus.org
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   kavvserver.bungkus.org
Address: 192.168.200.4

vboxuser@kavvclient:~$
```

Command: nslookup shaclient.bungkus.org

```
vboxuser@kavvclient:~$ nslookup Shaclient.bungkus.org
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   Shaclient.bungkus.org
Address: 192.168.200.5
```

Ping Method: Testing Our DNS Server and Client

One method to check our DNS Rocky server is working is to ping using FQDN

Command: ping shaserver.bungkus.org

```
vboxuser@kavvclient:~$ ping kavvserver.bungkus.org
PING kavvserver.bungkus.org (192.168.200.4) 56(84) bytes of data.
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=1 ttl=64 time=15.5 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=2 ttl=64 time=1.17 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=3 ttl=64 time=1.99 ms
64 bytes from kavvserver.bungkus.org (192.168.200.4): icmp_seq=4 ttl=64 time=1.12 ms
^C
--- kavvserver.bungkus.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 14948ms
rtt min/avg/max/mdev = 1.117/4.949/15.518/6.111 ms
vboxuser@kavvclient:~$
```

Command: ping shaclient.bungkus.org

```
vboxuser@kavvclient:~$ ping Shaclient.bungkus.org
PING Shaclient.bungkus.org (192.168.200.5) 56(84) bytes of data.
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=2 ttl=64 time=0.025 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=4 ttl=64 time=0.027 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=5 ttl=64 time=0.028 ms
64 bytes from Shaclient.bungkus.org (192.168.200.5): icmp_seq=6 ttl=64 time=0.040 ms
^C
--- Shaclient.bungkus.org ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 11391ms
rtt min/avg/max/mdev = 0.025/0.032/0.041/0.006 ms
```

Dig Method: Testing Our DNS Server and Client

One method to check our DNS Rocky server is working is to use dig

Command: dig shaserver.bungkus.org

```
vboxuser@kavvclient:~$ dig kavvserver.bungkus.org

; <>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <>> kavvserver.bungkus.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55914
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;kavvserver.bungkus.org.           IN      A

; ANSWER SECTION:
avvserver.bungkus.org. 6773    IN      A      192.168.200.4

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Jul 28 08:01:43 UTC 2025
;; MSG SIZE  rcvd: 67

vboxuser@kavvclient:~$
```

Command: dig shaclient.bungkus.org

```
vboxuser@kavvclient:~$ dig Shaclient.bungkus.org

; <>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <>> Shaclient.bungkus.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33128
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;Shaclient.bungkus.org.      IN      A

;; ANSWER SECTION:
Shaclient.bungkus.org.  6668    IN      A      192.168.200.5

;; Query time: 1 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Jul 28 08:04:41 UTC 2025
;; MSG SIZE rcvd: 66

vboxuser@kavvclient:~$
```

LAB 5B (Backup Our Server & Client)

