[ Answer questions from Mr. Lanh ]

**Question 1: How to check whether email is spam or not?**

**SPF (Sender Policy Framework)** is a way to check if the email server is allowed to send on behalf of that domain

ex:

* domain gmail.com only allows Google to send mail
* if an email says it is from abc@gmail.com but is sent from a strange IP → invalid, it may be fake

**DKIM (DomainKeys Identified Mail)** is a digital signature added to the email, used to verify that the message really came from the sender’s domain

* if DKIM is present and passes, it means the email is genuine
* if it’s missing or fails, the email might be forged

**DMARC (Domain-based Message Authentication, Reporting & Conformance)** is a policy set by the sender’s domain to require SPF and/or DKIM checks to pass *(if the email fails DMARC, it can be flagged or rejected by mail systems)*

**Sender IP address** is the actual IP address that sent the email

* if it belongs to a trusted provider (Google, Microsoft, …), that is a good sign
* if it’s an unknown IP or on a blacklist (RBL) → the email is likely spam

**Check if 'From' address matches the real sending IP**

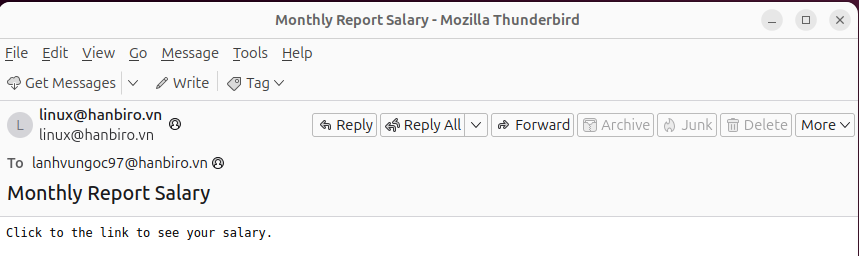
* some fake emails use a legitimate-looking “From” address but are actually sent from a different, unrelated IP
* if the IP and domain don’t match → it might be spoofed or phishing

***# A few things to consider, Content-based factors:***

**Vague or unclear content**

ex from Mr. Lanh: “Click the link to see your salary.”

No details, no context — that’s suspicious



**Contains strange or shortened links**

* links like **bit.ly**, or long weird URLs are common in phishing emails
* they may lead to **fake websites or malware**

**Uses emotional or urgent language**

ex: “You won a prize”, “Salary report available”, “Your account is locked”, ...

-> these messages try to **make the reader panic and click quickly** without thinking

**No signature or sender identity**

Real business emails usually include:

* Full name
* Job title or department
* Company info or contact details  
   => If missing, the email could be spam or a scam.

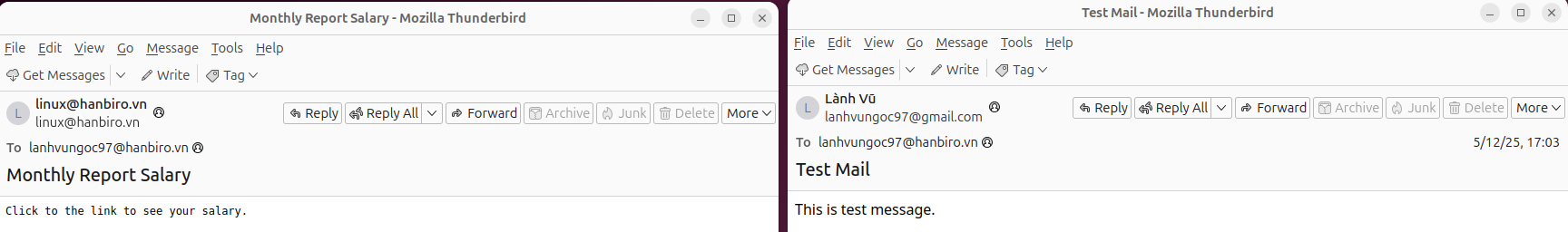
# If an email **lacks technical validation**, comes from a **suspicious IP**, and has **unclear or emotional content**, it’s most likely **spam or phishing**

**Question 2: I have an email please check if it is spam and why**

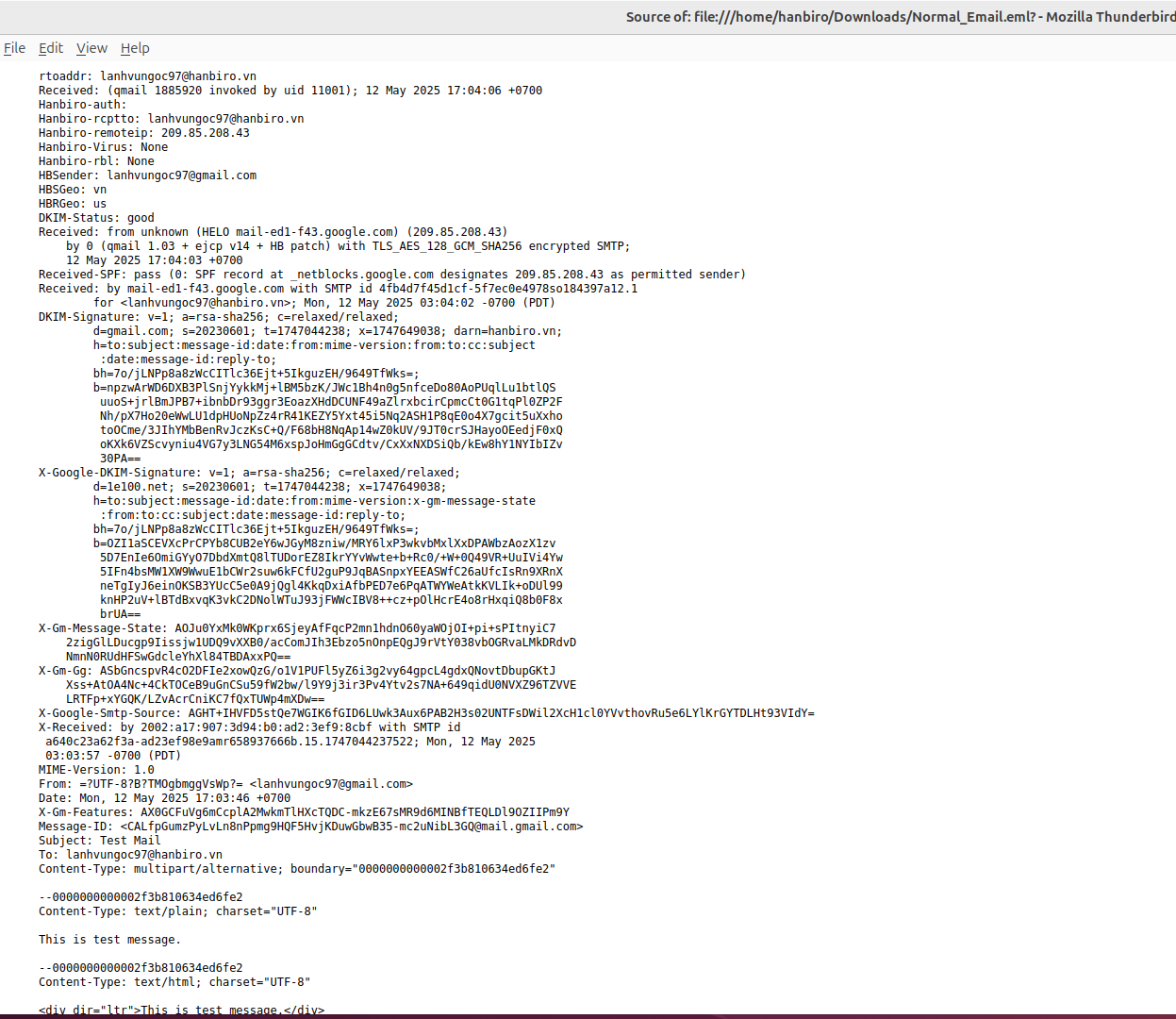
**linux@hanbiro.vn\_Monthly Report Salary\_1747044092.1847391.eml => Please check it**

**\* Normal\_Email.eml => This is normal email (not spam). Please refer it to check email above.**

emails when opened normally will be difficult to decide

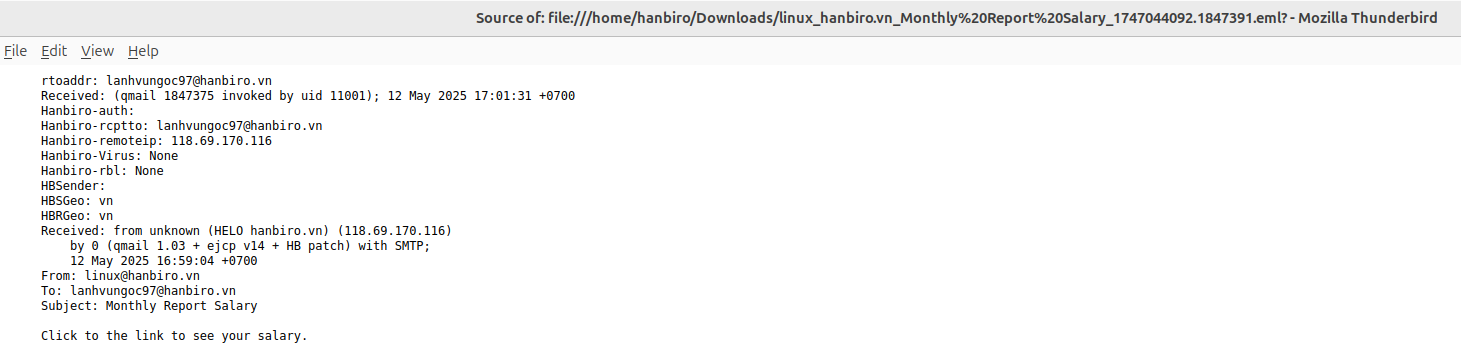


* When I view source of **Normal\_Email.eml**, I see:



| SPF | the sender's IP **209.85.208.43** is authorized to send on behalf of **gmail.com** |
| --- | --- |
| DKIM | email is digitally signed and verified as from **gmail.com** |
| DMARC | The sending domain policy is complied with (DKIM & SPF are both met) |
| Sender IP | 209.85.208.43 IP address belongs to Google, very reliable |
| TLS Encryption | sent over a secure encrypted connection |
| Sender identity | from: “Lành Vũ” <lanhvungoc97@gmail.com> - clearly identifiable |
| Email headers | contains proper headers: From, To, Subject, DKIM, … |
| Content | title "Test Mail" & content "This is test message." safe |

* When I view source of **linux\_hanbiro.vn\_Monthly Report Salary\_1747044092.1847391.eml**, I see:



| SPF | no SPF record found, can’t verify if the sender is authorized |
| --- | --- |
| DKIM | no digital signature, can't confirm authenticity |
| DMARC | without SPF or DKIM, DMARC checks are meaningless |
| Sender IP | unknown 118.69.170.116 IP, not related to Gmail, Microsoft,... could be fake |
| TLS Encryption | email was not encrypted during transmission |
| Sender identity | from: linux@hanbiro.vn, but no name or sender validation, not from major suppliers, easily counterfeited |
| Header validation | headers look minimal and don’t match trusted mail services (just write linux@hanbiro.vn, no authentication) |
| Content | Title: 'Monthly Report Salary' & content: 'Click the link to see your salary.' are lacks context, vague and potentially dangerous |

**linux@hanbiro.vn\_Monthly Report Salary** is likely spam or phishing due to several suspicious signs:

* no SPF/DKIM/DMARC authentication, cannot verify the sender is legitimate
* the sending IP (118.69.170.116) is not from a reputable email provider such as Google, Microsoft, …
* no clear sender name or email signature, lack of credibility
* vague content, containing links of unknown origin - “Click to the link to see your salary.”
* using psychological tricks (“salary”) to lure users into clicking on the link without thinking

1. **DNS**

**+ DNS Basic**

**Question: Role of a Name Server**

**Answer:** A name server is responsible for storing and managing information that maps domain names to corresponding IP addresses. This allow system to locate and connect to the correct the server hosting the desired service

**Question: DNS Query Process**

**Answer:** When a domain name is requested, the DNS resolution process typically follow these steps:

* dns client sends a query to a resolver, which may be provided by an ISP or third party service
* the resolver checks its local cache for the domain's IP address
* if the address is not cached, the resolver queries a root server to find the TLD server (".com", “.org”, …)
* the TLD server directs the resolver to the authoritative server for the domain
* the authoritative server responds with the domain’s IP address
* the resolver sends the IP address back to the device, allowing it to access the website

**Question: DNS security**

**Answer:** DNS security focused on protecting the DNS infrastructure from attacks such as DNS spoofing, cache poisoning and DDoD attack

Several security measure are commonly used to enhance DNS security:

* DNSSEC (dns security extensions) uses digital signatures to verify the authenticity of DNS data, preventing attacks like DNS spoofing and cache poisoning
* DoT encrypts DNS queries using the TLS protocol, protecting data from eavesdropping and tampering during transmission
* DoH encrypts DNS queries over HTTPS, enhancing user privacy and preventing third party surveillance

**# DNS Practice**

options {

listen-on port 53 { 127.0.0.1; 111.111.111.111; };

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";

allow-query { any; };

recursion yes;

allow-recursion { any; };

dnssec-enable yes;

dnssec-validation yes;

version "LOCAL DNS";

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

};

**Explanation:**

* listen-on port 53 { 127.0.0.1; 111.111.111.111; };

*-> This tells the DNS server to listen for DNS requests on IPv4 addresses 127.0.0.1 (localhost) and 111.111.111.111 (custom IP)*

* listen-on-v6 port 53 { ::1; };

-> similar to above, but for IPv6. ::1 is the IPv6 equivalent of 127.0.0.1

* directory "/var/named";

-> set the working directory for named. All zone files and logs will be located here by default

* dump-file "/var/named/data/cache\_dump.db";

-> define where to dump the current DNS cache when requested

* statistics-file "/var/named/data/named\_stats.txt";

-> this file stores DNS server statistics like queries received, answers sent,...

* memstatistics-file "/var/named/data/named\_mem\_stats.txt";

-> record memory usage statistics of the DNS server

* allow-query { any; };

-> allow all clients to send DNS queries to the server

* recursion yes;

-> enable recursive DNS lookups, the server can resolve names it is not authoritative for by querying other server

* allow-recursion { any; };

-> specify which clients are allowed to use recursion. { any; } means allow all

* dnssec-enable yes;

-> enable DNSSEC to allow secure DNS validation

* dnssec-validation yes;

-> instruct the server to validate DNSSEC-signed data

* version "LOCAL DNS";

-> hide the actual BIND version. This string is shown when someone queries the server version

* bindkeys-file "/etc/named.iscdlv.key";

-> point to the file that contains trust anchors for DNSSEC validation

* managed-keys-directory "/var/named/dynamic";

-> the folder where BIND stores keys it updates dynamically

**PRACTICE DNS CONFIG**

**# Setting DNS:**

VM1: CentOS9 - Master DNS server - IP 192.168.159.10 - DNS 192.168.159.20 8.8.8.8

VM2: CentOS9 - Slave DNS server - IP 192.168.159.20 - DNS 192.168.159.10 8.8.8.8

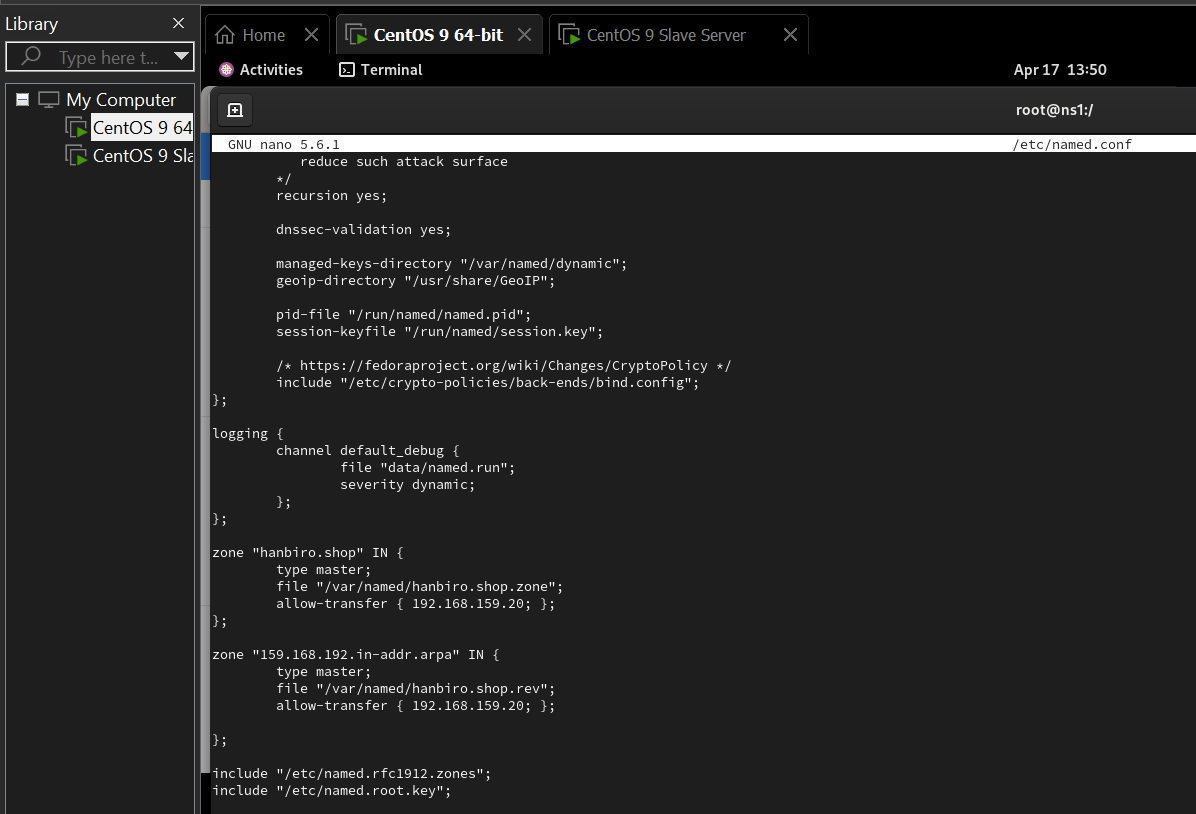
VM3: Window 10 client - IP 192.168.159.30 - DNS 192.168.159.10 192.168.159.20

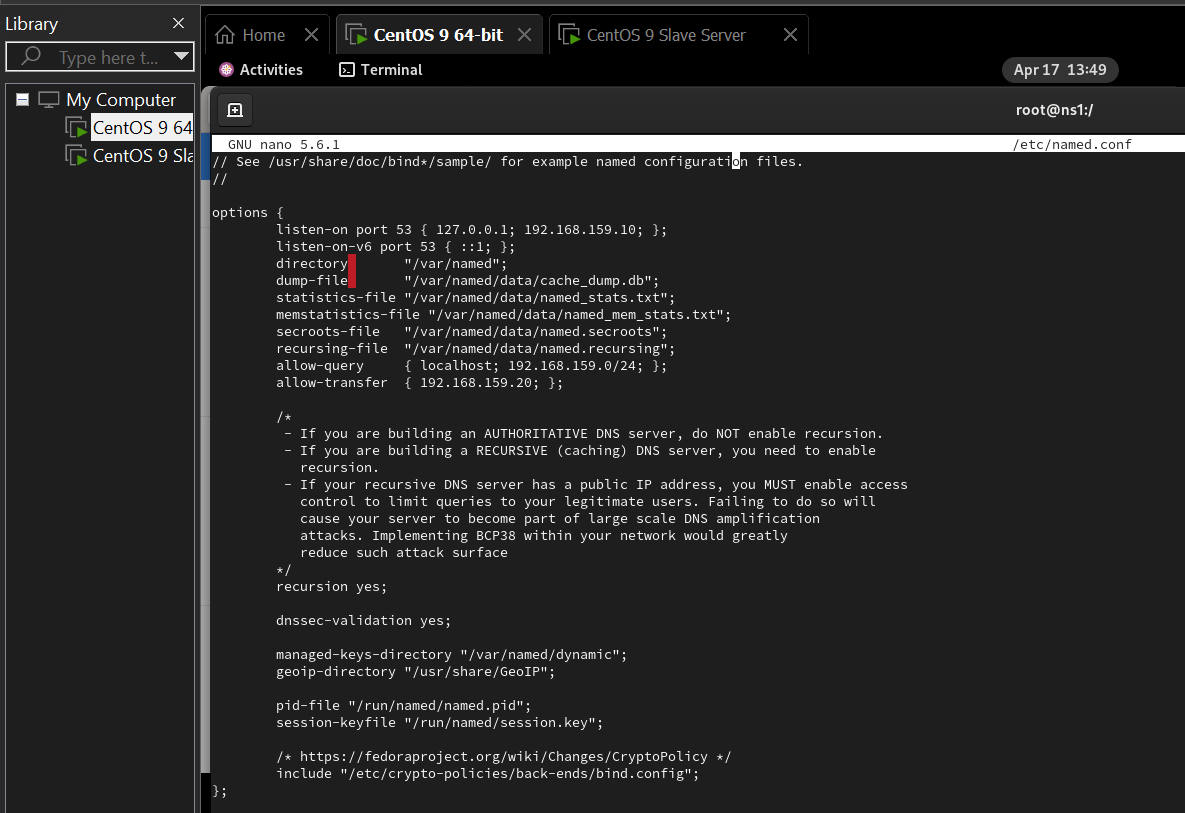
Domain name: hanbiro.shop

**# DNS Master setup**

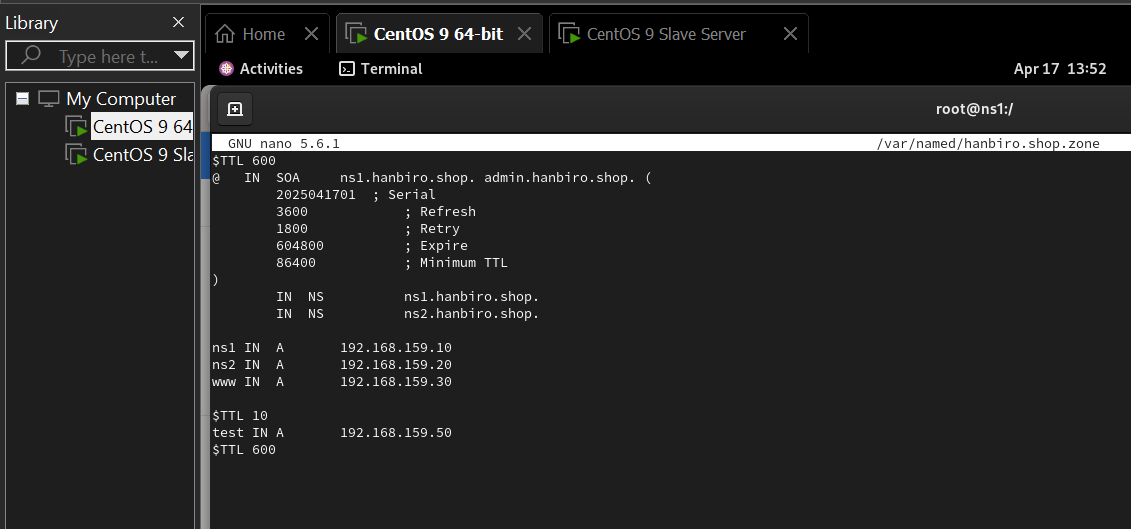
b1: download bind: sudo dnf install bind bind-utils -y

b2: setting ‘sudo nano /etc/named.conf’

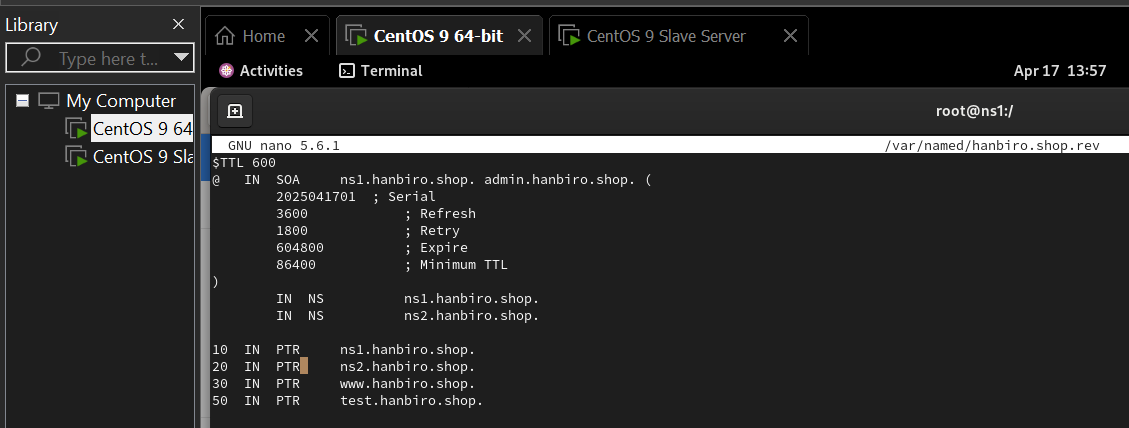




b3: create zone file (forward zone) ‘sudo nano /var/named/hanbiro.shop.zone’:



override TTL 10s for test

b4: create reverse zone ‘sudo nano /var/named/hanbiro.shop.rev‘

b5: authorize and enable service, add dns to firewall

sudo chown root:named /var/named/hanbiro.shop.\*

sudo restorecon -rv /var/named

sudo firewall-cmd --add-service=dns --permanent

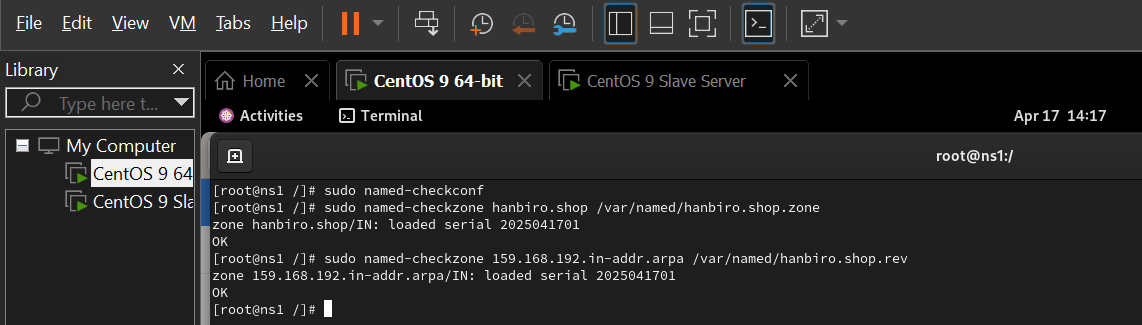
sudo firewall-cmd --reload

b6: check config and enable named:

sudo named-checkconf

sudo named-checkzone hanbiro.shop /var/named/hanbiro.shop.zone

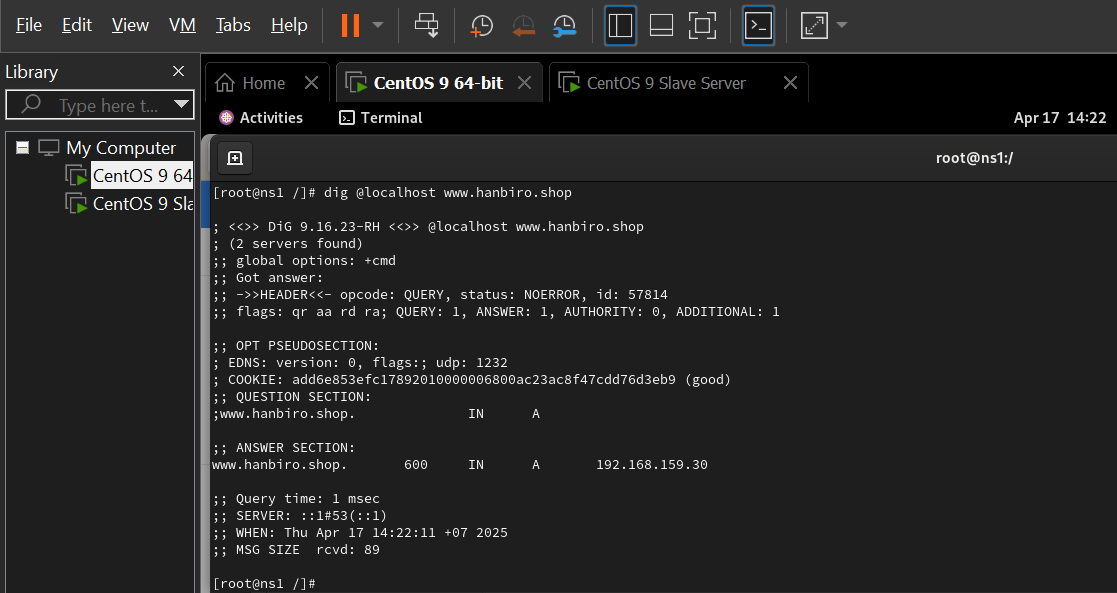
sudo named-checkzone 159.168.192.in-addr.arpa /var/named/hanbiro.shop.rev



sudo systemctl enable named

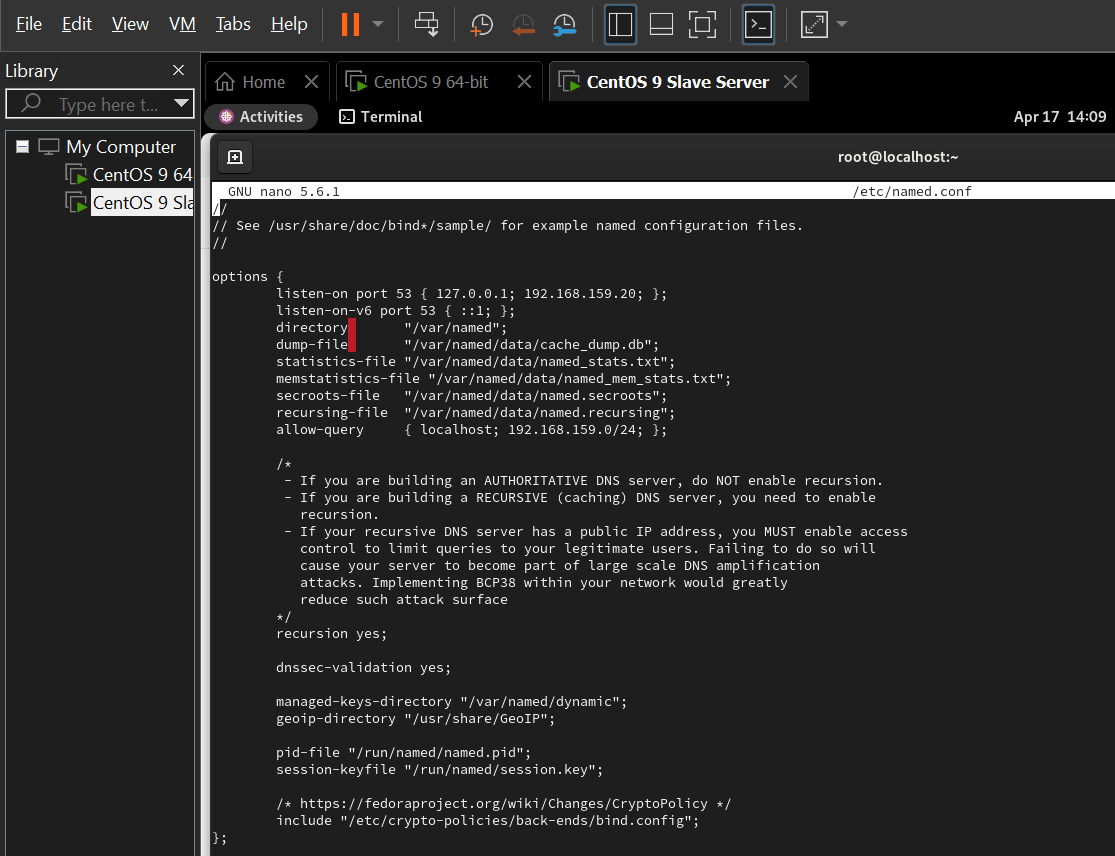
sudo systemctl start named

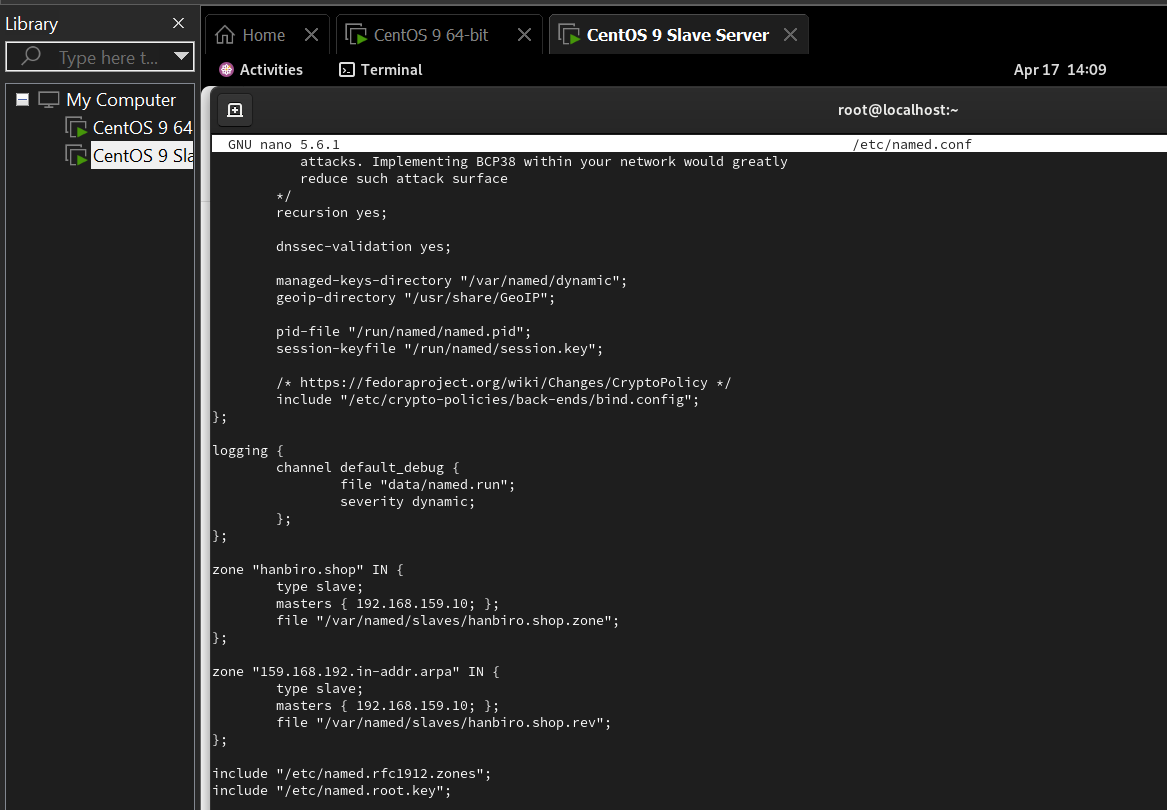
query at DNS Master server itself using the localhost address. Must return the correct IP (192.168.159.30) ‘dig @localhost www.hanbiro.shop’

****

**# DNS Slave setup**

b1: download bind: sudo dnf install bind bind-utils -y

b2: config file ‘sudo nano /etc/named.conf’



b3: create folder slaves, enable and config firewall, start named

sudo mkdir -p /var/named/slaves

sudo chown named:named /var/named/slaves

sudo restorecon -rv /var/named

check dns in firewall: sudo firewall-cmd --list-all

if without it:

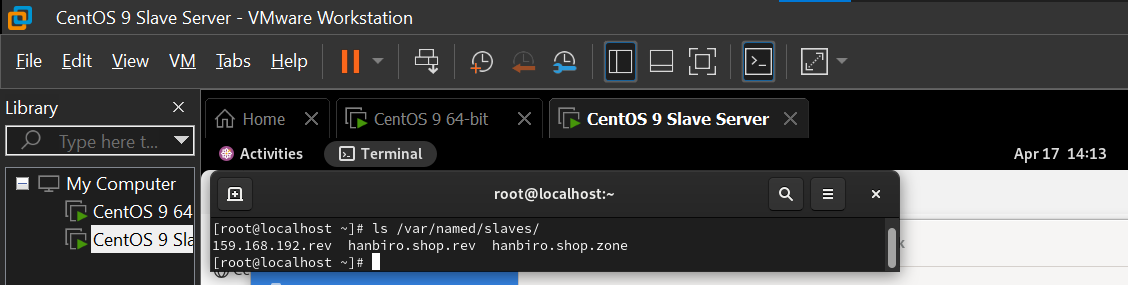
sudo firewall-cmd --add-service=dns --permanent

sudo firewall-cmd --reload

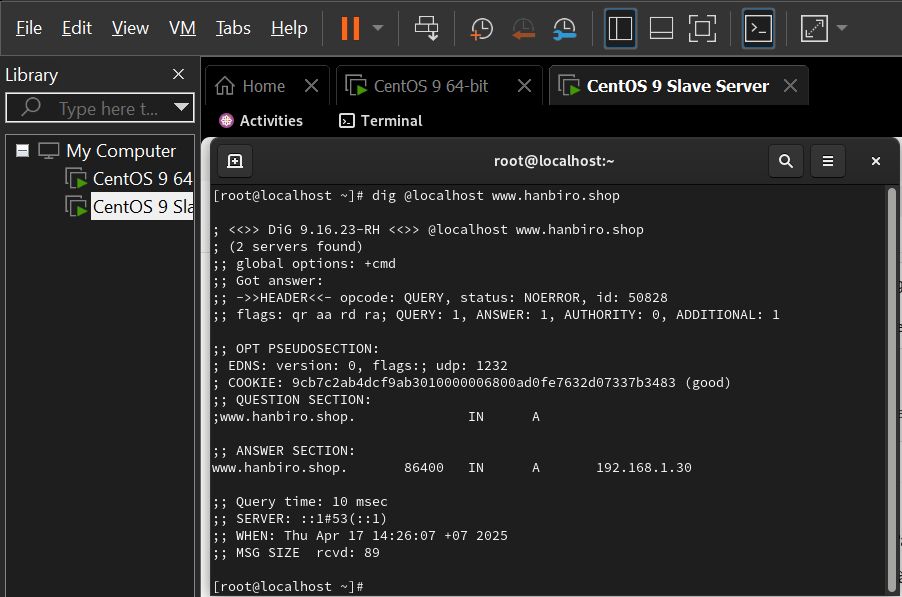
sudo systemctl enable named

sudo systemctl start named

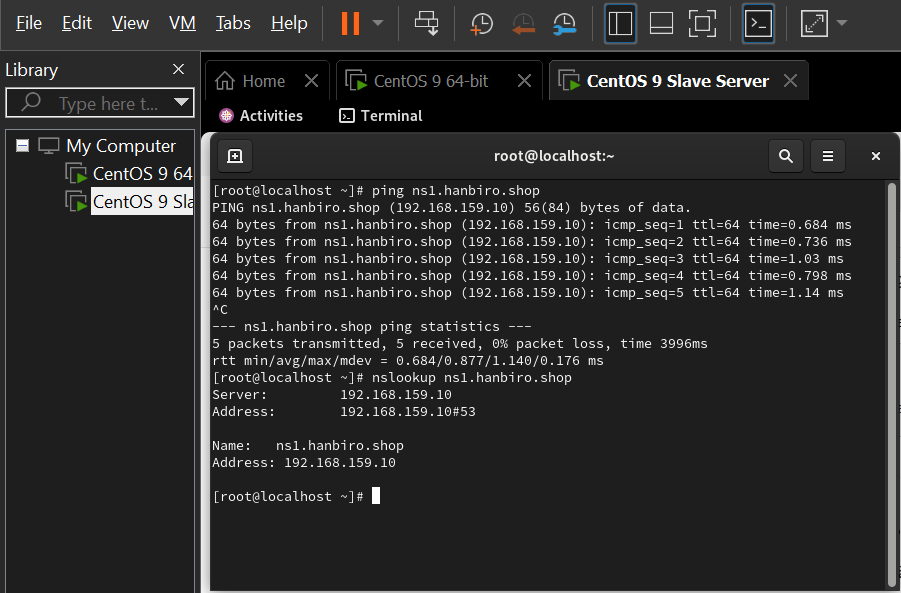
b4: check synchronization ‘ls /var/named/slaves’



b5: must see hanbiro.shop.zone file synchronized from Master ‘dig @localhost www.hanbiro.shop’

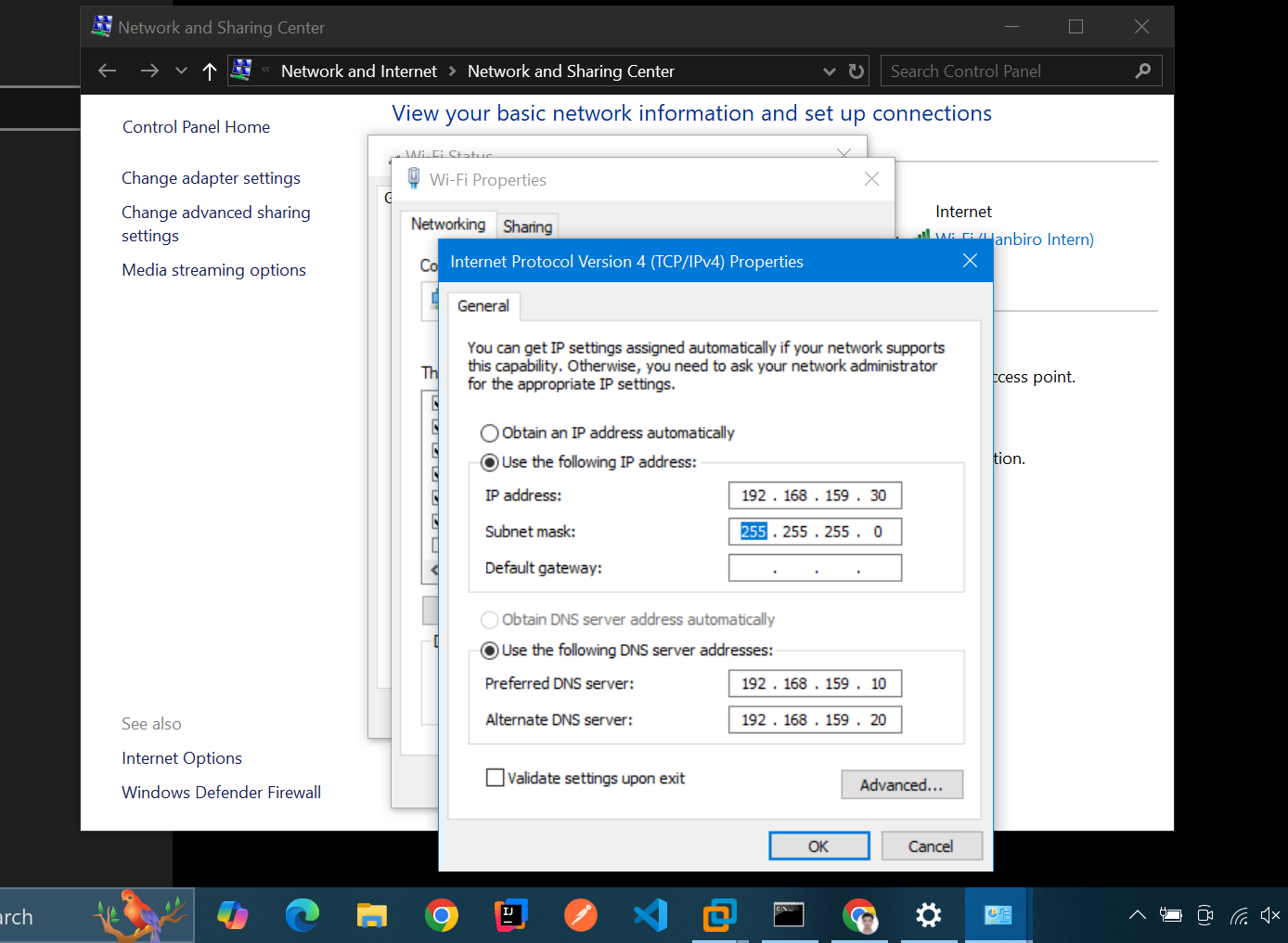


b6: try ‘ping ns1.hanbiro.shop’ and ‘nslookup ns1.hanbiro.shop’



**# Check with VM3 Window 10 - Client**

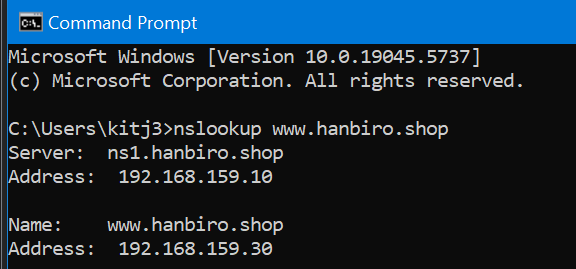
b1: setting network and sharing center



check IP & preferred DNS server address

b2: checking with

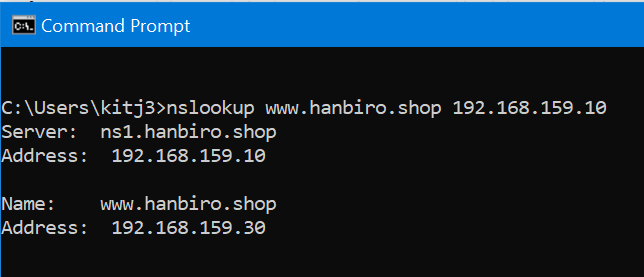
* check domain name resolution ‘nslookup www.hanbiro.shop’:



* check DNS query master and slave:

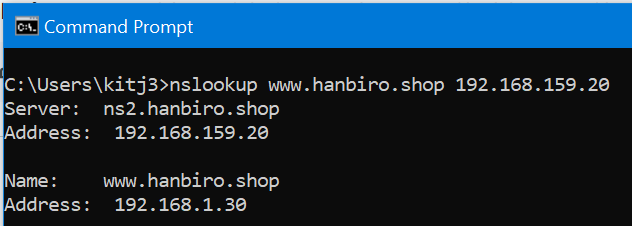
master:

nslookup www.hanbiro.shop 192.168.159.10



slave:

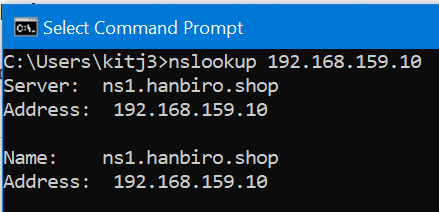
nslookup www.hanbiro.shop 192.168.159.20



* check reverse DNS:

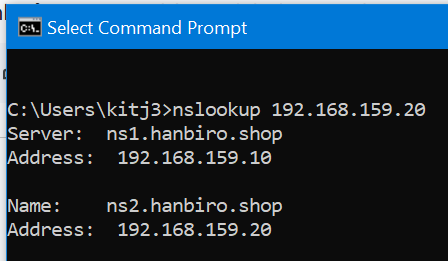
master:

nslookup 192.168.159.10

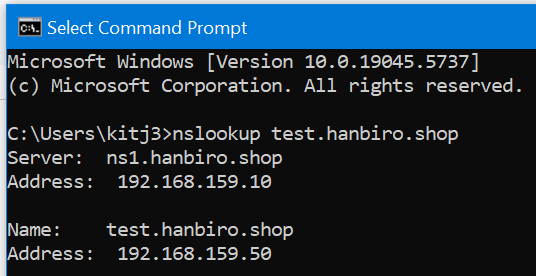


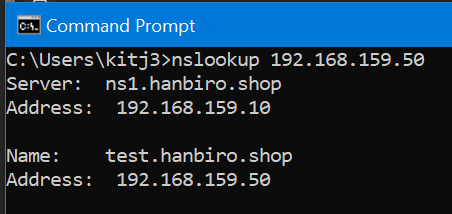
nslookup:

nslookup 192.168.159.20



* test $TTL 10s





**# Other Practice**

**Question: Purpose of settings long & short TTL values.**

**Answer:**

TTL (time to live) defines how long DNS records are cached by resolver before refreshing

Short TTL:

* fast update when DNS records change (change IP, ...)
* useful for testing or temporary configuration
* increase DNS traffic due to frequent lookups

Long TTL:

* reduce DNS queries, improve performance
* good for stable, rarely changed domain
* slow to reflect DNS changes

**Question: How to set if there are more than 2 MX records?**

**Answer:**

MX (mail exchange) record define mail server for receiving emails. Multiple MX records allow failover

Ex:

@ IN MX 10 mail1.hanbiro.shop.

@ IN MX 20 mail2.hanbiro.shop.

@ IN MX 30 mail-backup.hanbiro.shop.

* lower number = higher priority
* mail server tries in order: 10 -> 20 -> 30
* each target mail server must have A or AAAA record

**Question: Role of Rndc when configuring Master, Slave and way to rndc sync**

**Answer:**

rndc (remote name daemon control) is a command line tool to manage BIND without restarting the server

use cases:

* rndc reload reload zone / config files
* rndc flush clear caches
* rndc status show named status
* rndc notify hanbiro.shop trigger zone update to Slave

master - slave setup:

* after update zone and serial on Master, user ‘rndc notify’
* on Slave, use ‘rndc reload’ to sync zone immediately
* require proper ‘allow-transfer’ setting in named.conf

1. **MAIL**

**+ Mail server Basic**

**Question: What is a mail server?**

**Answer:**

mail server is a software system that sends, receives, stores and forwards emails between users over a network. it operates using protocols:

* SMTP (simple mail transfer protocol): for sending and forwarding outgoing email
* POP3 (post office protocol v3) / IMAP (internet message access protocol): for retrieving incoming email

There are 2 main types of mail server:

* SMTP server: delivers email to other server
* POP3 / IMAP server: stores emails and allows users to access them

Mail server works closely with DNS resolve domain names and locate the appropriate destination server via Mail Exchange record

**Question: Sending & Receiving Mail Process of (to be done in conjunction with DNS Query Process)**

**Answer:**

***Sending Mail Process:***

* **email composition:** the sender writes an email using mail client (outlook, thunderbird, gmail, …)
* **DNS query:** the sender’s mail server needs to find the destination server for the recipient’s domain (exam.com, …). It queries the DNS system to find Mail Exchange record of that domain
* **SMTP communication:** once the MX record is resolved to an IP address, the sender’s server establishes a connection using SMTP and transfer the email to recipient’s mail server
* **message storage:** the recipient’s mail server stores the email in the appropriate mailbox, waiting for the user to retrieve it

***Receiving Mail Process:***

* **mail client connection:** the recipient’s mail application (outlook, thunderbird, gmail, …) connect to their mail server. This server is already configured and doesn’t require DNS query at this point
* **retrieving email via IMAP / POP3:**
* IMAP: email stays on the server, allowing access from multiple devices
* POP3: email are downloaded to the device and usually removed from the server

**Question: How to create a db file using ‘*makemap’* such as virtusertable, access, ...**

**Answer:**

In sendmail configuration, .db files are used to store mappings for features like:

* virtusertable: Maps virtual email addresses to real ones
* access: Controls which IPs/domains can send mail
* mailertable: Defines mail routing rules for domains

Steps to create .db file:

* create a plain text mapping file: ex - /etc/mail/virtusertable

contact@yourdomain.com support@destination.com

sales@yourdomain.com salesbox@destination.com

-> this file maps virtual email address to real destination address

* generate .db file using makemap

makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable

-> create a file named /etc/mail/virtusertable.db

**Question: Meaning of each file, virtusertable, access, domaintable, mailertable?**

**Answer:**

**virtusertable**

this file is used to redirect or alias virtual email addresses (ex: info@domain.com) to actual mailbox destinations (either local or external). It allows one mail server to handle emails for multiple domains or users by remapping them

ex:

info@abc.com realuser1@localdomain.com

support@xyz.com support123@gmail.com

**access**

this file defines rules to control who can send mail to or relay through the server. It helps protect the server from spam and unauthorized use by allowing or rejecting access based on IP address or domain

ex:

191.168.159.100 RELAY

spamdomain.com REJECT

trusted.com OK

**domaintable**

this file is used to rewrite or alias domain names in email addresses. It’s useful when consolidating multiple domains or migrating users without changing their email addresses manually

ex: oldkiet.com newkiet.com

-> an email sent *user@oldkiet.com* will be rewritten to *user@newkiet.com*

**mailertable**

this file tells Sendmail ‘how to route email for specific domain’, such as sending them via a smart host or delivering locally. It defines the transport mechanism and destination for domain based mail delivery

ex:

ex.com smtp:[mail.ex.com]

internal.net local:

**Question: Building a sendmail.cf with the m4 of sendmail.cf and sendmail.mc.**

**Answer:**

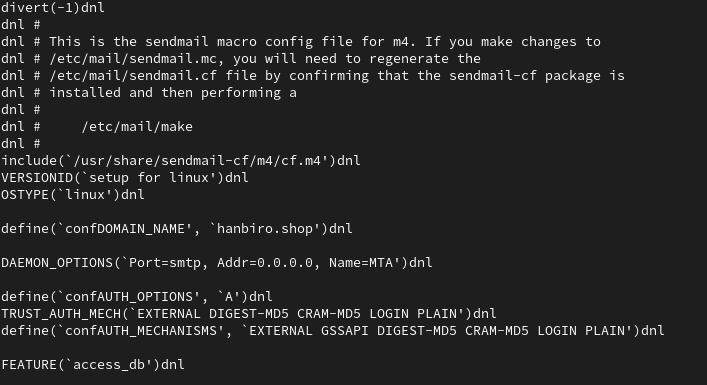
* install: sudo dnf install sendmail sendmail-cf m4 -y
* sendmail: main server program for handling emails
* sendmail-cf: configuration package for sendmail, allows to generate sendmail.cf file from sendmail.mc
* m4: macro preprocessor to generate final configuration file sendmail.cf
* configure file ‘sudo nano /etc/mail/sendmail.mc’:

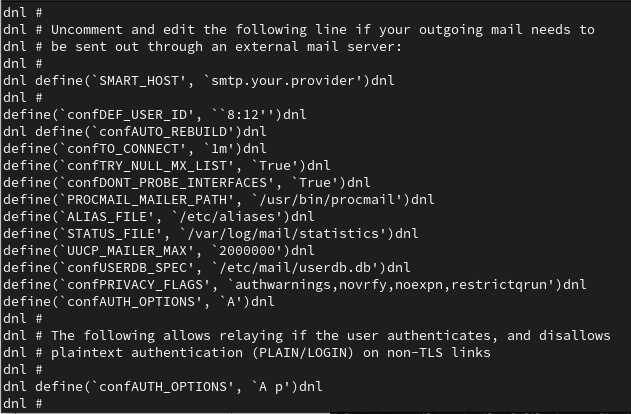
define hostname, allow receiving email from outside

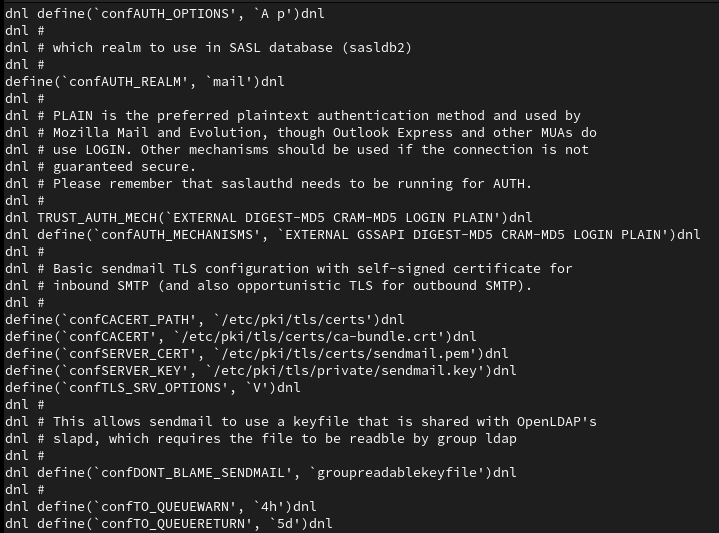
allow authentication even when the connection is not encrypted

list of "trusted" authentication methods

list of authentication mechanisms sendmail supports and accepts from clients





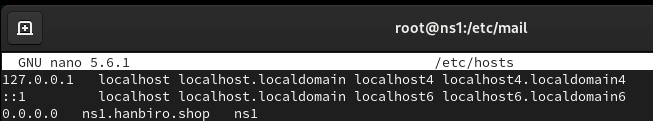


* use m4 to create sendmail.cf file

cd /etc/mail

sudo m4 sendmail.mc > sendmail.cf

* set-hostname with ‘sudo hostnamectl set-hostname ns1.hanbiro.shop’ command line. Update the /etc/hosts file to include the fully qualified domain name:



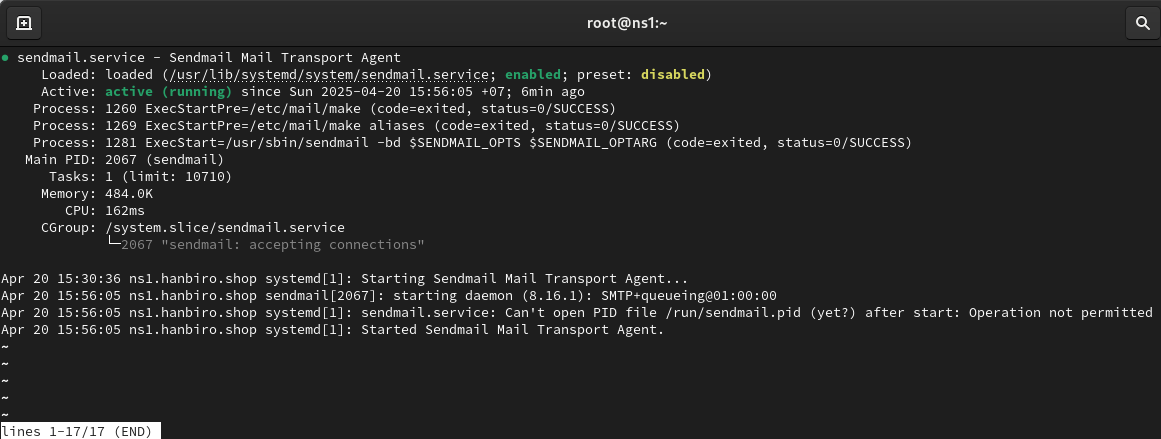
… and reboot

* restart the Sendmail service to apply the changes

sudo systemctl enable sendmail

sudo systemctl start sendmail

sudo systemctl status sendmail



**+ Mail server Practice**

**Question: Meaning of each file: virtusertable, access, domaintable, mailertable?**

**Answer:**

* */etc/mail/virtusertable* - map email address to local user. Used to redirect virtual email address to real Linux account
* */etc/mail/access* - control which IP addresses can relay mail through the server
* */etc/mail/domaintable* - map alias domain name to the canonical domain name
* /etc/mail/mailertable: forward mail for certain domains to a specified SMTP server

**Question: Building Sendmail using the domain purchased.**

**Answer: Building on VM2: DNS Slave server**

* Install Sendmail and required tools:

sudo dnf install sendmail sendmail-cf m4 -y

* Set the hostname for the server:

sudo hostnamectl set-hostname mail.hanbiro.shop

\*add the following line to map internal IP address to hostname in

/etc/hosts (if not already there):

192.168.159.20 mail.hanbiro.shop mail

* configure the /etc/mail/sendmail.mc file:

nano /etc/mail/sendmail.mc

dnl MASQUERADE\_AS(`mydomain.com')dnl

→ MASQUERADE\_AS(`hanbiro.shop')dnl

dnl FEATURE(masquerade\_envelope)dnl

→ FEATURE(masquerade\_envelope)dnl

dnl FEATURE(masquerade\_entire\_domain)dnl

→ FEATURE(masquerade\_entire\_domain)dnl

* Add domain to local-host-names, This file tells Sendmail which domains it is responsible for handling email. Without it, Sendmail will reject emails coming from that domain:

nano /etc/mail/local-host-names

\*add domains:

hanbiro.shop

mail.hanbiro.shop

* generate sendmail.cf from sendmail.mc using m4:

cd /etc/mail

sudo m4 sendmail.mc > sendmail.cf

* create internal user to test receiving mail

sudo useradd testuser

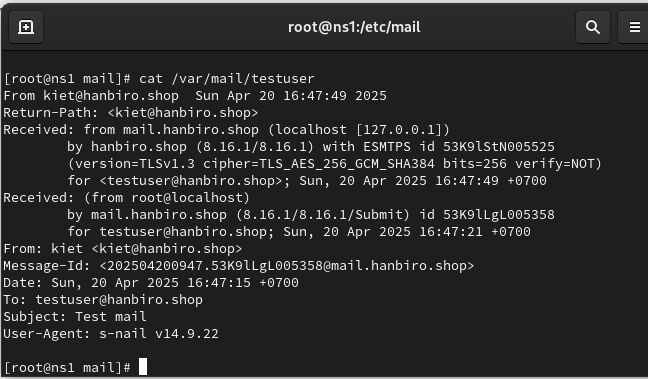
sudo passwd testuser (Tuankiet3172@3)

* send internal test email (download package mail)

echo "Hello from hanbiro.shop" | mail -s "Test mail" testuser@hanbiro.shop

check:

sudo cat /var/mail/testuser



**Question: Making the uid and password of the Linux account into the mail server account using saslauthd.**

**Answer:**

* install: dnf install cyrus-sasl cyrus-sasl-plain cyrus-sasl-md5 cyrus-sasl-lib -y
* enable and start saslauthd:

systemctl enable --now saslauthd

systemctl start saslauthd

systemctl status saslauthd

* check if saslauthd can authenticate Linux users (0: OK "Success."):

testsaslauthd -u testuser -p Tuankiet3172@3 -s smtp

* create /etc/mail/authinfo file:

AuthInfo: "localhost" "U:root" "I:testuser" "P:Tuankiet3172@3"

and then, create file authinfo.db:

makemap hash /etc/mail/authinfo < /etc/mail/authinfo

* open the file /etc/mail/sendmail.mc, add (or edit) the following lines if they don't exist:

define(`confAUTH\_OPTIONS', `A')dnl

TRUST\_AUTH\_MECH(`PLAIN LOGIN')dnl

define(`confAUTH\_MECHANISMS', `PLAIN LOGIN')dnl

DAEMON\_OPTIONS(`Port=smtp, Name=MTA')dnl

FEATURE(`authinfo')dnl

* recompile sendmail.cf from sendmail.mc:

m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

* modify the /etc/sysconfig/saslauthd file:

MECH=pam

* create socket directory:

mkdir -p /var/spool/postfix/var/run/saslauthd

chown -R sasl:sasl /var/spool/postfix/var/run/saslauthd

* restart service: systemctl restart sendmail
* check authentication:

testsaslauthd -u testuser -p Tuankiet3172@3 -s smtp

-> result: 0: OK "Success."

**Question: Allow RELAY for 121.254.171.248 on sendmail**

**Answer:**

* vi /etc/mail/access
* add the following line at the end of the file:

121.254.171.248 RELAY

* compile the access.db file. After saving the access file, rebuild it:

makemap hash /etc/mail/access < /etc/mail/access

* restart service: systemctl restart sendmail
* check if the /etc/mail/access.db file has the IP RELAY line:

strings /etc/mail/access.db | grep 121



**Question:Set sendmail to run witn 0.0.0.0:25 for maile xternal communication**

**Answer:**

* vi /etc/mail/sendmail.mc

\*Comment out this line by adding dnl at the beginning, so that Sendmail can listen to all IPs, not just 127.0.0.1:

dnl DAEMON\_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl

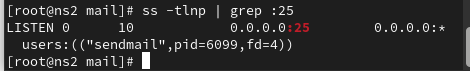
→ DAEMON\_OPTIONS(`Port=smtp,Addr=0.0.0.0, Name=MTA')dnl

* recompile sendmail.cf from sendmail.mc:

m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

* restart service: systemctl restart sendmail
* check Sendmail is listening properly:

ss -tunlp | grep :25



**Question: Set to send and receive mail via actual account such as Outlook or Thunderbird**

**Question: Capture the test result of mail sending and receiving with large portal using domain hanbirotest@xxx.shop**

**Question: Capture a record of mail sending and receiving on /var/log/maillog**

