Національний Технічний Університет України

«Київський Політехнічний Інститут Імені Ігоря Сікорського»

Факультет Інформатики Та Обчислювальної Техніки

Кафедра Обчислювальної Техніки

Лабораторна робота №1

Служба доменних імен (DNS)

Виконала

Шпак Діана

Група: ІО-з21мп

Київ 2023

2. Завдання на роботу.

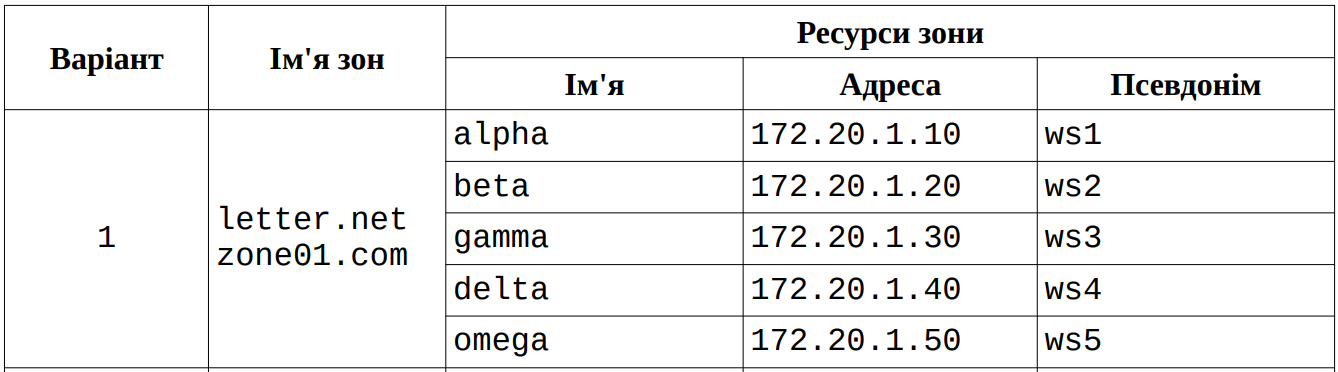
2.1. Встановити та налаштувати первинний та резервний сервери для двох прямих та зворотної зон, які відповідають наступним вимогам: - в прямих зонах обов'язково мають бути записи типів: SOA, NS, A, CNAME, які відповідають варіанту завдання; - в зворотній зоні обов'язково мають бути записи типів: SOA, NS, PTR, які відповідають варіанту завдання.

2.2. Перевірити результат передачі (transfer) зон на резервному сервері.

2.3. Перевірити працездатність первинного та резервного серверів для всіх зон за допомогою діагностичних утиліт, наприклад: nslookup, dig, host.

2.4. Рекомендується використовувати наступне програмне забезпечення: - DNS-сервер: bind9.

ВАРІАНТ №1



Для виконання даного завдання нам потрібно мати 2 днс сервера і 5 серверів ресурсів, які ми повинні розподілити між 2 зонами, відповідно для запуску даних серверів локально ми будемо використовувати віртуал бокс і вагрант який дозволяє описати наші ресурси скриптами відповідно дестроїти і запускати їх з консолі, що дасть нам можливість виконувати завдання набагато зручніше

Пишемо вагрант скрипт для запуску віртуальних машин на яких будим встановлювати днс сервери скрипт буде мати наступний вигляд

config.vm.define "dnsfirst" do |dnsfirst|

dnsfirst.vm.box\_download\_insecure = true

dnsfirst.vm.box = "ubuntu/xenial64"

dnsfirst.vm.network :private\_network, ip: "192.168.0.10"

dnsfirst.vm.network "forwarded\_port", guest: 5353, host: 5353

dnsfirst.vm.hostname = "dnsfirst"

dnsfirst.vm.post\_up\_message = "Run vm for dnsfirst"

dnsfirst.vm.provider :virtualbox do |vb|

vb.memory = "1024"

end

dnsfirst.vm.provision "shell" do |s|

s.path = 'dns\_first.sh'

end

end

config.vm.define "dnsreserve" do |dnsreserve|

dnsreserve.vm.box\_download\_insecure = true

dnsreserve.vm.box = "ubuntu/xenial64"

dnsreserve.vm.network :private\_network, ip: "192.168.0.20"

dnsreserve.vm.network "forwarded\_port", guest: 5353, host: 5354

dnsreserve.vm.hostname = "dnsreserve"

dnsreserve.vm.post\_up\_message = "Run vm for dnsreserve"

dnsreserve.vm.provider :virtualbox do |vb|

vb.memory = "1024"

end

dnsreserve.vm.provision "shell" do |s|

s.path = 'dns\_reserve.sh'

end

end

Так як у нас в завданні не вказані айпі адреси для днс серверів обираємо їх будь які, імена також задаємо самостійно, все налаштування основного і резервного днс виносимо в баш скрипт який запуститься автоматично при старті віртуальної машини.  
 Після того коли у нас є скрипти для підняття віртуальних машин переходимо до виконання самого завдання на лабораторну роботу а саме до пункту

**2.1. Встановити та налаштувати первинний та резервний сервери для двох прямих та зворотної зон, які відповідають наступним вимогам:**

- в прямих зонах обов'язково мають бути записи типів: SOA, NS, A, CNAME, які відповідають варіанту завдання;

- в зворотній зоні обов'язково мають бути записи типів: SOA, NS, PTR, які відповідають варіанту завдання.

для цього було написано 2 скрипти які запускаються на віртуальних машинах які ми виділили під наші днс сервери, скрипт dns\_first.sh для налаштування основного днс і скрипт dns\_reserve.sh  дані скрипти запускаються на відповідних машинах автоматично вагрантом під час запуску самої машини і виконують всі потрібні нам налаштування

Скрипт dns\_first.sh має наступний вигляд:

#!/bin/bash

echo "DNS first"

sudo apt-get update

sudo apt-get -y install bind9 bind9utils bind9-doc

sudo dpkg-preconfigure -f noninteractive -p critical

sudo dpkg --configure -a

sudo sed -i 's/OPTIONS="-u bind"/OPTIONS="-u bind -4"/' /etc/default/bind9

sudo systemctl restart bind9

# Так як по дефолту логи в демона Bind9 пишуться в /var/log/syslog, першим кроком ми це змінимо

# для цього додаємо секцію logging в файл конфігурації /etc/bind/named.conf, це нам дасть

# можливість виділити логи з демона Bind9 в окремий файл

cat <<EOT >> /etc/bind/named.conf

logging {

channel bind.log {

file "/var/lib/bind/bind.log" versions 10 size 20m;

severity notice;

print-category yes;

print-severity yes;

print-time yes;

};

category queries { bind.log; };

category default { bind.log; };

category config { bind.log; };

};

EOT

#Очищаємо файл конфігурації

cat /dev/null > /etc/bind/named.conf.options

cd /etc/bind/

#Далі над існуючим блоком options створюємо новий блок ACL (список контролю доступу)

#під назвою trusted. Саме тут ми створимо список клієнтів, для яких ми дозволятимемо

#рекурсивні DNS-запити (тобто запити від ваших серверів, що знаходяться в тому самому

#центрі обробки даних, що й ns1). За допомогою нашого прикладу приватних IP-адрес

#ми додамо dns1, dns2, ws1, ws2, ws3, ws4 і ws5 до нашого списку надійних клієнтів:

cat <<EOT >> named.conf.options

acl "trusted" {

192.168.0.10; # dns1

192.168.0.20; # dns2

172.20.1.10; # ws1

172.20.1.20; # ws2

172.20.1.30; # ws3

172.20.1.40; # ws4

172.20.1.50; # ws5

};

EOT

#Далі налаштовуємо блок options

cat <<EOT >> named.conf.options

options {

directory "/var/cache/bind";

recursion yes; # enables resursive queries

allow-recursion { trusted; }; # allows recursive queries from "trusted" clients

listen-on port 5353 { 192.168.0.10;}; # dns1 private IP address - listen on private network only

allow-transfer { none; }; # disable zone transfers by default

forwarders {

8.8.8.8;

8.8.4.4;

};

};

EOT

#Далі виконуємо налаштування локального конфіга, в якому задаємо зони для

#прямого і зворотнього перегляду, оскільки у нас за варіантом дві зони

# і айпі адреси які ми обрали для наших днс серверів і задаються в різних

# просторах IP-адрес то відповідно у нас буде 4 зони 2 для прямого перегляду

# і 2 для зворотнього

cat /dev/null > /etc/bind/named.conf.local

cat <<EOT >> named.conf.local

zone "zone01.com" {

type master;

file "/etc/bind/db.zone01.com"; # zone file path

allow-transfer { 192.168.0.20; }; # dns2 private IP address - secondary

also-notify { 192.168.0.20; }; # dns2 private IP address - secondary

};

zone "168.192.in-addr.arpa" {

type master;

file "/etc/bind/db.168.192"; # zone file path

allow-transfer { 192.168.0.20; }; # dns2 private IP address - secondary

also-notify { 192.168.0.20; }; # dns2 private IP address - secondary

};

zone "letter.net" {

type master;

file "/etc/bind/db.letter.net"; # zone file path

allow-transfer { 192.168.0.20; }; # dns2 private IP address - secondary

also-notify { 192.168.0.20; }; # dns2 private IP address - secondary

};

zone "20.172.in-addr.arpa" {

type master;

file "/etc/bind/db.20.172"; # zone file path

allow-transfer { 192.168.0.20; }; # dns2 private IP address - secondary

also-notify { 192.168.0.20; }; # dns2 private IP address - secondary

};

EOT

# Далі створюємо вже самі файли конфігурації самих зон для прямого перегляду, це місце де ми будемо

# оприділяти ДНС записи, CNAME, SOA

# Файли зони для зворотного перегляду є місцем, де ми будемо визначати PTR

# записів DNS для перегляду DNS, коли DNS отримує запит для IP-адреси, наприклад,

# “192.168.0.10”, вона буде шукати файл (файли) зони для зворотного перегляду, щоб отримати

# відповідне повне доменне ім'я dns1.zone01.com

# наші ресурси ми розділили по 2 зонах в зоні zone01.com маємо наступний список серверів

# dns1, dns2, alpha, beta а в зоні letter.net - gamma, delta, omega

sudo cp /etc/bind/db.local /etc/bind/db.zone01.com

cd /etc/bind/

cat /dev/null > /etc/bind/db.zone01.com

cat <<EOT >> db.zone01.com

\$TTL 604800

@ IN SOA dns1.zone01.com. admin.zone01.com. (

3 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

dns1 IN A 192.168.0.10

dns2 IN A 192.168.0.20

alpha IN A 172.20.1.10

beta IN A 172.20.1.20

ws1 IN CNAME alpha

ws2 IN CNAME beta

; name servers

@ IN NS dns1

IN NS dns2

IN NS alpha

IN NS beta

EOT

sudo cp /etc/bind/db.local /etc/bind/db.168.192

cd /etc/bind/

cat /dev/null > /etc/bind/db.168.192

cat <<EOT >> db.168.192

\$TTL 604800

@ IN SOA zone01.com. admin.zone01.com. (

3 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

; name servers

IN NS dns1.zone01.com.

IN NS dns2.zone01.com.

; PTR Records

10.0.168.192.in-addr.arpa. 33692 IN PTR dns1.zone01.com. ;192.168.0.10

20.0.168.192.in-addr.arpa. 33692 IN PTR dns2.zone01.com. ;192.168.0.20

EOT

sudo cp /etc/bind/db.127 /etc/bind/db.letter.net

cd /etc/bind/

cat /dev/null > /etc/bind/db.letter.net

cat <<EOT >> db.letter.net

\$TTL 604800

@ IN SOA dns1.letter.net. admin.letter.net. (

3 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

gamma IN A 172.20.1.30

delta IN A 172.20.1.40

omega IN A 172.20.1.50

ws3 IN CNAME gamma

ws4 IN CNAME delta

ws5 IN CNAME omega

;name servers

@ IN NS gamma

IN NS delta

IN NS omega

EOT

sudo cp /etc/bind/db.local /etc/bind/db.20.172

cd /etc/bind/

cat /dev/null > /etc/bind/db.20.172

cat <<EOT >> db.20.172

\$TTL 604800

@ IN SOA zone01.com. admin.zone01.com. (

3 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

; name servers

IN NS alpha.zone01.com.

IN NS beta.zone01.com.

IN NS gamma.letter.com.

IN NS delta.letter.com.

IN NS omega.letter.com.

; PTR Records

10.1.20.172.in-addr.arpa. IN PTR alpha.zone01.com. ;172.20.1.10

20.1.20.172.in-addr.arpa. IN PTR beta.zone01.com. ;172.20.1.20

30.1.20.172.in-addr.arpa. IN PTR gamma.letter.com. ;172.20.1.30

40.1.20.172.in-addr.arpa. IN PTR delta.letter.com. ;172.20.1.40

50.1.20.172.in-addr.arpa. IN PTR omega.letter.com. ;172.20.1.50

EOT

#Після створення всіх файлів конфігурації, і файлів зон робимо рестарт демона і

#і перевірку конфігів які ми створили

sudo systemctl restart bind9

sudo named-checkconf -z

sudo named-checkzone zone01.com /etc/bind/db.zone01.com

sudo named-checkzone letter.net /etc/bind/db.letter.net

sudo systemctl restart bind9

sudo ufw allow Bind9

скрипт dns\_reserve.sh  має наступний вигляд:

#!/bin/bash

echo "DNS reserve"

sudo apt-get update

sudo apt-get -y install bind9 bind9utils bind9-doc

sudo dpkg-preconfigure -f noninteractive -p critical

sudo dpkg --configure -a

sudo sed -i 's/OPTIONS="-u bind"/OPTIONS="-u bind -4"/' /etc/default/bind9

sudo systemctl restart bind9

# Налаштування додаткового сервіса виконується набагато простіше

# Так як по дефолту логи в демона Bind9 пишуться в /var/log/syslog, першим кроком ми це змінимо

# для цього додаємо секцію logging в файл конфігурації /etc/bind/named.conf, це нам дасть

# можливість виділити логи з демона Bind9 в окремий файл

cat <<EOT >> /etc/bind/named.conf

logging {

channel bind.log {

file "/var/lib/bind/bind.log" versions 10 size 20m;

severity notice;

print-category yes;

print-severity yes;

print-time yes;

};

category queries { bind.log; };

category default { bind.log; };

category config { bind.log; };

};

EOT

# У верхній частині файлу /etc/bind/named.conf.options додайте ACL із приватними IP-адресами всіх ваших довірених серверів:

cat /dev/null > /etc/bind/named.conf.options

cd /etc/bind/

cat <<EOT >> named.conf.options

acl "trusted" {

192.168.0.10; # dns1

192.168.0.20; # dns2

172.20.1.10; # ws1

172.20.1.20; # ws2

172.20.1.30; # ws3

172.20.1.40; # ws4

172.20.1.50; # ws5

};

EOT

cat <<EOT >> named.conf.options

options {

directory "/var/cache/bind";

recursion yes;

allow-recursion { trusted; };

listen-on port 5353 { 192.168.0.20; 127.0.0.1;}; # dns1 private IP address - listen on private network only

allow-transfer { none; }; # disable zone transfers by default

forwarders {

8.8.8.8;

8.8.4.4;

};

};

EOT

# Визначаємо slave-зони, що відповідають master-зонам основного DNS-сервера.

# Тип використовується slave, у файлі відсутня шлях, і існує директива masters,

# яка має бути налаштована на приватну IP-адресу основного DNS-сервера.

# ми кілька зон для зворотного перегляду на основному DNS-сервері, їх потрібно

# також додати:

cat /dev/null > /etc/bind/named.conf.local

cat <<EOT >> named.conf.local

zone "zone01.com" {

type slave;

file "db.zone01.com"; # zone file path

masters port 5353 { 192.168.0.10; }; # dns1 private IP address

};

zone "168.192.in-addr.arpa" {

type slave;

file "db.168.192"; # zone file path

masters port 5353 { 192.168.0.10; }; # dns1 private IP address

};

zone "letter.net" {

type slave;

file "db.letter.net"; # zone file path

masters port 5353 { 192.168.0.10; }; # dns1 private IP address

};

zone "20.172.in-addr.arpa" {

type slave;

file "db.20.172"; # zone file path

masters port 5353 { 192.168.0.10; }; # dns1 private IP address

};

EOT

#Після створення всіх файлів конфігурації, і файлів зон робимо рестарт демона і

#перевірку конфігів які ми створили

sudo systemctl restart bind9

sudo named-checkconf -z

sudo ufw allow Bind9

**2.2. Перевірити результат передачі (transfer) зон на резервному сервері.**

**у нас 2 днс сервери один основний і 1 резервний:**

dns1.zone01.com - 192.168.0.10

dns2.zone01.com - 192.168.0.20

такоє ми маємо 5 серверів ресурсів які розподілені по зонах:

ws1 alpha.zone01.com. 172.20.1.10

ws2 beta.zone01.com. 172.20.1.20

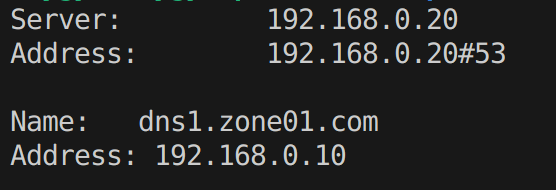
ws3 gamma.letter.com. 172.20.1.30

ws4 delta.letter.com. 172.20.1.40

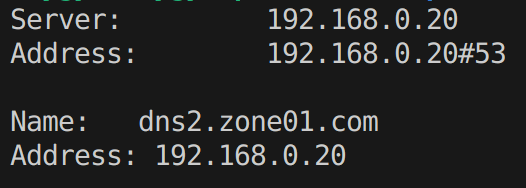
ws5 omega.letter.com. 172.20.1.50

перевіряємо роботу для прямої зони

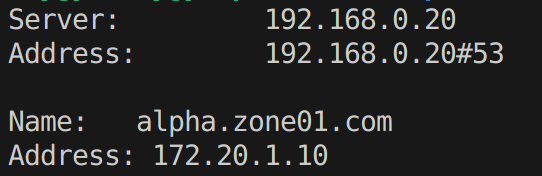
nslookup dns1.zone01.com 192.168.0.20



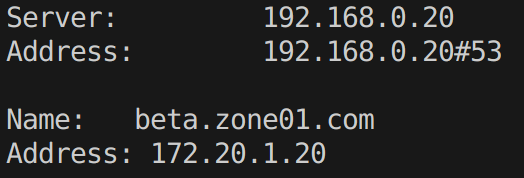
nslookup dns2.zone01.com 192.168.0.20



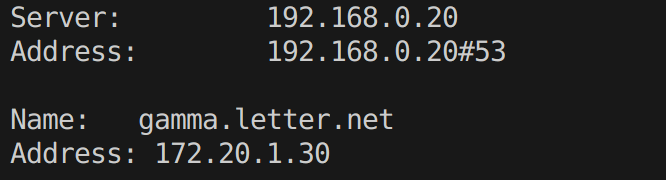
nslookup alpha.zone01.com 192.168.0.20



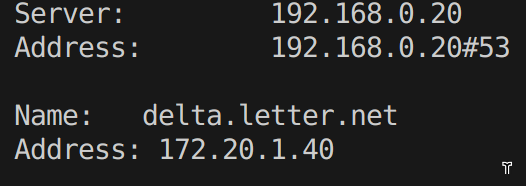
nslookup beta.zone01.com 192.168.0.20



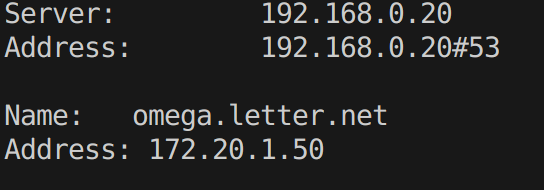
nslookup gamma.letter.net 192.168.0.20



nslookup delta.letter.net 192.168.0.20



nslookup omega.letter.net 192.168.0.20



**2.3. Перевірити працездатність первинного та резервного серверів для всіх зон за допомогою діагностичних утиліт, наприклад: nslookup, dig, host.**

в попередньому пункті ми перевірили всі прямі зони для резервного сервера днс  
діагностуємо роботу серверів за допомогою утиліти діг

dig @192.168.0.10 dns1.zone01.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @192.168.0.10 dns1.zone01.com

; (1 server found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32126

;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;dns1.zone01.com. IN A

;; ANSWER SECTION:

dns1.zone01.com. 604800 IN A 192.168.0.10

;; AUTHORITY SECTION:

zone01.com. 604800 IN NS dns1.zone01.com.

zone01.com. 604800 IN NS beta.zone01.com.

zone01.com. 604800 IN NS alpha.zone01.com.

zone01.com. 604800 IN NS dns2.zone01.com.

;; ADDITIONAL SECTION:

beta.zone01.com. 604800 IN A 172.20.1.20

dns2.zone01.com. 604800 IN A 192.168.0.20

alpha.zone01.com. 604800 IN A 172.20.1.10

;; Query time: 0 msec

;; SERVER: 192.168.0.10#53(192.168.0.10) (UDP)

;; WHEN: Sun Jun 18 17:27:32 EEST 2023

;; MSG SIZE  rcvd: 180

dig @192.168.0.10 omega.letter.net

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @192.168.0.10 omega.letter.net

; (1 server found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39640

;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;omega.letter.net. IN A

;; ANSWER SECTION:

omega.letter.net. 604800 IN A 172.20.1.50

;; AUTHORITY SECTION:

letter.net. 604800 IN NS omega.letter.net.

letter.net. 604800 IN NS delta.letter.net.

letter.net. 604800 IN NS gamma.letter.net.

;; ADDITIONAL SECTION:

delta.letter.net. 604800 IN A 172.20.1.40

gamma.letter.net. 604800 IN A 172.20.1.30

;; Query time: 4 msec

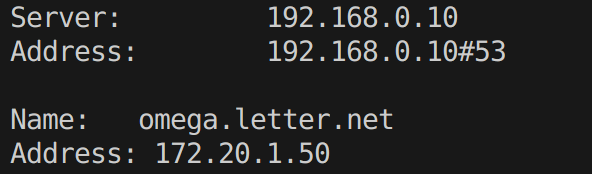
;; SERVER: 192.168.0.10#53(192.168.0.10) (UDP)

;; WHEN: Sun Jun 18 17:29:05 EEST 2023

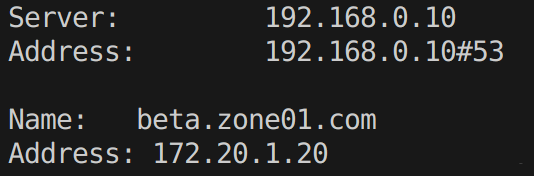
;; MSG SIZE  rcvd: 147

далі потрібно ще протестувати зворотні зони для резервного і основного сервера сервера,і прямі зони для основного

nslookup omega.letter.net 192.168.0.10



nslookup beta.zone01.com 192.168.0.10



як видно прямі зони працюють на обох серверах, тепер протестуємо зворотні

основний сервер

nslookup 172.20.1.30 192.168.0.10

30.1.20.172.in-addr.arpa name = gamma.letter.com.

nslookup 172.20.1.10 192.168.0.10

10.1.20.172.in-addr.arpa name = alpha.zone01.com.

nslookup 172.20.1.30 192.168.0.20

30.1.20.172.in-addr.arpa name = gamma.letter.com.

nslookup 172.20.1.10 192.168.0.20

10.1.20.172.in-addr.arpa name = alpha.zone01.com.

nslookup 192.168.0.10 192.168.0.20

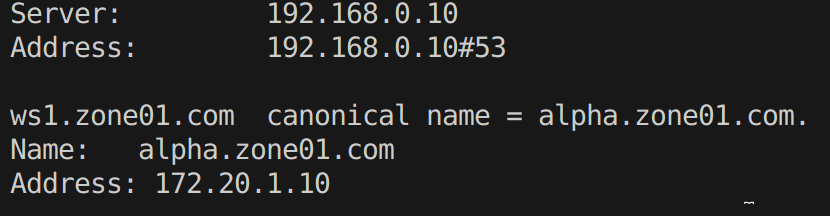
10.0.168.192.in-addr.arpa name = dns1.zone01.com.

nslookup 192.168.0.10 192.168.0.10

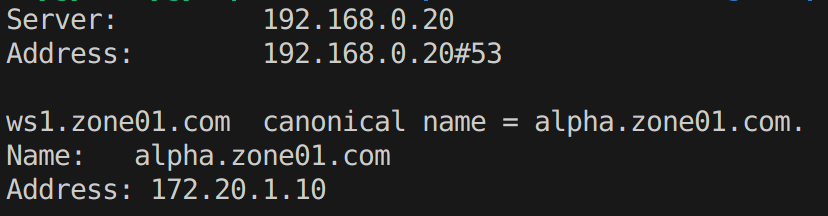
10.0.168.192.in-addr.arpa name = dns1.zone01.com.

як бачимо всі зворотні зони працюють і на зворотньому і остновному серверах, далі залишилось протестувати тільки CNAME для наших хостів

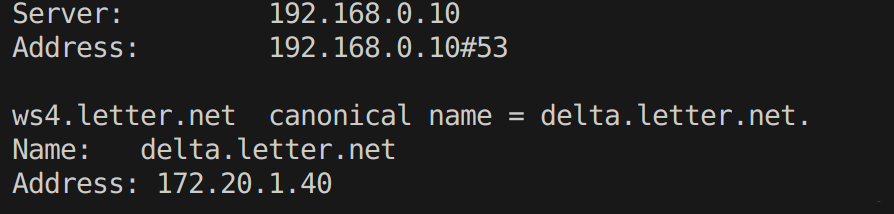
nslookup ws1.zone01.com 192.168.0.10



nslookup ws1.zone01.com 192.168.0.20



nslookup ws4.letter.net 192.168.0.10



nslookup ws4.letter.net 192.168.0.20

