ЛАБОРАТОРНА РОБОТА 3

Інсталяція та налаштування DNS серверу на ОС Linux

Мета: ознайомитись з основами налаштування DNS серверу на базі ОС Linux

Теоретичні відомості

Комп'ютери, що мають підключення до комп'ютерної мережі часто називають хостами (від англ. host), а для їх ідентифікації у мережі використовують *імена хоста*. Ім'я хоста – це певний мнемонічний, зручний для сприйняття людиною запис. Наприклад, znu.edu.ua, cnn.com, www.yahoo.com, gaia.cs.umass.edu, surf.eurecom.fr. У той же час для комп'ютерів прийнятними є тільки числові форми ідентифікації. Зараз для ідентифікації хостів в Інтернет широко використовується IP-адреса протоколу IPv4 – сукупність чотирьох однобайтових чисел з певною ієрархічною структурою. Символьне ім'я хоста є зручним для людини, вони легко запам'ятовується, але алгоритми обробки IP-адреси є більш простими для комп'ютерних систем, мереж та маршрутизаторів. Також важливим є те, що зв'язок (асоціація) між IP-адресою та символьним ім'ям робиться не назавжди. Зміна провайдера Інтернет або хостингової компанії буде змінювати IP-адресу але символьне (доменне) ім'я ресурсу, наприклад, вебсайту якоїсь організації скоріш за все буде залишатись тим самим, якщо, безумовно, ця організація не захоче його змінити. Крім того сам провайдер може інколи змінювати доступні йому діапазони IP-адрес, що також викликає зміну IP-адрес у споживачів його сервесів. Тому принципово важливим є здатність системи до зберігання символьної адреси при ймовірній зміни IP-адреси.

На початку створення та існування Internet (ARPANET) її зростання було помірним, тому реляція символьних імен хостів та IP-адреси могла централізовано контролюватись та підтримуватись Мережевим Інформаційним Центром (NIC – Network Information Center) за допомогою єдиного файлу (hosts.txt). Адміністратор будь-якого хоста або певної організації періодично робив копію цього файлу з використанням або електронного передавання файлів, або навіть якогось носія, наприклад, дискети. Нові ресурси додавались не часто, тому цей файл був невеликим і міг не змінюватись навіть декілька тижнів. Механізм визначення мережевої адреси за символьним ім'ям хоста полягав у звернені до файлу hosts.txt, свіжа копія якого переносилась у певний каталог на кожний комп'ютер, що підключався до мережі. Така система працювала достатньо добре поки ARPANET була порівняно невеликою мережею. Коли темп зростання та зміни складу мережі суттєво збільшились, частота звернення за оновленим файлом hosts.txt із NIC також суттєво зросла. Крім того стало необхідним відокремити управління локальними іменами і адресами у різних організаціях та компаніях, а також постійно вносити зміни до файлу hosts.txt. Тому централізована схема підтримки файлу із записами відповідності імен хостів та IP-адрес стала недостатньо практичною та повільною і потребувала зміни принципів у цьому питані. У 1983 році Пол Мокапетріс — науковець із Каліфорнійського університету, запропонував створити автоматизовану систему доменних імен DNS (Domain Name System).

Перші результати розробки DNS було опубліковано у 1983 році в RFC 882 та 883. Після експериментів з декількома реалізаціями, DNS було формально визначено у RFC 1034 та 1035 у 1987 році. Положення, специфікації, протоколи та принципи використання DNS продовжують розвиватись і у наш час, що закріплюється відповідними стандартами у документах RFC (https://datatracker.ietf.org/).

DNS базується на двох основних концепціях:

- на розподіленій базі даних, що зберігає узагальнені записи про ресурси мережі (resource records) та має децентралізоване управління;
- на певній схемі іменування, зо базується ієрархічно структурованих доменних іменах.

DNS є розподіленою базою даних. Це дозволяє локально контролювати окремі сегменти загальної бази даних. Дані у кожному сегменті мають доступ через мережу з використанням

технології клієнт-сервер. Адекватна продуктивність досягається через використання механізмів копіювання (replication) та кешування (caching).

Програми, що реалізують серверну частину DNS, звуться серверами імен (name servers). Сервер імен має інформацію про деякий сегмент загальної бази даних DNS, для якого він є повноважним сервером, та робить її доступною для клієнтів — програми вирішувачі (resolvers). Програми вирішувачі (resolvers) зазвичай надають бібліотечні функції, які генерують запити та насилають їх через мережу до серверів імен. DNS-перетворювач виконує дві основні функції. Функція *gethostbyname()* перетворює доменне ім’я у IP-адресу, а функція *gethostbyaddr()* перетворює IP-адресу у доменне ім’я. Перетворювач взаємодіє з одним або декількома DNS-серверами для виконання цих функцій від імені додатку. Для виконання цього процесу перетворювач повинен мати налаштування принаймні на один DNS-сервер. Для комп’ютера в мережі Internet зазвичай вказується список IP-адрес, кожна з яких відповідає певному DNS-серверу. Мережеві адміністратори намагаються розташовувати локальні DNS-сервери ближче до клієнтів, що використовують їх для запитів. Наприклад, в університеті комп’ютерна мережа, що об’єднує комп’ютери класів, кафедр, деканатів, відділів, також має у своєму складі локальні DNS-сервери, що оброблюють запити вирішувачів від цих комп’ютерів. Зазвичай провайдер Інтернет також розташовує свої DNS-сервери у мережі, до якої підключаються його клієнти.

Кожна одиниця даних розподіленої бази даних DNS індексується за ім’ям. Ці імена, за своєю суттю, є своєрідними вказівниками в інвертованому дереві простору доменних імен.

Кожний вузол (node) у просторі імен має власну мітку без крапки тому, що крапка (".") використовується в якості роздільників міток. Максимальна довжина мітки може складати 63 байта. Максимальна довжина доменного імені (сума усіх міток та роздільників) дорівнює 255 байтам. Кореневий домен має мітку нульової довжини (null). Повне доменне ім’я кожного вузлу у дереві — це послідовність міток на шляху від цього вузла до кореня (Рис.1). За згодою, мітки, що складають доменне ім’я читають зліва на право, починаючи з нижньої, найбільш віддаленої від кореня, і завершуючи самою верхньою, яка безпосередньо знаходиться перед коренем.

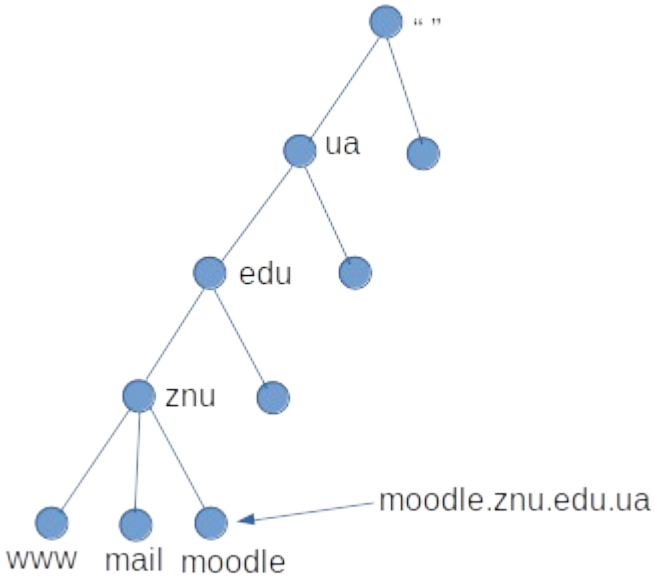


Рис.1. Приклад побудови доменного імені.

Для позначення кореневого домену в доменному імені вузла використовується символ крапки (".") наприкінці імені. Доменні імена, які закінчуються крапкою, звуться абсолютними доменними іменами (Рис.2).

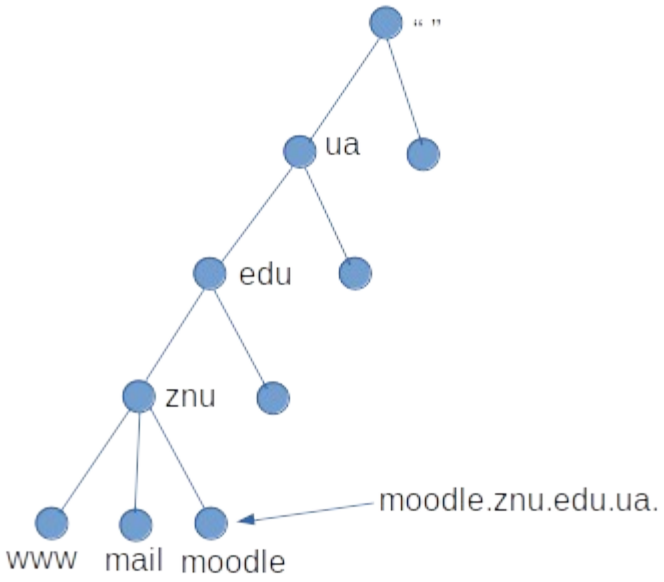


Рис.2. Приклад абсолютного доменного імені.

Абсолютне доменне ім’я зв’язане з коренем, тому воно однозначно специфікує місце знаходження вузла в ієрархії. Імена без крапки наприкінці інколи інтерпретуються як ті, що пов’язані з деяким доменом, який відрізняється від кореневого. Абсолютне доменне ім’я також зветься повністю кваліфікованим доменним ім’ям (fully-qualifed domain name або FQDN).

Домен - це деяке піддерево простору доменних імен. Доменне ім’я домену – це ім’я самого верхнього вузлу домену. Наприклад, верхнім вузлом домену "znu.edu.ua" є вузол, який має доменне ім’я "znu.edu.ua" (Рис.3).

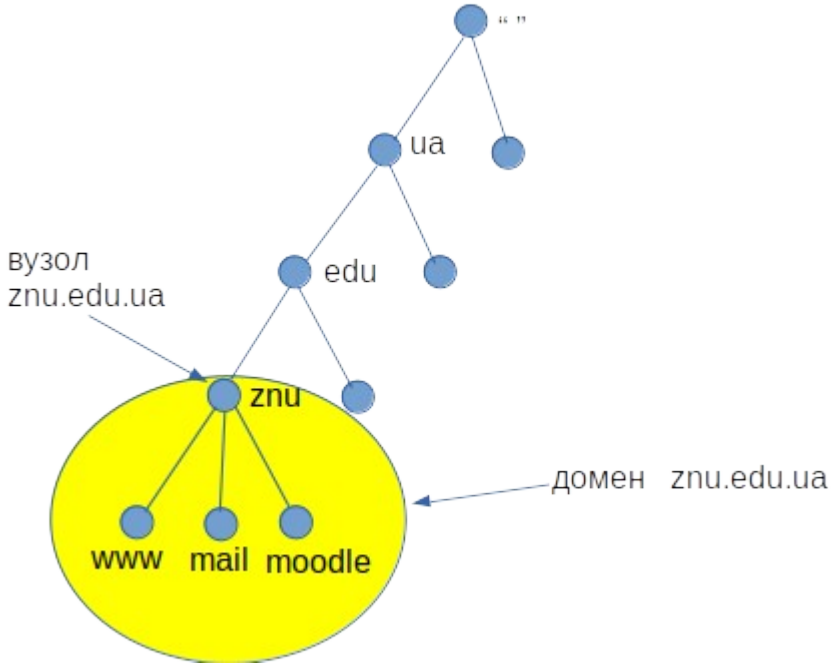


Рис.3. Верхній вузол в ієрархії має ім’я домену.

Кожне доменне ім’я може знаходитись у декількох піддеревах і, відповідно, у декількох доменах. Наприклад, доменне ім’я "moodle.znu.edu.ua" є частиною доменів "znu.edu.ua", "edu.ua", а також домену "ua". Хост-машини також є вузлами DNS-дерева, і також, за суттю доменної ієрархії, є доменами. Домен має усі хост-машини, доменні імена яких знаходяться всередині цього домену, тобто домен є суфіксом у їх іменах. Хост-машини у домені зв’язані логічно (інколи географічно або за організаційною ознакою) і не зобов’язані належати до певної мережі або класу

IP-адрес. В одному домені можуть знаходиться хост-машини із різних мереж, різних стран і навіть різних континентів. Доменні імена всередині домену можуть бути іменами хост-машин або посилатись на структурну інформацію про піддомени. Одне і те саме доменне ім’я може використовуватись як для позначення хост-машини, так і для домену. Наприклад, ім’я "hp.com" є іменем домену компанії Hewlett-Packard і іменем хост-машини, яка здійснює маршрутизацію пошти між HP та Internet. Або, ім’я "znu.edu.ua" є іменем домену компанії та іменем хост-машини, яка отримує запити на веб-сервер університету. Тип інформації, яку необхідно знайти про хост-машину або піддомен, залежить від контексту використання доменного імені.

Простір імен DNS має ієрархічну структуру і містить множину вкладених доменів, кожен з яких представляє адміністративно пов'язаний набір хост-машин Internet. Безпосередньо нижче кореня цієї ієрархії знаходиться набір доменів верхнього рівня, які відповідають або певним типам організацій, або пов’язані з географічним розташуванням ресурсу. Географічне розташування ресурсу є не стільки фізичним, скільки логічним з точки зору формуванням доменного імені. Наприклад, такий домен верхнього рівня як “tv” є географічним, позначає державу Тувалу і належить її уряду. Але цей домен за контрактами з урядом Тувалу використовується в усьому світі телевізійними та медійними компаніями, реальні ресурси яких розташовуються не в Тувалу.

Дані, які асоціюються з доменними іменами, знаходяться у ресурсних записах (resouce records або RR). Записи поділяються на класи. У сучасній DNS практично використовується єдиний клас – IN (Internet). Також записи поділяються на типи. Тип записи визначає тип інформації, яка може зберігатись у просторі доменних імен. Кожний тип записи має визначений формат. В Internet-класі є адресні записи, записи серверів імен, записи вказівників та інші. Серед них є дві обов’язкові, відповідні записи повинні бути у будь-якій зоні, інакше вона буде важитись некоректною. Обов’язковими типами є SOA та NS.

Ресурсний запис є одиницею зберігання та передавання інформації в DNS і має наступні поля:

- ім’я (Name), ім’я домену, до якого відноситься цей запис;
- TTL (Time Ti Live), заданий час зберігання запису неавторизованим сервером;
- тип (Type), параметр, який визначає призначення та формат запису в полі даних (Rdata);
- клас (Class), тип мережі передавання даних;
- довжина поля даних (Rdlen);
- поле даних (Rdata), вміст та формат залежать від типу запису.

Найчастіше використовуються такі типи ресурсних записів:

- SOA (Start of Authority), "початок повноважень" або початок зони відповідальності, вміщує базові параметри доменної зони: ім’я первинного DNS-серверу та контактну адресу електронної пошти;
- A (IPv4 Address Record), адресний запис, який зв’язує доменне ім’я з IPv4 адресою хоста;
- AAAA (IPv6 Address Record), адресний запис, який зв’язує доменне ім’я з IPv6 адресою хоста;
- CNAME (Canonical Name Record), канонічний запис імені, який використовується для перенаправлення на інше доменне ім’я;
- MX (Mail Exchange) поштовий обмінник, який посилається на поштовий сервер, що обслуговує домен;
- NS (Name Server), сервер імен, що посилається на DNS-сервер, який відповідає за домен;
- TXT, текстове описання домену, часто необхідно для виконання специфічних задач, наприклад, підтвердження права власності на домен при прив’язуванні його до поштового сервісу;
- PTR (Point to Reverse), запис вказівки, що зв’язує IP адресу хоста з доменом, часто використовується поштовими сервісами.

З 1995 р. в основі DNS знаходиться система кореневих (root) серверів, яка складається з первинного (primary) сервера a.root-servers.net та його реплік (secondary). З 1997 і по сьогодення таких реплік 12, з іменами b.root-servers.net, c.root-servers.net і т.д. до m.root-servers.net. Їх список, розподіл у світі та інша інформація доступна на офіційному сайті www.root-servers.org. Коли було

створено «скритий» мастер-сервер a.root-servers.net став одним із 13 рівноправних кореневих серверів, які управляються 12 незалежними організаціями-операторами, що несуть за них відповідальність:

a — VeriSign Global Registry Services;
b — University of Southern California, Information Sciences Institute;
c — Cogent Communications;
d — University of Maryland;
e — NASA Ames Research Center;
f — Internet Systems Consortium, Inc.;
g — US DoD Network Information Center;
h — US Army Research Lab;
i — Netnod;
j — VeriSign Global Registry Services;
k — RIPE NCC;
l — ICANN;
m — WIDE Project.

З метою підвищення продуктивності та стійкості системи, у відповідь на вибухове зростання інтернету і кількості звернень до доменів, з 2003 року було впроваджено технологію anycast, що дозволила операторам зробити множину дзеркал кореневих серверів, які розташовуються ближче до користувачів. Зараз число таких дзеркал становить 1467 (https://root-servers.org/).

Кореневі сервери DNS є критичним компонентом системи, тому що забезпечують доступ до кореневої зони DNS. Корнева зона має інформацію про усі домени верхнього (першого) рівня. Ця інформація вказує клієнту на які сервери DNS слід відправити запит для визначення повного доменного імені. Якщо інформація про домен у запиті не була раніше збережена у кеші клієнта, то його запит починається із звернення до кореневого серверу і буде обробленим найближчим його дзеркалом.

Якщо є необхідність змінити адресацію у кореневій зоні, наприклад, змінити склад серверів, які обслуговують домен, то адміністратор домену верхнього рівня надсилає підписаний ЕЦП запит, який ретельно перевіряється та авторизується спеціальним аудитором - IANA. Авторизований запит передається оператору VeriSign, який має право вносити зміни на скритому мастер-сервері DNS, після чого зміни розповсюджуються через захищений протокол на всі кореневі сервера.

Практична частина

В Linux більшість професійних серверів системи доменних імен (DNS) реалізовано за допомогою сервісу Berkeley Internet Name Domain (BIND). Налаштування реального DNS-серверу, який забезпечує обслуговування запитів як з внутрішньої локальної мережі, так і з глобальної мережі потребує наявності у адміністратора серверу прав реєстратора домену, а також певної конфігурації мережі. Вивчення та експерименти з налаштування DNS серверу слід проводити для невеликої локальної мережі, яка має обмеження за допомогою брандмауера. У цьому випадку DNS сервер буде грати роль внутрішнього серверу імен.

1. Приклад конфігурація експериментальної системи:

Ім’я	роль	Повне доменне ім’я	IP адреса
ns1	Первинний DNS сервер	ns1.lab.example.com	10.128.10.11
ns2	Вторинний DNS сервер	ns2.lab.example.com	10.128.10.12
host1	Вузол або клієнтській сервер host1	host1.lab.example.com	10.128.10.101
host2	Вузол або клієнтській сервер host2	host2.lab.example.com	10.128.10.102

2. Встановлення BIND на DNS сервер.

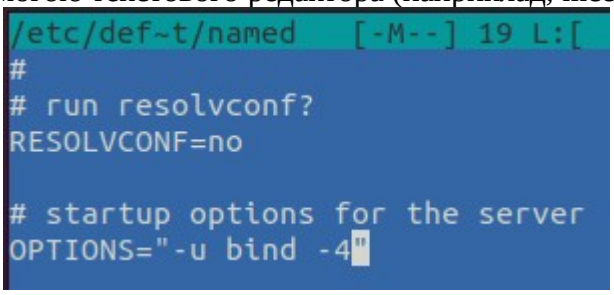
Виконайте оновлення пакетів системи, на яку встановлюється BIND:

```
sudo apt update
```

Виконайте інсталяцію BIND на сервер:

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

Якщо локальна мережа буде використовувати виключно IPv4, то слід встановити для BIND режим IPv4. На сервері необхідно відредагувати файл налаштувань за замовчуванням (/etc/default/named) за допомогою текстового редактора (наприклад, mcedit):



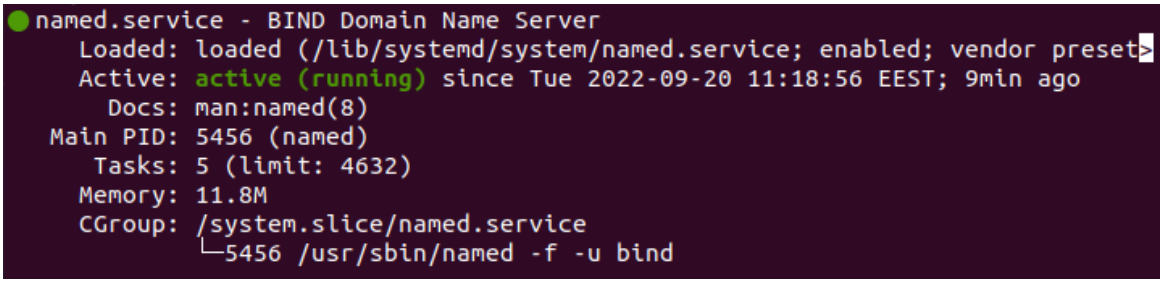
Перезапустить BIND для актуалізації налаштування:

```
sudo systemctl restart bind9
```

та перевірте його статус в системі

```
sudo systemctl status bind9
```

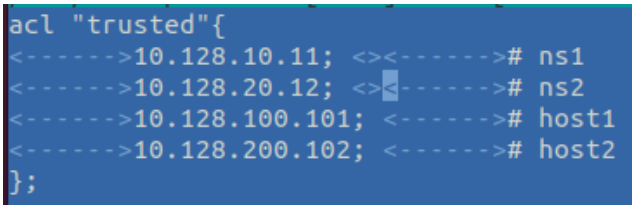
Сервер повинен бути активним:



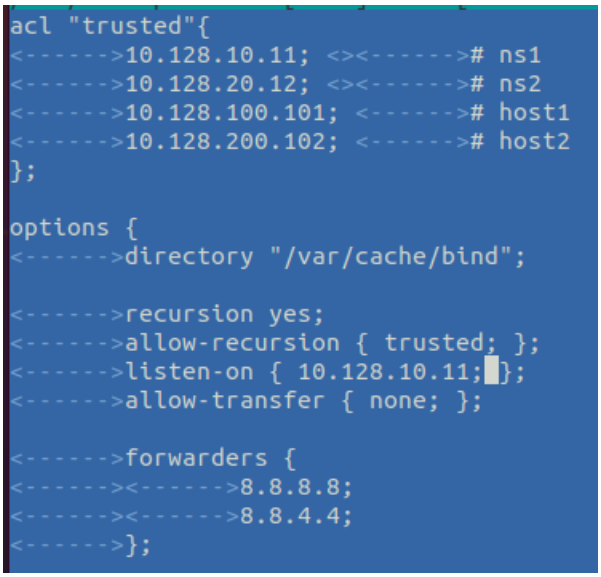
3. Налаштування первинного DNS серверу.

Конфігурація BIND складається із кількох файлів, які додаються в основному файлі конфігурації named.conf. Імена цих файлів починаються з named (скорочення від «name daemon»), тому що це ім’я процесу, який запускає BIND. Налаштування починаються з файлу named.conf.options за допомогою текстового редактору, наприклад, mcedit.

Над існуючим блоком параметрів створіть новий блок ACL (список контролю доступу) під назвою “trusted”. Визначте список клієнтів, яким дозволяються рекурсивні DNS-запити. Як правило, це ті сервери, які знаходяться у одній локальній мережі, що і первинний DNS сервер ns1. Додайте наступні рядки, щоб додати ns1, ns2, host1 і host2 до списку довірених клієнтів. Зробіть заміну прикладів IP-адрес на адреси ваших власних серверів:



Наступним, після додавання списку надійних клієнтів DNS, виконується редагування блоку параметрів. Під директивою каталогу додайте наступні рядки конфігурації (замініть відповідну IP-адресу ns1):



Блок переадресації містить дві IP-адреси: 8.8.8.8 і 8.8.4.4. Цей блок визначає пересилання - спеціальний механізм, який BIND використовує для зменшення трафіку через посилання на зовнішні сервери імен. BIND також може використовувати засоби пересилання, щоб дозволити запити серверів, які не мають прямого доступу до Інтернету. Це може допомогти пришвидшити відповіді на ці запити, зменшивши навантаження на локальну мережу. Дві IP-адреси в цьому блоці представляють загальнодоступні DNS-перетворювачі Google, але тут може бути IP-адреса будь-якого публічного рекурсивного сервера імен. Наприклад, замість цього можна використовувати IP-адресу DNS-сервера Cloudflare (1.1.1.1).

Після редагування збережіть і закрийте файл named.conf.options. Наведена вище конфігурація вказує, що лише власні сервери (довірені) зможуть запитувати DNS-сервер щодо зовнішніх доменів.

Далі вкажемо свої зони DNS, налаштувавши файл named.conf.local, для редагування якого використовуємо текстовий редактор.

Окрім кількох коментарів, файл буде порожнім. Додаємо зони прямого та зворотного ходу. Зони DNS призначають конкретну область для керування та визначення записів DNS. Вважаємо, що піддомен lab.example.com використовуємо як пряму зону. Оскільки IP-адреси прикладів серверів знаходяться в IP-просторі 10.128.0.0/16, то зворотну зону буде створено так, щоб можна було виконати зворотні пошуки в цьому діапазоні. Також додаємо зону пересилання в директиві allow-transfer.

```
zone "lab.example.com" {
<------>type primary;
<------>file "/etc/bind/zones/db.lab.example.com";
<------>allow-transfer { 10.128.20.12; };
};

zone "128.10.in-addr.arpa" {
<------>type primary;
<------>file "/etc/bind/zones/db.10.128";
<------>allow-transfer { 10.128.20.12; };
};
```

Якщо сервери охоплюють кілька підмереж, але обслуговуються тим самим DNS сервером, то вказується додаткова зона та файл зони для кожної окремої підмережі. Після додавання всіх потрібних зон, збережіть і закрийте файл named.conf.local.

Після визначення зон в BIND потрібно створити відповідні файли прямої та зворотної зони.

Файл прямої зони – це місце, де визначаються записи DNS для прямого пошуку DNS. Тобто, коли DNS отримує запит на ім’я, наприклад, host1.lab.example.com, він шукатиме у файлі зони пересилання, щоб визначити відповідну IP-адресу host1.

Для цього необхідно створити каталог, де будуть зберігатися файли зони. Відповідно до конфігурації named.conf.local це місце має бути /etc/bind/zones:

```
sudo mkdir /etc/bind/zones
```

Створимо файл прямої зони на основі зразка файлу зони db.local. Скопіюйте його в потрібне місце за допомогою таких команд:

```
sudo cp /etc/bind/db.local /etc/bind/zones/db.lab.example.com
```

Відкрийте скопійований файл у тестовому редакторі.

По-перше, відредагуйте запис SOA. Замініть перший локальний хост на FQDN ns1, потім замініть root.localhost на admin.lab.example.com. Кожного разу, коли редагуєте файл зони, потрібно збільшувати значення Serial перед перезапуском названого процесу. Тут збільште його до 3. Потім видаліть три записи в кінці файлу (після запису SOA):

```
;
@       IN      NS      localhost.      ; delete this line
@       IN      A       127.0.0.1       ; delete this line
@       IN      AAAA    ::1             ; delete this line
```

Додайте записи серверів імен, а також вузлів.

```
;
; BIND data file for local loopback interface
;
$TTL	604800
@	IN	SOA	ns1.lab.example.com. admin.lab.example.com. (
				      3		; Serial
				 604800		; Refresh
				  86400		; Retry
				2419200		; Expire
				 604800 )	; Negative Cache TTL
;
; name servers - NS records
;
	IN	NS	ns1.lab.example.com.
	IN	NS	ns2.lab.example.com.
;
; name servers - A records
;
ns1.lab.example.com.		IN	A	10.128.10.11
ns2.lab.example.com.		IN	A	10.128.20.12

; 10.128.0.0/16 - A records
host1.lab.example.com.		IN	A	10.128.100.101
host2.lab.example.com.		IN	A	10.128.200.102
```

Зберігаємо та закриваємо цей файл.

Наступним налаштовуємо файл зворотної зони, де визначаються записи PTR DNS для зворотного пошуку DNS. Тобто, коли DNS отримує запит за IP-адресою, наприклад, 10.128.100.101, він зробить пошук у файлі зворотної зони, щоб розпізнати відповідне FQDN, у цьому випадку host1.lab.example.com.

На ns1 для кожної зворотної зони, зазначеної у файлі named.conf.local, необхідно створити файл зворотної зони. Для цього можна скористатись прикладом - файлом зони db.127. BIND використовує цей файл для зберігання інформації для локального інтерфейсу loopback; 127 – це перший октет IP-адреси, яка представляє локальний хост (127.0.0.1). Скопіюйте цей файл у потрібне місце за допомогою наступних команд (замінивши назву цільового файлу таким чином, щоб вона відповідала визначенню зворотної зони, зазначеної у файлі named.conf.local):

```
sudo cp /etc/bind/db.127 /etc/bind/zones/db.10.128
```

Відкрийте файл db.10.128 у текстовому редакторі. Подібно до редагування файлу прямої зони необхідно внести зміни і привести їх у наступний вигляд:

```
/etc/bin~.10.128   [----]  0 L:[  1+23  24/ 24] *(557 / 557b) <EO
;
; BIND reverse data file for local loopback interface
;
$TTL	604800
@	IN	SOA	ns1.lab.example.com. admin.lab.example.co
				      3		; Serial
				 604800		; Refresh
				  86400		; Retry
				2419200		; Expire
				 604800 )	; Negative Cache TTL
;
; name server - NS records
;
	IN	NS	ns1.lab.example.com.
	IN	NS	ns2.lab.example.com.
;
; PTR records
;
11.10	IN	PTR	ns1.lab.example.com.	; 10.128.10.11
12.20	IN	PTR	ns2.lab.example.com.	; 10.128.20.12
101.100	IN	PTR	host1.lab.example.com.	; 10.128.100.101
102.200	IN	PTR	host2.lab.example.com.	; 10.128.200.102
```

Записи PTR додаються для всіх серверів та іменованих вузлів, IP-адреси яких знаходяться в підмережі файлу зони, який редагується. Важливо, що перший стовпець складається з двох останніх октетів IP-адрес серверів у зворотному порядку. Збережіть та закрийте файл.

Перевірити синтаксис файлів named.conf*:

```
sudo named-checkconf
```

Якщо конфігураційні файли не містять синтаксичних помилок, повідомлень про помилки не буде. Якщо є помилки, то їх слід виправити і повторити команду.

Команда named-checkzone може бути використана для перевірки правильності створених файлів зони. Його перший аргумент визначає назву зони, а другий аргумент визначає відповідний файл зони, обидва визначені в named.conf.local. Наприклад, щоб перевірити конфігурацію прямої зони lab.example.com, виконайте команду:

```
sudo named-checkzone lab.example.com /etc/bind/zones/db.lab.example.com
```

Очікувана відповідь:

```
zone lab.example.com/IN: loaded serial 3
OK
```

Для перевірки конфігурації зворотної зони виконайте команду:

```
sudo named-checkzone 128.10.in-addr.arpa /etc/bind/zones/db.10.128
```

Очікувана відповідь:

```
zone 128.10.in-addr.arpa/IN: loaded serial 3
OK
```

Перевантажте BIND:

```
sudo systemctl restart bind9
```

Перш ніж усі сервери в довіреному ACL зможуть запитувати DNS-сервери необхідно налаштувати кожен з них на використання ns1 та ns2 як серверів імен. Якщо припустити, що на клієнтських серверах працює Ubuntu, то потрібно визначити пристрій, який зв’язаний з потрібною локальною мережею. Необхідно виконати таку команду на кожному клієнтському комп’ютері:

```
ip address show to 10.128.0.0/16
```

Інтерфейс може мати, наприклад, ім’я eth1, або щось подібне:

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defau
    inet 10.128.100.101/16 brd 10.128.255.255 scope global eth1
           valid_lft forever preferred_lft forever
```

В Ubuntu 20.04 мережа налаштовується за допомогою Netplan. Щоб налаштувати DNS, необхідно написати файл конфігурації Netplan. Створіть новий файл у /etc/netplan під назвою

00-private-nameservers.yaml

та відкрийте його у текстовому редакторі.

Додайте наступний вміст, що змінює інтерфейс мережі, адреси DNS-серверів ns1 та ns2, а також зону DNS:

```
network:
  version: 2
  ethernets:
      eth1:
          nameservers:
              addresses:
              - 10.128.10.11      # Private IP for ns1
              - 10.128.20.12      # Private IP for ns2
              search: [ lab.example.com ]   # DNS zone
```

Збережіть та закрийте файл.
Наступним необхідно виконати оновлення налаштувань з новим файлом конфігурації за допомогою netplan try:

```
sudo netplan try
```

Якщо будуть виявлені проблеми, що спричинять втрату мережі, Netplan автоматично повернеться до попередніх налаштувань після тайм-ауту. Якщо зворотний відлік у нижній частині оновлюється правильно, нова конфігурація принаймні достатньо функціональна, щоб не порушити мережеве з’єднання і можна натиснути ENTER, щоб прийняти нову конфігурацію.

```
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in  88 seconds
```

Наступним слід перевірити, що системний резолвер DNS визначив нову конфігурацію DNS:

```
sudo systemd-resolve --status
```

Для перевірки роботи клієнта DNS слід використати утиліту nslookup. Для тестування роботи DNS сервера, щодо прямої зони слід виконати:

```
nslookup host1
```

Для тестування роботи DNS сервера, щодо зворотної зони команда задається з IP адресою:

```
nslookup 10.128.100.101
```

Завдання до лабораторної роботи

1. Виконайте вхід у систему під особистим логіном. Виконайте перемикання на акаунт root.
2. Встановіть в системі пакет BIND та виконайте налаштування, які вказано у практичній частині. Налаштування адаптуйте до мережевої конфігурації власної локальної мережі.
3. Перевірте правильність налаштувань DNS серверу. Виконайте тестування роботи DNS серверу.
4. Підготуйте звіт, в якому наведіть скрин-шоти етапів налаштування DNS серверу та результатів його перевірки та тестування.

Контрольні питання

1. Для чого призначено DNS систему?
2. Що таке пряма та зворотна зона?
3. Які типи ресурсних записів застосовуються у файлах опису зон?
4. Для чого призначено утиліту nslookup?