SH Dr. Soufiane Hourri

École Supérieure de Technologie de Safi · Cadi Ayyad University
Computer Science Department · 2026-2027

NETWORK ADMINISTRATION · DUT S3 · PRACTICAL SESSION 03

DNS: ns1 gives names to the lab

Since Practical 02, srv1's DHCP server tells every client "your DNS server is 192.168.50.2", but nothing answers at that address. Today you build that server: a new VM, ns1, running BIND. You write the zone of `lab.internal` and its reverse zone, test every answer with `dig`, then use the names from client1 and see caching at work.

BY THE END OF THE SESSION YOU WILL HAVE

- **ns1** (192.168.50.2): authoritative server for `lab.internal` and for the reverse zone of 192.168.50.0/24, and resolver for Internet names
- client1 reaching srv1, ns1 and client1 by name, with no line in `/etc/hosts`
- Real `dig` answers you can read: status, flags, TTL, SOA

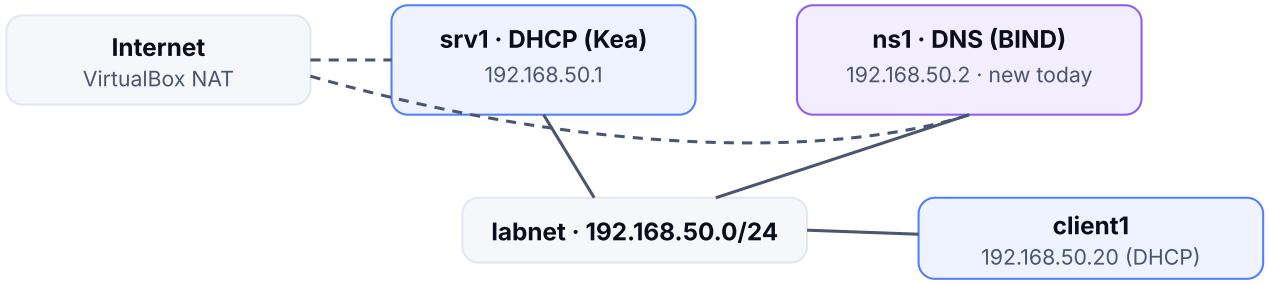
BEFORE YOU COME

- srv1 and client1 from Practical 02 (Kea running, client1 with its reserved address 192.168.50.20)
- srv1's snapshot **clean install** from Practical 01: ns1 is cloned from it
- Your notes from Masterclass 03 (zones, records, resolution, TTL)
- RAM for three VMs: srv1, ns1 and client1 (client2 stays off today)

Duration	4 hours, in class, including a 15-minute break. Work alone, except in Part 8 (pairs). Help your neighbour when you finish a part.
Deliverable	The answers to the **questions** (violet boxes), the output of each **checkpoint** (green boxes) and your troubleshooting table, in one PDF report. Show each checkpoint to the teacher during the session. The submission deadline is announced in class.
Conventions	`$` means "type this command" (do not type the `$`). Grey text after `#` is a comment. Grey numbers in front of a file are line numbers: do not type them. Work in the VirtualBox window of each VM.

Time	Part	Content
0:00	0	Starting point: client1 knows where to ask, but nobody answers
0:10	1	Create ns1 and give it 192.168.50.2
0:35	2	Install BIND and look at what Fedora provides
0:55	3	Configure named.conf, write the forward zone, start BIND
1:40	4	Test with dig from srv1: our zone, then the Internet
2:05		**Break (15 min)**
2:20	5	The reverse zone
2:45	6	client1 uses DNS
3:05	7	Changing a zone: reload, serial, negative caching
3:25	8	Troubleshooting challenge in pairs
3:50		Snapshots and what to hand in

THE LAB AT THE END OF TODAY



ns1 has two cards like srv1: NAT (to install BIND and to resolve Internet names) and labnet. Check the real interface names with `ip -br address`.

Part 0 Starting point

10 min

Start srv1, then client1. Check where Practical 02 ended:

BASH · SRV1

```
$ ip -br address
$ systemctl is-active kea-dhcp4
```

BASH · CLIENT1

```
$ ip -br address show enp0s3
$ nmcli -f IP4.DNS,IP4.DOMAIN device show enp0s3
$ resolvectl query srv1.lab.internal      # fails: nobody answers yet
$ ping -c 2 192.168.50.1                  # by address it works
```

QUESTIONS 1 AND 2

1. Which DNS server and which domain did client1 receive, and which lines of srv1's Kea configuration sent them (option names)?
2. client1 can ping 192.168.50.1 but cannot resolve srv1.lab.internal. In which layer is the problem, and why is the network itself fine?

Part 1 Create ns1

25 min

1. srv1 can keep running. In VirtualBox, select srv1, open the **Snapshots** view, right-click the snapshot **clean install ▸ Clone**. Name: `ns1`. **Full clone**. MAC address policy: **Generate new MAC addresses for all network adapters**.
2. Check ns1's network settings: Adapter 1 **NAT**, Adapter 2 **Internal Network** `labnet`. Give it 1536 MB of RAM if your laptop has 8 GB.
3. Start ns1, log in as `admin` and make it a different machine:

BASH · NS1

```
$ sudo hostnamectl set-hostname ns1.lab.internal
$ sudo rm /etc/machine-id
$ sudo systemd-machine-id-setup
$ sudo reboot
```

After the reboot, give ns1 its fixed address. A DNS server, like a DHCP server, needs an address that never changes: the clients are told this address.

BASH · NS1

```
$ ip -br address                    # the labnet card is usually enp0s8
$ sudo nmcli connection add type ethernet \
      ifname enp0s8 con-name labnet \
      ipv4.method manual ipv4.addresses 192.168.50.2/24 \
      ipv6.method disabled
$ sudo nmcli connection up labnet
$ ping -c 2 192.168.50.1               # srv1
$ ping -c 2 192.168.50.20              # client1 (or its address from DHCP)
$ ping -c 2 fedoraproject.org          # Internet through NAT
```

CHECKPOINT 1

☐ The prompt shows `ns1`. ☐ ns1 has `192.168.50.2/24` on labnet and pings srv1, client1 and fedoraproject.org.

QUESTION 3

Why do we clone ns1 from the snapshot **clean install** and not from srv1 as it is now? Name two things on today's srv1 that ns1 must not inherit.

Part 2 Install BIND and look around

20 min

`bind` is the DNS server; `bind-utils` gives the test tools, including `dig`. The service is called `named` (name daemon).

BASH · NS1

```
$ sudo dnf install bind bind-utils
$ sudo cp /etc/named.conf /etc/named.conf.orig
$ grep -n -E "listen-on|allow-query|recursion|directory" /etc/named.conf
$ sudo ls -l /var/named
$ sudo head -n 25 /var/named/named.ca
```

QUESTIONS 4 TO 6

4. In Fedora's default `/etc/named.conf`, on which addresses does BIND listen, and who may send it queries? Copy the two lines. With these defaults, could client1 use ns1?
5. `/var/named/named.ca` lists names such as `A.ROOT-SERVERS.NET.` with addresses. What are these servers, and why does a resolver need this file before it can answer any question about the Internet?
6. Who owns the files in `/var/named` (user and group)? Why does this matter for the zone files you will create?

Part 3 Configure BIND and write the forward zone

45 min

3.1 Who may ask: two lines in the options

Open the file with `sudo nano /etc/named.conf`. Do not rewrite it: change only the `listen-on` and `allow-query` lines of the `options` block, as below. Keep every other line of Fedora's file.

NAMED.CONF · THE TWO LINES TO CHANGE IN OPTIONS

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.50.2; };
        ...
        allow-query     { localhost; 192.168.50.0/24; };
        ...
        recursion yes;
```

Every address is followed by a semicolon, the list is in braces, and the line ends with a semicolon. A missing semicolon is the most common error in this file.

3.2 Declare our zones

At the very end of `/etc/named.conf`, add the two zones. The file names must be exactly the names of the files you create in `/var/named`.

NAMED.CONF · ADD AT THE VERY END OF THE FILE

```
1  zone "lab.internal" IN {
2      type primary;
3      file "lab.internal.zone";
4  };
5
6  zone "50.168.192.in-addr.arpa" IN {
7      type primary;
8      file "192.168.50.rev";
9  };
```

`type primary` means ns1 holds the original copy of the zone. Older tutorials, and this course in previous years, write `type master`; BIND 9.18 accepts both, but `primary` is the current term.

3.3 The forward zone

Create the zone file directly in `/var/named` with `sudo nano` (if you create it elsewhere and move it, run `sudo restorecon -v /var/named/*` so that SELinux lets `named` read it). For the serial on line 3, use **today's date** followed by `01`.

/VAR/NAMED/LAB.INTERNAL.ZONE

```
1  $TTL 3600
2  @  IN SOA ns1.lab.internal. admin.lab.internal. (
3             2026110401 ; serial
4             3600       ; refresh
5             900        ; retry
6             604800     ; expire
7             300 )      ; negative TTL
8  @        IN  NS     ns1.lab.internal.
9  ns1      IN  A      192.168.50.2
10 srv1     IN  A      192.168.50.1
11 web      IN  A      192.168.50.3
12 gw       IN  A      192.168.50.254
13 client1  IN  A      192.168.50.20
14 www      IN  CNAME  web
```

Lines	Meaning
1	Default TTL of every record: 3600 s
2 to 7	SOA: primary server, administrator (admin@lab.internal), serial, then refresh, retry and expire (for secondary servers) and the negative TTL (how long a "does not exist" answer may be cached)
8	`@` is the zone itself. Its name server is ns1, written as a full name with a final dot
9 to 13	A records. Names without a final dot are relative: `srv1` means `srv1.lab.internal.`
14	www is an alias of web. web (.3) is the future web server: the name is ready before the machine

client1 can have an A record because its address is fixed by the DHCP reservation of Practical 02. A client with an address from the pool cannot: its address may change. Later in the module, the DHCP server will register such clients automatically (dynamic DNS).

3.4 Check, protect, open, start

BASH · NS1

```
$ sudo nano /var/named/lab.internal.zone      # type the file above
$ sudo named-checkconf                         # nothing printed = correct
$ sudo named-checkzone lab.internal /var/named/lab.internal.zone
$ sudo chown root:named /var/named/lab.internal.zone
$ sudo chmod 640 /var/named/lab.internal.zone
$ sudo firewall-cmd --permanent --add-service=dns
$ sudo firewall-cmd --reload
$ sudo systemctl enable --now named
$ systemctl status named
```

THE FIREWALL MATTERS HERE

In Practical 02 you saw that Kea receives DHCP broadcasts through raw sockets, beside the firewall. BIND is different: it uses ordinary UDP and TCP sockets on port 53, so without `--add-service=dns` the firewall blocks every query coming from another machine. The `dns` service opens port 53 for both UDP and TCP.

CHECKPOINT 2

- [ ] `named-checkconf` prints nothing.
- [ ] `named-checkzone` prints `loaded serial ...` and `OK`.
- [ ] `named` is `active (running)` and `enabled`.

QUESTIONS 7 AND 8

7. Remove the final dot of line 8 (`ns1.lab.internal`), run `named-checkzone` again and copy its message. Explain the strange name it shows. Then put the dot back.
8. Why must the zone file belong to the group `named` and be readable by it? Which user does the `named` service run as (`ps -o user= -C named`)?

Part 4 Test with dig from srv1

25 min

client1 has no Internet access, so it cannot install `dig`. We test from srv1, which can, and ask ns1 directly with `@192.168.50.2`.

BASH · SRV1

```
$ sudo dnf install bind-utils          # once, on srv1
$ dig @192.168.50.2 srv1.lab.internal
```

Your answer should look like this real answer from a test server running the same zone (only the id and the time differ):

REAL OUTPUT OF THE SAME ZONE ON A TEST SERVER (SHORTENED)

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52742
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;srv1.lab.internal.        IN  A

;; ANSWER SECTION:
srv1.lab.internal.  3600   IN  A   192.168.50.1

;; Query time: 0 msec
;; SERVER: 192.168.50.2#53(192.168.50.2) (UDP)
```

Run the other tests and keep the outputs for your report:

BASH · SRV1

```
$ dig @192.168.50.2 +short www.lab.internal
$ dig @192.168.50.2 lab.internal SOA +noall +answer
$ dig @192.168.50.2 lab.internal NS +short
$ dig @192.168.50.2 pc9.lab.internal           # a name that does not exist
$ dig @192.168.50.2 fedoraproject.org          # outside our zone: recursion
$ dig @192.168.50.2 fedoraproject.org          # again, a few seconds later
```

CHECKPOINT 3

- [ ] From srv1, `dig @192.168.50.2 srv1.lab.internal` shows `status: NOERROR`, the flag `aa` and `192.168.50.1`.

QUESTIONS 9 TO 12

9. In the answer for srv1.lab.internal, what do the flags `qr`, `aa`, `rd` and `ra` mean? Why is `aa` present?
10. For www.lab.internal, `+short` prints two lines. Explain each one.
11. For pc9.lab.internal: give the status, and copy the record shown in the AUTHORITY section. Which of its numbers tells caches how long they may remember that pc9 does not exist?
12. Compare the two answers for fedoraproject.org: is `aa` present? How did the TTL and the query time change between the first and the second query? Explain with the words recursive resolver and cache.

BREAK · 15 MINUTES

Leave the three VMs running.

Part 5 The reverse zone

25 min

A reverse lookup finds the name of an address. The network 192.168.50 becomes the zone `50.168.192.in-addr.arpa`, already declared in `named.conf`. Create its file on ns1 (same serial as the forward zone):

/VAR/NAMED/192.168.50.REV

```
$TTL 3600
@  IN SOA ns1.lab.internal. admin.lab.internal. (
           2026110401 3600 900 604800 300 )
@    IN  NS   ns1.lab.internal.
1    IN  PTR  srv1.lab.internal.
2    IN  PTR  ns1.lab.internal.
3    IN  PTR  web.lab.internal.
20   IN  PTR  client1.lab.internal.
254  IN  PTR  gw.lab.internal.
```

On the left, `1` means `1.50.168.192.in-addr.arpa.`, that is 192.168.50.1. On the right, the names are full names in another zone, so each one ends with a dot.

BASH · NS1

```
$ sudo nano /var/named/192.168.50.rev
$ sudo named-checkzone 50.168.192.in-addr.arpa /var/named/192.168.50.rev
$ sudo chown root:named /var/named/192.168.50.rev
$ sudo chmod 640 /var/named/192.168.50.rev
$ sudo named-checkconf
$ sudo systemctl reload named
```

Test from srv1:

BASH · SRV1

```
$ dig @192.168.50.2 -x 192.168.50.1 +noall +answer
$ dig @192.168.50.2 -x 192.168.50.20 +short
$ dig @192.168.50.2 -x 192.168.50.77          # no PTR record for .77
```

CHECKPOINT 4

- [ ] `dig -x 192.168.50.1` returns `srv1.lab.internal.` and `dig -x 192.168.50.20` returns `client1.lab.internal.`

QUESTIONS 13 AND 14

13. Which name does `dig -x` build for 192.168.50.20? Why are the four numbers written in reverse order?
14. What status do you get for 192.168.50.77, and why? Which two records must you add, in which two files, when a new server with a fixed address joins labnet?

Part 6 client1 uses DNS

20 min

On Fedora, the stub resolver is the service `systemd-resolved`, managed with `resolvectl`. It sends client1's questions to the DNS server received from DHCP. If client1 still has no DNS answers, run `sudo nmcli connection up labnet` once to refresh its settings.

BASH · CLIENT1

```
$ resolvectl status enp0s3
$ resolvectl query srv1.lab.internal
$ resolvectl query www.lab.internal
$ resolvectl query 192.168.50.2                # reverse lookup
$ getent hosts srv1.lab.internal
$ ping -c 2 srv1                               # short name: lab.internal is added
$ ping -c 2 www.lab.internal
$ resolvectl statistics
```

CHECKPOINT 5

- [ ] On client1, `ping -c 2 srv1` works by name, and `resolvectl query 192.168.50.2` returns `ns1.lab.internal`.

QUESTIONS 15 TO 17

15. Which DNS server and which search domain does `resolvectl status enp0s3` show? Where do they come from?
16. `ping -c 2 srv1` works with the short name. Which part of client1's configuration completes `srv1` into `srv1.lab.internal`?
17. `ping -c 2 www.lab.internal` resolves the name but gets no reply. Explain both facts.

Part 7 Changing a zone: reload, serial, negative caching

20 min

7.1 Forgetting the serial

BASH · NS1

```
$ sudo nano /var/named/lab.internal.zone
  # add this line at the end, and do NOT change the serial:
  #   intranet IN  CNAME  web
$ sudo named-checkzone lab.internal /var/named/lab.internal.zone
$ sudo systemctl reload named
$ sudo journalctl -u named -n 5
```

On a test server, the same mistake produced these log lines:

REAL LOG LINES FROM THE TEST SERVER

```
zone lab.internal/IN: zone serial (2026110401) unchanged.
                      zone may fail to transfer to secondaries.
zone lab.internal/IN: loaded serial 2026110401
```

Test `dig @192.168.50.2 +short intranet.lab.internal` from srv1, then do it properly: increase the serial (last two digits 01 to 02), check, reload.

7.2 A cached "does not exist"

THREE MACHINES, IN THIS ORDER

```
# 1. client1: ask for a name that does not exist yet
$ resolvectl query pc9.lab.internal
# 2. ns1: add   pc9  IN  A  192.168.50.99   increase the serial, check, reload
# 3. srv1: ask ns1 directly
$ dig @192.168.50.2 +short pc9.lab.internal
# 4. client1: ask again at once, then after flushing its cache
$ resolvectl query pc9.lab.internal
$ sudo resolvectl flush-caches
$ resolvectl query pc9.lab.internal
```

QUESTIONS 18 AND 19

18. After the reload without a new serial, did ns1 answer for intranet.lab.internal? What does the warning mean, and which servers would miss the change? Why is increasing the serial still a rule you must follow?
19. In 7.2, at step 3 srv1 found pc9 at once, but at step 4 client1 may still have said it does not exist. Explain with the negative TTL of our SOA (line 7). What does `resolvectl flush-caches` change?

Part 8 Troubleshooting challenge in pairs

25 min

Take a snapshot of ns1 named **dns ok**. Student A creates **one** fault on their own ns1 while student B looks away; then B finds it **with commands only**, starting from the symptom "on srv1, `dig @192.168.50.2 srv1.lab.internal` no longer gives 192.168.50.1". Swap roles with another fault.

#	Fault to create on ns1 (student A)	How to undo it
a	Remove the final dot on line 8 of the forward zone, then `sudo systemctl reload named`	Put the dot back, check, reload
b	In `/etc/named.conf`, set `allow-query { localhost; };`, then `sudo systemctl restart named`	Restore labnet in allow-query, check, restart
c	`sudo systemctl stop named`	`sudo systemctl start named`
d	`sudo firewall-cmd --remove-service=dns` (without --permanent)	`sudo firewall-cmd --add-service=dns`

Useful commands for student B: on srv1, the full `dig` output (read the **status** line, or the error message); on ns1, `systemctl status named`, `sudo journalctl -u named -n 30`, `sudo named-checkconf`, `sudo named-checkzone ...`, `sudo firewall-cmd --list-services`, and `dig @127.0.0.1 srv1.lab.internal` (a query from ns1 to itself does not cross the firewall). If you are stuck, restore the snapshot **dns ok**.

> QUESTION 20
> For each fault you diagnosed: the symptom seen on srv1 (status or error message), the commands you ran on ns1 and what they showed, the cause, and the fix. Which faults give an answer with a status, and which give no answer at all? Why?

Before you leave

10 min

- Check that both zones load (`named-checkzone`), that `named` is enabled, and that client1 resolves srv1, ns1 and client1 by name.
- Shut srv1, ns1 and client1 down with `sudo poweroff` and take a snapshot of each named **dns final**.
- Next: the web server at 192.168.50.3. Its names `web.lab.internal` and `www.lab.internal` already exist in your zone.
- Your report contains the answers to questions 1 to 20, the output of checkpoints 1 to 5 and your challenge table, with your name and group on the first page.

If something goes wrong

Symptom	Likely cause	What to do
`named-checkconf` reports a line number	Missing `;` or `}` in named.conf	Look at that line and the one before it
`named-checkzone`: name repeated twice, e.g. `ns1.lab.internal.lab.internal`	Missing final dot on a full name	Add the dot at the end of the name
dig: `status: SERVFAIL`	The zone was not loaded: error in the file, wrong file name in named.conf, or named cannot read the file	`sudo journalctl -u named -n 30`; check owner `root:named` and `restorecon`
dig: `status: REFUSED`	allow-query does not include the client	Add `192.168.50.0/24` to allow-query, restart
dig: `connection refused`, then `no servers could be reached`	named stopped, or not listening on 192.168.50.2	`systemctl status named`; check `listen-on`
dig waits, then `no servers could be reached`	Firewall closed, or wrong address	`sudo firewall-cmd --list-services` on ns1
`status: NXDOMAIN` for a name you added	Not reloaded, typo in the name, or a cached negative answer	`sudo systemctl reload named`; `resolvectl flush-caches` on the client
dig on srv1 works, ping by name on client1 fails	client1 uses another DNS server, or has old DHCP settings	`resolvectl status enp0s3`; `sudo nmcli connection up labnet`

References: RFC 1034 and RFC 1035 (1987), Domain Names; RFC 2308 (1998), Negative Caching of DNS Queries; ISC, BIND 9 Administrator Reference Manual, 9.18; Fedora Project, package bind (9.18.50 in Fedora 44); Masterclass 03 slides of this module. The dig answer and the log lines in this handout come from real tests of these zone files with BIND 9.18.