Setup a DNS configuration on Rocky

DNS basic setup on Rocky.

You can run BIND (named) on both Rocky 9 boxes, one as master and one as slave, with one forward and one reverse zone for 192.168.4.0/24.

Below assumes:

- Domain: **roc.lan**
- Master DNS: 192.168.4.198
- Slave DNS: 192.168.4.199

  Adjust hostnames/IPs and domain to your actual values.

1. Install basics on both

On **both** servers:

(install bind and bind-utils. Make sure you use ipv4. Enable and start named and configure firewalld.)

```
sudo dnf install bind bind-utils -y
sudo sed -i 's/^OPTIONS=.*/OPTIONS="-4"/' /etc/sysconfig/named

sudo systemctl enable --now named
sudo firewall-cmd --add-service=dns --permanent
sudo firewall-cmd --reload
```

2. Master: /etc/named.conf

On **master (192.168.4.198)**:

```
sudo cp /etc/named.conf /etc/named.conf.orig
sudo vi /etc/named.conf
```

Example **minimal** content:

```
options {
    directory       "/var/named";
    pid-file        "/run/named/named.pid";
    allow-query     { any; };                    // or limit to your LAN
    recursion       yes;
    allow-recursion { 192.168.4.0/24; };

    // listen on all IPv4
    listen-on port 53 { any; };
    listen-on-v6 port 53 { none; };

    // upstream resolvers if you want recursion
    forwarders {
        1.1.1.1;
           8.8.8.8;
    };
    dnssec-validation auto;
};

// allow internal clients and the slave
acl "trusted" {
    127.0.0.1;
    192.168.4.0/24;
};

// forward zone
zone "roc.lan" IN {
    type master;
    file "roc.lan.zone";
    allow-transfer { 192.168.4.199; };           // slave IP
    also-notify { 192.168.4.199; };              // send NOTIFY to slave
```

```
};

// reverse zone for 192.168.4.0/24
zone "4.168.192.in-addr.arpa" IN {
    type master;
    file "4.168.192.rev";
    allow-transfer { 192.168.4.199; };
    also-notify { 192.168.4.199; };
};
```

Validate and restart:

```
sudo named-checkconf /etc/named.conf          # no output = OK
```

3. Master: create zone files

On **master**:

```
cd /var/named
sudo cp named.empty roc.lan.zone
sudo cp named.empty 4.168.192.rev
sudo chown root:named roc.lan.zone 4.168.192.rev
```

Edit **/var/named/roc.lan.zone**:

```
$TTL 86400
@   IN SOA  ns1.roc.lan. admin.roc.lan. (
        2026010301; Serial: yyyymmddnn
        3600        ; Refresh
        900         ; Retry
        604800      ; Expire
        86400       ; Minimum
)
    IN  NS  ns1.roc.lan.
    IN  NS  ns2.roc.lan.

; A records for name servers
```

```
ns1 IN  A   192.168.4.198
ns2 IN  A   192.168.4.199

; Some hosts
master  IN  A   192.168.4.198
slave   IN  A   192.168.4.199
host1   IN  A   192.168.4.197
host2   IN  A   192.168.4.196
```

Edit **/var/named/4.168.192.rev**:

```
$TTL 86400
@   IN SOA  ns1.roc.lan. admin.roc.lan. (
        2026010301
        3600
        900
        604800
        86400
)
    IN  NS  ns1.roc.lan.
    IN  NS  ns2.roc.lan.

; PTR records
198  IN PTR master.roc.lan.
199  IN PTR slave.roc.lan.
197  IN PTR host1.roc.lan.
196  IN PTR host2.roc.lan.
```

Check them:

```
sudo named-checkzone roc.lan /var/named/roc.lan.zone   # expect "loaded serial"
sudo named-checkzone 4.168.192.in-addr.arpa /var/named/4.168.192.rev
sudo systemctl restart
named
```

4. Slave: /etc/named.conf

On **slave (192.168.4.199)**:

```
sudo cp /etc/named.conf /etc/named.conf.orig
sudo vi /etc/named.conf
```

Example:

```
options {
    directory       "/var/named";
    pid-file        "/run/named/named.pid";
    allow-query     { any; };
    recursion       yes;
    allow-recursion { 192.168.4.0/24; };

    listen-on port 53 { any; };
    listen-on-v6 port 53 { none; };

    forwarders {
        1.1.1.1;
        8.8.8.8;
    };
    dnssec-validation yes;
};

acl "trusted" {
    127.0.0.1;
    192.168.4.0/24;
};

// slave copies zones from master
zone "roc.lan" IN {
    type slave;
    masters { 192.168.4.198; };             // master IP
    file "slaves/roc.lan.zone";        // stored under /var/named/slaves
};

zone "4.168.192.in-addr.arpa" IN {
    type slave;
```

```
    masters { 192.168.4.198; };
    file "slaves/4.168.192.rev";
};
```

Create slaves dir and set permissions:

```
sudo mkdir -p /var/named/slaves
sudo chown named:named /var/named/slaves
sudo named-checkconf /etc/named.conf
sudo systemctl restart named
```

On reload, the slave will AXFR the zones from the master into `/var/named/slaves`.

5. Point clients to DNS and test

On a client (or on the Rocky boxes themselves) configure `/etc/resolv.conf` or NetworkManager to use:

```
nameserver 192.168.4.198
nameserver 192.168.4.199
search roc.lan
```

Tests from anywhere in 192.168.4.0/24:

```
dig @192.168.4.198 master.roc.lan
dig @192.168.4.199 master.roc.lan

dig @192.168.4.198 -x 192.168.4.197
dig @192.168.4.199 -x 192.168.4.197

# Check serial on master vs slave
dig @192.168.4.198 roc.lan SOA +short
dig @192.168.4.199 roc.lan SOA +short
```

When you edit zone files on the master, increment the SOA serial, run `sudo systemctl reload named` on the master, and verify that the slave picks up the new serial via `dig SOA`.[1][2]

If you share your actual hostnames/IPs and whether you want recursion locked to the /24 or wider, a tailored set of ready-to-paste configs can be provided.

Additional exercises:

Add an A record and a PTR record to the master server, update the serial number
Force a retransfer if necessary from the slave:

```
rndc retransfer roc.lan
```

Run dig, to check the serialnumber on the slave.

```
dig @192.168.4.199 roc.lan SOA +short
```

Recursion?
When you run the following command:

```
dig google.com
```

Does your dns server take care of the recursion?

Config files explained.

named.conf

This configuration defines how BIND runs (options), who may use it (ACL), and which zones it serves (forward and reverse). It is close to a minimal authoritative-plus-recursive setup for one LAN and one slave, but a few things could still be trimmed or tightened.

options block
directory "/var/named";
Sets BIND's working directory, where it looks for zone files like roc.lan.zone and 4.168.192.rev unless you use absolute paths.

pid-file "/run/named/named.pid";
Tells BIND where to write its process ID so service scripts and tools like rndc can find and control the daemon.

allow-query { any; };
Allows any client on the Internet to query this server for data in zones it is authoritative for; for a LAN-only server you'd typically restrict this to your internal networks or the trusted ACL.

recursion yes; and allow-recursion { 192.168.4.0/24; };
Enables recursive lookups (acting as a resolver) and limits who can use that resolver function to your 192.168.4.0/24 subnet, which is important for security so the server cannot be abused as an open resolver.

listen-on port 53 { any; }; / listen-on-v6 port 53 { none; };
Makes BIND listen on all IPv4 interfaces on port 53 but not on IPv6 at all; you could set IPv6 to any; later if you add v6 connectivity.

forwarders { 1.1.1.1; 8.8.8.8; };
When doing recursion, BIND forwards queries it cannot answer locally to these upstream resolvers instead of going directly to the root servers; this effectively makes your server a caching forwarder for the LAN.

dnssec-validation auto;
Enables DNSSEC validation using the built-in root trust anchors, so answers obtained during recursion are cryptographically checked where possible.

acl "trusted" block
acl "trusted" { 127.0.0.1; 192.168.4.0/24; };
Defines a named address list for localhost and your LAN that you can reuse in allow-query, allow-recursion, or allow-transfer statements, which keeps the config cleaner and makes it easier to adjust access control later.

(At the moment you define the ACL but do not use it in options; you could change allow-query and allow-recursion to use trusted; for consistency.)

forward (primary) zone "roc.lan"
zone "roc.lan" IN { ... };
Declares an authoritative zone for the forward lookups of your local domain roc.lan.

type master;
This server is the primary source of truth for that zone and reads it from local zone files; slaves replicate from it via zone transfers.

file "roc.lan.zone";
Path (relative to directory) to the zone file containing SOA, NS, A, CNAME, etc. records for roc.lan.

allow-transfer { 192.168.4.199; };
Allows full or incremental zone transfers (AXFR/IXFR) only to the slave DNS at 192.168.4.199, preventing arbitrary hosts from pulling your entire zone data.

also-notify { 192.168.4.199; };
When this master reloads or the zone serial changes, it proactively sends NOTIFY messages to 192.168.4.199 so the slave can pick up changes quickly instead of waiting for the refresh timer.

reverse zone for 192.168.4.0/24
zone "4.168.192.in-addr.arpa" IN { ... };
Authoritative zone that maps IP addresses in 192.168.4.0/24 back to hostnames via PTR records, which some services and tools rely on.

type master; file "4.168.192.rev";
Same idea as the forward zone: this server is the primary source and the data lives in 4.168.192.rev under /var/named.

allow-transfer and also-notify
Mirror the forward zone settings so the slave gets the reverse zone via transfer and prompt NOTIFY-driven updates.