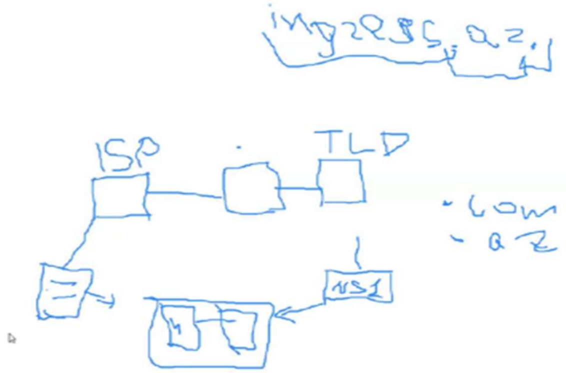
DNS

DNS nə üçündür? Biz browserdə saytlara girmək üçün İP yazmalıyıq. Amma IP çox uzun olduğu üçün onu yadda saxlamaq olmur. Çünki rəqəmlər yadda saxlamaq çətindir. Buna görədə biz DNS istifadə edirik. DNS IP-ni ada, adıda IP-yə resolv edir. DNS necə işləyir?

1. İlk olaraq browser-in cache-nə baxır. (TTL vaxtı qədər cache-də qalır.)
2. Daha sonra local host faylına baxır.
3. OS səviyyəsində cache baxır.
4. DNS server varsa ona baxır.
5. Daha sonra ISP-yə baxır.
6. 13 root DNS server-ə baxır. (Burada Top-Level-Domain məlumatlarını öyrənir.)
7. Name Server-ə yönləndirilir.



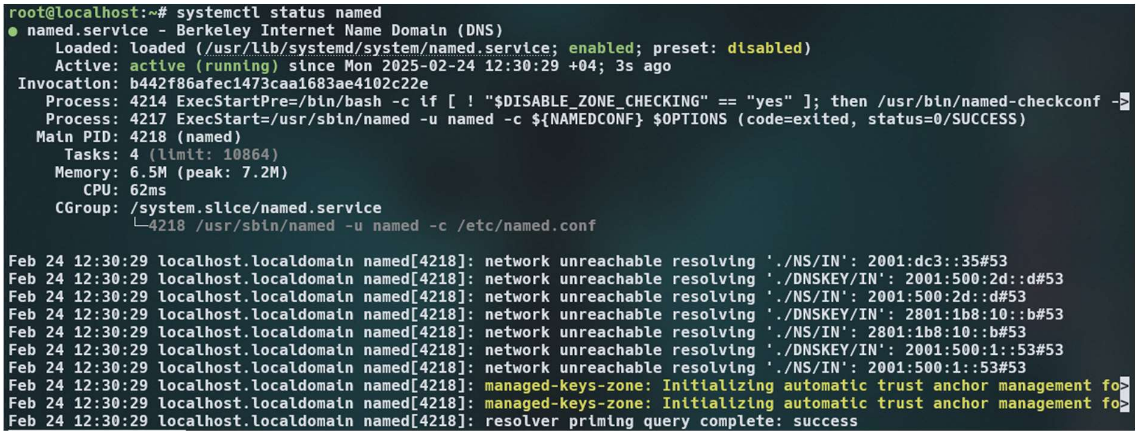
Biz burada Name Server-in Public IP-sini Domain aldığımız sayta qeyd edirik. Təhlükəsizlik üçün master Name Server qurulur. Və bu global-a çıxarılmır. Bunun yerinə Slave Name Server qurulur, və global-a çıxarılır.

Install DNS (BIND)

yum -y install bind bind-utils

firewall-cmd --add-service=dns --permanent;firewall-cmd --reload

systemctl status named (distroya görə dəyişkənlik göstərir)



```
root@localhost:~# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
     Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
     Active: active (running) since Mon 2025-02-24 12:30:29 +04; 3s ago
 Invocation: b442f86afec1473caa1683ae4102c22e
    Process: 4214 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/bin/named-checkconf ->
    Process: 4217 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 4218 (named)
      Tasks: 4 (limit: 10864)
     Memory: 6.5M (peak: 7.2M)
        CPU: 62ms
     CGroup: /system.slice/named.service
             └─4218 /usr/sbin/named -u named -c /etc/named.conf

Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './NS/IN': 2001:dc3::35#53
Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './DNSKEY/IN': 2001:500:2d::d#53
Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './NS/IN': 2001:500:2d::d#53
Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './DNSKEY/IN': 2801:1b8:10::b#53
Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './NS/IN': 2801:1b8:10::b#53
Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './DNSKEY/IN': 2001:500:1::53#53
Feb 24 12:30:29 localhost.localdomain named[4218]: network unreachable resolving './NS/IN': 2001:500:1::53#53
Feb 24 12:30:29 localhost.localdomain named[4218]: managed-keys-zone: Initializing automatic trust anchor management fo>
Feb 24 12:30:29 localhost.localdomain named[4218]: managed-keys-zone: Initializing automatic trust anchor management fo>
Feb 24 12:30:29 localhost.localdomain named[4218]: resolver priming query complete: success
```

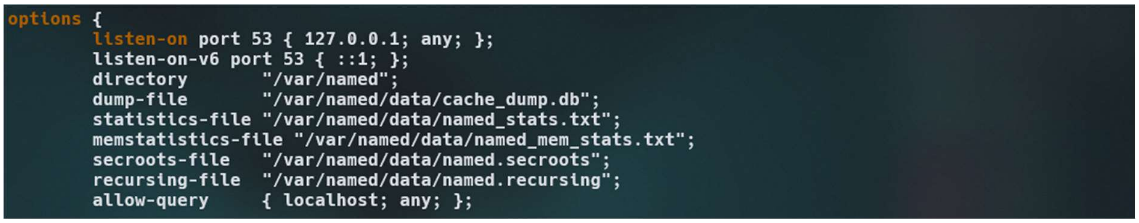
Configure DNS (BIND)

vim /etc/named.conf

listen-on port 53 { 127.0.0.1; 1any; };

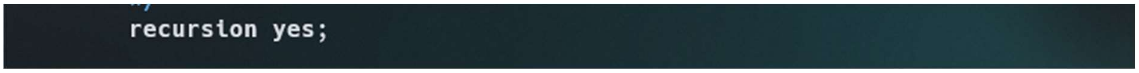
allow-query { localhost; any; };

(Allow-query kimlərdən gələcək olan requestləri resolve edəcəyini göstərir. Any yazıldığı üçün istənilən hostdan gələn request resolve ediləcək.*)*.



```
options {
        listen-on port 53 { 127.0.0.1; any; };
        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";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { localhost; any; };
```

Recursion – Əgər no olarsa, "məndə əgər A recordu varsa cavab qaytar, əgər yoxdusa ilişib qalır". Əgər yes olarsa DNS flow recursion sayılır. Forwarder – Əgər məndə yoxdursa təyin olunmuş server-ə gedir. Məsələn 8.8.8.8



```
        recursion yes;
```

Create Forward Zones

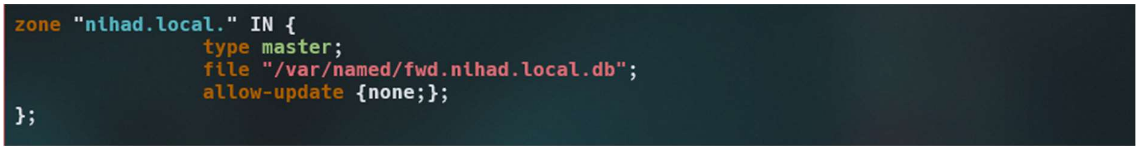
vim /etc/named.conf

zone "nihad.local." IN {

type master;

file "/var/named/fwd.nihad.local.db";

allow-update {none;};
};



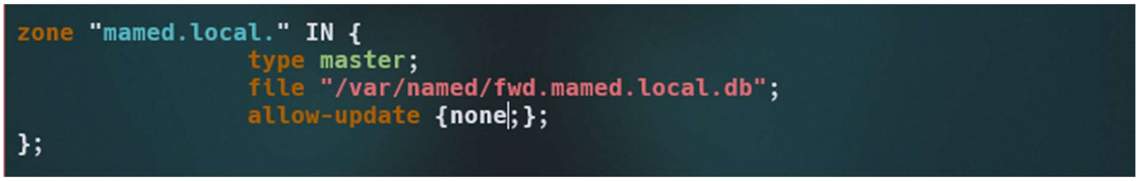
```
zone "nihad.local." IN {
                type master;
                file "/var/named/fwd.nihad.local.db";
                allow-update {none;};
};
```

zone "mamed.local." IN {

type master;

file "/var/named/fwd.mamed.local.db";

allow-update {none;};
};



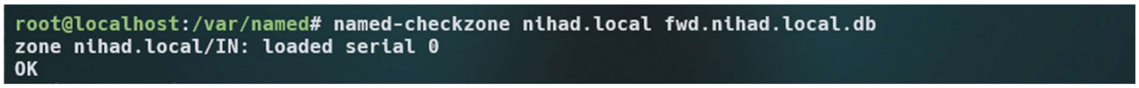
```
zone "mamed.local." IN {
                type master;
                file "/var/named/fwd.mamed.local.db";
                allow-update {none;};
};
```

Servisin check olunması üçün istifadə edilir

named-checkconf

cd /var/named

nihad.local domain-ə aid fwd.nihad.local.db zone-sini kontrol et.

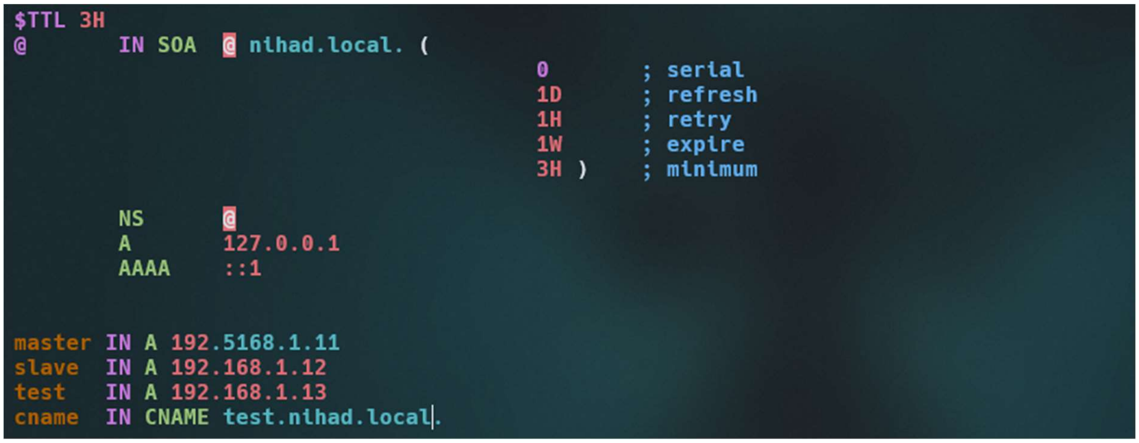


```
root@localhost:/var/named# named-checkzone nihad.local fwd.nihad.local.db
zone nihad.local/IN: loaded serial 0
OK
```

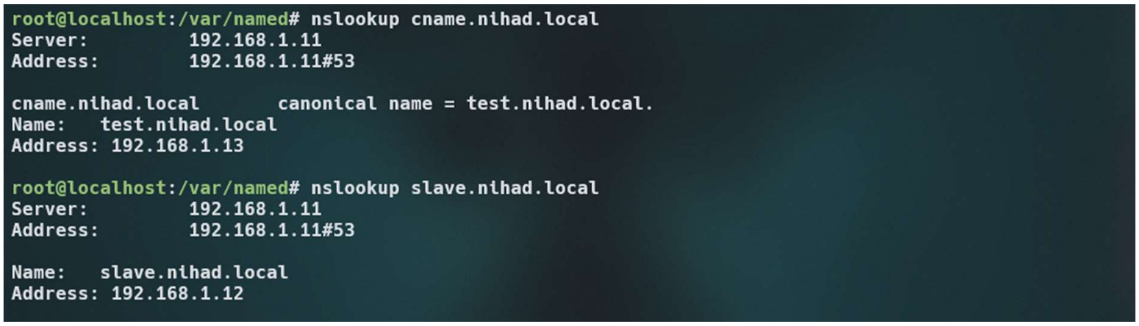
Create Forward Zone File

```
vim  /var/named/fwd.nihad.local.db
```

```
$TTL 3H
@       IN SOA  @ nihad.local. (
                        0       ; serial
                        1D      ; refresh
                        1H      ; retry
                        1W      ; expire
                        3H )    ; minimum

                        NS      @
                        A       127.0.0.1
                        AAAA    ::1

master IN A 192.168.1.11
slave  IN A 192.168.1.12
test   IN A 192.168.1.13
cname  IN CNAME test.nihad.local.
```



```
$TTL 3H
@       IN SOA  @ nihad.local. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum


        NS      @
        A       127.0.0.1
        AAAA    ::1


master IN A 192.5168.1.11
slave  IN A 192.168.1.12
test   IN A 192.168.1.13
cname  IN CNAME test.nihad.local.
```

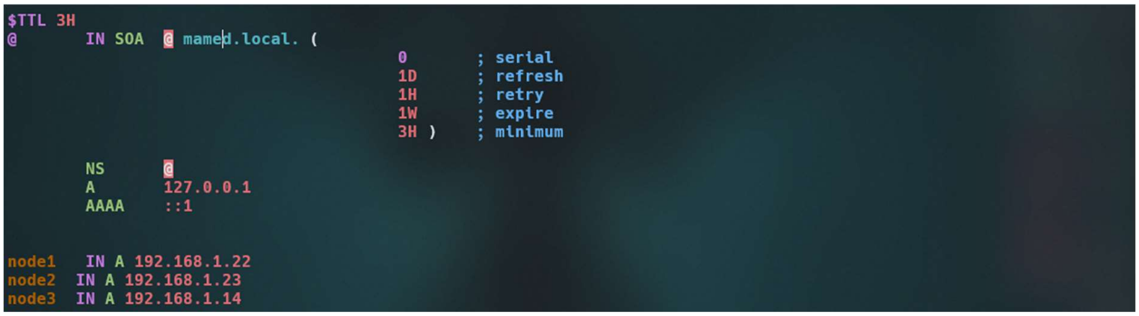


```
root@localhost:/var/named# nslookup cname.nihad.local
Server:         192.168.1.11
Address:        192.168.1.11#53

cname.nihad.local       canonical name = test.nihad.local.
Name:   test.nihad.local
Address: 192.168.1.13

root@localhost:/var/named# nslookup slave.nihad.local
Server:         192.168.1.11
Address:        192.168.1.11#53

Name:   slave.nihad.local
Address: 192.168.1.12
```

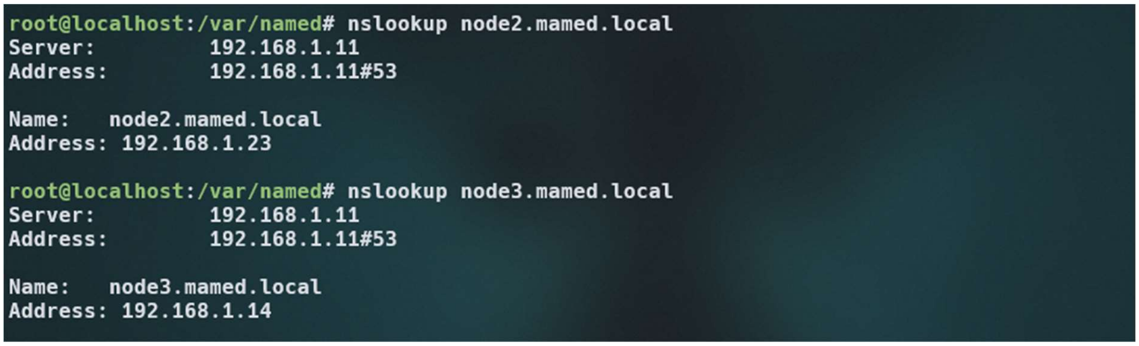
Create Second Forward Zone File

vim /var/named/fwd.mamed.local.db

```
$TTL 3H
@       IN SOA  @ mamed.local. (
                    0       ; serial
                    1D      ; refresh
                    1H      ; retry
                    1W      ; expire
                    3H )    ; minimum

        NS      @
        A       127.0.0.1
        AAAA    ::1

node1   IN A 192.168.1.22
node2  IN A 192.168.1.23
node3  IN A 192.168.1.14
```



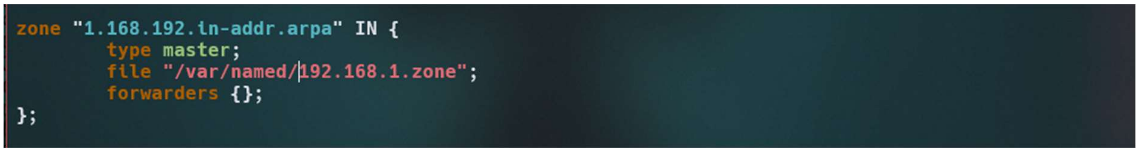
```
$TTL 3H
@       IN SOA  @ mamed.local. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum

        NS      @
        A       127.0.0.1
        AAAA    ::1

node1   IN A 192.168.1.22
node2  IN A 192.168.1.23
node3  IN A 192.168.1.14
```



```
root@localhost:/var/named# nslookup node2.mamed.local
Server:         192.168.1.11
Address:        192.168.1.11#53

Name:   node2.mamed.local
Address: 192.168.1.23

root@localhost:/var/named# nslookup node3.mamed.local
Server:         192.168.1.11
Address:        192.168.1.11#53

Name:   node3.mamed.local
Address: 192.168.1.14
```

Create Reverse Zone

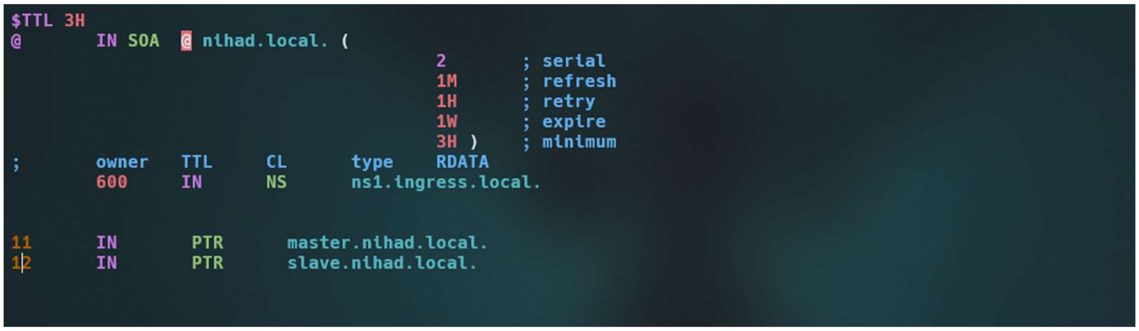
```
zone "1.168.192.in-addr.arpa" IN {
type master;
file "/var/named/192.168.1.zone";
forwarders {};
};
```



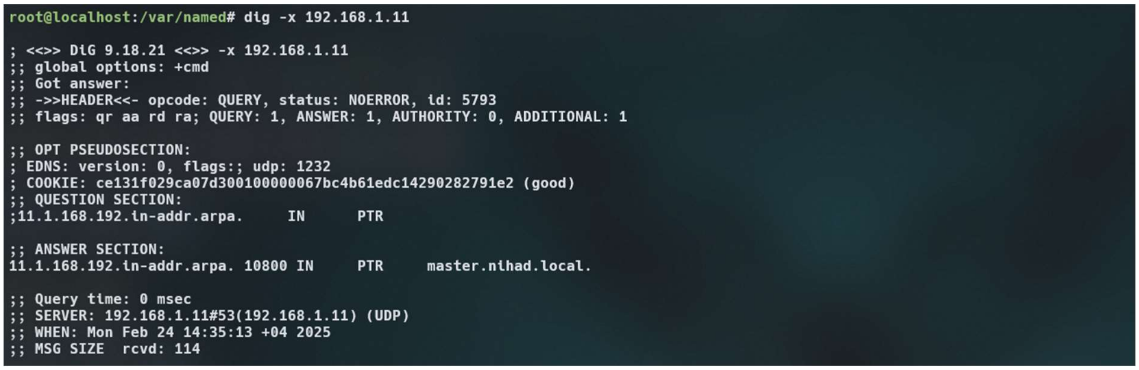
```
zone "1.168.192.in-addr.arpa" IN {
        type master;
        file "/var/named/192.168.1.zone";
        forwarders {};
};
```

Create Reverse Zone Files

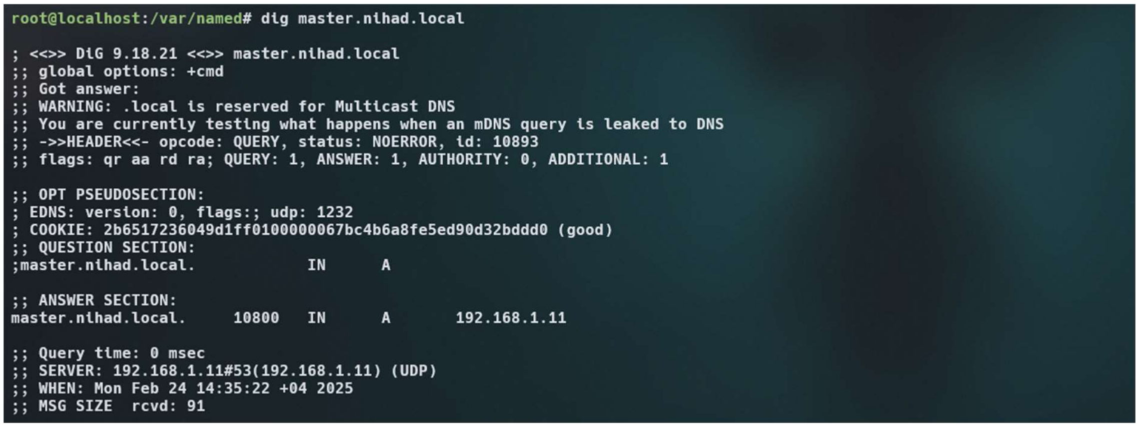
```
$TTL 3H
@      IN SOA  @ nihad.local. (
                2      ; serial
                1M     ; refresh
                1H     ; retry
                1W     ; expire
                3H )   ; minimum
;     owner  TTL    CL    type   RDATA
  600    IN     NS     ns1.ingress.local.

11    IN      PTR     master.nihad.local.
  12    IN      PTR     slave.nihad.local.
```



```
$TTL 3H
@       IN SOA  @ nihad.local. (
                                        2       ; serial
                                        1M      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
;       owner   TTL     CL      type    RDATA
        600     IN      NS      ns1.ingress.local.


11      IN       PTR     master.nihad.local.
12      IN       PTR     slave.nihad.local.
```



```
root@localhost:/var/named# dig -x 192.168.1.11

; <<>> DiG 9.18.21 <<>> -x 192.168.1.11
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5793
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: ce131f029ca07d300100000067bc4b61edc14290282791e2 (good)
;; QUESTION SECTION:
;11.1.168.192.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
11.1.168.192.in-addr.arpa. 10800 IN     PTR     master.nihad.local.

;; Query time: 0 msec
;; SERVER: 192.168.1.11#53(192.168.1.11) (UDP)
;; WHEN: Mon Feb 24 14:35:13 +04 2025
;; MSG SIZE  rcvd: 114
```



```
root@localhost:/var/named# dig master.nihad.local

; <<>> DiG 9.18.21 <<>> master.nihad.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10893
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 2b6517236049d1ff0100000067bc4b6a8fe5ed90d32bddd0 (good)
;; QUESTION SECTION:
;master.nihad.local.            IN      A

;; ANSWER SECTION:
master.nihad.local.     10800   IN      A       192.168.1.11

;; Query time: 0 msec
;; SERVER: 192.168.1.11#53(192.168.1.11) (UDP)
;; WHEN: Mon Feb 24 14:35:22 +04 2025
;; MSG SIZE  rcvd: 91
```

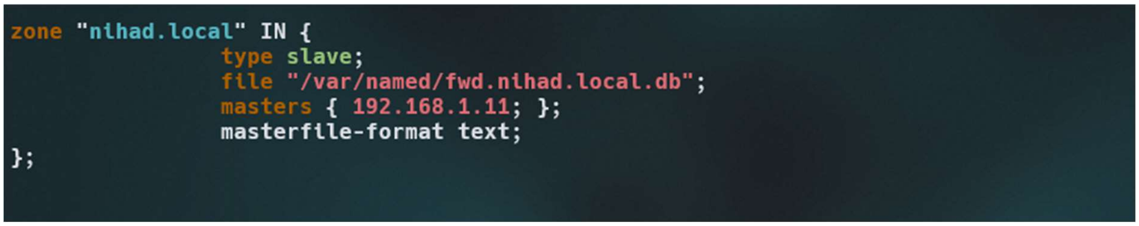
Slave configuration

allow-transfer { localhost; 192.168.149.129; }; ***add to master BIND***

yum -y install bind bind-utils

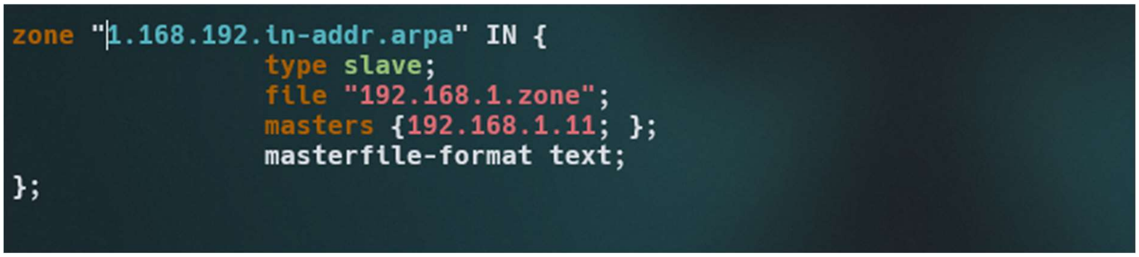
firewall-cmd --add-service=dns --permanent;firewall-cmd --reload

```
zone "nihad.local" IN {
type slave;
file "/var/named/fwd.nihad.local.db";
masters { 192.168.1.11; };
masterfile-format text;
};
```

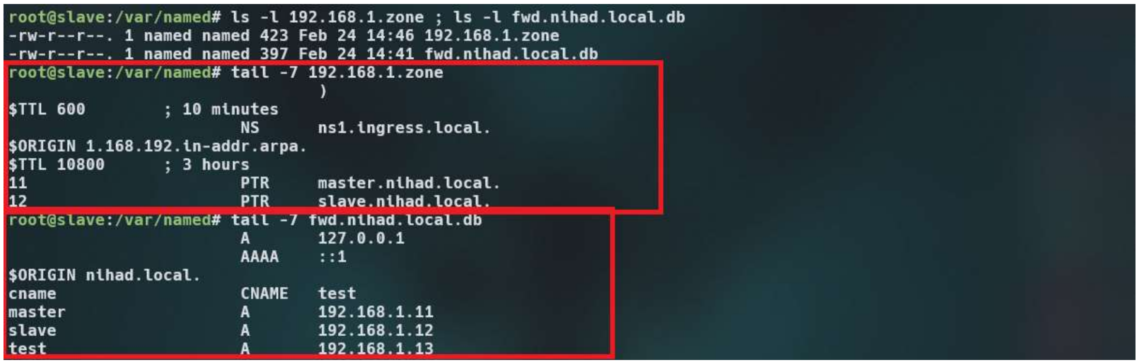


```
zone "nihad.local" IN {
                type slave;
                file "/var/named/fwd.nihad.local.db";
                masters { 192.168.1.11; };
                masterfile-format text;
};
```

```
zone "1.168.192.in-addr.arpa" IN {
type slave;
file "192.168.1.zone";
masters {192.168.1.11; };
masterfile-format text;
};
```



```
zone "1.168.192.in-addr.arpa" IN {
                type slave;
                file "192.168.1.zone";
                masters {192.168.1.11; };
                masterfile-format text;
};
```



```
root@slave:/var/named# ls -l 192.168.1.zone ; ls -l fwd.nihad.local.db
-rw-r--r--. 1 named named 423 Feb 24 14:46 192.168.1.zone
-rw-r--r--. 1 named named 397 Feb 24 14:41 fwd.nihad.local.db
root@slave:/var/named# tail -7 192.168.1.zone
                                )
$TTL 600        ; 10 minutes
                        NS      ns1.ingress.local.
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 10800      ; 3 hours
11                      PTR     master.nihad.local.
12                      PTR     slave.nihad.local.
root@slave:/var/named# tail -7 fwd.nihad.local.db
                        A       127.0.0.1
                        AAAA    ::1
$ORIGIN nihad.local.
cname                   CNAME   test
master                  A       192.168.1.11
slave                   A       192.168.1.12
test                    A       192.168.1.13
```

- **Primary Name Server** – The nameserver that contains the original zone file and not an AXFR transferred copy.
- **Hostmaster Email** – Address of the party responsible for the zone. A period "." is used in place of an "@" symbol. For email addresses that contain a period, this will be escaped with a slash "/".
- **Serial Number** – Version number of the zone. As you make changes to your zone file, the serial number will increase.
- **Time To Refresh** – How long in seconds a nameserver should wait prior to checking for a Serial Number increase within the primary zone file. An increased Serial Number means a transfer is needed to sync your records. Only applies to zones using secondary DNS.
- **Time To Retry** – How long in seconds a nameserver should wait prior to retrying to update a zone after a failed attempt. Only applies to zones using secondary DNS.
- **Time To Expire** – How long in seconds a nameserver should wait prior to considering data from a secondary zone invalid and stop answering queries for that zone. Only applies to zones using secondary DNS.
- **Minimum TTL** – How long in seconds that a nameserver or resolver should cache a negative response.