

# Relazione esercitazione 3 Laboratorio di reti Server DNS con BIND e backend con OpenLDAP

Franco Masotti      Danny Lessio

April 12, 2015

## Contents

<b>I</b>	<b>Configurazione ed installazione del server DNS</b>	<b>3</b>
1	Ambiente di lavoro	3
2	Definizione delle sottoreti e degli indirizzi IP	3
3	Definizione delle zone	4
4	Definizione degli host	4
5	Test del server DNS - Queries	4
6	File di configurazione	8
7	Script	8
<b>II</b>	<b>LDAP Backend</b>	<b>9</b>
8	Ambiente di lavoro	9
9	Installazione	9
10	Configurazione	9
11	Albero LDAP	10
12	Problemi riscontrati	10
<b>III</b>	<b>Listati</b>	<b>12</b>
13	domain.conf	12
14	setup_dns.sh	15

<b>15</b>	<b>chg_nw_settings.sh</b>	<b>28</b>
<b>16</b>	<b>chg_resolv_settings.sh</b>	<b>30</b>
<b>17</b>	<b>Zona diretta zone file</b>	<b>32</b>
<b>18</b>	<b>Zona inversa IPv4 zone file</b>	<b>34</b>
<b>19</b>	<b>Zona inversa IPv6 zone file</b>	<b>35</b>
<b>20</b>	<b>named.conf per i file di zona</b>	<b>36</b>
<b>21</b>	<b>Zona diretta LDAP</b>	<b>38</b>
<b>22</b>	<b>Zona inversa IPv4 LDAP</b>	<b>41</b>
<b>23</b>	<b>Zona inversa IPv6 LDAP</b>	<b>43</b>
<b>24</b>	<b>named.conf per LDAP</b>	<b>46</b>
<b>25</b>	<b>named.service per systemd</b>	<b>48</b>

## Part I

# Configurazione ed installazione del server DNS

## 1 Ambiente di lavoro

Per la realizzazione di questo progetto ci siamo basati su repository di **Arch Linux**, nei quali abbiamo trovato i due pacchetti fondamentali per implementare ed interrogare correttamente il server DNS:

- Il pacchetto **bind**, ci fornisce un server DNS configurabile (grazie all'eseguibile **named**).
- **dnsutils** ci fornisce un set di comandi per poter interrogare il server, come il comando **dig** ed il comando **host**.

Per l'installazione dei pacchetti é quindi sufficiente fare:

```
sudo pacman -Sy dnsutils bind
```

Per maggiori informazioni riguardo il funzionamento di entrambi ci siamo basati sia sulla guida ufficiale <sup>1</sup>, sia su altre guide come (soprattutto) quella di **Gentoo Linux** <sup>2</sup>. Per quanto riguarda configurazioni del client ci siamo basati sul wiki **Arch Linux** <sup>3</sup>.

## 2 Definizione delle sottoreti e degli indirizzi IP

Abbiamo definito la struttura della nostra rete in questo modo:

- Indirizzamento IPv4
  - La nostra sottorete é **192.168.2.0/24**. Questo significa che abbiamo  $2^{(32-24)} = 256$  indirizzi per gli host
  - Considerando che non é possibile utilizzare gli indirizzi che terminano con **0** e **255** (perché **192.168.2.0** rappresenta la sottorete mentre **192.168.0.255** rappresenta l'indirizzo di broadcast) rimangono a disposizione 254 indirizzi realmente utilizzabili per gli host.
  - Come indirizzo del server DNS, per comodità abbiamo scelto **192.168.2.1** in quanto primo indirizzo disponibile.
- Indirizzamento IPv6
  - Quasi tutti i concetti espressi per l'IPv4 possono essere usati per l'IPv6. Tuttavia IPv6 non ha il concetto di indirizzo di broadcast.
  - La nostra sottorete IPv6 é **2002::/64**, quindi abbiamo a disposizione circa  $2^{(128-64)} = 2^{64}$  indirizzi, escludendo **2002::0** che identifica la sottorete.
  - L'indirizzo del server DNS é, sullo stesso concetto dell'IPv4, **2002::1**.

---

<sup>0</sup>networks-lab Copyright (C) 2016 frnmst (Franco Masotti), dannylessio (Danny Lessio). This document comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under certain conditions; see LICENSE file for details.

<sup>1</sup><http://www.isc.org/software/bind/>

<sup>2</sup><https://wiki.gentoo.org/wiki/BIND/Tutorial>

<sup>3</sup><https://wiki.archlinux.org/index.php/BIND>

### 3 Definizione delle zone

Per effettuare correttamente tutte le ricerche DNS abbiamo definito tre zone: una di indirizzamento diretto e due di indirizzamento inverso. Le zone dicono sia come tradurre da un host name ad un indirizzo IP (e viceversa), sia quali informazioni autoritative dare (ad esempio qual é il nome del server dns, server mail, informazioni testuali, ecc...) oltre che informare sui vari tempi in cui scadono le richieste. Le nostre tre zone sono:

- `gruppo2.labreti.it`
  - Questa é la zona diretta, cioè quella che serve a tradurre da hostname ad indirizzo IPv4 ed IPv6. Ogni host infatti ha due entry. Ogni entry ha una serie di attributi. Nel nostro caso i più importanti sono il campo **name** che identifica un'host (o altri oggetti) ed il campo **type** che ci dice se si tratta di un host (tipo **IN** cioè **IN**ternet), oppure di un name server (tipo **NS**). Il campo **class** invece ci definisce se si tratta di un indirizzo IPv4 (classe **A**) oppure IPv6 (classe **AAAA**).
  - Il server DNS ha quindi due entry: una di tipo **NS** e l'altra di tipo **IN**.
- `2.168.192.in-addr.arpa`
  - Questa é la zona inversa IPv4. Ha lo scopo di tradurre un indirizzo IPv4 nel corrispondente host name. Da notare l'uso della classe **PTR** cioè **PoinTeR** in cui la parte host degli indirizzi *punta* ad un nome di macchina.
- `0.2.0.0.2.ip6.arpa`
  - Questa é la zona inversa IPv6, cioè traduce gli indirizzi IPv6 nei corrispondenti host names. É concettualmente uguale alla precedente.

### 4 Definizione degli host

Abbiamo definito quattro host di esempio ed il name server stesso (**ns**) per testare il corretto funzionamento del server DNS. Abbiamo associato i seguenti indirizzi IP alle relative macchine:

Hosts		
Host name	Indirizzo IPv4	Indirizzo IPv6
ns	192.168.2.1	2002::1
vaio	192.168.2.10	2002::10
lenovo	192.168.2.20	2002::20
asus	192.168.2.30	2002::30
acer	192.168.2.40	2002::40

### 5 Test del server DNS - Queries

A questo punto possiamo avviare lo script `setup_dns.sh` in modo che crei i file di zona, come indicato nei listati.

Di seguito mostriamo l'esecuzione del comando `dig` per verificare il corretto funzionamento. I primi esempi mostrano una risoluzione diretta (IPv4 ed IPv6) grazie alla zona `gruppo2.labreti.it`,

In entrambi i casi eseguiamo prima un semplice comando `dig`, poi useremo l'opzione `+short` che permette di stampare a video solamente il corpo della sezione `ANSWER SECTION`. Successivamente lo switch `+short` sarà abilitato per evitare ridondanze.

- `dig ns.gruppo2.labreti.it` , in questo modo otteniamo la risoluzione IPv4 del nome associato al server DNS (il comando `dig`, se non specificato diversamente, effettua solamente la risoluzione IPv4 di default):

- `dig ns.gruppo2.labreti.it AAAA`, Attraverso lo switch AAAA otteniamo la risoluzione IPv6 del nome associato al server DNS:

5

```
ns.gruppo2.labreti.it. 172800 IN AAAA 2002::1
```

```
;; AUTHORITY SECTION:  
gruppo2.labreti.it. 172800 IN NS ns.gruppo2.labreti.it.
```

```
;; ADDITIONAL SECTION:  
ns.gruppo2.labreti.it. 172800 IN A 192.168.2.1
```

```
;; Query time: 0 msec  
;; SERVER: 192.168.2.1#53(192.168.2.1)  
;; WHEN: Wed Apr 08 14:58:07 CEST 2015  
;; MSG SIZE rcvd: 108
```

- `dig asus.gruppo2.labreti.it +short` , in questo modo otteniamo la risoluzione IPv4 dell'host `asus`, da notare l'utilizzo dello switch `+short` descritto all'inizio del paragrafo:

```
192.168.2.30
```

- `dig asus.gruppo2.labreti.it AAAA +short` , in questo modo otteniamo la risoluzione IPv6 dell'host `asus`:

```
2002::30
```

- `dig lenovo.gruppo2.labreti.it +short` ,

```
192.168.2.20
```

- `dig lenovo.gruppo2.labreti.it AAAA +short` ,

```
2002::20
```

- Esempi con risoluzione inversa in IPv4

- `dig -x 192.168.2.20` , in questo modo otteniamo la risoluzione inversa del nome host associato a questo indirizzo IPv4. Il nome restituito é `lenovo`:

```
;<<>> DiG 9.9.7 <<>> -x 192.168.2.20  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42862  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3
```

```
;; OPT PSEUDOSECTION:  
;; EDNS: version: 0, flags:; udp: 4096
```

```
;; QUESTION SECTION:  
20.2.168.192.in-addr.arpa. IN PTR
```

```
;; ANSWER SECTION:  
20.2.168.192.in-addr.arpa. 172800 IN PTR lenovo.gruppo2.labreti.it.
```

```
;; AUTHORITY SECTION:  
2.168.192.in-addr.arpa. 172800 IN NS ns.gruppo2.labreti.it.
```

```
;; Query time: 0 msec
;; SERVER: 192.168.2.1#53(192.168.2.1)
;; WHEN: Wed Apr 08 14:47:04 CEST 2015
;; MSG SIZE rcvd: 154
```

- Esempi di risoluzione inversa in IPv6

```
; <<>> DiG 9.9.7 <<>> -x 2002::1
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 923
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3
```

```
;; Query time: 0 msec
;; SERVER: 192.168.2.1#53(192.168.2.1)
;; WHEN: Wed Apr 08 15:25:31 CEST 2015
;; MSG SIZE rcvd: 194
```

```
– dig -x 2002::10 +short , in questo modo otteniamo la risoluzione inversa del nome
  associato a questo indirizzo IPv6, infatti ci viene restituito il nome dell’host associato
  cioè vaio:

  vaio.gruppo2.labreti.it.

– dig -x 2002::30 +short ,

  asus.gruppo2.labreti.it.
```

## 6 File di configurazione

Abbiamo creato un file di configurazione chiamato **domain.conf** in cui andiamo a definire tutte le variabili necessarie per il funzionamento del server DNS. Qui abbiamo definito il nostro dominio (**gruppo2.labreti.it**), le informazioni IPv4 ed IPv6 della nostra rete e tutti gli host. In questo modo é possibile definire nuovi computer semplicemente cambiando alcuni array.

## 7 Script

Lo script **setup\_dns.sh** ci consente di caricare automaticamente i dati contenuti in **domain.conf** e di creare automaticamente le tre zone (sia sotto forma di zone-file sia come file **ldif** da usare con LDAP). Viene anche creato il file di configurazione di **named**. Il funzionamento dettagliato con le varie opzioni é rimandato alla parte dei listati.

Gli altri script vengono chiamati direttamente da **setup\_dns.sh** a seconda delle operazioni indicate dall’utente.



## Part II

# LDAP Backend

## 8 Ambiente di lavoro

Grazie ad LDAP é possibile salvare le informazioni riguardanti le zone e gli host in modo che **named** possa collegarsi direttamente. Utilizzando LDAP é piú semplice fare manutenzione e gestire gli host soprattutto quando sono in quantità elevata.

Per effettuare il collegamento del nostro server DNS al database LDAP siamo ricorsi al pacchetto<sup>4</sup> **bind-sdb**. Questo permette di inserire all'interno di **named.conf** una entry, per ogni zona, contenente l'URI LDAP corrispondente.

## 9 Installazione

Utilizzando **Arch Linux** il pacchetto **bind** non é precompilato per poter utilizzare il backend e non esiste il pacchetto **bind-sdb** (come su **CentOS**, **fedora**, ecc...). Siamo dunque stati costretti a scaricare il sorgente dal sito ufficiale<sup>5</sup> la versione **9.9.7**<sup>6</sup> ed a ricompilare l'intero pacchetto collegandogli manualmente l'estensione **bind-sdb**.

É necessario inoltre scaricare lo schema **dnszone**<sup>7</sup> da inserire nella cartella degli schemi di LDAP, senza dimenticare della direttiva **include** in **/etc/openldap/slapd.conf**. Con questo schema é possibile definire le zone e gli host, quindi risulta fondamentale.

## 10 Configurazione

Una volta verificata la corretta compilazione, abbiamo lanciato lo script **setup\_dns.sh** che genera e carica nel database LDAP i file **ldif** necessari per le interrogazioni DNS. Questi file sfruttano lo schema del punto precedente il quale definisce object class ed attributi per il nostro scopo (ad esempio vengono forniti gli attributi **A** ed **AAAA** necessari per l'inserimento di campi IPv4 ed IPv6 rispettivamente).

---

<sup>4</sup><http://www.bind9-ldap.bayour.com/bind-sdb-ldap-1.0.tar.gz>

<sup>5</sup><https://www.isc.org>

<sup>6</sup><ftp://ftp.isc.org/isc/bind9/9.9.7/bind-9.9.7.tar.gz>

<sup>7</sup><http://www.bind9-ldap.bayour.com/dnszone-schema.txt>

## 11 Albero LDAP

Questa é la struttura di una parte dell'albero LDAP dopo aver lanciato lo script.

- gruppo2.labreti.it
  - HOST
    - \* gruppo2.labreti.it
      - @
      - ns
      - vaio
      - lenovo
      - asus
      - acer
    - \* 2.168.192.in-addr.arpa
      - @
      - 1
      - 10
      - 20
      - 30
      - 40
    - \* 0.2.0.0.2.ip6.arpa
      - @
      - 1.0.0.0
      - 0.1.0.0
      - 0.2.0.0
      - 0.3.0.0
      - 0.4.0.0

## 12 Problemi riscontrati

Nella precente configurazione, LDAP era stato implementato con il protocollo `ldaps` (LDAP over TLS/SSL). Abbiamo tentato l'allacciamento DNS attraverso questa configurazione ma non abbiamo ottenuto alcun risultato utilizzando l'URI `ldaps://indirizzo/` in `named.conf`. Per questo motivo é necessario cambiare `slapd.service` modificando l'opzione `ExecStart` così:

```
ExecStart=/usr/bin/slapd -u ldap -g ldap -h "ldap:/// ldaps://"
```

Aggiungendo `ldap://`, `slapd` rimane in ascolto anche sulla porta standard oltre che su quella sicura. A questo punto verifichiamo il corretto funzionamento di `named` con la nuova configurazione attraverso questi comandi:

```
sudo systemctl daemon-reload
sudo systemctl restart slapd.service
sudo systemctl status slapd.service
sudo systemctl restart named.service
sudo systemctl status named.service
```

e controlliamo l'assenza di errori.

É anche possibile abilitare il `tls` mediante l'opzione `!x-tls` in coda all'URI ma anche questa soluzione non ha prodotto risultati. Navigando in rete, sembra che in molti abbiano riscontrato il nostro stesso problema. Il backend con `ldaps` esula dalle richieste di progetto.

## Part III

# Listati

Per problemi di spazio siamo andati a capo dove necessario. Per il corretto funzionamento, riferirsi direttamente al codice sorgente.

### 13 domain.conf

File di configurazione per il server DNS, in cui vengono anche definiti tutti gli host ed alcune configurazioni per il backed LDAP.

```
#
# domain.conf
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.
#   masotti@student.unife.it>
#           dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or
#   modify
#   it under the terms of the GNU General Public License as
#   published by
#   the Free Software Foundation, either version 3 of the License,
#   or
#   (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public
#   License
#   along with networks-lab. If not, see <http://www.gnu.org/
#   licenses/>.
#

# BIND DNS configuration file.
#
#

#
# File locations.
#
```

```

namedConfFile="/etc/named.conf"
zoneFilesDir="/var/named"

#
# DNS config.
#
# Server administrator.
admin="root"

DNSPort="53"
domain="gruppo2.labreti.it"

DNSIPv4Addr="192.168.2.10"
# IPv6 address must be written fully.
DNSIPv6Addr="2002:0000:0000:0000:0000:0000:0000:0001"

IPv4Subnet="192.168.2.0"
IPv4NetMask="24"

IPv6Subnet="2002:0000:0000:0000:0000:0000:0000:0000"
IPv6NetMask="64"

#
# DNS Expiration Times
#
TTL="2d"
refresh="3h"
retry="1h"
expiry="1w"
minimum="1d"
# TTL used for undefined TTLs.
defaultTTL="86400"

#
# LDAP Backend
#

ldapServerIPv4Addr="192.168.2.10"
ldapServerIPv6Addr="2002::1"
ldapRootPwd="gruppo2"

#
# Host infos.
#
hostsOu="HOST"
hostNames="vaio lenovo asus acer"
IPv4HostAddr=(192.168.2.50 192.168.2.20 192.168.2.30 192.168.2.40)
# IPv6 addresses must be written fully.

```

```
IPv6HostAddr=(2002:0000:0000:0000:0000:0000:0000:0010
  2002:0000:0000:0000:0000:0000:0000:0020
  2002:0000:0000:0000:0000:0000:0000:0030
  2002:0000:0000:0000:0000:0000:0000:0040)
hostTTL=(86400 86400 85400)

#
# Resolv files.
#
resolvconfFile="/etc/resolv.conf"
resolvconfconfFile="/etc/resolvconf.conf"
```

## 14 setup\_dns.sh

Script per il setup di BIND. Qui si può scegliere tra cinque opzioni:

- `./setup_dns -az`
  - Esegue l'installazione automatica creando i file di zona e modificando `/etc/named.conf`.
- `./setup_dns -al`
  - Esegue l'installazione automatica creando i file `ldif` e caricandoli nel database LDAP. Viene inoltre modificando `/etc/named.conf` in modo che utilizzi il server LDAP come backend.
- `./setup_dns -aaz` ed `./setup_dns -aal`
  - Come le due operazioni sopra ma vengono modificate le impostazioni di rete automaticamente.
- `./setup_dns -d`
  - Cancella la configurazione per `/etc/resolvconf.conf` in modo che al riavvio del computer vengano ripristinate le precedenti impostazioni di `/etc/resolv.conf`

```
#!/bin/bash
```

```
#
# setup_dns.sh
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.masotti@student.unife.it>
#           dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
#

# This script reads the configuration from domain.conf file and creates a DNS
# server in a few seconds.
# You can choose between bind zone files or the ldap backend.
```

```

#

function disableResolvconfconf ()
{
    rcc="$1"
    rc="$2"
    echo -en "Disabling "$rcc" entries... "

    echo -en "\
resolv_conf="$rc"\n\
" > "$rcc"

    # Force resolvconf to update all its subscribers.
    resolvconf -u

    echo -en "[DONE]\n"

    return
}

function checkZones ()
{
    zoneName="$1"
    zoneFile="$2"

    echo -en "Checking zone file of $zoneName... \n"

    named-checkzone "$zoneName" "$zoneFile"
    if [ "$?" -ne 0 ]; then
        echo -en "[ERROR] Invalid zone file configuration \
("$zoneFile"). Aborting.\n"
        exit 1
    fi

    return
}

# Function that adds the generated ldif file to the LDAP server.
function addZoneToLDAP ()
{
    echo -en "Adding "$currentZone".zone.ldif to LDAP database... "
    ldapadd -H ldap://"ldapServerIPv4Addr" -x -w "$ldapRootPwd" -D \
"cn=$admin","$domainDn" -f "$currentZone".zone.ldif
    #if [ "$?" -ne 0 ]; then
    #    echo -en "[FAILED]\n"
    #    exit 1
}

```



```

        #fi
        echo -en "[DONE]\n"

        return
    }

    # Function that extracts the IPv4 host zone component of an address.
    function getIPv4AddrComp ()
    {

        IPv4Addr="$1"
        tmp=""

        IFS='.' read -ra tmp <<< "$IPv4Addr"
        IPv4HostAddrComp="${tmp[3]}"
        echo "$IPv4HostAddrComp"

        return
    }

    # Function that extracts the IPv4 zone name.
    function getIPv4ZoneName ()
    {

        IPv4Addr="$1"
        type="$2"
        tmp=""

        # Get current zone name.
        IFS='.' read -ra tmp <<< "$IPv4Addr"
        currentZone="${tmp[2]}"."${tmp[1]}"."${tmp[0]}.in-addr.arpa"
        echo "$currentZone"

        return
    }

    # Function that extracts the IPv6 host zone component of an address.
    function getIPv6AddrComp ()
    {

        IPv6Addr="$1"
        tmp=""

        tmp=$(echo "$IPv6Addr" | rev)

```

```

IFS=: read -ra tmp <<< "$tmp"
tmp=$(echo "$tmp" | sed 's/.\{1\}/&./g')
IPv6HostAddrComp="${tmp:0:(-1)}"
echo "$IPv6HostAddrComp"

return

}

# Function that extracts the IPv6 zone name.
function getIPv6ZoneName ()
{

    IPv6Addr="$1"
    type="$2"
    tmp=""

    # Get current zone name.
    tmp="${IPv6Addr:0:(-5)}"
    tmp=$(echo "$tmp" | rev)
    IFS=: read -ra tmp <<< "$tmp"
    tmp=""${tmp[0]}""${tmp[1]}""${tmp[2]}"\
"${tmp[3]}""${tmp[4]}""${tmp[5]}""${tmp[6]}""
    tmp=$(echo "$tmp" | sed 's/.\{1\}/&./g')
    if [ "$type" == "file" ]; then
        # Zone files have the '.' after arpa.
        currentZone=""$tmp"ip6.arpa."
    else
        # ldap db.
        currentZone=""$tmp"ip6.arpa"
    fi
    echo "$currentZone"

    return

}

# Function that writes common information in /etc/named.conf in both cases.
function writeGloblToNamedConf ()
{

    echo -en "\
options {\n\
\tldirectory \"$zoneFilesDir\";\n\
\tlisten-on port \"$DNSPort\" { any; };\n\
\tlisten-on-v6 port \"$DNSPort\" { any; };\n\
\tlisten-on port \"$DNSPort\" { \"$DNSIPv4Addr\"; \"$DNSIPv6Addr\"; };\n\
\tallow-recursion { \"$IPv4Subnet\"/\"$IPv4NetMask\"; \"$DNSIPv4Addr\"; \

```

```

"$IPv6Subnet"/"$IPv6NetMask"; "$DNSIPv6Addr";};\n\
};\n
zone \".\" {\n\
\ttype hint;\n\
\tfile \"root.hint\";\n\
};\n\n\
" > "$namedConfFile"

        return
}

# Function which writes ldap type zone entries.
function writeZoneInfoToNamedConfLDAP ()
{

        currentZone="$1"
        currentZoneDn="$2"

        echo -en "\
zone \"\"$currentZone\" IN {
\ttype master;
\tatabase \"ldap \
ldap://\"$ldapServerIPv4Addr\"/zoneName=\"$currentZone\",ou=\"$hostsOu\", \
\"$currentZoneDn\" \"$defaultTTL\"";
\tallow-query { any; };
};\n\n\
" >> "$namedConfFile"

        return
}

# Function which writes classical zone entries.
function writeZoneInfoToNamedConfFILE ()
{

        currentZone="$1"

        echo -en "\
zone \"\"$currentZone\" IN {\n\
\ttype master;\n\
\tfile \"\"$currentZone\"";\n\
\tallow-query { any; };;\n\
};\n\n\
" >> "$namedConfFile"

```

```

        return
    }

# Core functions.
# All, zones.
function az ()
{
    # Try to create dir.
    mkdir -p "$zoneFilesDir"

    echo -en "Writing configuration to files... "
    echo "$currentZone"

    # Write common config to named.conf file.
    writeGloblToNamedConf

    # Write direct zone info to named.conf file.
    writeZoneInfoToNamedConfFile "$currentZone"

    # Change file permissions.
    chown root:named "$namedConfFile"

    # If zonefile is changed on the fly serial must be incremented each
    # time.
    # But since we restart named, we don't care about this problem.
    serial="01"
    i=0
    echo -en "\
\$TTL \"$TTL\"\n\
@IN SOA tns.\"$domain\".\t\"$admin\".\"$domain\". (\n\
$(date "+%Y%m%d")\"$serial\" \t; serial\n\
\"$refresh\" \t; refresh\n\
\"$retry\" \t; retry\n\
\"$expiry\" \t; expiry\n\
\"$minimum\" )\t; minimum\n\
@IN NS tns.\"$domain\".\n\
ns.\"$domain\".\tIN AAAA\t\"$DNSIPv6Addr\"\n\
ns.\"$domain\".\tIN A\t\"$DNSIPv4Addr\"\n\
$(for host in $hostNames; do
    echo -en "\"$host\".\"$domain\".\tIN A\t"
    echo -en "${IPv4HostAddr[$i]}\n"
    echo -en "\"$host\".\"$domain\".\tIN AAAA\t"
    echo -en "${IPv6HostAddr[$i]}\n"
    i=$((i+1))
done)
" > "$zoneFilesDir/$currentZone"

    checkZones "$currentZone" "$zoneFilesDir/$currentZone"

```

```

# Reverse zones.

# Reverse zone IPv4.

currentZone=$(getIPv4ZoneName "$IPv4Subnet")
writeZoneInfoToNamedConfFile "$currentZone"

i=0
echo -en "\
\$ORIGIN "$currentZone".\n\
\$TTL "$TTL"\n\
@ \tIN \tSOA \tns."$domain".\t"$admin"."$domain". (\n\
$(date "+%Y%m%d")"$serial"\t; serial\n\
"$refresh"\t; refresh\n\
"$retry"\t; retry\n\
"$expiry"\t; expiry\n\
"$minimum" )\t; minimum\n\
"$currentZone".\tIN \tNS \tns."$domain".\n\
$(DNSIPv4AddrComp=$(getIPv4AddrComp "$DNSIPv4Addr")
echo -en "\"$DNSIPv4AddrComp"\tIN \tPTR \tns."$domain".\n")
$(for host in $hostNames; do
    IPv4HostAddrComp=$(getIPv4AddrComp "${IPv4HostAddr[$i]}")
    echo -en "\"$IPv4HostAddrComp"\tIN \tPTR \t"$host"."$domain".\n"
    i=$((i+1))
done)
" > "$zoneFilesDir/$currentZone"

checkZones "$currentZone" "$zoneFilesDir/$currentZone"

# Reverse zone IPv6

currentZone=$(getIPv6ZoneName "$IPv6Subnet" "file")
writeZoneInfoToNamedConfFile "$currentZone"

# N.B. Absence of '.' at the end of "$currentZone" because already included.
i=0
echo -en "\
\$ORIGIN "$currentZone"\n\
\$TTL "$TTL"\n\
@ \tIN \tSOA \tns."$domain".\t"$admin"."$domain". (\n\
$(date "+%Y%m%d")"$serial"\t; serial\n\
"$refresh"\t; refresh\n\
"$retry"\t; retry\n\
"$expiry"\t; expiry\n\
"$minimum" )\t; minimum\n\
"$currentZone".\tIN \tNS \tns."$domain".\n\
$(DNSIPv6AddrComp=$(getIPv6AddrComp "$DNSIPv6Addr")
echo -en "\"$DNSIPv6AddrComp"\tIN \tPTR \tns."$domain".\n")

```

```

$(for host in $hostNames; do
    IPv6HostAddrComp=$(getIPv6AddrComp "${IPv6HostAddr[$i]}")
    echo -en "$IPv6HostAddrComp\tIN\tPTR\t"$host"."$domain".\n"
    i=$((i+1))
done)
" > "$zoneFilesDir/$currentZone"

    checkZones "$currentZone" "$zoneFilesDir/$currentZone"

    echo -en "[DONE]\n"

    return
}

# All, LDAP backend.
function al ()
{

    # Direct zone.

    serial="01"

    # Write initial configuration to named.conf.
    writeGloblToNamedConf

    # Write zone info
    writeZoneInfoToNamedConfLDAP "$currentZone" "$currentZoneDn"

    echo -en "Generating zone "$currentZone" on file \
"$currentZone".zone.ldif... "
    i=0
    echo -en "\
# Zone definition.\n\
dn: zoneName="$currentZone",ou="$hostsOu","$currentZoneDn"\n\
objectClass: dNSZone\n\
zoneName: "$currentZone"\n\
relativeDomainName: "$currentZone"\n\n\
# SOA.\n\
dn: relativeDomainName=@,zoneName="$currentZone",ou="$hostsOu",\
"$currentZoneDn"\n\
objectClass: dNSZone\n\
relativeDomainName: @\n\
zoneName: "$currentZone"\n\
dNSTTL: "$defaultTTL"\n\
DNSClass: NS\n\
SOARecord: ns."$domain". "$admin"."$domain". $(date "+%Y%m%d")"$serial"\
"$TTL" "$refresh" "$retry" "$expiry"\n\
NSRecord: ns."$domain".\n\n\

```

```

# Name server.\n\
dn: relativeDomainName=ns,zoneName="$currentZone",ou="$hostsOu",\
"$currentZoneDn"\n\
objectClass: dNSZone\n\
relativeDomainName: ns\n\
zoneName: "$currentZone"\n\
dNSTTL: "$defaultTTL"\n\
dNSClass: IN\n\
aRecord: "$DNSIPv4Addr"\n\
aAAARRecord: "$DNSIPv6Addr"\n\n\
$(for host in $hostNames; do
    echo -en "\
# Host "$host".\n\
dn: relativeDomainName="$host",zoneName="$currentZone",ou="$hostsOu",\
"$currentZoneDn"\n\
objectClass: dNSZone\n\
relativeDomainName: "$host"\n\
zoneName: "$currentZone"\n\
$(if [ "${#hostTTL[@]}" -le "$i" ]; then
    echo -en "dNSTTL: "$defaultTTL"\n"
else
    echo -en "dNSTTL: "${hostTTL[$i]}\n"
fi)
dNSClass: IN\n\
aRecord: "${IPv4HostAddr[$i]}\n\
aAAARRecord: "${IPv6HostAddr[$i]}\n\n"
    i=$((i+1))
done)
" > "$currentZone".zone.ldif

    echo -en "[DONE] \n"

    addZoneToLDAP

    # IPv4 reverse zone.
    currentZone=$(getIPv4ZoneName "$IPv4Subnet")
    currentZoneDn="$domainDn"

    # Write zone info
    writeZoneInfoToNamedConfLDAP "$currentZone" "$currentZoneDn"

    echo -en "Generating zone "$currentZone" on file \
"$currentZone".zone.ldif... "
    i=0
    echo -en "\
# Zone definition.\n\
dn: zoneName="$currentZone",ou="$hostsOu","$currentZoneDn"\n\
objectClass: dNSZone\n\
zoneName: "$currentZone"\n\

```

```

relativeDomainName: "$currentZone"\n\n\
# SOA.\n\
dn: relativeDomainName=@,zoneName="$currentZone",ou="$hostsOu",\
"$currentZoneDn"\n\
objectClass: dNSZone\n\
relativeDomainName: @\n\
zoneName: "$currentZone"\n\
dNSTTL: "$defaultTTL"\n\
dNSClass: NS\n\
sOARRecord: ns."$domain". "$admin"."$domain". $(date +%Y%m%d)"$serial"\
"$TTL" "$refresh" "$retry" "$expiry"\n\
nSRecord: ns."$domain".\n\n\
# Name server.\n\
dn: relativeDomainName=ns,zoneName="$currentZone",ou="$hostsOu",\
"$currentZoneDn"\n\
objectClass: dNSZone\n\
$(IFS='.' read -ra tmp <<< "$DNSIPv4Addr"
DNSIPv4AddrComp="${tmp[3]}"
echo -en "relativeDomainName: "$DNSIPv4AddrComp"\n")
zoneName: "$currentZone"\n\
pTRRecord:ns."$domain".\n\n\
$(for host in $hostNames; do
    IPv4HostAddrComp=$(getIPv4AddrComp "${IPv4HostAddr[$i]}")
    echo -en "\
# Host "$host".\n\
dn: relativeDomainName="$host",zoneName="$currentZone",ou="$hostsOu",\
"$currentZoneDn"\n\
objectClass: dNSZone\n\
relativeDomainName: "$IPv4HostAddrComp"\n\
zoneName: "$currentZone"\n\
pTRRecord:"$host"."$domain".\n\n\
    i=$((i+1))
done)
" > "$currentZone".zone.ldif

echo -en "[DONE] \n"

addZoneToLDAP

# Get current Zone Name.
currentZone=$(getIPv6ZoneName "$DNSIPv6Addr" "ldap")
currentZoneDn="$domainDn"

# Write zone info
writeZoneInfoToNamedConfLDAP "$currentZone" "$currentZoneDn"

echo -en "Generating zone "$currentZone" on file \
"$currentZone".zone.ldif... "
i=0

```



```

        echo -en "\
# Zone definition.\n\
dn: zoneName=\"$currentZone\",ou=\"$hostsOu\",\"$currentZoneDn\"\n\
objectClass: dNSZone\n\
zoneName: \"$currentZone\"\n\
relativeDomainName: \"$currentZone\"\n\n\
# SOA.\n\
dn: relativeDomainName=@,zoneName=\"$currentZone\",ou=\"$hostsOu\", \"\
\"$currentZoneDn\"\n\
objectClass: dNSZone\n\
relativeDomainName: @\n\
zoneName: \"$currentZone\"\n\
dNSTTL: \"$defaultTTL\"\n\
dNSClass: NS\n\
sOARRecord: ns.\"$domain\". \"$admin\".\"$domain\". $(date +%Y%m%d)\"$serial\" \"\
\"$TTL\" \"$refresh\" \"$retry\" \"$expiry\"\n\
nSRecord: ns.\"$domain\".\n\n\
# Name server.\n\
dn: relativeDomainName=ns,zoneName=\"$currentZone\",ou=\"$hostsOu\", \"\
\"$currentZoneDn\"\n\
objectClass: dNSZone\n\
$(DNSIPv6AddrComp=$(getIPv6AddrComp \"$DNSIPv6Addr\")
echo -en "relativeDomainName: \"$DNSIPv6AddrComp\"\n")
zoneName: \"$currentZone\"\n\
pTRRecord:ns.\"$domain\".\n\n\
$(for host in $hostNames; do
    IPv6HostAddrComp=$(getIPv6AddrComp "${IPv6HostAddr[$i]}")
    echo -en "\
# Host \"$host\".\n\
dn: relativeDomainName=\"$host\",zoneName=\"$currentZone\",ou=\"$hostsOu\", \"\
\"$currentZoneDn\"\n\
objectClass: dNSZone\n\
relativeDomainName: \"$IPv6HostAddrComp\"\n\
zoneName: \"$currentZone\"\n\
pTRRecord:\"$host\".\"$domain\".\n\n\
    i=$((i+1))
done)
" > "$currentZone".zone.ldif

    echo -en "[DONE] \n"

    addZoneToLDAP

    return

}

# Check if root is running this script.
if [ "$UID" -ne 0 ] || [ "$1" == "-h" ]; then

```

```

        echo -en "[ERROR] You must be root to run this script.\n"
        echo -en "sudo -u root $0\n"
        echo -en "OR\n"
        echo -en "sudo $0\n"
        echo -en "Help:\n"
        echo -en "\t$0 -d ;deletes resolv settings.\n"
        echo -en "\t$0 -az ;makes all the necessary configuration with the \
zone files.\n"
        echo -en "\t$0 -al ;makes all the necessary configuration with LDAP \
as database.\n"
        echo -en "\t$0 -aaz ;makes all the necessary configuration with the \
zone files and modifies network parameters.\n"
        echo -en "\t$0 -aal ;makes all the necessary configuration with LDAP \
as database and modifies network parameters.\n"
        exit 1
    fi

# Load configuration from domain.conf file
if [ -f "domain.conf" ]; then
    source "domain.conf"
else
    echo -en "[ERROR] No domain.conf file found.\n"
    exit 1
fi

# Global variables
tmp=""

# Get domain Distinguished Name.
IFS='.' read -ra tmp <<< "$domain"
domainDn="dc="${tmp[0]}",dc="${tmp[1]}",dc="${tmp[2]}"

# Get current zone values.
currentZone="$domain"
currentZoneDn="$domainDn"

# End of global variables.

if [ "$1" == "-d" ]; then
    disableResolvconfconf "$resolvconfconfFile" "$resolvconfFile"
    exit 0
fi

# stop BIND daemon.
systemctl stop named.service

systemctl disable named.service

```

```

# Make all.
if [ "$1" == "-aaz" ] || [ "$1" == "-aal" ]; then
    # Call the network changing script.
    ./chg_nw_settings.sh
    # Call the resolv.conf script
    ./chg_resolv_settings.sh
fi

# Call az.
if [ "$1" == "-az" ] || [ "$1" == "-aaz" ]; then
    az
fi

# Call al.
if [ "$1" == "-al" ] || [ "$1" == "-aal" ]; then
    al
fi

# Enable named.service.
systemctl enable named.service

# Start BIND daemon.
systemctl start named.service

exit 0

# Another way instead of using read.
#reverse=$(echo "$DNSIPv4Addr" | rev)
#indexPoint=$(expr index "$reverse" .)
#indexPoint=$((indexPoint-1))
#reverseString=${reverse:0:indexPoint}
#domainString=$(echo "$reverseString" | rev)

```

## 15 chg\_nw\_settings.sh

Script per l'assegnazione degli indirizzi di rete (validi fino al riavvio della macchina) sull'interfaccia cablata.

Lo script `chg_nw_settings.sh` ci permette di assegnare gli indirizzi di tipo IPv4 ed IPv6, le sottoreti e l'indirizzo di broadcast IPv4 all'interfaccia di rete cablata. Questi indirizzi corrispondono agli indirizzi su cui `named` ascolta le richieste.

Per quanto riguarda l'assegnazione di indirizzi IPv6 abbiamo riscontrato problemi inizialmente per la struttura stessa del protocollo di rete. Anche se si assegna un indirizzo di questo tipo ad un'interfaccia il computer aspetta la *portante* in modo da evitare collisioni con altri computer. Quindi un modo di risolvere il problema é quello di attaccare un cavo di rete all'interfaccia.

```
#!/bin/bash

#
# chg_nw_settings.sh
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.masotti@student.unife.it>
#                               dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
#

# This script changes the network configuration for the current session.
# At the next reboot these settings will be forgotten.

# Check if root is running this script.
if [ "$UID" -ne 0 ]; then
    echo -en "[ERROR] You must be root to run this script.\n"
    echo -en "sudo -u root $0\n"
    echo -en "OR\n"
    echo -en "sudo $0\n"
    exit 1
fi

# Load configuration from domain.conf file
```

```

if [ -f "domain.conf" ]; then
    source "domain.conf"
else
    echo -en "[ERROR] No domain.conf file found.\n"
    exit 1
fi

# If Network Manager exists and it's running.
ps aux | grep NetworkManager | grep -v grep > /dev/null
if [ "$?" -eq 0 ]; then
    killall NetworkManager
    sleep 1
fi

# Get dynamic addresses for all interfaces.
# If dhcpcd is not running start it.
ps aux | grep dhcpcd | grep -v grep > /dev/null
if [ "$?" -ne 0 ]; then
    dhcpcd > /dev/null
else
    killall dhcpcd
    dhcpcd > /dev/null
fi

# Get a static ip address for one interface.
# Set prefix of network interface (e.g. enp).
nwCardNamePrefix="enp"
# Get wired ethernet name.
interface=$(ip addr | grep "$nwCardNamePrefix" -m 1 | awk ' {print $2} ')
interface=${interface:0:(-1)}
# Set v4 and v6 addresses, valid until reboot.
# Remove previous nw config on device.
ip addr flush dev "$interface"
# Change network interface's ip with ip command.
# Set dev up.
ip link set up dev "$interface"
# v4
ip addr add "$DNSIPv4Addr"/"$IPv4NetMask" brd + dev "$interface"
# Wait to avoid problems setting IPv6 address.
sleep 1
# v6
ip addr add scope global "$DNSIPv6Addr"/"$IPv6NetMask" dev "$interface"
ip link set multicast on dev "$interface"

exit 0

```

## 16 chg\_resolv\_settings.sh

Script per cambiare il file `/etc/resolvconf.conf` in modo che la risoluzione dei nomi host avvenga con il nostro server DNS.

Per fare funzionare le query su un client é necessario cambiare alcune impostazioni di rete in modo che le ricerche con `dig` vengano fatte con il nostro server (senza bisogno di esplicitarlo) é necessario modificare il file `/etc/resolv.conf`, mettendogli in testa gli indirizzi del nostro server. Nel nostro caso gli indirizzi sono `192.168.2.1` e `2002::1`. Per fare un modo di non esplicitare sempre il dominio di ricerca accanto all'hostname (nel nostro caso `gruppo2.labreti.it`), quando usiamo il comando `host`, é necessario aggiungere anche questo nel file. Spesso, tuttavia, le informazioni contenute in `/etc/resolv.conf` vengono sovrascritte ad ogni riavvio, o addirittura ad ogni evento provocato dai piú disparati network manager. Per questo motivo abbiamo deciso di agire su `/etc/resolvconf.conf` che ci permette di fare tutte le cose scritte precedentemente senza che le informazioni vengano sovrascritte.

```
#!/bin/bash

#
# chg_resolv_settings.sh
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.masotti@student.unife.it>
#                               dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
#

# This script changes the resolvconf.conf file so that DNS queries are
# possible without explicitly specifying the DNS server.

# Check if root is running this script.
if [ "$UID" -ne 0 ]; then
    echo -en "[ERROR] You must be root to run this script.\n"
    echo -en "sudo -u root $0\n"
    echo -en "OR\n"
    echo -en "sudo $0\n"
    exit 1

```

```

fi

# Load configuration from domain.conf file
if [ -f "domain.conf" ]; then
    source "domain.conf"
else
    echo -en "[ERROR] No domain.conf file found.\n"
    exit 1
fi

# Write new /etc/resolvconf.conf
echo -en "Overwriting $resolvconfconfFile... "
echo -en "\
search_domains=\"$domain\"\n\
name_servers=\"\" $DNSIPv4Addr" "$DNSIPv6Addr"\" \"\n\
resolv_conf=\"$resolvconfFile\"\n\
" > "$resolvconfconfFile"
echo -en "[DONE]\n"

# Force resolvconf to update all its subscribers.
resolvconf -u

exit 0

```

## 17 Zona diretta zone file

Di seguito troviamo le tre zone sotto forma di zone file.

```
;
; gruppo2.labreti.it
;
; Copyright (C) 2016 frnmst (Franco Masotti) <franco.
;   masotti@student.unife.it>
;               dannylessio (Danny Lessio)
;
; This file is part of networks-lab.
;
; networks-lab is free software: you can redistribute it and/or
;   modify
;   it under the terms of the GNU General Public License as
;   published by
;   the Free Software Foundation, either version 3 of the License,
;   or
;   (at your option) any later version.
;
; networks-lab is distributed in the hope that it will be useful,
;   but WITHOUT ANY WARRANTY; without even the implied warranty of
;   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
;   GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public
;   License
;   along with networks-lab. If not, see <http://www.gnu.org/
;   licenses/>.
;
```

```
$TTL 2d
@      IN      SOA      ns.gruppo2.labreti.it.  root.gruppo2.
; labreti.it. (
2015040901      ; serial
3h      ; refresh
1h      ; retry
1w      ; expiry
1d )      ; minimum
@      IN      NS       ns.gruppo2.labreti.it.
ns.gruppo2.labreti.it.  IN      AAAA
; 2002:0000:0000:0000:0000:0000:0000:0001
ns.gruppo2.labreti.it.  IN      A       192.168.2.1
vaio.gruppo2.labreti.it.      IN      A       192.168.2.10
vaio.gruppo2.labreti.it.      IN      AAAA
; 2002:0000:0000:0000:0000:0000:0000:0010
lenovo.gruppo2.labreti.it.     IN      A       192.168.2.20
```



```

lenovo.gruppo2.labreti.it.      IN      AAAA
    2002:0000:0000:0000:0000:0000:0000:0020
asus.gruppo2.labreti.it.       IN      A      192.168.2.30
asus.gruppo2.labreti.it.       IN      AAAA
    2002:0000:0000:0000:0000:0000:0000:0030
acer.gruppo2.labreti.it.       IN      A      192.168.2.40
acer.gruppo2.labreti.it.       IN      AAAA
    2002:0000:0000:0000:0000:0000:0000:0040

```

## 18 Zona inversa IPv4 zone file

```
;
; 2.168.192.in-addr.arpa.
;
; Copyright (C) 2016 frnmst (Franco Masotti) <franco.
;   masotti@student.unife.it>
;               dannylessio (Danny Lessio)
;
; This file is part of networks-lab.
;
; networks-lab is free software: you can redistribute it and/or
;   modify
; it under the terms of the GNU General Public License as
;   published by
; the Free Software Foundation, either version 3 of the License,
;   or
; (at your option) any later version.
;
; networks-lab is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public
;   License
; along with networks-lab. If not, see <http://www.gnu.org/
;   licenses/>.
;

$ORIGIN 2.168.192.in-addr.arpa.
$TTL 2d
@      IN      SOA      ns.gruppo2.labreti.it.  root.gruppo2.
      labreti.it. (
2015040901      ; serial
3h      ; refresh
1h      ; retry
1w      ; expiry
1d )      ; minimum
2.168.192.in-addr.arpa. IN      NS      ns.gruppo2.labreti.it.
1      IN      PTR      ns.gruppo2.labreti.it.
10     IN      PTR      vaio.gruppo2.labreti.it.
20     IN      PTR      lenovo.gruppo2.labreti.it.
30     IN      PTR      asus.gruppo2.labreti.it.
40     IN      PTR      acer.gruppo2.labreti.it.
```

## 19 Zona inversa IPv6 zone file

```
;
; 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.arpa
;
;
; Copyright (C) 2016 frnmst (Franco Masotti) <franco.
;   masotti@student.unife.it>
;               dannylessio (Danny Lessio)
;
; This file is part of networks-lab.
;
; networks-lab is free software: you can redistribute it and/or
;   modify
; it under the terms of the GNU General Public License as
;   published by
; the Free Software Foundation, either version 3 of the License,
;   or
; (at your option) any later version.
;
; networks-lab is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public
;   License
; along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
;

$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa.
$TTL 2d
@      IN      SOA      ns.gruppo2.labreti.it.  root.gruppo2.
      labreti.it. (
2015040901      ; serial
3h      ; refresh
1h      ; retry
1w      ; expiry
1d )      ; minimum
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.arpa.
      IN      NS      ns.gruppo2.labreti.it.
1.0.0.0 IN      PTR    ns.gruppo2.labreti.it.
0.1.0.0 IN      PTR    vaio.gruppo2.labreti.it.
0.2.0.0 IN      PTR    lenovo.gruppo2.labreti.it.
0.3.0.0 IN      PTR    asus.gruppo2.labreti.it.
0.4.0.0 IN      PTR    acer.gruppo2.labreti.it.
```

## 20 named.conf per i file di zona

Nota bene: con la notazione IN nell'instazione delle zone si identifica la classe di dati INternet. Se omessa nella definizione é comunque assunta di default.

```
/*
 * named.conf.zones
 *
 * Copyright (C) 2016 frnmst (Franco Masotti) <franco.
 *   masotti@student.unife.it>
 *
 *   dannylessio (Danny Lessio)
 *
 * This file is part of networks-lab.
 *
 * networks-lab is free software: you can redistribute it and/or
 *   modify
 * it under the terms of the GNU General Public License as
 *   published by
 * the Free Software Foundation, either version 3 of the License,
 *   or
 * (at your option) any later version.
 *
 * networks-lab is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public
 *   License
 * along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
 *
 */

options {
    directory "/var/named";
    listen-on port 53 { any; };
    listen-on-v6 port 53 { any; };
    listen-on port 53 { 192.168.2.1;
        2002:0000:0000:0000:0000:0000:0000:0001; };
    allow-recursion { 192.168.2.0/24; 192.168.2.1;
        2002:0000:0000:0000:0000:0000:0000:0000/64;
        2002:0000:0000:0000:0000:0000:0000:0001; };
};

zone "gruppo2.labreti.it" IN {
    type master;
    file "gruppo2.labreti.it";
    allow-query { any; };
};
```

```
};

zone "2.168.192.in-addr.arpa" IN {
    type master;
    file "2.168.192.in-addr.arpa";
    allow-query { any; };
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
arpa." IN {
    type master;
    file
        "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
        ip6.arpa.";
    allow-query { any; };
};
```

## 21 Zona diretta LDAP

Di seguito troviamo le tre zone sotto forma di file ldif.

```
#
# gruppo2.labreti.it.zone.ldif
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.
#   masotti@student.unife.it>
#           dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or
#   modify
#   it under the terms of the GNU General Public License as
#   published by
#   the Free Software Foundation, either version 3 of the License,
#   or
#   (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
#   GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public
#   License
#   along with networks-lab. If not, see <http://www.gnu.org/
#   licenses/>.
#

# Zone definition.
dn: zoneName=gruppo2.labreti.it,ou=HOST,dc=gruppo2,dc=labreti,dc=
  it
objectClass: dNSZone
zoneName: gruppo2.labreti.it
relativeDomainName: gruppo2.labreti.it

# SOA.
dn: relativeDomainName=@,zoneName=gruppo2.labreti.it,ou=HOST,dc=
  gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: @
zoneName: gruppo2.labreti.it
dNSTTL: 86400
dNSClass: NS
sOARRecord: ns.gruppo2.labreti.it. root.gruppo2.labreti.it.
  2015051901 2d 3h 1h 1w
```

nSRecord: ns.gruppo2.labreti.it.

# Name server.

dn: relativeDomainName=ns,zoneName=gruppo2.labreti.it,ou=HOST,dc=gruppo2,dc=labreti,dc=it  
objectClass: dNSZone  
relativeDomainName: ns  
zoneName: gruppo2.labreti.it  
dNSTTL: 86400  
dNSClass: IN  
aRecord: 192.168.2.10  
aAAARRecord: 2002:0000:0000:0000:0000:0000:0000:0001

# Host vaio.

dn: relativeDomainName=vaio,zoneName=gruppo2.labreti.it,ou=HOST,dc=gruppo2,dc=labreti,dc=it  
objectClass: dNSZone  
relativeDomainName: vaio  
zoneName: gruppo2.labreti.it  
dNSTTL: 86400  
dNSClass: IN  
aRecord: 192.168.2.50  
aAAARRecord: 2002:0000:0000:0000:0000:0000:0000:0010

# Host lenovo.

dn: relativeDomainName=lenovo,zoneName=gruppo2.labreti.it,ou=HOST,dc=gruppo2,dc=labreti,dc=it  
objectClass: dNSZone  
relativeDomainName: lenovo  
zoneName: gruppo2.labreti.it  
dNSTTL: 86400  
dNSClass: IN  
aRecord: 192.168.2.20  
aAAARRecord: 2002:0000:0000:0000:0000:0000:0000:0020

# Host asus.

dn: relativeDomainName=asus,zoneName=gruppo2.labreti.it,ou=HOST,dc=gruppo2,dc=labreti,dc=it  
objectClass: dNSZone  
relativeDomainName: asus  
zoneName: gruppo2.labreti.it  
dNSTTL: 85400  
dNSClass: IN  
aRecord: 192.168.2.30  
aAAARRecord: 2002:0000:0000:0000:0000:0000:0000:0030

# Host acer.

dn: relativeDomainName=acer,zoneName=gruppo2.labreti.it,ou=HOST,dc=gruppo2,dc=labreti,dc=it

```
objectClass: dNSZone
relativeDomainName: acer
zoneName: gruppo2.labreti.it
dNSTTL: 86400
dNSClass: IN
aRecord: 192.168.2.40
aAAARRecord: 2002:0000:0000:0000:0000:0000:0040
```



## 22 Zona inversa IPv4 LDAP

```
#
# 2.168.192.in-addr.arpa.zone.ldif
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.
#   masotti@student.unife.it>
#           dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or
#   modify
#   it under the terms of the GNU General Public License as
#   published by
#   the Free Software Foundation, either version 3 of the License,
#   or
#   (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
#   GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public
#   License
#   along with networks-lab. If not, see <http://www.gnu.org/
#   licenses/>.
#

# Zone definition.
dn: zoneName=2.168.192.in-addr.arpa,ou=HOST,dc=gruppo2,dc=labreti,
   dc=it
objectClass: dNSZone
zoneName: 2.168.192.in-addr.arpa
relativeDomainName: 2.168.192.in-addr.arpa

# SOA.
dn: relativeDomainName=@,zoneName=2.168.192.in-addr.arpa,ou=HOST,
   dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: @
zoneName: 2.168.192.in-addr.arpa
dNSTTL: 86400
dNSClass: NS
sOARRecord: ns.gruppo2.labreti.it. root.gruppo2.labreti.it.
           2015051901 2d 3h 1h 1w
nSRecord: ns.gruppo2.labreti.it.
```

```

# Name server.
dn: relativeDomainName=ns,zoneName=2.168.192.in-addr.arpa,ou=HOST,
   dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 10
zoneName: 2.168.192.in-addr.arpa
pTRRecord:ns.gruppo2.labreti.it.

# Host vaio.
dn: relativeDomainName=vaio,zoneName=2.168.192.in-addr.arpa,ou=
   HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 50
zoneName: 2.168.192.in-addr.arpa
pTRRecord:vaio.gruppo2.labreti.it.

# Host lenovo.
dn: relativeDomainName=lenovo,zoneName=2.168.192.in-addr.arpa,ou=
   HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 20
zoneName: 2.168.192.in-addr.arpa
pTRRecord:lenovo.gruppo2.labreti.it.

# Host asus.
dn: relativeDomainName=asus,zoneName=2.168.192.in-addr.arpa,ou=
   HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 30
zoneName: 2.168.192.in-addr.arpa
pTRRecord:asus.gruppo2.labreti.it.

# Host acer.
dn: relativeDomainName=acer,zoneName=2.168.192.in-addr.arpa,ou=
   HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 40
zoneName: 2.168.192.in-addr.arpa
pTRRecord:acer.gruppo2.labreti.it.

```

## 23 Zona inversa IPv6 LDAP

```
#
# 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.arpa
#   .zone.ldif
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.
#   masotti@student.unife.it>
#           dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or
#   modify
# it under the terms of the GNU General Public License as
#   published by
# the Free Software Foundation, either version 3 of the License,
#   or
# (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public
#   License
# along with networks-lab. If not, see <http://www.gnu.org/
#   licenses/>.
#

# Zone definition.
dn: zoneName
   =0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
   arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
zoneName: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa
relativeDomainName:
   0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
   arpa

# SOA.
dn: relativeDomainName=@,zoneName
   =0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
   arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: @
```

```

zoneName: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa
dNSTTL: 86400
dNSClass: NS
sOARRecord: ns.gruppo2.labreti.it. root.gruppo2.labreti.it.
2015051901 2d 3h 1h 1w
nSRecord: ns.gruppo2.labreti.it.

# Name server.
dn: relativeDomainName=ns,zoneName
=0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 1.0.0.0
zoneName: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa
pTRRecord:ns.gruppo2.labreti.it.

# Host vaio.
dn: relativeDomainName=vaio,zoneName
=0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 0.1.0.0
zoneName: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa
pTRRecord:vaio.gruppo2.labreti.it.

# Host lenovo.
dn: relativeDomainName=lenovo,zoneName
=0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 0.2.0.0
zoneName: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa
pTRRecord:lenovo.gruppo2.labreti.it.

# Host asus.
dn: relativeDomainName=asus,zoneName
=0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it
objectClass: dNSZone
relativeDomainName: 0.3.0.0
zoneName: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
ip6.arpa
pTRRecord:asus.gruppo2.labreti.it.

# Host acer.

```



## 24 named.conf per LDAP

Nota bene: L'URI ha il dn (Distinguished Name) diverso per ogni zona, identificata da ZoneName.

```
/*
 * named.conf.ldap
 *
 * Copyright (C) 2016 frnmst (Franco Masotti) <franco.
 *   masotti@student.unife.it>
 *
 *   dannylessio (Danny Lessio)
 *
 * This file is part of networks-lab.
 *
 * networks-lab is free software: you can redistribute it and/or
 *   modify
 * it under the terms of the GNU General Public License as
 *   published by
 * the Free Software Foundation, either version 3 of the License,
 *   or
 * (at your option) any later version.
 *
 * networks-lab is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public
 *   License
 * along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
 *
 */

options {
    directory "/var/named";
    listen-on port 53 { any; };
    listen-on-v6 port 53 { any; };
    listen-on port 53 { 192.168.2.1;
        2002:0000:0000:0000:0000:0000:0000:0001; };
    allow-recursion { 192.168.2.0/24; 192.168.2.1;
        2002:0000:0000:0000:0000:0000:0000:0000/64;
        2002:0000:0000:0000:0000:0000:0000:0001; };
};

zone "gruppo2.labreti.it" IN {
    type master;
    database "ldap ldap://192.168.2.1/zoneName=gruppo2.labreti
        .it,ou=HOST,dc=gruppo2,dc=labreti,dc=it 86400";
    allow-query { any; };
};
```

```

};

zone "2.168.192.in-addr.arpa" IN {
    type master;
    database "ldap ldap://192.168.2.1/zoneName=2.168.192.in-
        addr.arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it 86400";
    allow-query { any; };
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.ip6.
    arpa" IN {
    type master;
    database "ldap ldap://192.168.2.1/zoneName
        =0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.2.
        ip6.arpa,ou=HOST,dc=gruppo2,dc=labreti,dc=it 86400";
    allow-query { any; };
};

```

## 25 named.service per systemd

Poiché abbiamo compilato il pacchetto bind é necessario cambiare alcune configurazioni per il sistema di init.

```
#
# named.service
#
# Copyright (C) 2016 frnmst (Franco Masotti) <franco.masotti@student.unife.it>
#           dannylessio (Danny Lessio)
#
# This file is part of networks-lab.
#
# networks-lab is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# networks-lab is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with networks-lab. If not, see <http://www.gnu.org/licenses/>.
#

# To be copied to etc/systemd/system/named.service and/or
# /usr/lib/systemd/system/named.service
[Unit]
Description=Internet domain name server
After=network.target

[Service]
ExecStart=/usr/local/sbin/named -f -u named
ExecReload=/usr/local/sbin/rndc reload
ExecStop=/usr/local/sbin/rndc stop

[Install]
WantedBy=multi-user.target
```