Bind

概要

本項目では，DNSサーバとして，Bindの構築とログの確認手法を習得する．
事前知識として，名前解決の仕組みを復習する．

目次

DNSとは

インターネットに接続されるホストは **IPアドレス** で認識される．
しかし，IPアドレスは単なる数値であり，人間には扱いにくい．
そこで，ホストに名前をつけ，IPアドレスを対応付ける，DNS (Domain Name System) が生まれた．

例えば，IPアドレス `133.92.145.49` は，ホスト名 `sleepingbeauty.eng.kagawa-u.ac.jp` に対応している．

```
$ host 133.92.145.49
49.145.92.133.in-addr.arpa domain name pointer sleepingbeauty.eng.kagawa-u.ac.jp.
```

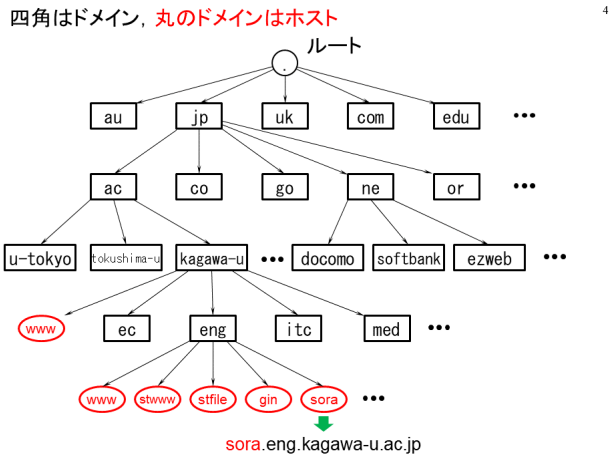
旧来は，hostsファイル(Linuxでは `/etc/hosts`)で全てのホスト情報を管理していた．
更新があると，メールで通知し，各ユーザがFTPサーバからダウンロードする形である．
しかし，ホスト数の増加に伴い，以下の問題が発生した．

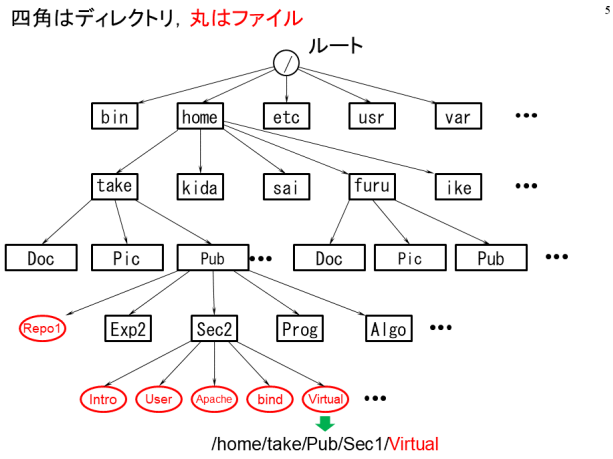
1. hostsファイルの配布によるトラフィック負荷の増大
2. 同じホスト名の名前の衝突
3. アドレスの追加や変更による一貫性の消失

これらを解決するために，階層構造の分散型データベースである，DNSが開発された．

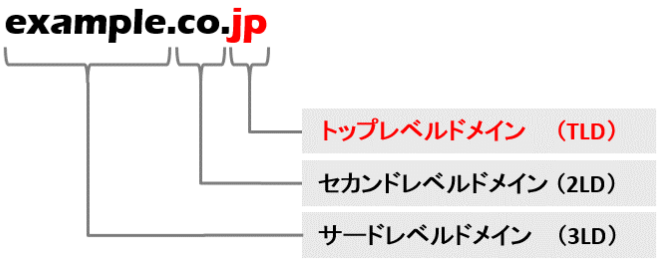
DNSの仕組み

DNSでは，階層構造の実現に，ドメイン名やホスト名という考え方が導入されている．
ドメイン名は組織を表し，ホスト名はその組織が管理しているコンピュータを指している．
これは，多分木で表すことができ，ディレクトリの構図と似ている．



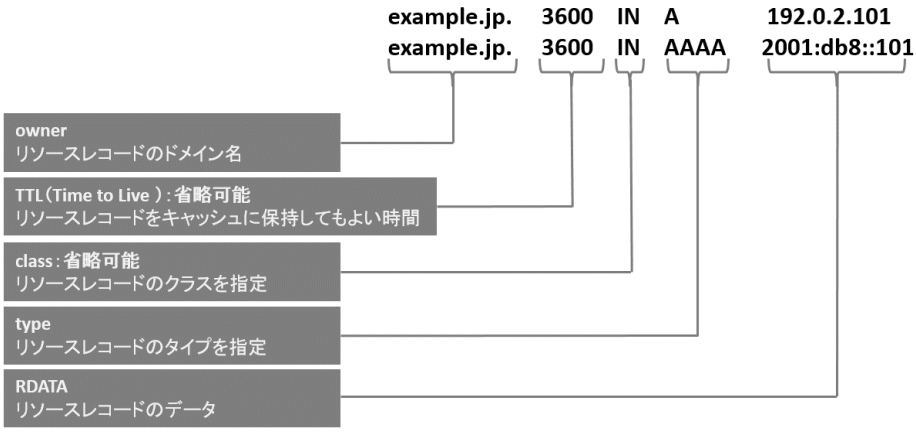


DNSでは，ドメイン名に関する情報を検索することを**名前解決**と呼んでおり，名前解決で起点となるルートゾーンを管理するサーバは**ルートサーバ**という．
ルートゾーンにはCOMドメインやJPドメインといった各TLDの情報を保持するサーバがそれぞれどのような名前で，どういったIPアドレスがつけられているかなどの情報が保存されている．



また，COMドメインの情報を保持するサーバの名前や，IPアドレスの情報が保存されている．
これらのドメインの情報を保持するDNSサーバを **権威サーバ** と呼んでいる．
名前解決は，まずルートサーバに知りたいドメイン名を問い合わせることでTLDサーバの情報を得て，次にTLDサーバに問い合わせてと，繰り返すことで，最終的に目的のデータを持つサーバを特定し問い合わせることでドメイン名の情報(リソースレコード: Resource Record)を得ることができる．

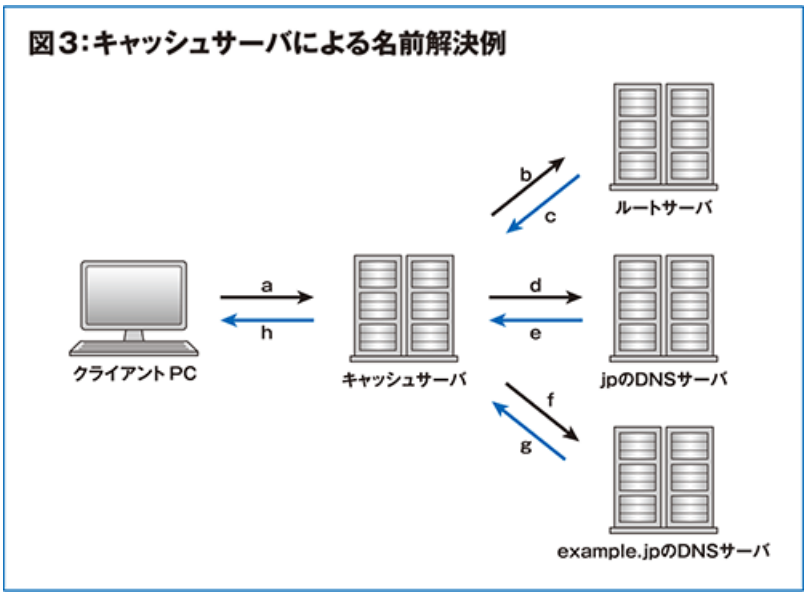
リソースレコードとは，ドメイン名に関連付けられた情報のことである．
様々な種類があり，それぞれに役割とルールが決められている．



名前解決の流れ

`www.example.jp` のIPアドレスを得たいとする.
クライアントPCが直接ルートサーバへ問い合わせることはなく，キャッシュサーバと呼ばれるDNSサーバがクライアントPCの代わりに名前解決を行う.

a. クライアントPCから予め設定されたキャッシュサーバに対して，問い合わせを行う.
b. 問い合わせを受けたキャッシュサーバは，ルートサーバへ `www.example.jp` のIPアドレスについて問い合わせる.
c. ルートサーバは，`a.dns.jp` , `b.dns.jp` , ..., `g.dns.jp` のいずれかのサーバへ問い合わせるよう回答する.
d. キャッシュサーバは，ルートサーバからの回答の中から `jp` のDNSサーバを1つ選びそのサーバへ再び問い合わせる.
e. `jp` のDNSサーバサーバは `example.jp` のDNSサーバの情報を回答する.
f. キャッシュサーバは同様にして回答の中からexample.jpのDNSサーバ(`dns.example.jp`)を1つ選び，そのサーバへ問い合わせる.
g. example.jpのDNSサーバは，キャッシュサーバにIPアドレスの情報を回答する.
h. キャッシュサーバは，クライアントPCにIPアドレスの情報を回答する.



以上の流れでIPアドレスを取得する.
しかし，検索の度に問い合わせるのでは，ルートサーバへ大量のアクセスが発生する.
そのため，クライアントPCのキャッシュやキャッシュサーバのキャッシュを用いてIPアドレスを取得する.

キャッシュと生存期間(TTL)

キャッシュとして保存されたデータはずっと保持されるわけではない.
DNSは分散データベースであるため，クライアントPCやキャッシュサーバが手元に残したキャッシュをずっと使用していると，実際の状態と不整合が起こることがある.
そのため，DNSでは，**どのくらいの期間までキャッシュとして利用してよいか** というTTLと呼ばれるパラメータがそれぞれのデータに設定されている.

TTLが短いと，ルートサーバ等に頻繁に問い合わせるため整合性は高くなるが，キャッシュネームサーバへの負荷は大きくなり，問合せ平均時間が長くなる.
TTLが長いと，問合せ平均時間が短いが一貫性が低くなる.

DNSキャッシュを用いた攻撃・脆弱性

1. キャッシュポイズニング
   DNSの問い合わせ結果は，キャッシュサーバやクライアントPCなどにキャッシュとして一時的に保持されている．

   ```
   Q．このキャッシュを何らかの方法で意図的に変更した場合，どうなるか
   A．名前解決ができないようにしたり，悪意のあるサイトへ誘導できる
   ```

   このような手法をキャッシュポイズニングという．

2. 幽霊ドメイン名 (ghost domain names)
   キャッシュの仕組みに対する脆弱性として，ghost domain namesと呼ばれるものがある．
   ある条件では，保持するキャッシュのレコードを上書きするかどうかの判定が実装依存となる．
   そのため，強制的にレコードを保持させ続けることができてしまうという脆弱性である．

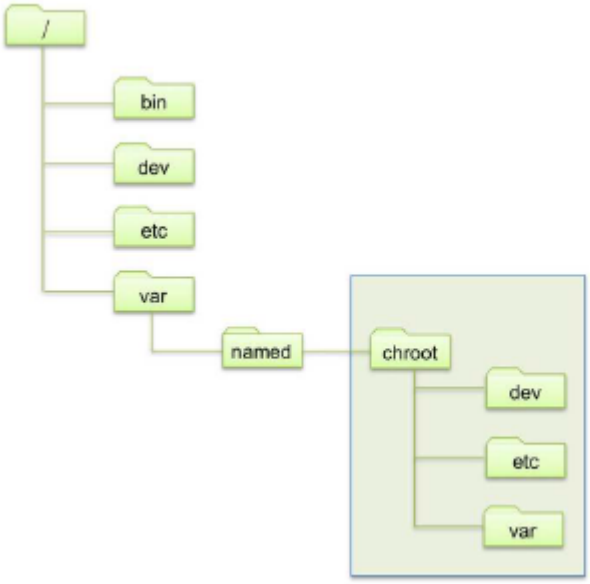
DNSサーバ

DNSサーバはDNSの名前解決の機能が実装されたサーバである．
DNSサーバは**権威DNSサーバ**と**DNSキャッシュサーバ**に分けられる

権威DNSサーバの設定

次に権威DNSサーバの設定を行う．
権威DNSサーバのソフトウェアとしてBINDをインストールして設定する．
BINDはインターネット上の多くのサーバで実行されており，セキュリティ攻撃を受けやすくなっている．
そのため，万が一BINDがセキュリティ攻撃を受けて乗っ取られてもBINDがアクセスできるディレクトリを限定することで被害を最小限に食い止める．
そのためにchroot機能を用いる．

chroot機能はプログラムに対して特定のディレクトリ以外へのアクセスを制限する機能である．
chroot機能を使ってBINDを実行するとbindプロセスは/var/named/chrootをルートディレクトリとして動作する．



BINDのインストールを行う． 必要なモジュールは`bind, bind-chroot, bind-utils`である．

```
$ sudo yum install bind bind-chroot bind-utils
```

次にゾーンの設定を行う．
ゾーンとは，1台の権威DNSサーバが管理する範囲のことであり，そのゾーンの名前解決のための情報を**ゾーン情報**と言う．
また，ゾーン情報を構成する1件1件の情報を**DNSレコード**と呼ぶ．代表的なレコードには以下のようなものがある．

レコード名	説明	例
Aレコード	ドメイン名に対応するIPアドレスが書かれたレコード	www.example.com IN A 192.168.1.1
MXレコード	メールサーバのホスト名を定義するレコード	www.example.com IN MX 10 mail.example.com
NSレコード	ネームサーバのサーバ名を定義するレコード	www.example.com IN NS 01.dnsv.jp.

まず，最初に，/etc/named.confファイルに権威DNSサーバとして動作する場合の，基本設定を行う．

```
$ sudo vi /etc/named.conf
```

`listen-on port 53 { 127.0.0.1; }`を`listen-on port 53 { 127.0.0.1; 10.0.2.15; }`に書き換える．
`allow-query { localhost; };`を`allow-query { any; };`に書き換える．
`recursion yes;`を`recursion no;`に書き換える．
ファイルの最後に以下を追加．

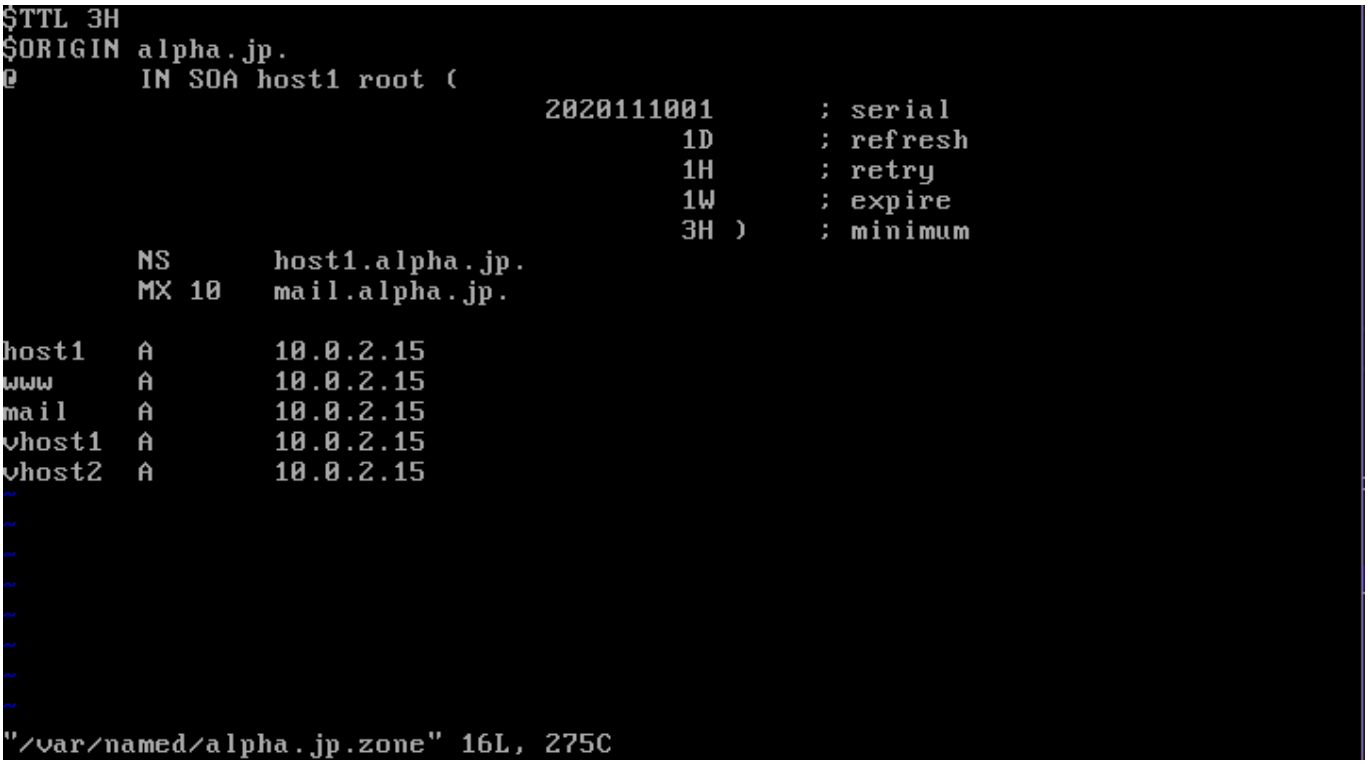
```
zone "alpha.jp" IN {
    type master;
    file "alpha.jp.zone";
    allow-update { none; };
};

zone "15.2.0.10.in-addr.arpa" {
    type master;
    file "15.2.0.10.in-addr.arpa.rev";
    allow-update { none; };
};
```

次にゾーンファイルの基になる`/var/named/named.empty`ファイルをコピーし，修正する．

```
# cp -p /var/named/named.empty /var/named/alpha.jp.zone
```

```
# vi /var/named/alpha.jp.zone
```

alpha.jp.zoneは正引き用のファイルである.



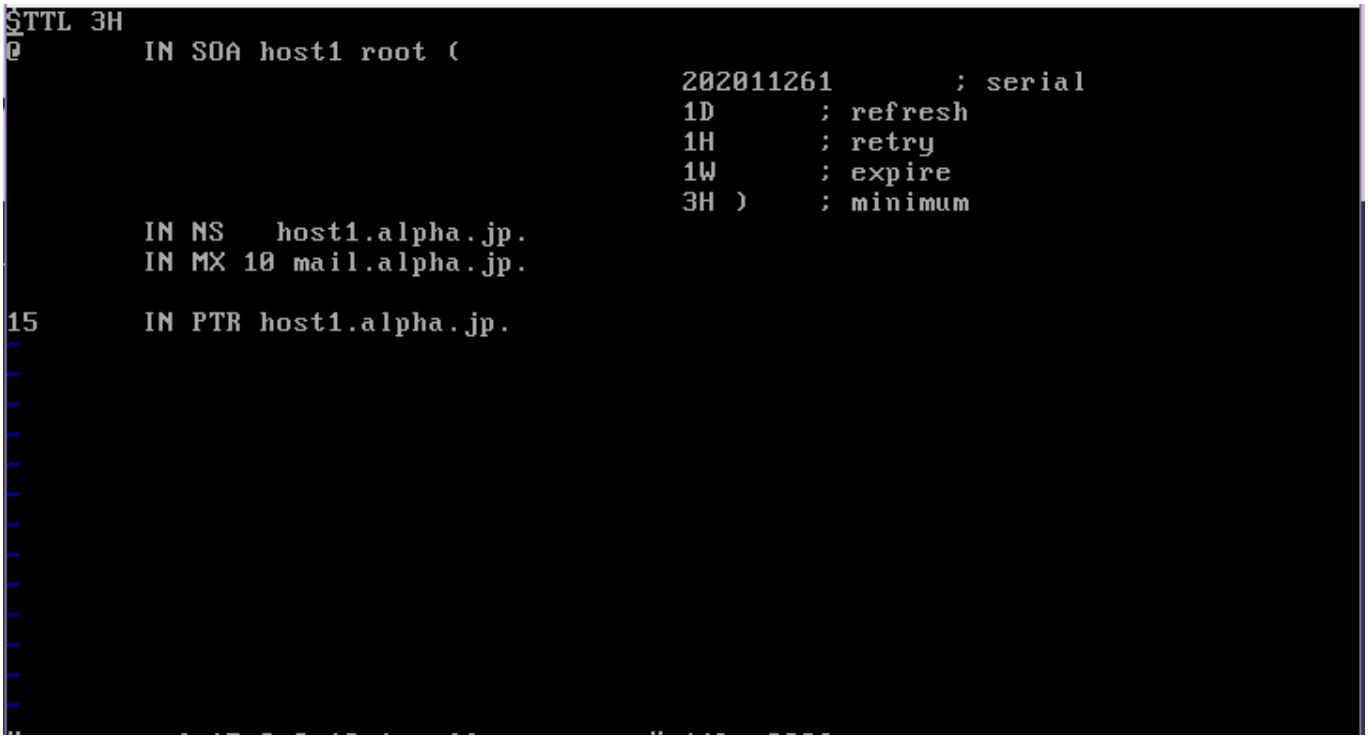
@から始まるゾーンファイルの最初のレコードはSOAレコードで，ゾーンの管理ポリシーについて設定している.
SOAレコードの先頭の@は$ORIGINで指定しているゾーン(alpha.jp)に置き換えられる.
MXレコードはメールサーバを指している．また，NSレコードやMXレコードの定義では，右側にFQDNを入れるため，最後は.で終わる.
先頭が空白になっているのは，前の行と同じものを省略しているためである．ここでは，SOAレコードの先頭である@を省略している.

下のAレコードでは，名前とIPアドレスの対応を定義している．左側がホスト名，右側がIPアドレスとなっている.

同様の手順で逆引き用のファイルを作成する.

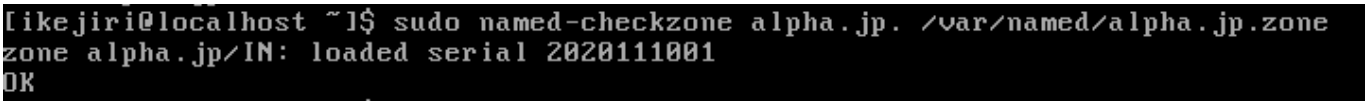
```
# cp -p /var/named/named.empty /var/named/15.2.0.10.in-addr.arpa.rev
```

```
# vi /var/named/15.2.0.10.in-addr.arpa.rev
```



ゾーンファイルの書式確認を行う.

```
$ sudo named-checkzone alpha.jp. /var/named/alpha.jp.zone
$ sudo named-checkzone 15.2.0.10.in-addr.arpa /var/named/15.2.0.10.in-
addr.arpa.rev
```



設定ファイルの書式確認を行う.

```
$ sudo named-checkconf
```

ファイアウォールの設定を行う.

```
$ sudo firewall-cmd --add-service=dns
$ sudo firewall-cmd --runtime-to-permanent
```

BINDの起動と確認

```
$ sudo systemctl start named-chroot
$ systemctl status named-chroot
```

BINDの自動起動の設定

```
$ sudo systemctl enable named-chroot
```

名前解決の確認を行う．
名前解決にはhostコマンドとdigコマンドを用いる． hostコマンドで名前からIPアドレスを確認する．hostコマンドの最初の引数は調査するアドレス，2つめの引数は調査対象サーバのアドレスである．

```
$ host host1.alpha.jp 10.0.2.15
```

```
[ikejiri@localhost ~]$ host host1.alpha.jp 10.0.2.15
Using domain server:
Name: 10.0.2.15
Address: 10.0.2.15#53
Aliases:

host1.alpha.jp has address 10.0.2.15
```

digコマンドでドメインの確認を行う．digコマンドでは，ゾーン全体の情報も確認できる．ドメイン名の後にaxfrを指定するとゾーンに登録されているすべての情報が表示される．

```
$ dig alpha.jp axfr @10.0.2.15
```

```
[ikejiri@localhost ~]$ dig alpha.jp axfr @10.0.2.15

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-16.P2.el7_8.6 <<>> alpha.jp axfr @10.0.2.15
;; global options: +cmd
alpha.jp.               10800   IN      SOA     host1.alpha.jp. root.alpha.jp. 2
020111001 86400 3600 604800 10800
alpha.jp.               10800   IN      NS      host1.alpha.jp.
alpha.jp.               10800   IN      MX      10 mail.alpha.jp.
host1.alpha.jp.         10800   IN      A       10.0.2.15
mail.alpha.jp.          10800   IN      A       10.0.2.15
vhost1.alpha.jp.        10800   IN      A       10.0.2.15
vhost2.alpha.jp.        10800   IN      A       10.0.2.15
www.alpha.jp.           10800   IN      A       10.0.2.15
alpha.jp.               10800   IN      SOA     host1.alpha.jp. root.alpha.jp. 2
020111001 86400 3600 604800 10800
;; Query time: 1 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Tue Nov 10 17:22:21 JST 2020
;; XFR size: 9 records (messages 1, bytes 253)
```

次にリゾルバの変更を行う．
リゾルバとは，ドメイン名を基にIPアドレスの情報を検索したり，IPアドレスからドメイン名の情報を検索する名前解決を行うソフトウェアのこと．
DNSサーバの設定を変更し，すべてのドメインに問い合わせることができる．
今回はminimalインストールなのでNetworkManagerを用いてDNSの設定を行う．

```
$ nmcli device
```

```
[ikejiri@localhost ~]$ nmcli device
DEVICE  TYPE      STATE      CONNECTION
enp0s3  ethernet  connected  enp0s3
```

```
$ nmcli connection modify enp0s3 ipv4.dns 10.0.2.15
$ nmcli connection modify enp0s3 ipv4.ignore-auto-dns yes
$ sudo systemctl restart NetworkManager
$ cat /etc/resolv.conf
```

```
[ikejiri@localhost ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.0.2.15
```

以上でDNSの設定が終了．hostコマンドでDNSサーバを指定しなくてもDNSの問い合わせができることを確認する．

```
$ host -t ns alpha.jp
```

```
[ikejiri@localhost ~]$ host -t ns alpha.jp
alpha.jp name server host1.alpha.jp.
```

権威DNSサーバのセキュリティに関して BINDのデフォルトでは，すべてのホストへのゾーン転送が許可されている．zoneの設定ファイルの`allow-transfer`オプションを記述することで設定を上書きできる．
そのため，`allow-transfer { localhost; 10.0.2.0/24; };`とすることで，ゾーン転送をlocalhostと10.0.2.0以下にある端末からのみ許可できる．

BINDのlogの出力・確認

今回のlogの出力はdefaultのままの出力を行う．
`/etc/named.conf`の48行目にloggingの設定が書かれている．

```
48 logging {
49         channel default_debug {
50                 file "data/named.run";
51                 severity dynamic;
52         };
53 };
```

channelはログデータの出力先(syslog，ファイル，namedの標準エラー出力，ビットバケット)を指定する．
各channelはメッセージの重要度によってデータを識別できる．
それぞれの重要度は，syslogで使う重要度のレベルと同じものである．
今回は設定していないが，`print-time`で時刻の表示，`print-category`でカテゴリーの出力を行える．

では，実際にログをみてみる．
ログはloggingに設定したとおり`/var/named/data/named.run`である．

```
# tail /var/named/data/named.run
```

このままでは，network unreachableというエラーが大量に発生する．
エラー原因としては，IPV6を使用しようとしているためである．
そのため，IPV4のみを使用するように設定を変更する必要がある．
/etc/sysconfig/namedの最後の行にOPTIONS="-4"を追加し，named-chrootを再起動する．

事後処理

本稿の演習では，CentOSで取り扱うパケットの名前解決を，DNSサーバを構築したCentOS自身にさせる．
CentOSが名前解決先として選んでいるDNSサーバのIPアドレスは， /etc/resolv.conf に記録される．

```
nameserver 10.0.2.15
```

本来であれば，CentOSはキャッシュサーバとして，他のDNSサーバに問い合わせる．
しかし，演習の都合から，権威DNSとして構築し，再帰的な問い合わせを行わない．
他の演習では，名前解決は外部のDNSサーバにさせなければならない．

そこで，以下の手順で，名前解決を外部のDNSサーバにさせるようにする．
DNSサーバのアドレスは，DHCPサーバにより機器のIPアドレスと共に，配布される．

```
$ nmcli connection modify enp0s3 ipv4.ignore-auto-dns no
# reboot (再起動)
```

大学内であれば，おそらく /etc/resolv.conf の中身は，以下のアドレスが指定される．

```
domain eng.kagawa-u.ac.jp
nameserver 192.168.7.2
nameserver 192.168.10.17
nameserver 192.168.10.18
```

以上で，通常の名前解決の経路に戻る．

また，ping コマンドや curl コマンド，設定ファイルの確認等により，どこで設定ミスが起こっているのか，調査できる．
例えば，IPレベルでネットワークが接続されているか確認するならば，以下で疎通を確認する．

```
$ ping 192.168.1.2
```

名前解決ができているか確認するならば，以下である．

```
$ ping hoge.eng.kagawa-u.ac.jp
```

上記のIPアドレスやホスト名は，学内限定であることを注意すること．
また，香川大学のFirewallでは，学外へのpingは不可能である．

参考文献

- Linuxシステム管理標準教科書 ダウンロード LinuCレベル2対応 | Linux技術者認定試験 リナック | LPI-Japan https://linuc.org/textbooks/admin/ , 2020/11/07 .
- DNS & BIND 第4版 | O'REILLY
- インターネット10分講座：DNSキャッシュ - JPNIC | https://www.nic.ad.jp/ja/newsletter/No51/0800.html , 2020/11/09 .
- CentOS 7でDNSの設定を変えるメモ - Qiita | https://qiita.com/h_tyokinuhata/items/0257ab4b94a632e393ba , 2020/11/09 .
- DNSサーバとゾーン | https://manual.iij.jp/dns/help/1480651.html , 2020/11/09 .
- @IT：DNS Tips：リゾルバとは | https://www.atmarkit.co.jp/fnetwork/dnstips/010.html , 2020/11/09 .
- JPRS用語辞典｜TLD（トップレベルドメイン） https://jprs.jp/glossary/index.php?ID=0058 , 2020/11/11 .
- JPRS用語辞典｜リソースレコード（RR） https://jprs.jp/glossary/index.php?ID=0165 , 2020/11/11
- DNSって何？DNSレコードって？イチから理解しよう！ |｜Webマーケティングの「remacre-リマケ」栃木県宇都宮市 https://remacre.jp/homepage/7478#DNS-3, 2020/11/17 .

奥付

```
名前     池尻 圭佑 ( Keisuke Ikejiri ), 竹原 一駿 ( Ichitoshi TAKEHARA )
所属     香川大学大学院 工学研究科 信頼性情報システム工学専攻 喜田研究室 M1
--------------------------------------------------------------------------------
2020/12/03 初版
2021/05/25 2版
--------------------------------------------------------------------------------
```