

# ■■ 超入門・サーバー構築勉強会 ■■

VPSリモート構築ハンズオンセミナー ～WEBサーバー立上げ、バーチャルHOSTにも挑戦～

場所 Zoomミーティング

内容

- 1) VPSからGentOS8構築および各設定確認
- 2) 超入門 OpenSSH(リモート接続環境)
- 3) 超入門 vi操作
- 4) MyDNS.JP無料ドメインの取得
- 5) オンラインマニュアルの作成
- 6) オンラインマニュアルでサーバー構築実習  
SSH接続、Apache他必須パッケージインストール  
バーチャルHOST他設定
- 7) おわりに

付録1 ネットワークの設定

付録2 DNSサーバーの役割・  
インターネットが繋がる仕組み

主催 MyDNS.JPユーザー会  
<http://usg.mydns.jp/>

後援 Future Versatile Group  
<https://www.fvg-on.net/>  
MyDNS.JP  
<http://www.mydns.jp>

## 会場準備

- ・ConoHaアカウント  
ログインできる状態でお待ちください
  - ・MyDNS.JPアカウント  
ログインできる状態でお待ちください
  - ・Zoomミーティングに参加しつつ、作業ができる環境
    1. Zoomミーティングが閲覧できるマイク&スピーカー付属デバイス  
(WEBカメラは無くてもOK)  
講師の作業状態(ターミナル画面など)をZoomミーティングでモニターしますので閲覧できるタブレットやPCなどご用意ください  
※ターミナル画面のコマンドが読める程度のモニターサイズを推奨します
- デバイスのスペックや回線帯域によっては  
セミナー内容が十分閲覧できない場合があります  
Zoomテスト環境が用意されていますので、一度確認しておいてください
2. 作業するPC  
ConoHa-VPSに対し、外部からSSHでサーバー構築していきますので、  
インターネットへ繋がられるPC(Windows10またはMacOSX)を  
ご用意ください。

### \*\*\*サーバー構築勉強会内容\*\*\*

- 1) VPSからCentOS8構築および各設定確認
- 2) 超入門 OpenSSH(リモート接続環境)
- 3) 超入門 vi操作
- 4) MyDNS.JP無料ドメインの取得
- 5) オンラインマニュアルの作成
- 6) オンラインマニュアルでサーバー構築実習  
SSH接続、Apache他必須パッケージインストール、  
バーチャルHOST他設定
- 7) おわりに

付録1 ネットワークの設定

付録2 DNSサーバーの役割・  
インターネットが繋がる仕組み

### \*\*\*注意事項\*\*\*

- ・オンラインの性格上、各自セキュリティには配慮ください  
WEBカメラやマイクを接続しますので背景や音など  
他人に知られたくないものがありましたら予め対策等をお願いします  
(うしろにシーツ等置くと、部屋の中は見えないと思います)
- ・他の人のご迷惑となる行為はご遠慮ください
- ・急用などで途中辞退は可能ですが、払戻はできませんのでご注意ください
- ・その他当方は一切責任を負いませんので、各自自己責任でお願いします

\*\*\*メモ\*\*\*

---

---

---

---

---

---

---



簡単ではありますが、確認していきます。



左メニューの「サーバー」を選択してください。  
サーバーリストの中から、VPSの名タグをクリックしてください。



では、1つずつ確認していきましょう。

上部の各ボタンが大元の操作となります。

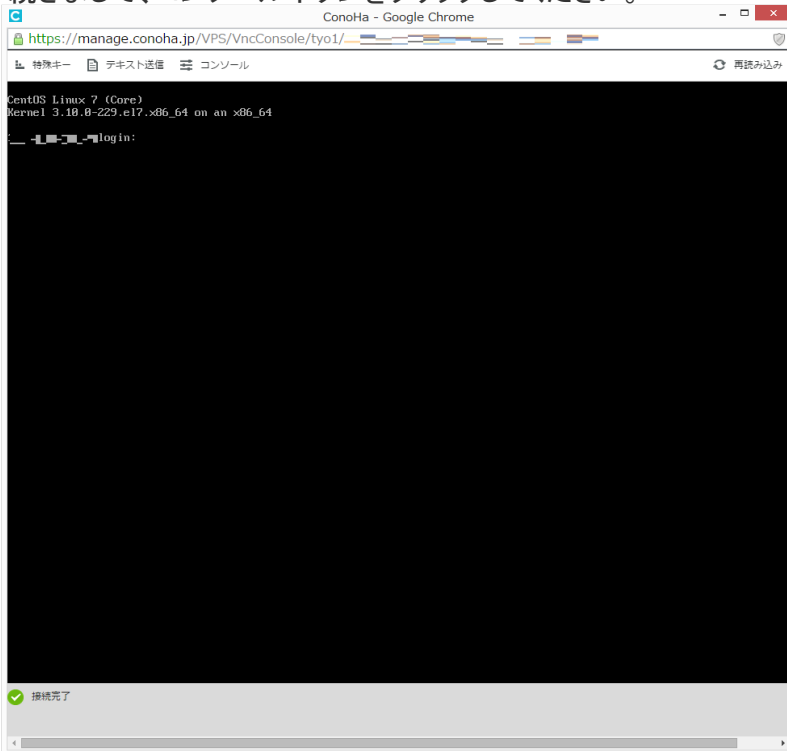
起動、停止と言った「マシン自体の操作」に相当する部分だったり、モニターに相当するコンソールボタンとなります。

その下のリソースはCPUやトラフィックなどのリソースになります。

続きましてネットワーク情報です。  
サーバー自体のIPアドレスだったり、VPS側のゲートウェイだったり、あるいは、参照するDNSサーバーのアドレスだったり、サーバーを運用する上で、重要になる部分です。

最後にVPS設定です。ここは、ざっと確認するだけで大丈夫です。

続きまして、コンソールボタンをクリックしてください。

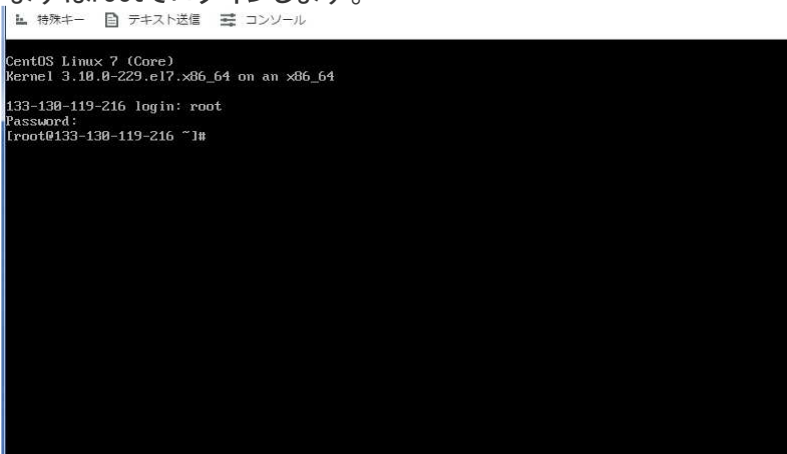


コンソール要求画面となります。  
こちらがいわゆるVNC接続(Virtual Network Computing)による  
コンソールとなります。

… login:

が確認できますでしょうか。

まずはrootでログインします。



… login: **root**

password: → 先ほど設定しましたrootパスワードを入力してください  
なお、パスワード入力はマスクされますので、表示はされません。

**[root@…]**#

このように変わったのが確認できますでしょうか。

## 2) 超入門 OpenSSH(リモート接続環境の構築)

sshとは、物理的に離れているサーバーに対し、ネットワークを介して、アクセスするプロトコルです。リモートログインと言う呼び方もされています。

リモートログインには長い間、telnetと言うプロトコルが利用されていました。しかし、telnetにはデータを素のまま(平文)で流してしまう欠点があります。

ネットワーク上を流れるデータが盗聴され、ログイン情報など非常に重要なデータさえも、第三者に知られてしまう場合があります。

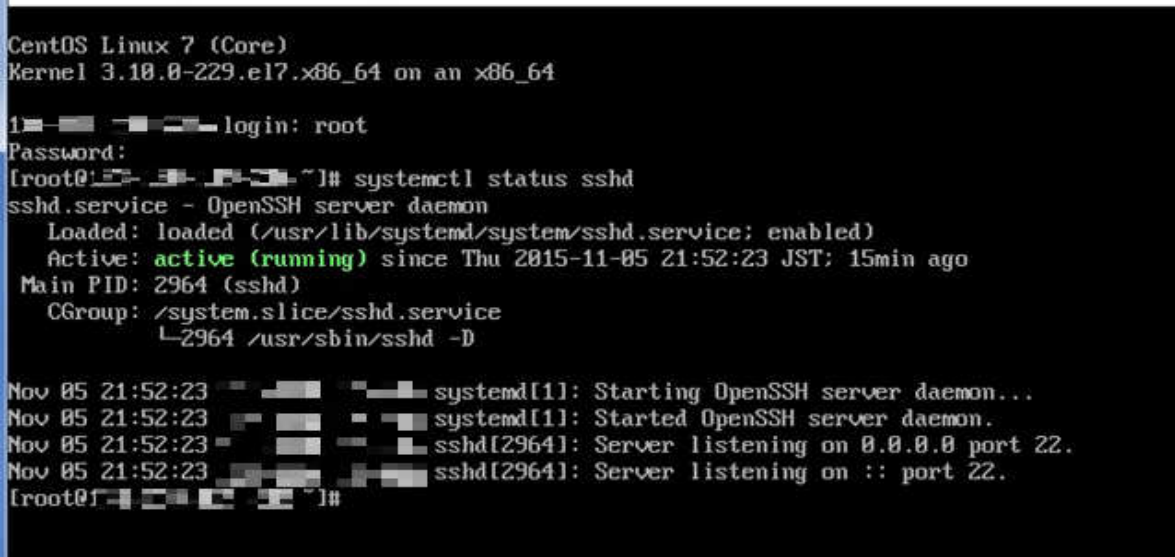
そこで、安全な運用を実現するために、データの暗号化と確実なホスト認証が必要とされ、これを実現したのが「ssh」です。

このConoHaコントロールパネルのWEB(VNC)コンソールだけによるサーバー構築も可能ですが、ブラウザによって様々なクセが出たり、画面イメージをそっくりキャプチャするため、各マシン、トラフィックなどにリソースが使われ、非常に扱いにくい場合があります。

と言う事で、TeraTerm(MacOSの方はターミナル)を使ってssh接続するのですが、その前にVPS(サーバー)側のsshd(デーモン)が起動されているか、このWEBコンソールで確認してみましょう。

まずは、下記コマンドを入力してください。sshdの状態を確認するコマンドです。

```
# systemctl status sshd
```



```
CentOS Linux 7 (Core)
Kernel 3.10.0-229.el7.x86_64 on an x86_64

login: root
Password:
[root@ ~]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Thu 2015-11-05 21:52:23 JST; 15min ago
   Main PID: 2964 (sshd)
   CGroup: /system.slice/ssh.service
           └─2964 /usr/sbin/sshd -D

Nov 05 21:52:23 systemd[1]: Starting OpenSSH server daemon...
Nov 05 21:52:23 systemd[1]: Started OpenSSH server daemon.
Nov 05 21:52:23 sshd[2964]: Server listening on 0.0.0.0 port 22.
Nov 05 21:52:23 sshd[2964]: Server listening on :: port 22.
[root@ ~]#
```

active (running) と出ていればサービスは起動しています。

また、listening on ... port 22 が確認できますでしょうか。

22番ポート(sshデフォルト)ですべてのIPアドレスから、待ち状態と言うことです。

もし、サービスが停止している場合は下記コマンドで起動しましょう。

```
# systemctl start sshd
```

サービスそのものがインストールされていない場合、  
yum -y openssh-serverなどでパッケージをインストールしましょう。

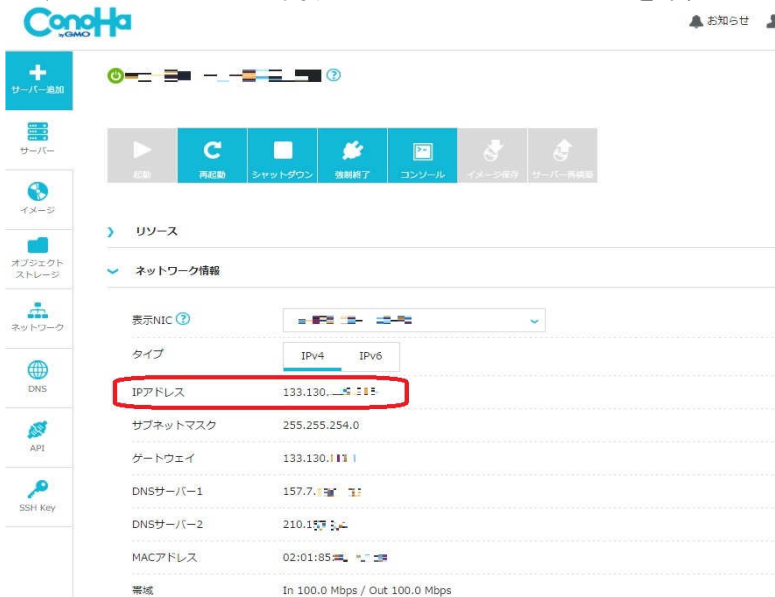
これだけ確認できれば、もうWEB(VNC)コンソールは使いません。  
下記コマンドでログアウトし、WEBコンソールは[×]で閉じてください。

```
# exit
```

※補足 なんらかの原因でssh接続が出来なくなった場合、  
このConoHaコントロールパネルから復旧してください。

では、VPSに対してssh接続していきましょう

まずはConoha-VPSで割り当てられているIPアドレスを確認してください。



IPアドレス 133.130.###.###とあります。(各自で違うアドレスが割り当てられます)

続いてターミナルを用意しましょう

**MacOSの方:**

「Finder」→「アプリケーション」→「ユーティリティ」→「ターミナル」  
 ssh (ユーザー名)@(サーバーIPv4アドレスまたはホスト名)  
 具体的には root@IPv4アドレス と入力します

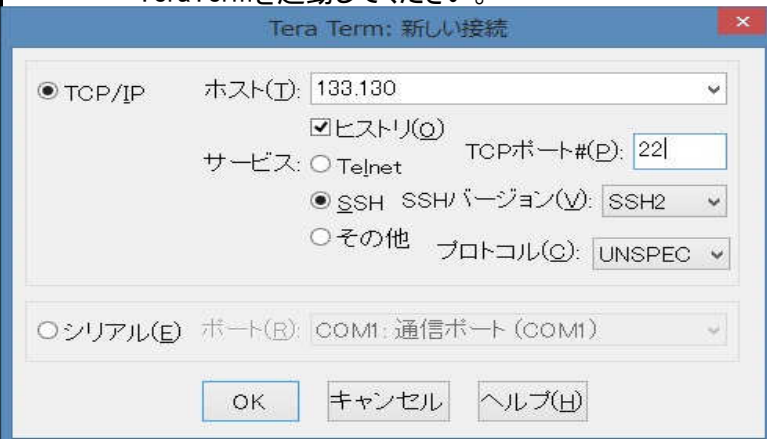
または ssh (サーバーIPv4アドレス) -l (一般ユーザーID)

**Linuxクライアントの方:**

コンソール(端末)からコマンドでSSH接続します  
 ssh (ユーザー名)@(サーバーIPv4アドレスまたはホスト名)  
 具体的には root@IPv4アドレス と入力します

**Windowsの方:** TeraTermを起動してください。

TeraTermを起動してください。

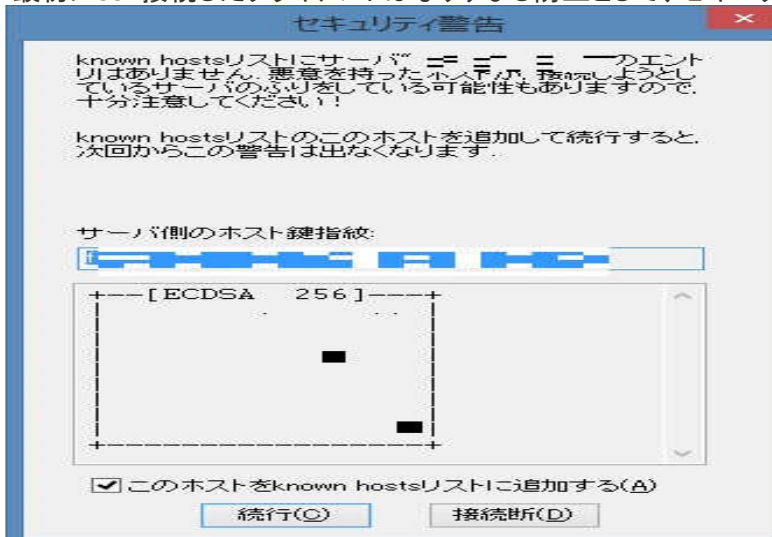


「新しい接続」が出ましたら  
 ホスト部分にそれぞれ割り当てられた  
 IPアドレスを入力してください

TCPポートは22番です

OKをクリックしてください

最初にssh接続したクライアントはなりすまし防止として、セキュリティ警告が出ます。



これはホスト認証と呼ばれるもので、クライアントがサーバーの正当性を確認しています。

sshで接続するたびにクライアント、サーバー間で公開鍵が一致しているか確認しています。

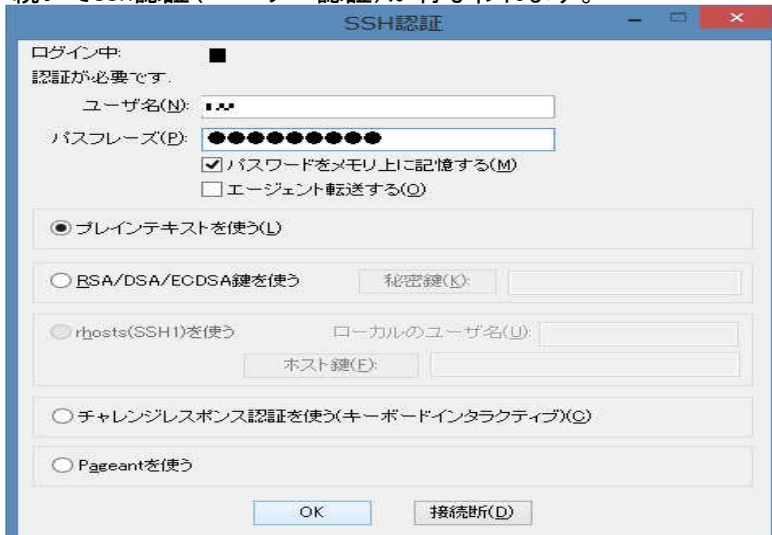
1度ホスト認証されるとクライアント側に公開鍵が登録されるので次回以降はセキュリティ警告は出ません。

最初にssh接続した場合、公開鍵が登録されていないのでこのような警告がでます。

続行をクリックしてください。

※MacOS(Linux)ターミナルでも、同様に、確認メッセージがでます。YESで返してあげれば、鍵が登録され次回以降メッセージは出ません。

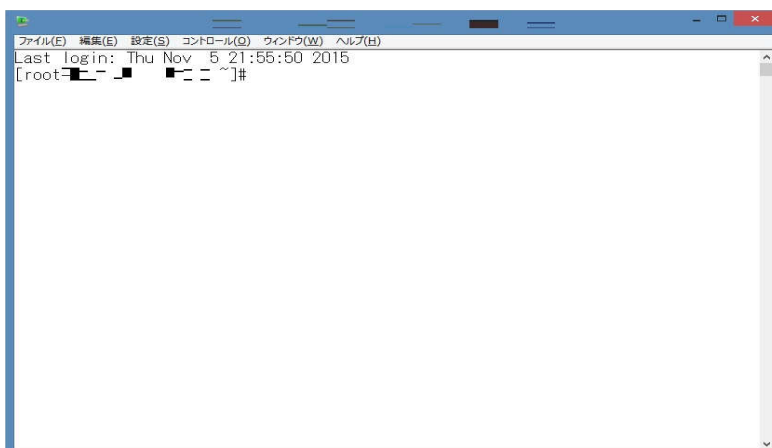
続いてssh認証(ユーザー認証)が行なわれます。



現在ログインできる有効なユーザーはrootユーザーなので、rootでログインしてください

ユーザー名: root  
 パスフレーズ: root/パスワード  
 ※ConoHaで設定したパスワードです

OKをクリックしてください



[root@---- ~]#  
 [現在のユーザー名@ホスト名 場所]#

現在のユーザー :ユーザー名  
 ホスト名 :ログインしているサーバー  
 場所 :カーレントディレクトリ  
 ~ :ユーザーホームディレクトリ  
 # :root  
 \$ :一般ユーザー

ログインができましたでしょうか。



### 3) 超入門 viエディタ

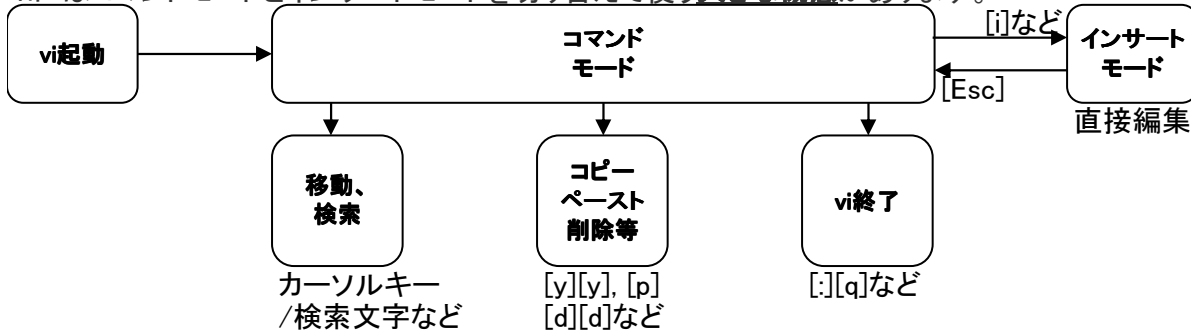
viとはVisual Editorの略です。

Linux標準のテキストファイルを編集する「エディタ」で、これを使って各種ファイルを設定していきます。

エディタには他にもEmacsやNanoなどがありますが、viはLinux(に限らずUNIX系OS)であれば、必ず実装されていますので、これをまず使えるようにした方が良いでしょう。

ここでは、vi機能のうちファイルの開閉やテキスト検索、ページ移動や編集方法など基本的な使い方について練習していきましょう。

viにはコマンドモードとインサートモードを切り替えて使う**大きな概念**があります。



viを起動すると最初は「**コマンドモード**」になります。検索や削除、そしてコピー・ペースト、終了と言った操作を実行するモードです。まだこの状態では、文字入力(編集)ができません。

内容を編集するには検索コマンドやカーソルキーなどで変更箇所まで移動しインサートモードへ切替えてから編集します。

編集が終わったら[Esc]で「**コマンドモード**」へ戻ります。全ての編集が終わったら終了コマンドを実行してviを終了します。

このように「**モード切替え**」と言う概念を把握しましょう。

では、TeraTerm(ターミナル)を使って、実際のファイルをviで開いて動作を確認してみましょう。

#### vi起動

```
# vi /var/log/messages      システムに関する一般的なメッセージが格納される場所
```

#### ・コマンドモード

##### 移動

カーソルキーでカーソルの動きを確認しましょう。

大きなファイルの場合、ページ単位で移動する方が良いでしょう。

次ページ [Ctrl]+[f]

戻る [Ctrl]+[b]

行頭や行末への移動はカーソルキーより下記コマンドの方が良いでしょう。

行頭 [^]

行末 [ \$ ]

##### 検索

検索するのは[/]検索キー(一部でも可、大文字小文字区別あり)となります。実際に検索して、その動きを確認しましょう。

[/][s][s][h] と入力してください。"ssh"を含む文字がカーソル位置より下方向へ検索されます。

[n] と入力してください。順次、下方向へ検索していきます。

##### vi終了

[Esc][:][q] と入力してください。

・**インサートモード**

こちらでファイルを作りながら、どのように動くか見てみましょう。  
 まずviでtest.txtと言うファイルを作ってください。

```
# vi test.txt
```

そのファイルがある場合はファイルを開きますが、無い場合はファイルを作ってくれます。

例文を入力するため、**[i]** を押してインサートモードにし、下記のように入力してください。

```
Apple
Oreng
```

保存してみましょう	[Esc]	[:]	[w]	実行
終了してみましょう	[Esc]	[:]	[q]	実行

もう1度 viでtest.txtを開いて、保存されているか確認してみてください。  
 さらに編集してみましょう。"Oreng"の下に"Banana"を入力します。

上下キーでOreng行まで行ったら、**[o]**を押します。

該当行の下にカーソルが行きインサートモードになるので"Banana"を入力します。

そのまま終了してみましょう	[Esc]	[:]	[q]	実行
---------------	-------	-----	-----	----

→ 保存せずに終了しようとするとワーニングが出ます

強制終了してみましょう	[Esc]	[:]	[q]	[!]	実行 ※[q]の後ろに[!]を付ける
-------------	-------	-----	-----	-----	--------------------

```
余談...
保存終了を一度にするには [Esc] [:] [w] [q] 実行
また、コマンドモードが分かっているなら[Esc]は不要です
```

・**コピー、ペースト、削除等コマンド**

では、もう一度viで開いてみましょう。

```
# vi test.txt
```

さきほど入力した文字が確認できるかと思えます。  
 カーソルキーでOreng行まで移動しましょう。

```
コピー
カーソル上で[y][y]と押すとその行をバッファします。
Oreng行で [y][y] と押してみましょう。
コピーは行範囲が可能です
カーソル行を含め3行コピーしたい場合は [3][y][y]
30行コピーは [30][y][y] となります。
```

```
ペースト
カーソル行で[p]と押すとバッファをカーソル位置の次の行へ貼付けます。
[y][y]した行で[p]と押してみましょう。
```

```
削除
カーソル上で [d][d] と押すと、その行を削除してくれます。
同時にバッファされるので、[p]との組み合わせで、カット&ペーストとしても使えます。

削除も行範囲が可能です、
カーソル行を含め3行削除は [3][d][d]
30行削除は [3][0][d][d] となります。

また、1文字だけ削除したい場合、[x] を押すとカーソル部分の文字だけが削除されます。
コメントアウト記号(#など)のみを削除したい場合などは便利だと思います。
```

```
編集の取消し
実行した作業を戻したい場合、[u] を押します。
キャッシュされてる部分まで戻る事が可能ですので、誤って削除した場合などはこれで戻せます。
```

各種の動作確認が終わりましたら、viを[Esc][w][q]などで終了してください。



コマンド	概要
<b>&lt;挿入コマンド&gt;</b>	
i	カーソル前に挿入
o	下に1行加える
<b>&lt;移動コマンド&gt;</b>	
k	1行上へ移動
j	1行下へ移動
h	1文字左へ移動
l	1文字右へ移動
0 (ゼロ)	行先端へ移動
\$	行末へ移動
:10	10行目へ移動 :0なら開始行へ移動
G ([Shift] + [g])	最終行へ移動
<b>&lt;削除コマンド&gt;</b>	
x	カーソル文字を削除
dd	カーソル行を削除 2ddで2行削除
<b>&lt;修正コマンド&gt;</b>	
r文字	カーソル位置の文字を置き換える
u	直前コマンド取り消し(Undo)
yy	1行コピー 2yyで2行コピー
p	yy(dd)した行を貼り付け
. (小数点)	直前入力を繰り返す
J	下の行を連結する
<b>&lt;ファイル操作と終了&gt;</b>	
:w	現在の内容を上書き
:w file	現在の内容をファイル名fileとして書込み
:r file	現在の内容をfileに追加書込み
:wq	保存して終了
:q!	保存せず終了
:e!	編集内容を破棄して最初から

オプション名	機能
autoindent	自動段付け
autowrite	自動でファイル保存
all	現在設定されているオプションを表示
background	背景色に応じた表示色設定 darkまたはlight
hlserch	検索結果の反転表示
ignorecase	検索時、大文字小文字の区別をしない
number	行番号表示
tabstop	[TAB]キーで移動する大きさ(文字列 省略は8)
wrapscan	検索中、最後まで達した時、最初に戻って再検索
※コマンドモードで :set オプション名で実行 :set noオプション名で無効	
例1 :set number	
例2 :set nonumber	

## 5) MyDNS.JP無料ドメインの取得

ドメインでwebアクセスできるよう  
MyDNS.JPで無料で使えるサブドメインを取得します。

事前にMyDNS.JPアカウント登録をお願いしていましたが、皆さん準備はできてますでしょうか。

では、MyDNS.JPへアクセスしログインをしてください。

<http://www.mydns.jp>

やり方はこちらのページの真ん中くらいに

[https://blog.mydns.bz/?page\\_id=366](https://blog.mydns.bz/?page_id=366)

MyDNS.JPへのドメインの設定(DOMAIN INFO)  
がありますので、そこをみてください。

左側がメニューに変わりますので、その中の「DOMAIN INFO」を押してサブドメインを取得します。  
MyDNS.JPでは、十数個の無料で使えるサブドメインがありますので、  
「DOMAIN」のところに例えば「beginner.mydns.jp」とか、「beginner.0am.jp」というように、  
任意のサブドメインをFQDNで最後まで入力します。

さらに「HOSTNAME」のところに「\*」(アスタリスク)を1文字だけ入力してください。

これは、上記の例でいうと「www.beginner.0am.jp」とか「mail.beginner.0am.jp」とか  
「ftp.beginner.0am.jp」といった、いろいろなホスト名について問い合わせると、  
すべて同じIPアドレスを答える「ワイルドカード」という設定になります。

「DOMAIN」と「HOSTNAME」を入力したら、下の「CHECK」ボタンを押してみてください。

そのドメインが取得できるなら、「OK」ボタンが押せるようになっていますので押します。

取得できない場合(他の誰かがすでに使っている場合や使えないドメインなど)には、  
別のドメイン名で再度CHECKしてみてください。

取得できた旨が表示されたら、もう一度「DOMAIN INFO」を押して  
サブドメインが取得&設定できていることを確認してください。

\*\*\*コメント\*\*\*

サーバーを運用するにあたってドメイン取得は必要となってきますが、  
ここでは構築実習の意味もあり、無料で使えるMyDNS.JPサブドメインを使っています。

ご存じのとおり、MyDNS.JPはDDNSを提供するサービスです。  
そのため、固定IPでも定期的なIP通知が必要となります。

その方法はオンラインマニュアルで実習していきますが、  
通常はドメインを取得した時に、ドメイン登録代行業者などが展開しているDNSサーバーに  
取得ドメインとサーバーIPアドレスを登録することによって  
ドメインでサーバーにアクセスできるようになります。

## 6) オンラインマニュアルの作成

では、これからサーバー構築の実習をするためのオンラインマニュアルを作成します。

皆さんごとにサーバーアドレスやドメインなどが違うため、共通のマニュアルですと、読み替える部分が頻繁にあり、サーバー構築がスムーズに行きませんのでここで、皆さん専用のオンラインマニュアルを作成します。

### 作成と言っても超簡単！

MyDNS.JPアカウント情報と  
今取得したドメインと  
ConoHa-VPSサーバーアドレス、  
そしてサーバーの名前であるホスト名を決めてもらい入力するだけです。

下記URLへアクセスしてください

[https://usg.mydns.jp/centos8\\_online/](https://usg.mydns.jp/centos8_online/)

オンラインマニュアルが作成できましたら、当テキストはもう使用しません。  
「オンラインマニュアルでサーバー構築実習！！」へ進みましょう。

巻末にネットワーク設定などの付録をご用意しましたので  
興味ある方はご参照ください。

## 付録1 ネットワークの設定

CentOS7ではコマンドでのネットワーク設定が推奨されております。

基本は設定ファイルをvi編集なので、本当のところを知りたいと言う方は是非、**「ネットワーク設定入門 vi編集編」**をご覧ください。

<http://usg.mydns.jp/?p=259>

コマンド実行でもvi編集でも基本的な流れは同じです。

- 1) サービス起動状況の確認
- 2) 現在のネットワーク状態を確認
- 3) ネットワーク設定方法
- 4) ネットワーク再起動

VPSベンダーから割り当てられたネット情報をメモしといてください。  
(IPv6は割愛します)

<例>

サーバー.IPv4 222.222.222.222 /xx  
サブネットマスク.IPv4 〇〇〇.〇〇〇.〇〇〇.×××  
ゲートウェイ.IPv4 222.222.222.1  
DNSサーバー1.IPv4 222.222.222.100  
DNSサーバー2.IPv4 222.222.222.200

では、サーバーの設定を確認しましょう。

### 1) サービス起動状況の確認

使うサービスはNetworkManagerなので、NetworkManagerの状態を確認します。

```
# systemctl status NetworkManager
```

active(running) と出ていれば動作しています。

動作してなければ、NetworkManagerを起動させます。

```
# systemctl start NetworkManager
```

## 2)現在のネットワーク状態を確認

ネットワークの状態を確認するコマンドは以下の通りです。

```
# nmcli d
```

DEVICE	TYPE	STATE	CONNECTION
eth0	ethernet	connented	eth0
lo	loopback	unmanaged	—

デバイスにeth0があることがわかります。

では、このデバイスeth0がどんな状態か確認してみましょう。

```
# nmcli d show eth0
```

GENERAL	.デバイス		:eth0
GENERAL	.タイプ		:Ethernet
GENERAL	.HWADDR		:FF:FF:xx:xx:xx:xx
GENERAL	.MTU		:1500
GENERAL	.状態		:100 (接続済み)
GENERAL	.接続		:eth0
GENERAL	.CONパス	:/org/freedesktop/NetworkManager/ActiveConnecti	
WIRED-	.キャリア		:オン
IP4	.アドレス[1]	:ip = 222.222.222.222 /xx, gw = 222.222.222.1	
IP4	.DNS[1]		:222.222.222.100
IP4	.DNS[2]		:222.222.222.200
IP6	.アドレス[1]	:ip = xx::f:xx:xx:xx/xx, gw = xx::xx:1	

続いて以下のコマンドでも確認してみましょう。

```
# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    qdisc pfifo_fast state UP qlen 1000
    link/ether 02:01:a3:2c:ad:d8 brd ff:ff:ff:ff:ff:ff
    inet ###.###.###.###/23 brd ###.###.###.### scope global dynamic eth0
        valid_lft 71881sec preferred_lft 71881sec
```

サーバーから外部へpingが通るか確認してみましょう。

```
# ping 8.8.8.8
```

※終わる時は[Ctrl] + [c]

また外部からサーバーへpingが通るか確認してみましょう。

Windowsのコマンドプロンプトから  
ping サーバーIPv4アドレス

これで、問題なくネットワークがつながっているのが確認できます。



### 3) ネットワーク設定方法

設定が必要なサーバーだったり、変更などをする場合は以下の項目をご参照ください。

NICに対してネットワーク設定

#### ●NICのアクティブ化

# nmcli d でeth0:CONNECTIONが“-”だった場合

```
# nmcli conention up eth0
```

#### ●サーバーIPv4アドレスとゲートウェイIPv4アドレスの設定

```
# nmcli c modify eth0 ipv4.addresses "サーバーIPv4アドレス" "ゲートウェイIPv4アドレス"
```

#### ●DNSの設定

```
# nmcli c modify eth0 ipv4.dns "DNSサーバー1IPv4アドレス" "DNSサーバー2IPv4アドレス"
```

#### ●固定IPの設定

```
# nmcli c modify eth0 ipv4.method manual
```

※DHCPなら“auto”になります

#### ●ホスト名設定

ホスト名をtestに変える場合

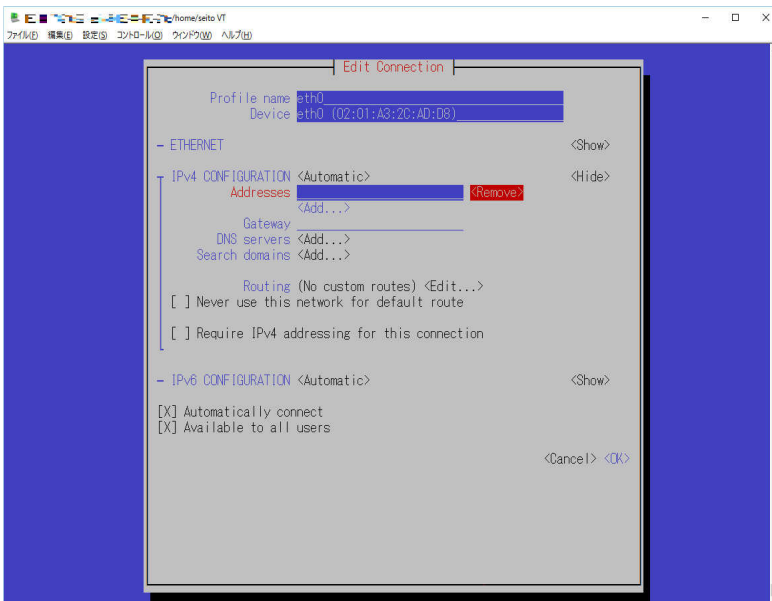
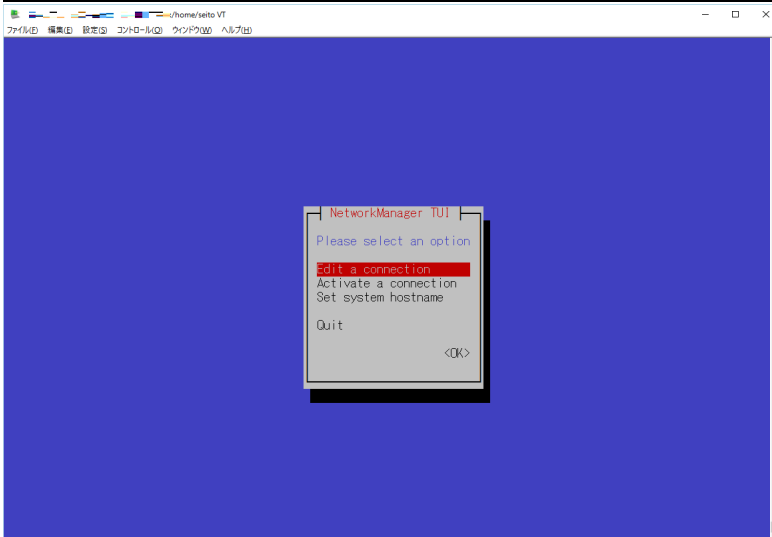
```
# hostnamectl set-hostname test
```

#### ●サーバー起動時に自動でeth0をActiveにするには

```
# nmcli con mod eth0 conention.autoconnect "yes"
```

これらコマンドを憶えるが大変！という方には下記ツールを使ってみてください。

```
# nmtui
```



このようにツールによる設定も可能です。

#### 4) ネットワーク再起動

各設定が終わりましたら、ネットワークを再起動して設定を有効にしましょう。

```
# systemctl restart NetworkManager
```

コマンドによって設定がファイルがどのように記述されたか見てみるのも良いでしょう。

主にIPアドレスを設定するファイル

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

主にネットワーク機能を設定するファイル

```
vi /etc/sysconfig/network
```

主にDNSサーバーアドレスを設定するファイル

```
vi /etc/resolv.conf
```

主にホストネームを設定するファイル

```
vi /etc/hostname
```

実はこれら4つのファイルでの設定方法を憶えてしまえば、ネットワークの設定は完了してしまいます。

新しいディストリビューションでは様々なパッケージ(NetworkManagerなど)で、これらファイルを便利に編集してくれますが、実はそんな新しいコマンドなど知らなくても、この基本的な4つのファイルをエディタ(viなど)で編集すればオッケーなんです。

サーバーによってはCentOS5や6と言った古いバージョンで稼働しているのも少なくありません。

こう言った古いサーバーをメンテナンスする際に、新しいコマンドなどは用意されていない場合があるので、基本的な設定方法が必要とされます。

知っていて損はないかと思えますよ。

ただし、バージョンや構築者のポリシーで、必ずしもご紹介した設定ファイル通りには記述されてない場合があります。(その際は適時読み替えが必要になりますのでご注意ください)

基本的なviでの設定方法が知りたい場合は簡単ではありますが、「ネットワーク設定入門 vi編集編」(<http://usg.mydns.jp/?p=259>)に記載しましたので、興味のある方は参考になさってください。

## 付録2 DNSサーバーの役割・インターネットが繋がる仕組み

サーバーを構築する上で、どうしても知って置きたい基本中の基本がDNSサーバーの役割。

ここでは、PC同士が通信しあう仕組みについて、DNSサーバーの役割を確認しながら、インターネットが繋がる仕組みについて勉強しましょう。

NSレコードって何でしょう？  
ドメインとIPアドレスの関係って？  
解ってるようで本当のところはどうなのかな？

などなど疑問があると、サーバー構築がぼや～っとしちやいます。

ここでは、インターネットが繋がる仕組みを理解しながら、DNSサーバーの役割について、おさらいしていきたいと思います。

そもそも、インターネットが繋がる仕組みとは本来、コンピューター同士の通信はIPアドレスで行います。

IPv4(xxx.xxx.xxx.xxx)では32桁の2進数

2進数の8桁の集まりを"."で区切り、それを4組でIPv4としています。

2進数の8桁 は"00000000" ~ "11111111"  
コンピューターの言葉である2進数(on/off)は人間にとって非常に理解しにくいので、これを10進数に変換し "0" ~ "255" としてます。  
そのため 192.168.1.1 などと言う表記になります。

2進数の 8桁 × 4 → 2<sup>32</sup> → 約43億  
つまり、IPアドレスで通信を行おうとするコンピューターは、約43億通りあることが解ります。

これをいちいち数字の羅列で特定のコンピューターへ通信するのは大変だ！  
と言うことで、これを人間が理解しやすい文字列に変換するようになりました。

例えば、yahooのwebページを開くとき、  
**http://www.yahoo.co.jp** と入力しますが  
本来のやり方 =**ipアドレス** で開いてみましょう。

Linuxコンソールから

```
# dig www.yahoo.co.jp
```

```
:: ANSWER SECTION:
```

```
www.yahoo.co.jp. 895 IN CNAME www.g.yahoo.co.jp.
```

```
www.g.yahoo.co.jp. 55 IN A 124.83.203.233
```

DNSサーバーに問い合わせをして

www.yahoo.co.jpのアドレスが返ってきました

これをブラウザで**http://124.83.203.233/**とするより

**http://www.yahoo.co.jp/**とした方が便利です。

このようにホスト名(ドメイン)をIPアドレスに変換(名前解決)する役目がDNSサーバーとなります。

DNSサーバーはクライアントから名前解決の問い合わせを受けるとそれに対応するIPアドレスを返答します。

インターネットがあまりに巨大なネットワークとなった現在、  
1台のDNSサーバーでは全ての名前解決するレコードを持つのは不可能となりました。

そのため、複数のDNSサーバーを使って名前解決を行っています。

各々のDNSサーバーには受け持つ範囲が決められていて、この範囲を「ゾーン」と言います。  
DNSサーバーは階層構造で連携しています。

クライアントPCから "www.yahoo.co.jp" でDNSサーバーへ問い合わせします。

例えば、あるDNSサーバーは「jp」のドメインを受け持っています。  
これを 仮に DNS1としましょう。

また別のDNSサーバーは「co.jp」のドメインを受け持つとします。  
これを 仮に DNS2とします。

さらに「yahoo.co.jp」のドメインを受け持つDNSサーバーを  
仮に DNS3とします。

さて、ここでクライアントPCが例えば DNS4と言うDNSサーバーへ  
「www.yahoo.co.jp」の問い合わせをします。

しかし、DNS4は「www.yahoo.co.jp」のIPアドレスを持っていません。

そこでDNS4はyahoo.co.jpのDNSサーバーであるDNS3へ問い合わせしたいのですが、  
あいにくDNS3のIPアドレスも持っていません。

そのため、DNS4はIPアドレスを持っていないドメインの問い合わせがあると、  
「ルートサーバー」(DNSサーバー)に対し、  
「jp」などを管理するDNSサーバーのIPアドレスを問い合わせます。

するとルートサーバーから「jp」を管理するDNSサーバー (= DNS1)のIPアドレスが返ってきます。

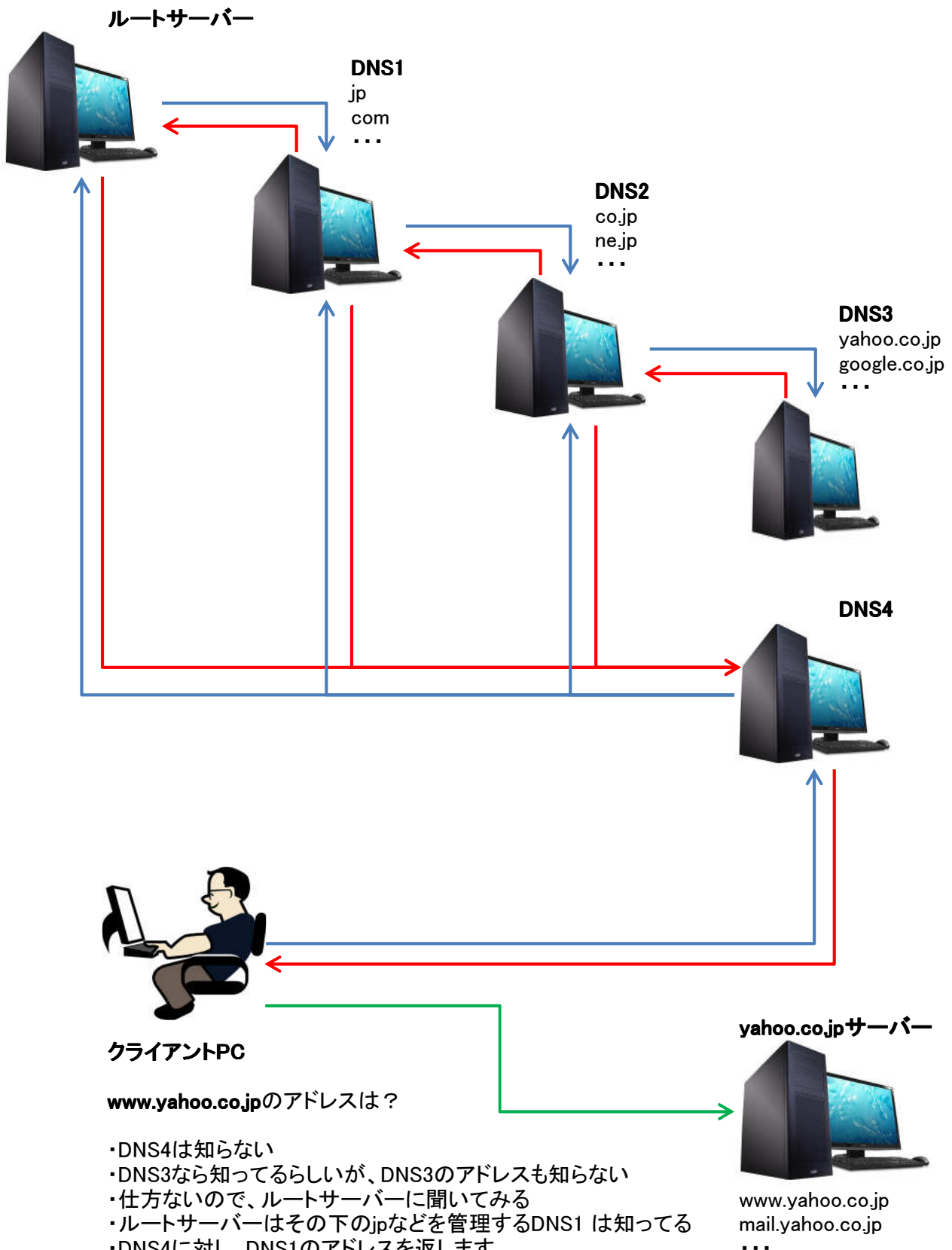
次にDNS4はDNS1のアドレスがわかったので、  
そのサーバーに対し「co.jp」を受け持つDNS2のIPアドレスを問い合わせます。

するとDNS2のIPアドレスが判明するので、  
さらにDNS4はDNS2に、DNS3のIPアドレスを問い合わせます。

そして判明したDNS3のIPアドレスを使って、「yahoo.co.jp」のIPアドレスを問い合わせます。

DNS4はこうして判明したIPアドレスをクライアントPCに返します。

このような仕組みで各DNSサーバーは連携しながら名前解決を行っています。



- ・DNS4は知らない
- ・DNS3なら知ってるらしいが、DNS3のアドレスも知らない
- ・仕方ないので、ルートサーバーに聞いてみる
- ・ルートサーバーはその下のjpなどを管理するDNS1は知ってる
- ・DNS4に対し、DNS1のアドレスを返します
- ・DNS4はDNS1アドレスが判明したので、DNS1に対し、「co.jp」を受け持つDNS2アドレスを問い合わせる
- ・DNS4はDNS2アドレスが判明したので、DNS2に対し、「yahoo.co.jp」を受け持つDNS3アドレスを問い合わせる
- ・DNS4はこうして判明したIPアドレスをクライアントに返します
- ・クライアントPCはyahoo.co.jpサーバーのアドレスが判明したので、アクセスする

DNSサーバーは名前解決専用のデータベースとも言えます。

個々ホストのIPアドレスや下部ドメインのDNSサーバーIPアドレスなど、様々な情報を管理しています。

これら情報はレコードと呼ばれており、DNSサーバーには様々なレコードが存在します。この様々なレコードの事を リソースレコードタイプ(RRタイプ)と呼んでます。

#### RRタイプ

NS	Name Server	ドメインを管理するサーバーを指定	
MX	Mail eXchanger	ドメイン内のメールサーバーを指定	
A	Address	ドメイン内ホストのIPv4アドレスを記述	正引き参照
AAAA	Quad A(ddress)	ドメイン内ホストのIPv6アドレスを記述	正引き参照 (クアッドAレコード)
CNAME	Canonical Name	あるホストの持つ正規名を表す	1つのホストに複数FQDN割当可能
PTR	PoinTeR	ドメイン内ホストのFQDNを記述	逆引きはPTRレコード参照

と、ここまで名前からIPアドレスの引き方を説明してきました。これを通常 正引きと言います。

逆に IPアドレスから FQDNを引くのを逆引きと呼んでます。

実際にドメインを取得しても、DNSサーバーに対してIPアドレス情報を入力しないと、そのドメインからはアクセスできません。

このドメインはこのIPアドレスですよ という事を適切なRRタイプと共にDNSサーバーに登録しなければなりません。

また、アドレスを入力しても、反映するのに少し時間がかかります。

DNSサーバーは一度アクセスが成功するとその情報をキャッシュし、次にアクセスが来た時に、キャッシュレコードを使って返答するようになります。

最初、時間がかかっていたアクセスが、次第にすぐにアクセス出来るようになるのはこの仕組みのおかげです。

ここまで理解できるとDNSサーバーの設定もそんなに難しくは ないと思います。

またDNSサーバーが理解できればサーバー構築もイメージ強く出来るものですよ。

■■ 超入門・サーバー構築勉強会 ■■

著作者・発行者

MyDNS.JPユーザー会  
<http://usg.mydns.jp>  
master@usg.mydns.jp

2015年5月初版  
2018年10月第7版

