LinuC

Linuxサーバー構築標準教科書

（Ver.3.0.2）

LPI-JAPAN

目次

まえがき

このたび、特定非営利活動法人エルピーアイジャパンは、Linux 技術者教育に利用していただくことを目的とした教材、「Linux サーバー構築標準教科書」を開発し、インターネット上にて公開し、提供することとなりました。

この「Linux サーバー構築標準教科書」は、多くの教育機関から、Linux によるサーバーの構築を「基礎」から学習するための教材や学習環境の整備に対するご要望があり、開発したものです。

公開にあたっては、「Linux サーバー構築標準教科書」に添付されたライセンス（クリエイティブ・コモンズ・ライセンス）の下に公開されています。

本教科書は、最新の技術動向に対応するため、随時アップデートを行っていきます。

また、テキスト作成やアップデートについては、意見交換のメーリングリストで、誰でもオープンに参加できます。詳しくは Linux サーバー構築標準教科書のホームページをご覧ください。

Linux サーバー構築標準教科書

```
https://linuc.org/textbooks/server/
```

執筆者・制作者紹介

岡田 賢治（バージョン 1 執筆担当）
UNIX/Linux を初めて触ってから 15 年、ユーザー、管理者そして育成に携わってきました。本テキストは、実際に用いた説明方法などを使い、今までのノウハウを集約して執筆したつもりです。認定試験を通しての Linux 技術者の発展と、育成に関わる先生方のお役に立てれば幸いです。

川井 義治（バージョン 1 執筆担当）
Linux の講義をするとき、多肢に渡る知識と解説が必要で、前後の内容が複雑に絡み合うため、教材選定に苦労したり、自作教材を使っていました。多くの内容の中から実践として使える項目を選び、使いやすい並びを目指して書きました。学校教材の選択肢の一つや、個人教材として使って頂ければ幸いです。

宮原 徹（バージョン 2 執筆/バージョン 3 監修担当）
本教科書は、Linux/オープンソースソフトウェアをこれから勉強する皆さんと、熱心に指導に当たられている先生方の一助になればと思い、作成いたしました。バージョン 3 の改訂にあたっては、新しいディストリビューションへの対応と合わせて、冗長だった部分を減らして学習しやすい構成にしてあります。

遠山 洋平（バージョン 2 校正・図版作成担当）
Linux サーバー構築標準教科書のバージョン 1 のリリースから 3 年がたちました。本教科書は CentOS6.2 を使ったサーバー構築のノウハウが集約されています。これからサーバー構築を実践してみたい！ そんなチャレンジ精神のある皆さんのお役に立てれば幸いです。

田口 貴久（バージョン 2 技術検証担当）
改版にあたり、わかりやすい教科書になるよう、構築でつまづきやすい部分を重点的に検証いたしました。Linux 技術者を目指す方のお役に立てれば幸いです。

高橋 征義（バージョン 2 PDF ／ EPUB 版制作担当）
本教科書の PDF・EPUB 作成のお手伝いをいたしました。サーバーエンジニアの方々の技術力向上に貢献できれば幸いです。

恒川 裕康（バージョン 3 執筆担当）
バージョン 3 では、CentOS7 で採用された systemd、NetworkManager、firewalld などに合わせて内容を変更しました。近年のセキュリティの動向にも配慮して改訂を行っています。さらに、仮想マシン上で実習をする場合に合わせて、Windows からの SSH 接続なども盛り込んでいます。本書を少しでも多くの方に活用していただければ幸いです。

著作権

使用に関する権利

本教科書の使用に関するお問合せ先
特定非営利活動法人エルピーアイジャパン（LPI-Japan）事務局
〒100-0011 東京都千代田区内幸町 2-1-1 飯野ビルディング 9 階
TEL: 03-6205-7025
E-Mail: info@lpi.or.jp

本教科書の目的

本教科書の目的は、LinuC レベル 2 の 201 試験と 202 試験の学習範囲に含まれるサーバー構築の知識を、構築の実習を通しながら学習することにあります。サーバーを構築した環境で、実際に Web アクセスをしたりメールの送受信をしたりすることで、サーバーの動作原理やプロトコルの仕組みを理解することも可能です

想定している実習環境

本教科書での実習環境として、以下の環境を構築しています。

講師と受講生

講師 1 名と受講生が 2 名以上存在すること前提とします。これは、実習のなかで受講生同士 2 名でペアを組み、お互いの設定したサーバーにアクセスをする作業を行うためです。

教室と割当

実習はコンピューター実習室のような教室で行うことを想定してます。講師の指示に従いながら、受講生が実習を行う形式になります。マシンは、講師、受講生に各名 1 台ずつを想定しています。

1 名で学習する場合

1 名で学習する場合は、マシンは最低 3 台必要になります。講師用マシン 1 台と受講生用マシン 2 台です。後述する仮想マシン環境を活用すれば、1 台で実習を行うことも可能です。

仮想マシン環境

仮想マシン環境を利用すると、Windows や Linux、Mac OS X 上の仮想マシンに Linux をインストールし、動作させることができます。仮想マシンは複数同時に動作させることができるので、3 台必要となる実習環境を 1 台のマシンでまかなうこともできます。仮想マシン環境を実現するソフトウェアとして、たとえば VMware 社の VMware Workstation(Windows) や VMware Fusion(Mac OS X)、Parallels 社の Parallels Desktop(Mac OS X) や Oracle 社の VirtualBox（Windows、Linux、Mac OS X）などが挙げられます。

マシンの構成とハードディスク

マシンの構成は、市販されている一般的な構成の PC を想定しています。その PC に Linux をインストールします。ハードディスクの内容は完全に消去されるので、ハードディスクの内容を消去しても良いマシンを用意するか、必要に応じてハードディスクの内容をあらかじめバックアップしておく必要があります。

OS

本教科書では、Linux ディストリビューションとして CentOS のバージョン 7.6（64 ビット版）を利用します。

ネットワーク

実習で利用するマシンは、すべて 1 つのネットワークで接続されていることを前提とします。インターネットへの接続は任意です。

全体の流れ

本教科書では、以下の通りに実習を進めます。

- 1 章 Linux のインストール準備と事前学習
- 2 章 Linux のインストールと設定を行う
- 3 章 DNS サーバーのインストールと設定を行う
- 4 章 Web サーバーのインストールと設定を行う
- 5 章 メールサーバーのインストールと設定を行う
- 6 章 サーバーのネットワークを管理しセキュリティの設定を行う

まず、1 章を学習してから、2 章から 5 章でサーバーのインストールと設定を行います。6 章は、インストール後に設定を変更したい場合に参考にして頂くための付録です。IP アドレスを変更したい場合、リモート管理をしたい場合、ファイアウォールの設定変更が必要な場合などに参考にしてください。

1 Linux のインストール準備と事前学習

本教科書では、Linux をインストールし、サーバー環境を構築する実習をしながら、LinuC レベル 2 の範囲の理解と取得を目指します。第 1 章では、2 章以降に行う実習に必要な環境の確認と知識の確認を行います。

1.1 用語集

Linux

Linus Torvalds 氏により開発された、UNIX 互換を目指した OS の総称を Linux といいます。ソースコードは公開されており、世界中の開発者の協力により、日々開発が継続されています。

ディストリビューション

Linux は狭い意味では OS の中心部（＝カーネル）のみを指しますが、Linux カーネルだけではシステムは動作しません。カーネル以外のさまざまなソフトウェアやインストーラを追加して、利用できるようにしたのがディストリビューションです。ディストリビューションごとに開発方針があり、それに沿ってソフトウェアがまとめられたり、リリースが行われています。

CentOS

Linux のディストリビューションの 1 つです。Red Hat Enterprise Linux という商用のディストリビューション互換の環境を無償で提供しているディストリビューションで、CentOS コミュニティによってリリースされています。

ネットワークアドレス

IP ネットワークを小さく分割して利用するときに、ネットワークアドレス部とホストアドレス部に分かれます。ネットワークアドレスの識別に利用されるのが、ネットワークアドレス部です。

IP アドレス

インターネットにおいて、IP で通信が行われる場合、端末一つ一つに IP アドレスが割り当てられます。IP アドレスとはインターネット上での端末の所在地を示す” 住所” にあたります。

サブネットマスク

IP アドレスのうちネットワークアドレス部とホストアドレス部を識別するための数値のことをいいます。通常 8 ビット毎に.(ドット) で区切って入力されます。

DNS サーバーアドレス

IP アドレスと FQDN（＝ホスト名＋ドメイン名）の変換を行うのが DNS サービスであり、そのサービスを提供する DNS サーバーの IP アドレスのことです。

ホスト名

ネットワークに接続されたコンピューターに割り当てられた名称のことをいいます。特定のホストを識別するために使われます。

ドメイン名

インターネット上に存在するコンピューターやネットワークを識別するために付けらている名前の一種のことをいいます。ドメイン名はアルファベット、数字、一部の記号の組み合わせで構成されますが、日本語.jp のような国際化ドメインも使われるようになっています。

DVD

光学メディアの 1 種類で、ビデオ再生での利用で普及し、現在ではデータ記録の用途でも利用されています。約 700MB の CD-ROM に比べ、約 4.7GB と大容量でも利用できることから、OS のインストールディスクとしても利用されています。

ハードディスク

磁気を用いた記憶媒体であり、パソコンの記憶媒体の他、音楽プレーヤー、ビデオなどの記憶媒体としても用いられています。

SSD

半導体メモリであるフラッシュメモリを用いた記憶媒体であり、ハードディスクよりも読み書きの処理性能に優れています。

LVM

LVM（Logical Volume Manager）とは，複数のハードディスクやパーティションにまたがった記憶領域を一つの論理的なディスクとして扱うことのできるディスク管理の機能のことです。Linux をはじめとした UNIX 系 OS 上で利用できます。

RAID

複数のハードディスクをまとめて 1 台のハードディスクとして管理する技術のことです。RAID を使うことによりデータを分散して記録するため、高速化や安全性の向上が期待できます。RAID の方法には、専用のハードウェアを使う方法（ハードウェア RAID）とソフトウェアで実現する方法（ソフトウェア RAID）があり、高速性や安全性のレベルにより、RAID 0 や RAID 5 などいくつかの種類があります。

1.2 実習で利用するハードウェア

本教科書の実習では、市販されているような一般的な構成の PC に Linux を導入して、その環境上に様々なサーバーを導入し実際に動作させます。この実習で必要な、ハードウェアの仕様は次の通りです。

マシン本体

Windows や Linux が動作する、いわゆる「パソコン」を想定しています。ただ、実習用のパソコンを用意することが難しい場合には、「VirtualBox」のような仮想マシンソフトウェアを利用して実習環境を用意することもできます。VirutalBox は、次の URL からダウンロードすることができます。

```
https://www.virtualbox.org/
```

実装メモリ

CentOS 7 では 1024MB のメモリが推奨されています。実装メモリが少ない場合はテキストインストールが実行されることがあります。

DVD 光学ドライブ

本教科書の実習では、インストール用 DVD を利用するので、光学ドライブが DVD を読み取りできる必要があります。DVD の光学ドライブがない機種では、USB ドライブを利用することもできます。また一部のノートパソコン等で、光学ドライブが無いマシンもあります。そういったときは、USB 等で接続する DVD ドライブを用意します。それを利用することにより、インストール DVD を起動することができます。

USB メモリ

最近の PC では USB メモリからのブートもサポートしています。そのため、インストール DVD の代わりに、USB メモリを利用することができます。

ハードディスク

Linux をインストールするためには記憶装置が必要です。ここでは記憶装置としてハードディスクを使います。インストールにはハードディスクに約 8GB の空き領域があれば十分なので、一般的な構成の PC では十分満たしていると思います。またハードディスクをフォーマット（初期化）して Linux をインストールします。従って、ハードディスクの中身は全部消去されます。その為、ハードディスクを削除していい PC を利用するか、バックアップを取ってから作業を行ってください。

その他周辺機器

本体や DVD、光学ドライブ、ハードディスクドライブの他にも、一般的に利用するためにはキーボード、マウス、ディスプレイ等の周辺機器が必要です。キーボードは、日本語か英語かで設定が異なりますので、日本語キーボード、英語キーボードの区別を「確認シート」に記述します。

1.3 利用する Linux のディストリビューション

本教科書では、CentOS のバージョン 7.6、64 ビット版を利用します。

CentOS 公式サイト

```
http://www.centos.org/
```

CentOS は、商用ディストリビューションである Red Hat Enterprise Linux の互換ディストリビューションとして提供されています。本家 Red Hat とのバイナリ互換を保ちながら、サポートも同等を目指すという方針で開発されています。利用に際し費用が発生することはない、無償で提供されているディストリビューションです。

1.3.1 インストール DVD/USB の入手方法

今回インストールには、DVD または USB のインストーラーを利用します。CentOS のインストール用 DVD、USB の入手方法には次の 2 通りが存在します。

ISO イメージをダウンロードする

CentOS が配布している ISO イメージを、ダウンロードします。ダウンロード元の URL は以下のとおりです。CentOS の最新版の、ISO イメージのページへのリンクになっています。

ダウンロードサイト

```
http://isoredirect.centos.org/centos/7.6.1810/isos/x86_64/
```

この URL をクリックすると、多くのミラーサイトが表示されます。その中でバージョン 7.6 の ISO イメージをダウンロードします。例えば、riken（理化学研究所）が提供しているミラーサイトの URL であれば次のようになります。

ダウンロードする ISO イメージ

```
http://ftp.riken.jp/Linux/centos/7.6.1810/isos/x86_64/CentOS-7-x86_64-DVD-1810.iso
```

この DVD イメージ (CentOS-7-x86_64-DVD-1810.iso) をダウンロードします。ISO イメージは合計で 4.3GB あり、転送に時間がかかります。

また、BitTorrent を使ってのダウンロードも行えます。ダウンロードサイトに.torrent という拡張子のファイル名が置かれているので、このファイルをダウンロード後、BitTorrent に対応したソフトウェアを使ってダウンロードが行えます。

ダウンロードした ISO イメージは、DVD や USB のライティングソフトウェアを使って DVD/USB に書き込んでください。データとしてではなく、イメージとして書き込む点に注意してください。

雑誌や解説書の付録

CentOS は雑誌に付属していたり、CentOS の解説書が多く出版されています。それらに付属しているインストール DVD を利用してもかまいません。

図 1　CentOS7.6 のダウンロードサイト

1.3.2　バージョン

本教科書では、本教科書の作成時点で最新であった CentOS 7.6 を利用した構築方法について解説しています。雑誌や解説書などの付録など、入手の方法によっては 7.6 ではない、より新しいバージョンの CentOS を手にすることがあるかもしれません。しかし、バージョン 7.x 系であれば大きな差は無いようです。従って CentOS の 7.x 系であれば、同様の手順でサーバーの構築ができるようになっています。

1.4　ネットワーク環境について

本教科書での実習ではネットワークを利用します。ネットワークの設定項目は複数ありますので、次の「ネットワークの設定項目」の説明を参考にしてあらかじめ別紙「確認シート」を作成した上で設定を行いましょう。

利用するネットワークの各種設定情報が組織のネットワーク管理担当者から指示されている場合は、その内容を「確認シート」に記述します。本教科書では特定の IP アドレスを用いて設定しますが、それを適宜指示された内容に読み替えてください。

ネットワークを自由に設定できる場合は、本教科書で用いているネットワークの設定を利用して下さい。本教科書では、ネットワーク環境としてコンピューター実習教室を想定しています。教室には PC が 3 台以上あり、講師用 PC が 1 台と受講生用 PC が 2 台以上あるとします。ネットワークは、講師用、受講生用 PC の区別無く、すべての PC が 1 つのネットワークに接続されていることを想定しています。

CentOS7 は、インターネットからパッケージを自動的にダウンロードする機能など、インターネットに接続できる環境で利用することを想定しています。そのため、実際にインターネットに接続できる IP アドレスと DNS サーバの設定をすると、簡単に設定を行うことができます。できるだけ、そのような環境を用意することをお勧めします。

1.4.1　ネットワークの設定項目

ドメイン名

ドメイン名は、DNS サーバーを設定するときに必要になります。受講生同士が同じドメイン名にならなければ、各自自由なドメイン名をつけてかまいません。このドメイン名は、あくまでこのネットワーク内のみで有効なドメイン名で、外部の DNS とは隔離された状態にあります。本教科書では、受講生用に alpha.jp と beta.jp の 2 つを使用します。

ホスト名

自分の PC に設定するホスト名です。今回は host ＋受講生番号 ＋ ドメイン名とします。確認のため「確認シート」に記述します。

IP アドレス

IP アドレスは、PC の IP アドレスです。できれば、実際にインターネットに接続できる IP アドレスを設定してください。なお、本教科書では、講師用 PC の IP アドレスを 192.168.1.10、受講生用 PC の IP アドレスを 192.168.1.101 と 192.168.1.102 としています。適宜テキストを読み替えて利用してください。

サブネットマスク

サブネットマスクは、IP アドレスのネットワーク部とホスト部を分ける値です。本教科書では 255.255.255.0(/24) とします。実際のネットワークに合わせた値を設定してください。

ネットワークアドレス

ネットワークアドレスは、PC が含まれているネットワーク全体を示すアドレスです。本教科書では 192.168.1.0 とします。

デフォルトゲートウエイ

異なるサブネットとの通信に必要な値です。本教科書では 192.168.1.1 とします。実際のネットワークに合わせた値を設定してください。

DNS サーバーアドレス

ホスト名と IP アドレスの対応を解決する、DNS（ドメインネームシステム）という機構があります。DNS を利用するためには DNS サーバーの IP アドレスが必要です。インストール時には、できるだけ実際のネットワークで利用可能な DNS サーバーのアドレスを設定してください。

なお、本教科書では 3 章で、実際に DNS サーバーを設定し動作させます。実習では、最初は外部の DNS サーバを使ってパッケージをインストールし、DNS サーバを構築します。その後は、自分自身で動作させている DNS サーバーを参照するため、DNS サーバーアドレスを自分の IP アドレスに変更します。

1.4.2 ネットワークの設定項目の確認シートの例 (受講者 1 用)

設定項目	本教科書の例	備考
ドメイン名	alpha.jp	
ホスト名	host1.alpha.jp	
IP アドレス	192.168.1.101	
サブネットマスク	255.255.255.0(/24)	
ネットワークアドレス	192.168.1.0	
デフォルトゲートウエイ	192.168.1.1	
DNS サーバーアドレス	192.168.1.10	

1.5 高度なストレージ管理

ここでは、高度なストレージ管理として LVM(Logical Volume Manager) と RAID(Redundant Arrays of Inexpensive Disks) について説明します。インストールを開始すると、LVM の設定が施されるところがあり、それについての説明です。少々高度な内容になるため、インストール作業後に読んでもかまいません。

1.5.1 LVM

Logical Volume Manager（LVM）という機構は、一言で言えば「ディスク管理操作を非常に便利にしてくれる機構」と言えます。ハードディスクを利用する際には、いくつかの「パーティション」に分割します。Windows のみならず、Linux でもパーティションを分割する作業を行います。パーティション分割作業は、容量を決めるのに非常に困難が伴います。「思ったより利用が多く、足りなくなってしまった」「念のため多めにパーティションを割り当てたら、あまり利用されず、大半が未使用になってしまった」といった具合です。だからといって、パーティションを再分割することは非常に手間がかかります。ハードディスクの内容を、一度全部消してしまうからです。再度インストール作業を行なった後、設定を行いデータを復元する作業は、相当な時間や手間を使います。LVM を用いると、パーティションを柔軟に取り扱うことができます。

1.5.2 LVM の仕組み

LVM の仕組みは、次のようになっています。

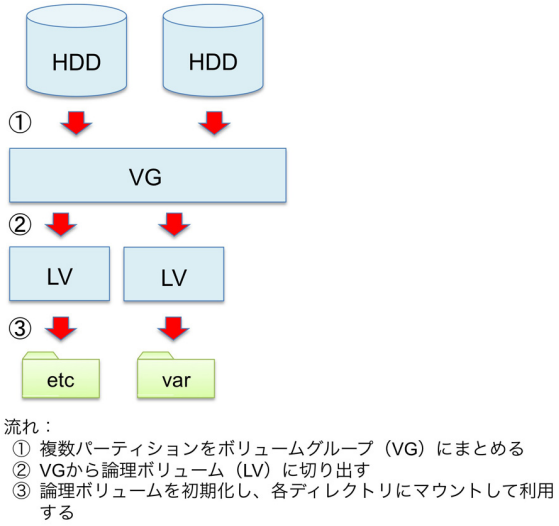


図 2 LVM の仕組み

PV(Physical Volume)

ディスクの物理領域です。パーティションの 1 区画であったり、ディスク 1 台丸ごと PV ということもありえます。

VG(Volume Group)

一つ以上の PV の集まりが Volume Group です。

LV(Logical Volume)

VG から、LV 領域を切り出して利用します。LV は自由に容量を増やしたり減らしたりできます。LV の容量の合計が、切り出し元の VG より大きくなることはありません。LV の領域にファイルシステムを作成して、ファイルやディレクトリなどのデータが格納されます。

1.5.3 LVM の利点

LVM を用いると、どんな点が便利なのでしょうか?実際にケーススタディで学習してみましょう。

ディスク領域が不足した場合にディスク容量の増加が容易

LV に保存したデータが増加し、割り当てた LV の容量では足りなくなった場合は、LV の大きさを増やすことで対応可能です。使用しているファイルシステムによっては、OS を止めることなく容量を増やすこともできます。逆に LV を大きく切り出しすぎて余ってしまった場合には、ファイルシステムを縮めた後に LV を小さくします。これで VG の未使用領域が増えるので、他の LV を増やしたり、新しく LV を切り出したりするときに利用できます。

ハードディスクを増設するのも容易

LV の利用率も増え、その元である VG の空き容量が少なくなったとします。そのときは、ハードディスクを増設しますが、LVM を用いると作業は簡単です。ハードディスクを取り付けて、そのハードディスクを PV とします。その PV を VG に追加すると、VG 全体の容量が増えます。増えた VG から新しく LV を切り出したり、既存の LV のサイズを増やしたりすることに利用できる領域を増やすことができます。

1.6 RAID

1.6.1 RAID とは

RAID とは Redundant Arrays of Inexpensive Disks の略で、ディスクの耐障害性を高めたり、機能を高めたりすることに用いられます。ディスクのアクセス性能を上げたい場合や、重要なデータを置いておく場合に利用します。

1.6.2 RAID の種類

RAID にはその機能でさまざまな種類があります。ここでは広く用いられる RAID 0,1,5,6,1+0 について説明します。

RAID 0(ストライピング)

2 台以上のディスクを用意し、書き込み時にそれぞれのディスクに分散書き込みを行います。ディスクの処理が分散するので、読み書きの速度が高速になります。また、使用できるディスク容量はすべてのディスクの容量の合計となります。欠点は、1 台でもディスクが壊れるとすべてのデータが読み書きできなくなることです。

RAID 1(ミラーリング)

ディスクを 2 台用意し、それぞれに同じ内容を書き込みます。一方のディスクが壊れても、もう一方のディスクが正常であればデータは失われません。そのため、ディスク障害に強い構成を実現できます。欠点は、利用できる容量が総容量の半分になってしまうことです。例えば容量 500GB のディスクを 2 台用意しても、使用できる容量は 500GB のままです。

RAID 5(パリティ分散)

ディスクを 3 台以上用意し、パリティという特別な仕組みを一緒に書き込むことでディスクの冗長化を図っています。ディスクが 1 台故障してもデータを失うことはありません。RAID 5 は 1 台あたりのディスク容量　×　(台数-1) の容量が使えますので、ディスクの利用効率も良いことになります。たとえば 500GB のディスクを 3 台用意すれば、合計 1TB のディスク容量になります。欠点は、データ書き込み時のパリティ計算の負荷が高いため書き込み性能が高くないことや、故障に耐えられるディスクが 1 台までなので、運悪く 2 台以上同時に壊れると元データの復元ができないといった点が挙げられます。

RAID 6（複数分散パリティ）

パリティを 2 重に計算し書き込むことで、ディスクが 2 台まで故障しても大丈夫にした RAID 構成です。欠点は、より高度なパリティ計算を高速に行うために専用のハードウェアが必要となる点です。

RAID1+0（ミラーリング＋ストライピング）

RAID 1+0 はミラーリング (RAID 1) したディスクをストライピング (RAID 0) する RAID の構成です。ストライピングはディスクが 1 台でも壊れるとすべてのデータが失われてしまいますが、RAID 1+0 ではミラーリングが行われているので、ディスクが 1 台壊れてもデータは失われません。ただし、ミラーリングされた両方のディスクが壊れてしまうと、通常の RAID 0 と同様にデータは失われてしまいます。欠点は、ディスクが最低でも 4 台必要であることと、ミラーリングされているため容量が半分になってしまうという点です。

その他の RAID

RAID には、他にも 2,3,4 がありますが、あまり使われていません。

1.6.3 ハードウェア RAID とソフトウェア RAID

RAID ではハードウェア RAID とソフトウェア RAID が存在します。

ハードウェア RAID

ハードウェア RAID は、RAID の処理をハードウェアが行います。従って、OS、マザーボード側から見ると、ディスクが 1 台存在しているように見えるだけです。RAID コントローラは OS、マザーボードにディスクが 1 台と「見せかけながら」、その背後で RAID の処理を行っています。ハードウェア RAID を使う利点は、OS は一般的なハードディスクとして認識されるために特

別なドライバーを導入する必要がないことと、OS やハードウェアに負荷がかからないことです。欠点として特別なハードウェア（RAID コントローラ）が必要であるため、費用がかかる点が挙げられます。

ソフトウェア RAID

ソフトウェア RAID は、OS やドライバーが RAID 作業を行います。ソフトウェア RAID は特別なハードウェア（RAID コントローラ）が必要ではないため、コストを抑えて RAID を組むことができますが、欠点として OS の対応やドライバーの対応が必要であることや、ハードウェア RAID と比べて OS、ハードウェア（特に CPU）に負荷がかかる点が挙げられます。

1.6.4 高度なストレージの利用

高度なストレージとして、LVM と RAID を紹介しました。ではどのような場面で利用するのが好ましいでしょうか？

Linux では（後に紹介しますが)、ディスクを使用するためにパーティションに対して特定のディレクトリをマウントさせます。デフォルトの構成ではパーティションが一つ作成され、そこにすべてのディレクトリの大元となるルートディレクトリ（”/”）がマウントされ、そのサブディレクトリとして/var や/home といったディレクトリが作成されます。

/var には、ログファイルなどシステムが様々なデータを書き出します。/home は、ユーザーが作成したデータが置かれます。この 2 つのディレクトリは非常に重要であり、なおかつ利用量が非常に変化しやすいディレクトリなので、このようなディレクトリはパーティションを分け、ルートディレクトリと切り離してマウントし、そのパーティションを LVM を用いて可変としたり、RAID を用いて冗長化されるよう構成することが望ましいと言えます。

2 Linux のインストール

本章では、実際に Linux のインストールと設定を行います。ネットワークの設定や、インストールするソフトウェアの選択など、次章以降に影響する重要な内容ですので、しっかり学習しましょう。

2.1 用語集

メディアの整合性

作成されたメディアが、配布されたオリジナルの内容と違いが無いかどうかはメディアの整合性が取れているかどうかで確認できます。何らかの原因により整合性が取れていない場合、ソフトウェアのインストールに失敗してしまいます。CentOS ではインストール手順の開始時、メディアの整合性が取れているかどうかチェックが行えるようになっています。

BIOS

PC の周辺機器を制御するプログラムのことをいいます。PC には必ずこの BIOS が内蔵され、BIOS が起動後、OS が起動します。内蔵の時計や、起動デバイスの選択等を設定できます。設定は、マザーボード上のフラッシュメモリに保存されています。

起動順序

どの記憶装置から OS を起動するか、起動デバイスの優先順位をつけることをいいます。起動順序は、BIOS で設定することができます。ハードディスクの他、CD/DVD 等の光学ドライブ、USB のストレージデバイス、FDD 等を選ぶことができます。

タイムゾーン

Linux の動作時に時刻を設定します。通常の時刻を設定するほか、そのマシンが起動している場所の時間帯を設定できます。日本で動作させるときは、日本標準時 (=JST) に設定します。

フォーマット

ハードディスク等を OS で読み書きできる状態にすることで初期化ともいいます。フォーマットを実行すると、ディスクのデータはすべて削除されます。

ファイアウォール

インターネットにコンピューターを直接接続すると不正にアクセスされるおそれがあるため、ファイアウォールを構築します。ファイアウォールを動作させることで、ネットワークのセキュリティ機能を高めることができます。通常の利用では有効化することが推奨されます。

2.2 インストールの前に用意するもの

確認シート

インストールの前に、1 章で記入した「確認シート」を手元に用意します。この内容を確認しながら、インストール作業を行います。

インストール DVD/USB

CentOS 7.6 のインストール DVD/USB を用意します。

マシンの設定

インストールを開始するにあたり、マシンの設定を確認します。確認する内容は、BIOS で設定する「起動順序」です。DVD からインストールする場合には、起動順序で必ず DVD ドライブを優先にします。同様に、USB からインストールする場合には、

起動順序で USB デバイスを優先にします。これらのデバイスよりハードディスクの優先度が高いと、ハードディスクにインストールされている OS が起動してしまいます。

ハードディスク

本教科書では、CentOS をインストールする際にハードディスクの中身を消去します。従ってハードディスクの中身を消去しても良い PC を利用するか、ハードディスクの中身のバックアップを取ってから作業を行います。

2.3 インストールの開始

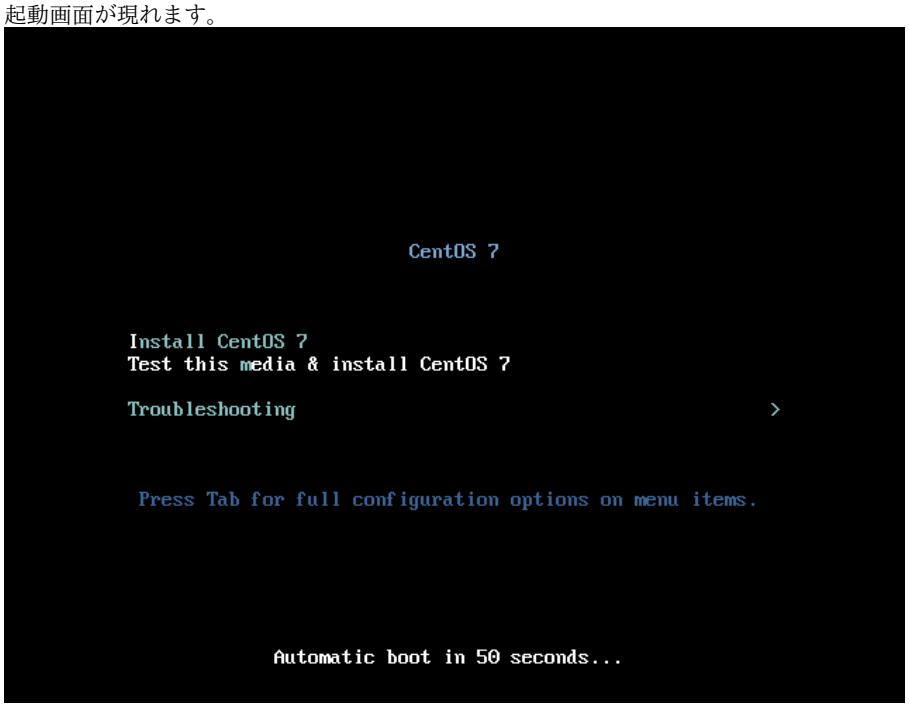
それでは、インストールを開始します。

2.3.1 インストールメディアの読み込み

1. DVD ドライブ、USB デバイスの起動優先順位をハードディスクより高くします。

2. インストール DVD/USB をセットし、マシンを起動します。

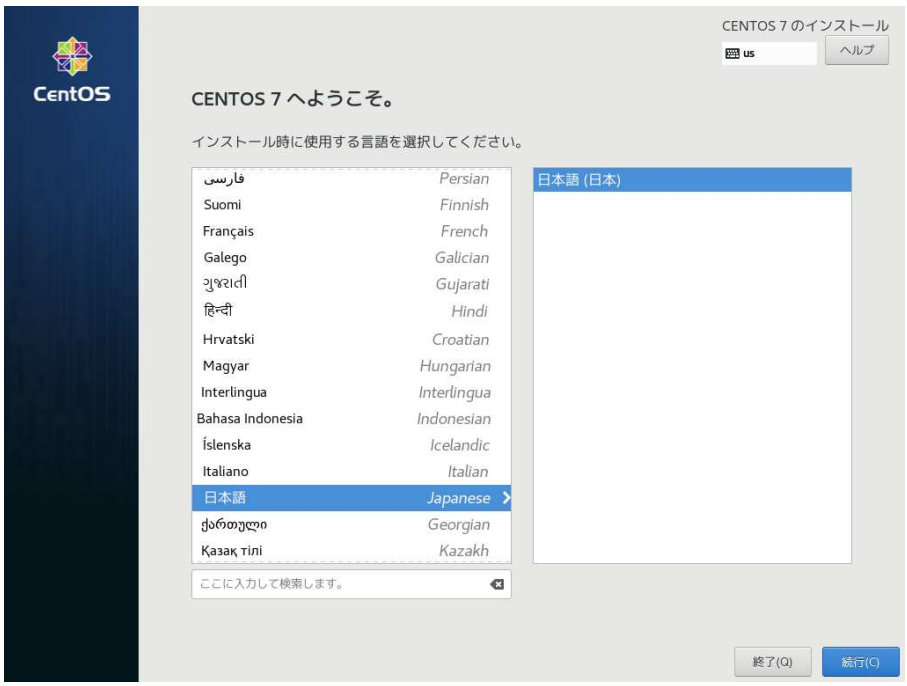
3. 起動画面が現れます。



4. ↓↑などのキーを使ってメニューを選択します。「Install CentOS 7」を選択し、Enter キーを押すとインストールが始まります。
「Test this media & Install CentOS 7」を選択すると、メディアの内容の整合性をチェックし、問題がなければインストールが始まります。

2.3.2 言語設定

1. 言語の選択画面が表示されるので、左側のメニューから「日本語 Japanese」を選択します。画面右側のメニューに「日本語（日本）」と表示されます。



2.「続行」をクリックします。すると、インストールメニュー画面が表示されます。

「!」が付いた項目は、必ず設定を行わなければならない項目です。

2.3.3 インストール先ディスクの設定

1.「インストール先」のアイコンをクリックします。すると、「インストール先のディスク設定」画面が表示されます。

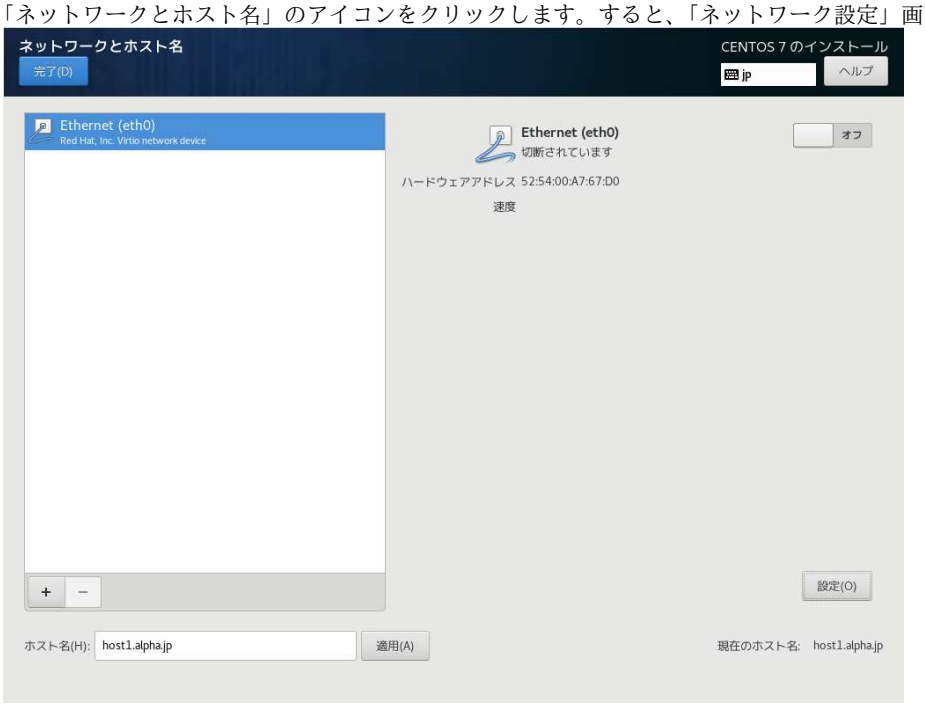
2.「ローカルの標準ディスク」の欄で、利用するディスクを選択します。

3.「パーティション構成」の欄の「パーティションを自動構成する」を選択します。

4.「完了」をクリックします。すると、インストールメニュー画面に戻ります。

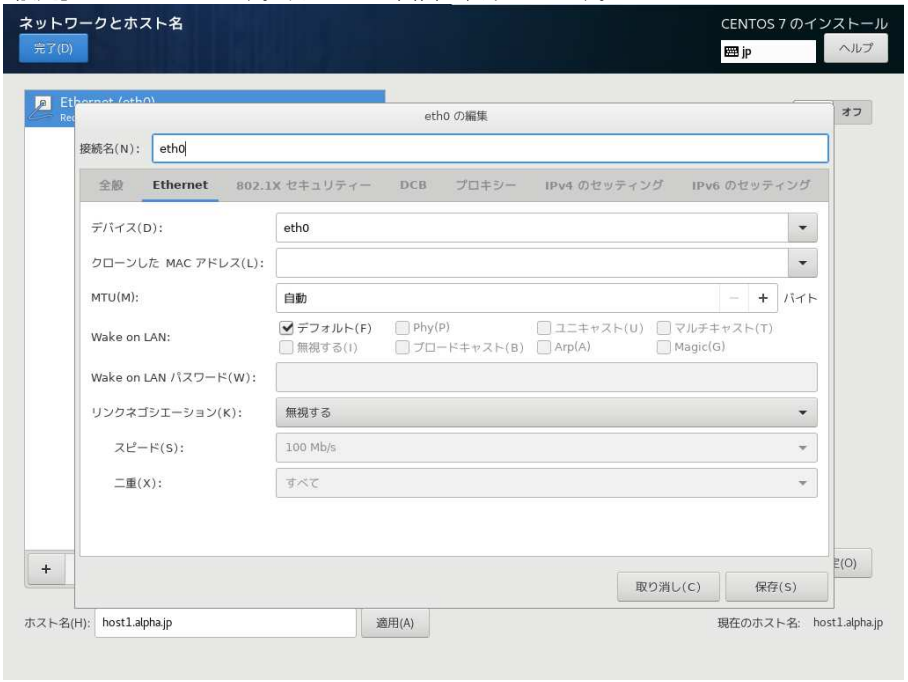
2.3.4 ネットワークとホスト名の設定

1.「ネットワークとホスト名」のアイコンをクリックします。すると、「ネットワーク設定」画面が表示されます。

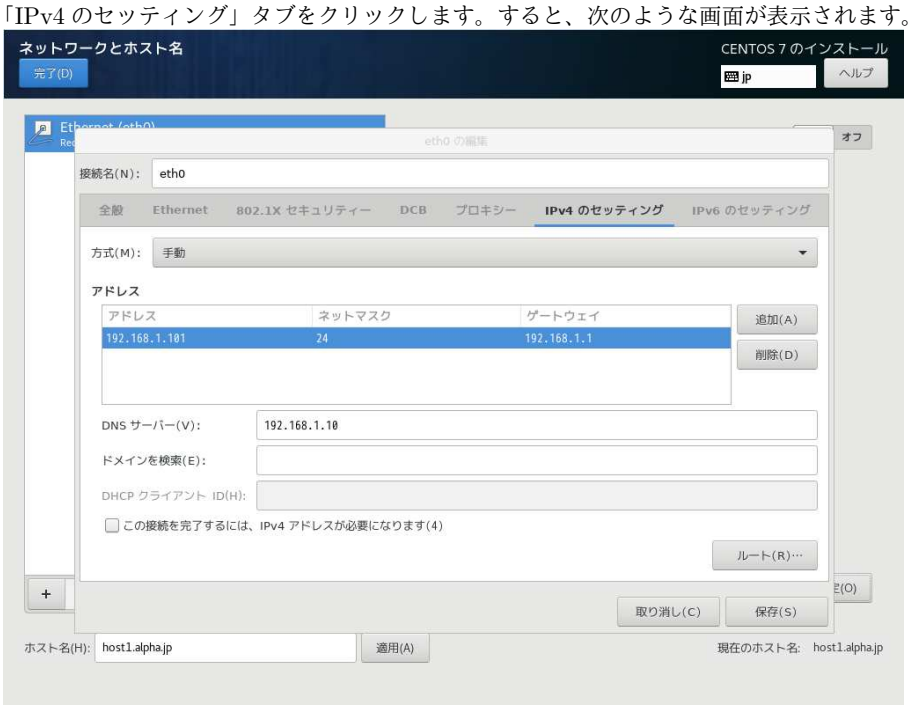


2.「ホスト名」の欄に、ホスト名を入力します。入力後、「適用」をクリックします。適用されると、右側の「設定」ボタンの下にホスト名が表示されます。

3.「設定」をクリックします。次のような画面が表示されます。



4.「IPv4 のセッティング」タブをクリックします。すると、次のような画面が表示されます。



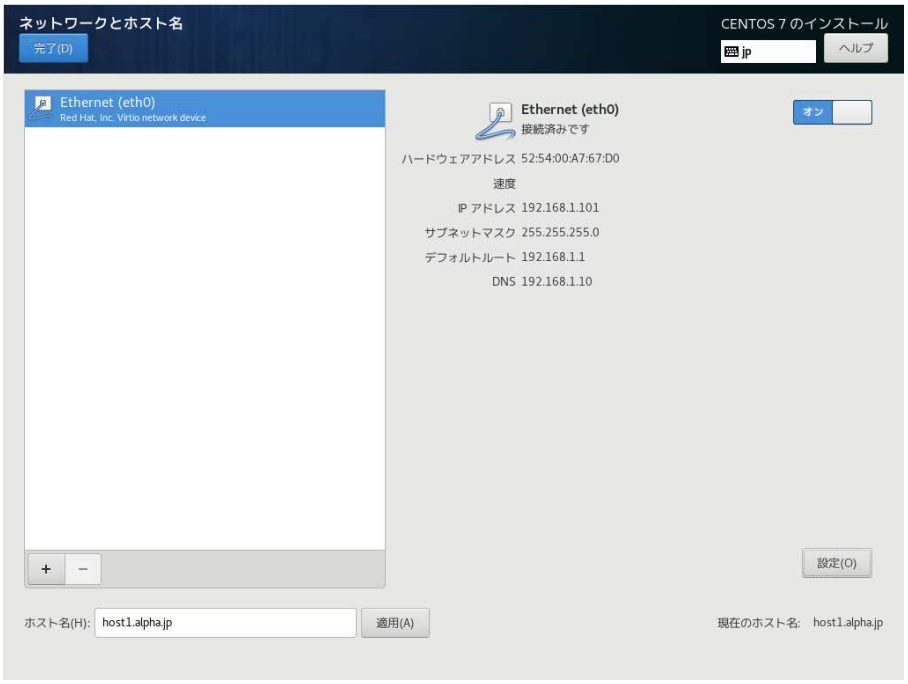
5.「方式」を「手動」に変更します。

6. アドレスの欄の「追加」をクリックすると、アドレス、ネットマスク、ゲートウェイの項目が入力できるようになります。各項目に設定を行います。

7.「DNS サーバー」に DNS サーバーのアドレスを入力します。

8. 設定が終わったら「保存」をクリックします。

9. ネットワークとホスト名の設定画面に戻ります。「オフ」になっているスイッチをクリックして、「オン」に変更します。



先ほど設定したアドレス、サブネットマスク、デフォルトゲートルート、DNS の値が表示され、ネットワークが有効になります。

10.「完了」をクリックして、インストールメニュー画面に戻ります。

2.3.5 ソフトウェアの選択

1.「ソフトウェアの選択」のアイコンをクリックします。「ソフトウェアの選択」画面が表示されます。

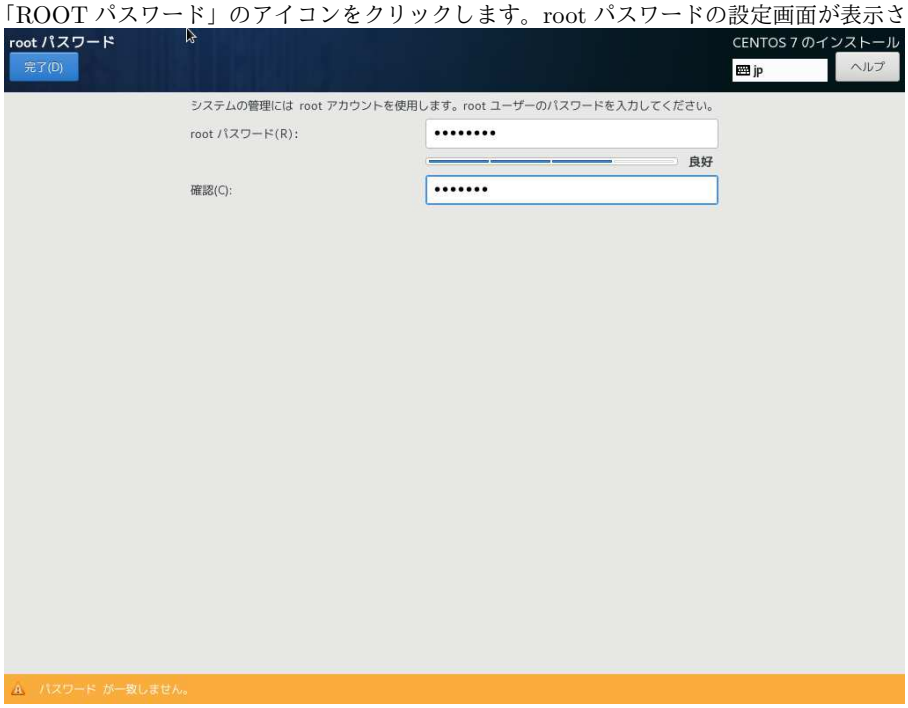
2. ベース環境メニューから「サーバー（GUI 使用）」を選択します。また、選択した環境のアドオンのメニューから「開発ツール」を選択します。

3. 設定ができたら、「完了」をクリックして、インストールメニューに戻ります。
自動的にソフトウェアの依存関係のチェックが行われます。チェックが終わると「！」がはずれ、「インストール開始」がクリックできるようになります。

2.3.6 インストールの開始とパスワード設定

1.「インストール開始」をクリックすると、インストールが始まり、「インストール状況の表示」画面が表示されます。

2.「ROOT パスワード」のアイコンをクリックします。root パスワードの設定画面が表示されます。



3.「root パスワード」「確認」の項目に、root のパスワードを入力します。英文字の大文字、小文字、数字、記号などを組み合わせて、複雑なパスワードを設定します。パスワードの強度が足りない場合や、２つのパスワードが一致しない場合には、画面の最下部に警告メッセージが表示されます。

4. 設定ができたら「完了」をクリックして、「インストール状況の表示」画面に戻ります。

5.「ユーザの作成」のアイコンをクリックします。「ユーザの作成」画面が表示されます。ここでは、管理ユーザを設定します。

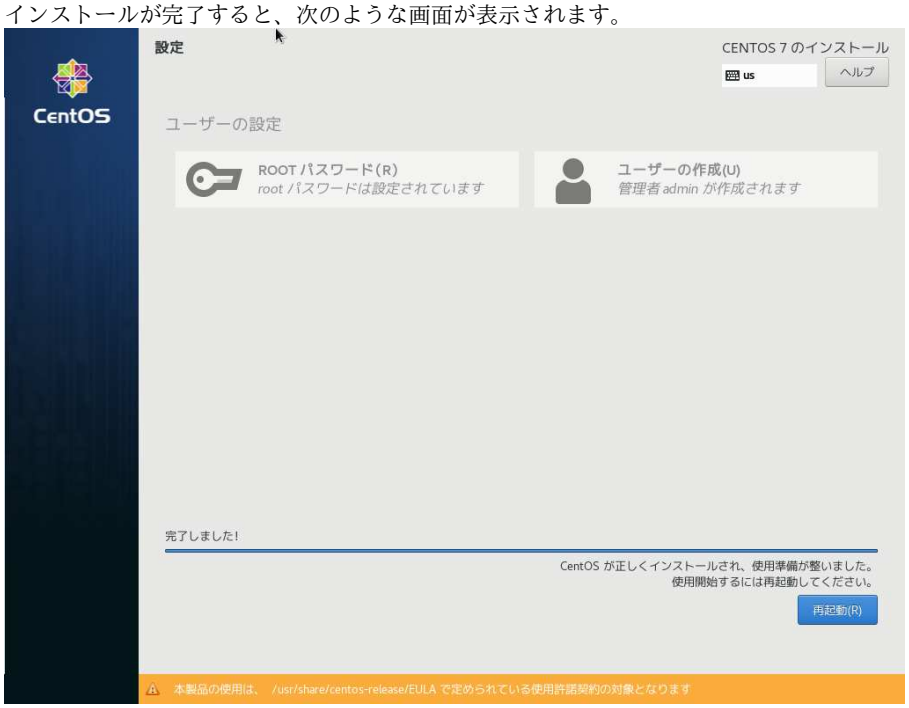
6. ユーザ名に管理ユーザの名前を入力します。どのような名前でも構いませんが、本書では、admin を入力します。

7.「このユーザを管理者にする」「このアカウントを使用する場合にパスワードを必要とする」をクリックし、チェックします。

8.「パスワード」「パスワードの確認」に、管理ユーザのパスワードを入力します。パスワードの強度が足りない場合や、2 つのパスワードが一致しない場合には、画面の最下部に警告メッセージが表示されます。

9.「完了」をクリックして、インストール状況の表示画面に戻ります。

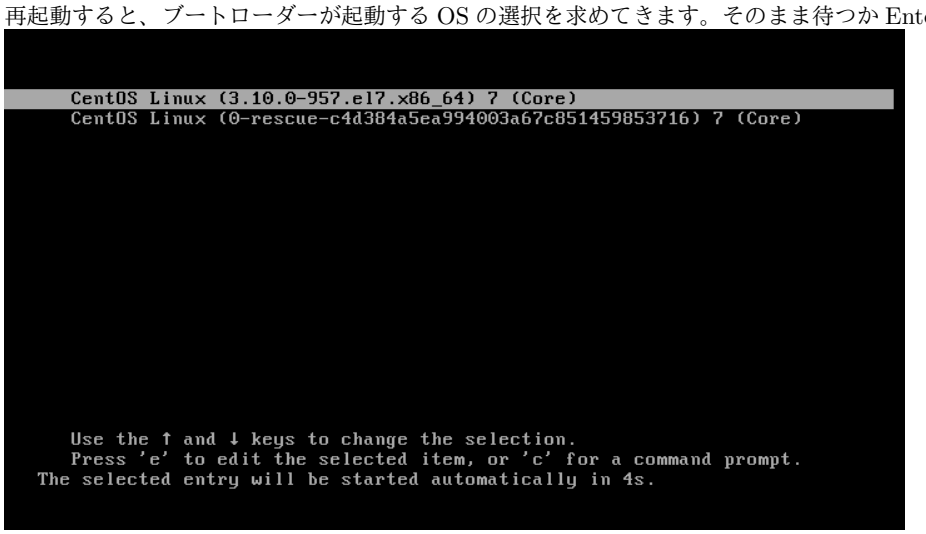
10. インストールが完了すると、次のような画面が表示されます。



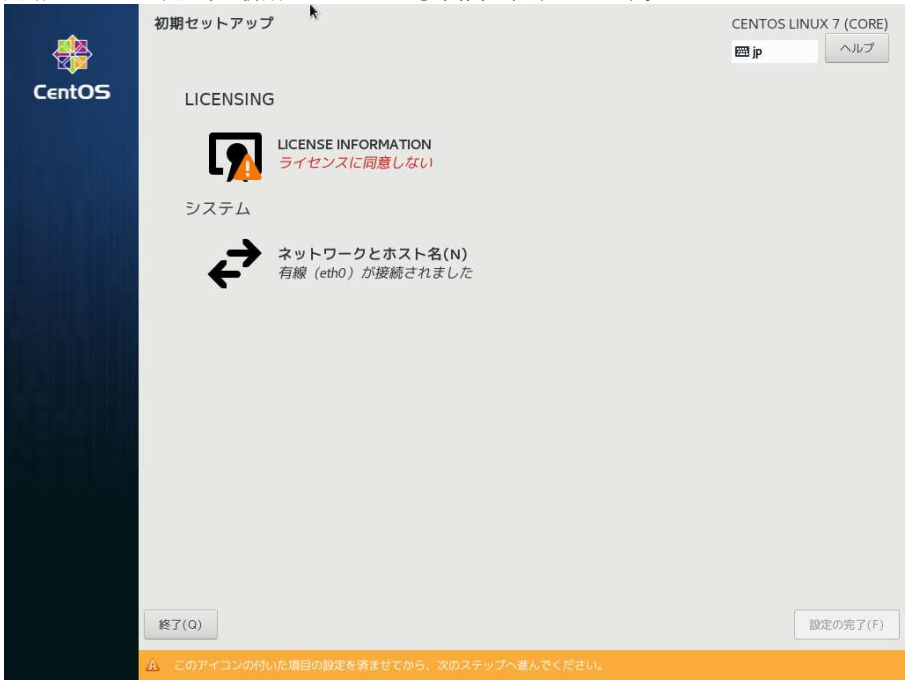
「再起動」をクリックします。

2.3.7 インストール後の初期設定

1. 再起動すると、ブートローダーが起動する OS の選択を求めてきます。そのまま待つか Enter キーを押します。



2. 起動プロセスが進み、「初期セットアップ」画面が表示されます。



3.「LICENSE INFORMAITON」のアイコンをクリックします。すると、「ライセンス情報」画面が表示されます。

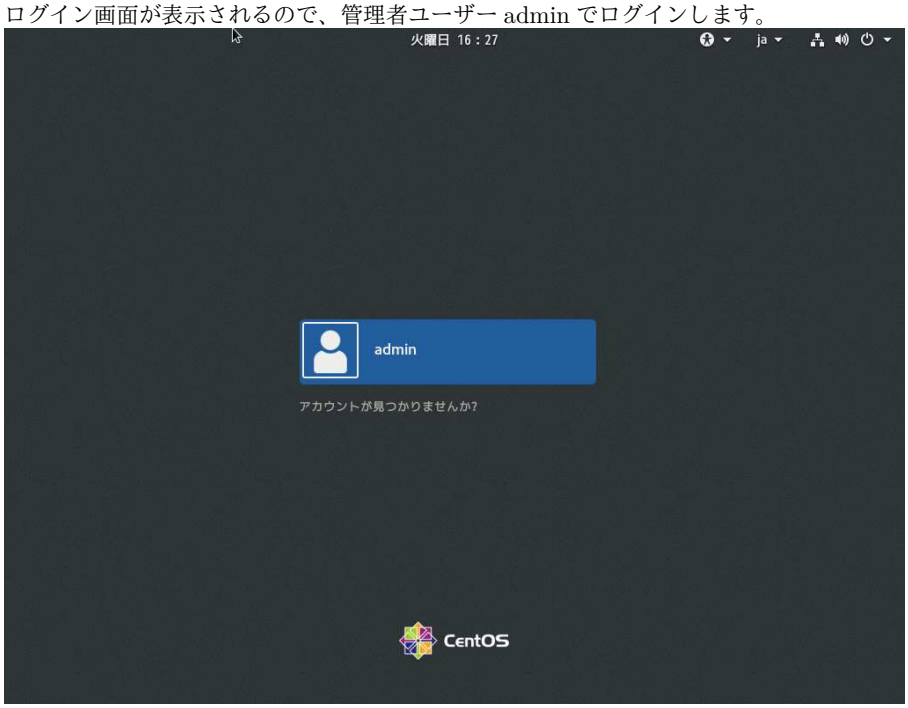
4.「ライセンス契約に同意します。」のチェックボックスをクリックします。

5.「完了」をクリックし、「ライセンス情報」画面に戻ります。

6.「設定の完了」ボタンをクリックすると、初期設定作業はすべて完了です。

2.4 ログインする

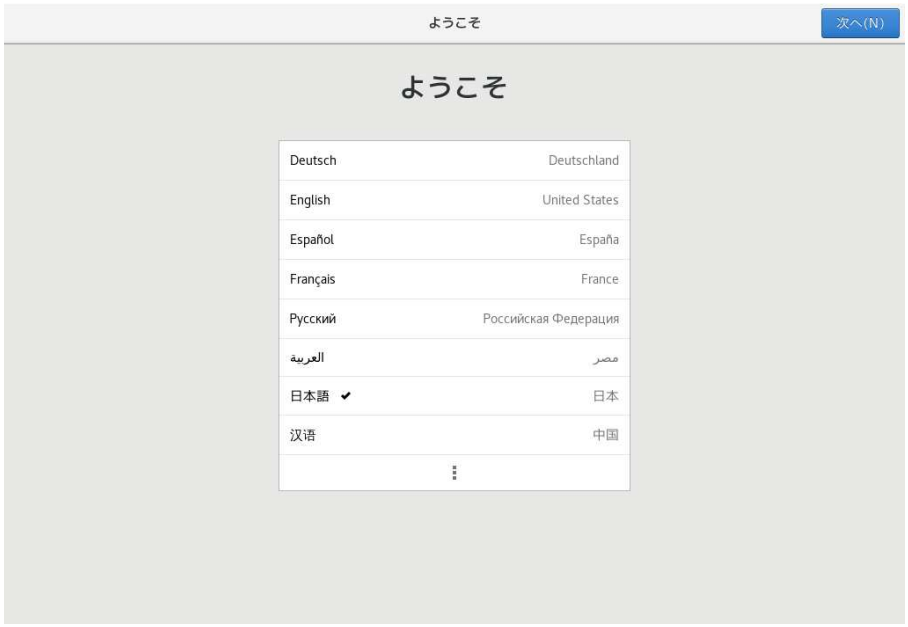
ログイン画面が表示されるので、管理者ユーザー admin でログインします。



1. ユーザ名をクリックします。すると、パスワード入力欄が表示されます。

2. admin のパスワードを入力します。

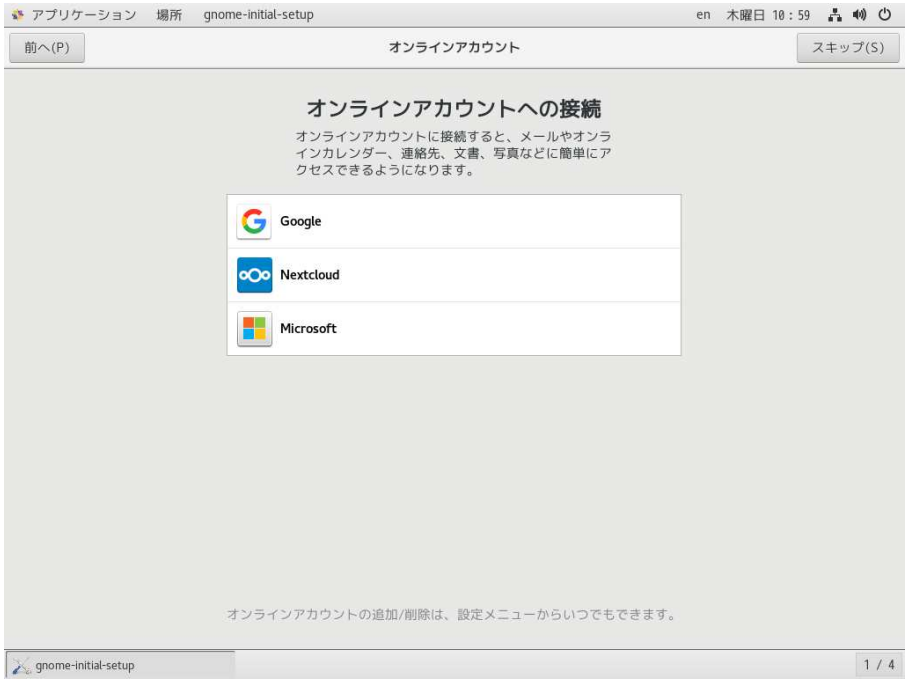
3. 最初にログインすると、「ようこそ」の画面が表示されます。



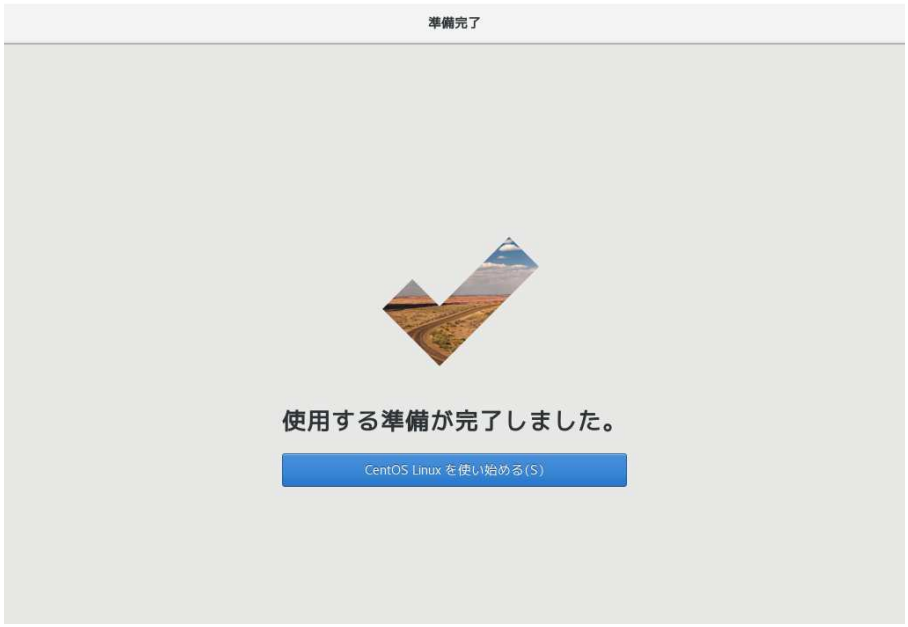
4. 使用言語が「日本語」になっていることを確認して、「次へ」をクリックします。すると、「入力」画面が表示されます。

5.「日本語（かな漢字）」を選択して、「次へ」をクリックします。すると「プライバシー」画面が表示されます。

6. 「次へ」をクリックします。「オンラインアカウントへの接続」画面が表示されます。



7. サーバーの管理には不要な項目ですので、「スキップ」をクリックします。すると、「準備完了」画面が表示されます。



8.「CentOS Linux を使い始める」ボタンをクリックすると、ウィザードが終了します。

2.5 コマンドの実行

GUI でログインすると、Windows などと同じように様々なグラフィカルなアプリケーションが利用できますが、「端末」を実行すると Linux のコマンドを実行できます。

2.5.1 端末を利用する

端末を起動するには以下の方法があります。

- 「アプリケーション」メニューから「システムツール」、「端末」を選択する
- デスクトップ上を右クリックし、ポップアップメニューから「端末を開く」を選択する

「端末」ウインドウは複数起動できるので、一方で操作をしながら一方でログを表示したり、ユーザーを切り替えて操作することもできます。

2.5.2 Windows から SSH を使って接続する

VirtualBox などの仮想環境で実習を行っている場合には、Windows からネットワーク経由で接続することで、Linux のコマンドを使う方法が便利です。この方法を使う場合には、Windows に SSH 用のアプリケーションをインストールする必要があります。Windows で利用できる SSH のソフトウェアとしては、TeraTerm がよく使われています。TeraTerm は、telnet や SSH に対応したオープンソースソフトウェアです。日本語の表示にも対応しています。

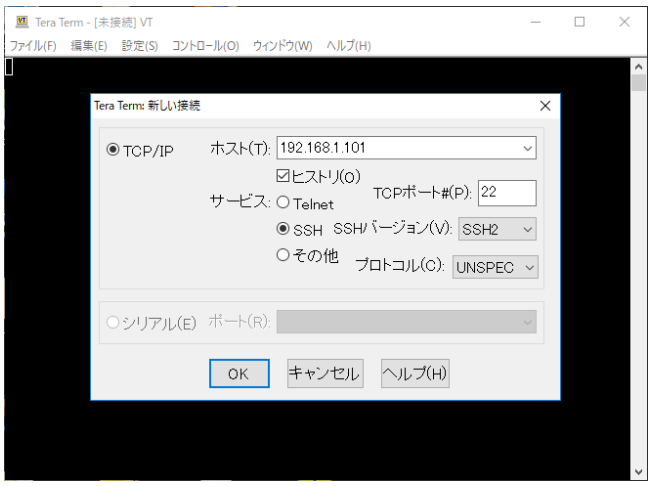
TeraTerm は、次の URL から入手することができます。

```
https://ja.osdn.net/projects/ttssh2/
```

ソフトウェアをダウンロードし、PC にインストールします。

TeraTerm の使い方

1. TeraTerm を起動すると、次のような画面が表示されます。



2.「ホスト」の項目に SSH で接続したいホスト名か IP アドレスを入力します。サービスとバージョンは、標準で SSH、SSH2 になっていますので、変更不要です。「OK」をクリックすると、SSH 接続が行われます。

3. 初めて SSH 接続を行った場合には、SSH サーバーの電子証明書が送られてきて、接続してよいか尋ねられますので、「続行」をクリックします。

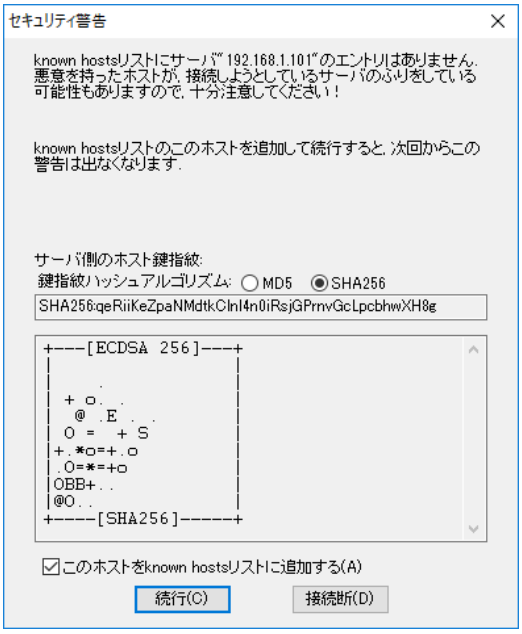
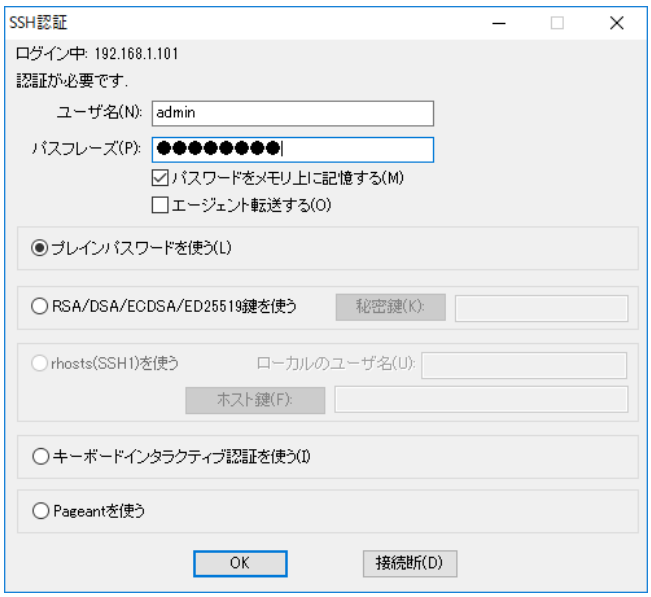
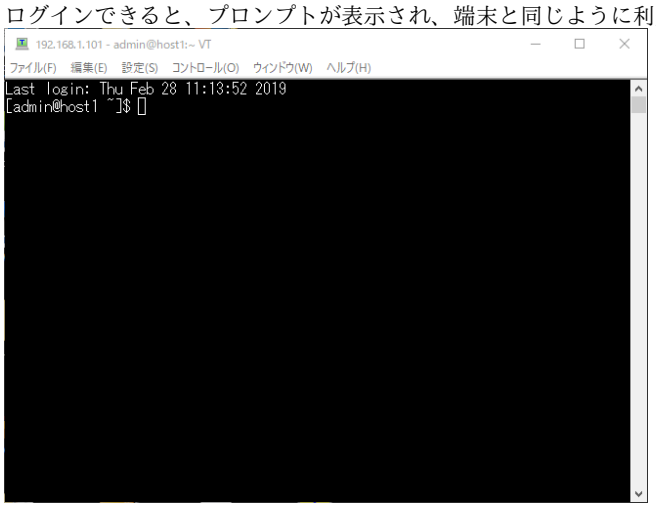


図 3 TeraTerm の起動画面

4. ユーザ認証の画面が表示されますので、ユーザ名に admin、パスワードに admin に設定したパスワードを入力して、「OK」をクリックします。



5. ログインできると、プロンプトが表示され、端末と同じように利用できるようになります。



TeraTerm も、端末と同様に複数起動し、必要に応じて切り替えて使うことができます。

2.5.3 root で設定を行う

実際に、サーバーの設定を行う場合には、root ユーザで設定を行う必要があります。root ユーザになるには、次のようにコマンドを実行します。パスワードには、インストール時に設定した root パスワードを入力します。

root ユーザになる

```
[admin@host1 ~]$ su -
パスワード: ******** ← rootパスワードを入力
[root@host1 ~]#
```

先頭の「[root@host1 ~]#」はプロンプトで、入力待ちであることを示しています。プロンプトには、ユーザ名、ホスト名、カレントディレクトリが表示されています。admin ユーザの場合には、その後ろに「$」が付きます。root ユーザになっている場合には、必ず「#」になります。確認しながら作業を行います。なお、本書では特に理由がない限り、プロンプトは省略して「# 」とだけ表記しています。

2.6 ローカルリポジトリの設定

CentOS7 では、標準では、ソフトウェアのパッケージはインターネットからダウンロードするようになっています。もし、インターネットに接続していない状態で演習を行う場合には、ローカルリポジトリの設定を行います。

ローカルリポジトリの設定ファイルは、/etc/yum.repos.d/CentOS-Media.repo です。このファイルを編集し、ローカルリポジトリを有効にし、DVD からインストールが行われるようにします。

/etc/yum.repos.d/CentOS-Media.repo の編集

```
# vi /etc/yum.repos.d/CentOS-Media.repo
```

/etc/yum.repos.d/CentOS-Media.repo

```
[c7-media]
name=CentOS-$releasever - Media
baseurl='file:///run/media/admin/CentOS 7 x86_64'          ←  変更する
gpgcheck=1
enabled=1                                              ←  変更する
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

この設定を行うと、admin ユーザでログインし、DVD を挿入したときにローカルリポジトリが使えるようになります。また、base や updates など、インターネットに接続した時にしか使えないリポジトリを無効にしておきます。

/etc/yum.repos.d/CentOS-Base.repo の編集

```
# vi /etc/yum.repos.d/CentOS-Base.repo
```

/etc/yum.repos.d/CentOS-Base.repo

```
# CentOS-Base.repo
#
# The mirror system uses the connecting IP address of the client and the
# update status of each mirror to pick mirrors that are updated to and
# geographically close to the client.  You should use this for CentOS updates
# unless you are manually picking other mirrors.
#
# If the mirrorlist= does not work for you, as a fall back you can try the
# remarked out baseurl= line instead.
#
#

[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearc
h&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
enabled=0   ←  追加

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearc
h&repo=updates&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
enabled=0   ←  追加
```

```
#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearc
h&repo=extras&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
enabled=0   ← 追加

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearc
h&repo=centosplus&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

3 DNS サーバーの構築

ネットワークサービスを使うための土台となる名前解決のサービス (DNS) を設定します。自分の DNS サーバーを他のコンピューターから参照できるように設定をします。DNS に問い合わせを行うコマンドに慣れ、ドメインを管理する BIND プログラムの設定ファイルを扱います。

3.1 用語集

ドメイン名とゾーン

組織に割り当てられてインターネットで使用する名前をドメイン名と呼びます。ドメイン名は ICANN(Internet Corporation for Assigned Names and Numbers) により管理されています。DNS でドメイン名を設定するときは、ドメインではなく「ゾーン」と呼びます。

FQDN

ドメイン名表記で、一番右に「.」（ドット）でルートドメインまでを記述する方式を FQDN と呼びます。

DNS

DNS(Domain Name System) は、IP アドレスと対応するホスト名を登録しておき、プログラムからの問い合わせに応じて IP アドレスやホスト名を返答するシステムです。

ゾーン

DNS の名前空間の一部分を取り出したものをゾーンとよびます。ゾーンは、DNS を管理する単位として使われます。ゾーンには、ドメイン、サブドメイン、ホスト名などが含まれています。DNS 名前空間はツリー構造で表されますが、ゾーンは特定のノード以下の一部または全部を含む部分です。

DNS キャッシュサーバー

プログラムからの名前問い合わせを代行して、様々な DNS サーバーへ名前問い合わせを行って結果を返却するサーバーです。調べた結果をキャッシュしておき、次回問い合わせ時にキャッシュの情報を返すことから、DNS キャッシュサーバーと呼ばれます。単純に、DNS サーバーと呼んだときには、DNS キャッシュサーバーのことを指します。

DNS コンテンツサーバー

委託されたゾーンの IP アドレスやホスト名を管理する DNS サーバーです。

リゾルバ

ドメイン名をもとに IP アドレス情報の検索をしたり、IP アドレスからドメイン情報の検索を行う、名前解決を行うプログラムのことです。

BIND

BIND(Berkeley Internet Name Domain) は、Linux と組み合わせて多く使用されている DNS サーバーのソフトウェアです。DNS キャッシュサーバーとしても、DNS コンテンツサーバーとしても利用することができます。

unbound

unbound は、高機能で高速な DNS キャッシュサーバーです。攻撃に強いことから、最近は BIND に変わって利用されることが多くなっています。

グルーレコード

管理を委任しているゾーンについての問合せに対して、DNS サーバーが委任先のゾーンの DNS コンテンツサーバーのアドレスを返す際に、追加情報として必要となる委任先 DNS コンテンツサーバーの A レコードをグルーレコードといいます。

A レコード

名前に対して IP アドレスを指定するためのレコードです。

NS(Name Server) レコード

ゾーンの権威を持つ DNS コンテンツサーバーを指定するためのレコードです。

MX(Mail eXchange) レコード

メールアドレスに利用するドメイン名を定義するためのレコードです。メールサーバーの障害にも対応するために、複数個のメールサーバーを記述でき、プリファレンス値の低いサーバーにメール配信が優先されます。

3.2 DNS の仕組み

インターネットでのコンピューター同士の通信は、IP(Internet Protocol) を使って行われています。IP 通信には相手の IP アドレスが必要ですが、インターネット上の大量のコンピューターを IP アドレスで識別するのは困難です。そこでドメイン名やホスト名という考え方が導入されました。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ーです。表記するときは「ホスト名. ドメイン名」とドット区切りで表記しますが、両方を合わせてホスト名と呼ぶこともあります。

インターネットの研究が始まった当初は IP アドレスが割り当てられたコンピューターの数も数えるほどだったので、ホスト名と IP アドレスの対応関係はファイルに記述されて、定期的に更新されていました。この仕組みは今でも残っており、Linux では/etc/hosts がそのファイルです。しかし、インターネットが広まるに従って、ホストファイルでは管理しきれなくなってきました。そこで登場したのが DNS(Domain Name System) です。

DNS **コンテンツサーバー**は、ドメイン名を割り当てられた組織毎に用意します。DNS コンテンツサーバーの管理者は、そのドメインに所属しているホスト名と割り当てられた IP アドレスを DNS コンテンツサーバー登録します。ホストにアクセスしたい利用者は、そのホストが所属するドメインの DNS コンテンツサーバーに問い合わせを行うことで、IP アドレスを得ることができます。しかし、ユーザが、アクセスする毎にどの DNS コンテンツサーバーに問い合わせをするのかを調べることは面倒です。*DNS* **キャッシュサーバー**は、その調査を自動的に行ってくれます。また、調査結果を一定時間キャッシュし、毎回調べなくても良いようにしてくれます。

DNS の仕組みでは、ゾーンの管理権限がそれぞれの DNS 管理者に権限委譲されているので、ホストファイルのような一元管理ではなく、分散管理となります。管理作業が分担されていて更新も頻繁に行われるので、リアルタイムにホスト名と IP アドレスの対応関係を調べることができる仕組みとなっています。

3.3 ドメインの構造

DNS が取り扱うドメイン名は設計上、ルートドメインを頂点とした階層型のツリー構造となっています。ちょうど、コンピューターのファイルシステムがルートディレクトリを頂点としたツリー構造になっているのと同じだと考えてよいでしょう。そして、その配下にあるドメインはサブドメインとよばれます。ドメインの階層型のツリーは、ルートドメインとたくさんのサブドメインから構成されています。

3.3.1 ルートドメイン

ルートドメインは、ドメイン名の開始点です。通常は省略されますが、DNS 名として記述する際には「.」（ドット）で表されます。

トップレベルドメイン

トップレベルドメインには、.com や.org のような組織別ドメインや、.jp のような国別ドメインがあります。また、日本の場合には.co.jp のような組織種別型ドメインと、example.jp のような汎用 JP ドメインなどがあります。

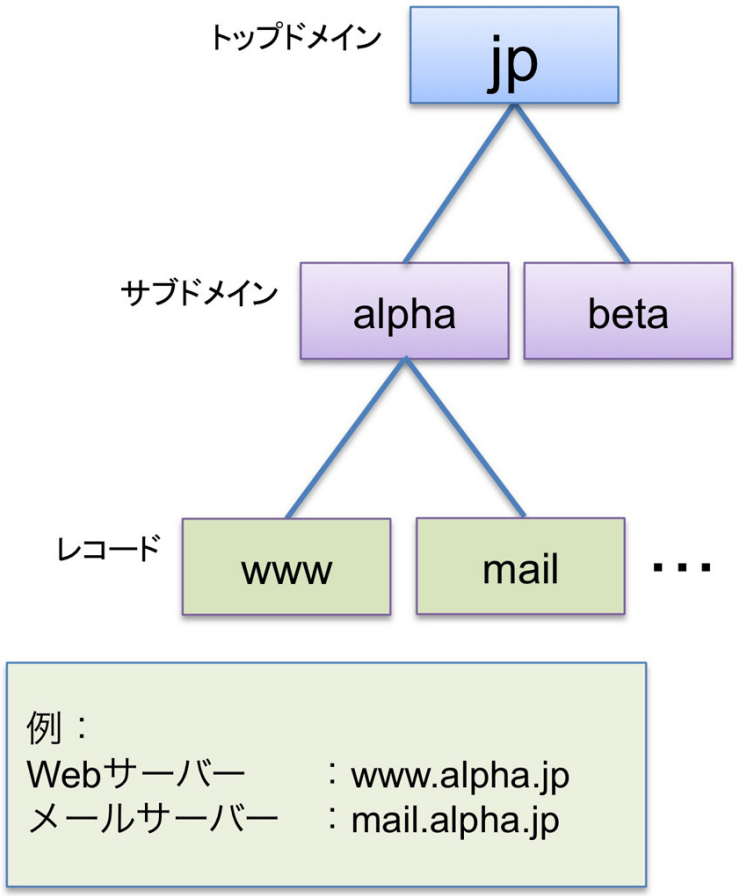


図 4　DNS の構造

3.3.2　ドメイン名の記述

ドメイン名の記述は、右側から記述していきます。FQDN(Fully Qualified Domain Name) であれば一番右にルートドメイン、そしてトップレベルドメインを記述し、さらに左側に各組織毎に割り当てられたドメイン名を記述していきます。各要素の間は「.」（ドット）で区切っていきます。

ドメイン名の記述例

- example.com.
- example.jp.
- example.co.jp.

トップレベルドメイン以降のドメイン名は、ドメイン取得者が独自にドメイン名を決めることができます。上記の例では example の部分が独自のドメイン名にあたります。なお、最後の「.」は、トップレベルドメインを示していますが、省略して表記することがあります。

3.3.3　サブドメイン

記述例のようにドメイン名の左側にさらにドメイン名を記述していくことを「サブドメイン化」と呼びます。たとえば、example.co.jp ドメインをさらに東京と大阪の 2 つに分けて表記したいような場合には、以下の例のように記述します。

- tokyo.example.co.jp.
- osaka.example.co.jp.

サブドメイン化は、上位の（右側の）ドメインを管理している管理者が行います。たとえば、tokyo.example.co.jp ドメインまでのサブドメインの階層は次のようになっています。

1. jp ドメインはルートドメインのサブドメイン
2. co.jp ドメインは jp ドメインのサブドメイン
3. example.co.jp ドメインは co.jp ドメインのサブドメイン
4. tokyo.example.co.jp ドメインは example.co.jp ドメインのサブドメイン

3.3.4　ドメイン名の取得

ドメイン名を取得するということは、上位のドメイン名の管理者にサブドメインを作ってもらい、管理権限を委譲してもらうということになります。短いドメイン名を取得したいのであればトップレベルドメインを管理している管理組織からサブドメイン化してもらうことになりますが、既にドメイン名を取得している管理者からサブドメインの管理権限を委譲してもらうこともできます。

3.4　DNS を使った名前解決

DNS を使って名前を解決する、すなわち名前から IP アドレスを調べる時には、次のような手順で調査が行われます。

1. PC は、自分の組織やプロバイダーの DNS キャッシュサーバーへ問い合わせます。
2. DNS キャッシュサーバーは、ルートサーバーに「jp」を管理する DNS コンテンツサーバーのアドレスを問い合わせます。ルートサーバーは、「jp」を管理する DNS コンテンツサーバーのアドレスを DNS キャッシュサーバーへ返します。
3. DNS キャッシュサーバーは、「jp」を管理する DNS コンテンツサーバーへ、サブドメイン「alpha.jp」を管理する DNS コンテンツサーバーのアドレスを問い合わせます。「alpha.jp」を管理する DNS コンテンツサーバーは、サブドメイン「alpha」を管理する DNS コンテンツサーバーのアドレスを DNS キャッシュサーバーへ返します。
4. DNS キャッシュサーバーは、「alpha.jp」を管理する DNS コンテンツサーバーへ、「www.alpha.jp」のアドレスを問い合わせます。「alpha.jp」を管理する DNS コンテンツサーバーは、「www.alpha.jp」のアドレスを DNS キャッシュサーバーへ返します。
5. DNS キャッシュサーバーは、PC に「www.alpha.jp」のアドレスを返します。

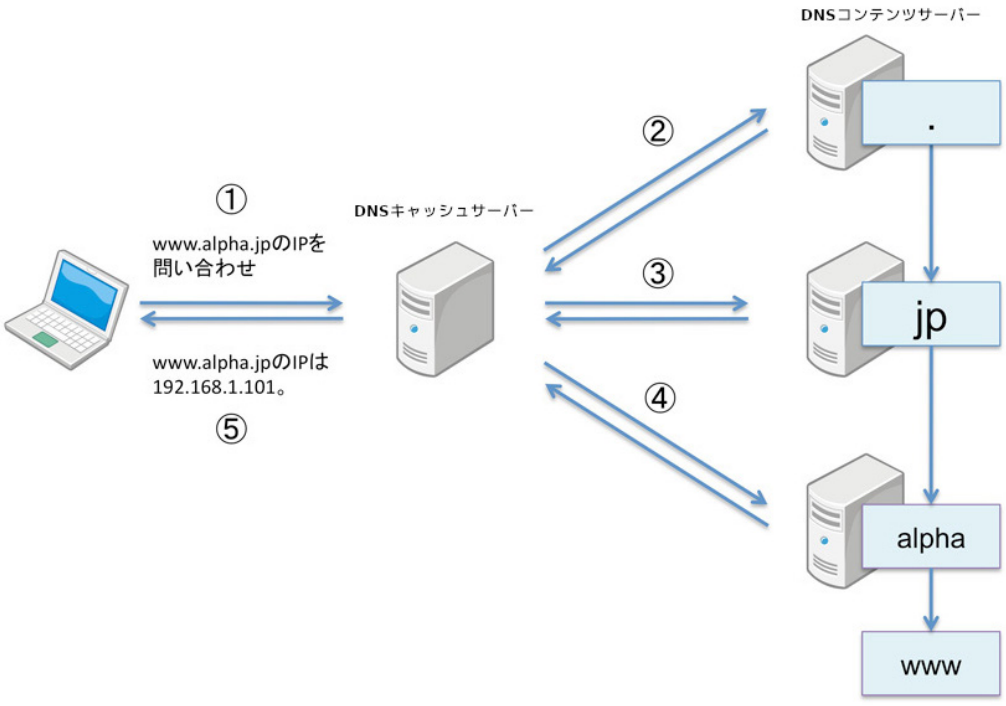


図 5　DNS の名前解決の仕組み

3.5　これから構築する DNS の概略

各自が jp. ドメインのサブドメインを管理する DNS サーバーを作る演習を進めてもらうので、ドメインを管理する次の 3 台以上のマシンがある環境が望ましいです。

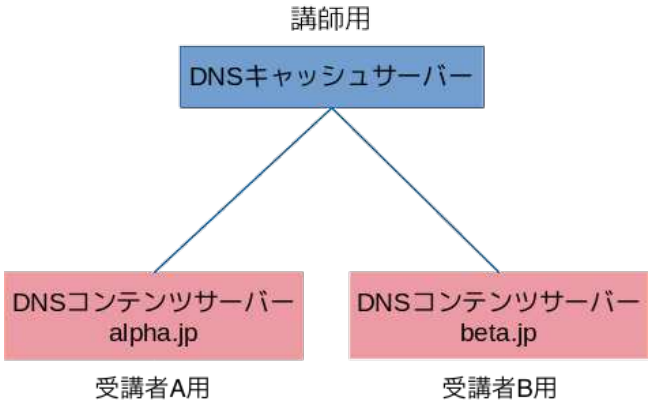


図 6　演習の環境

各マシンには、以下のような役割を割り当てます。

- 講師のマシン
  DNS キャッシュサーバー
- 受講生 A のマシン
  alpha.jp ドメインを受け持つ DNS コンテンツサーバー
- 受講生 B のマシン
  beta.jp ドメインを受け持つ DNS コンテンツサーバー

教室の環境はインターネットに接続されている必要はありません。

以降の章 (特にメール) では DNS サーバーが正しく設定されていることを前提としているので、この章の演習内容が完全に終わっている必要があります。

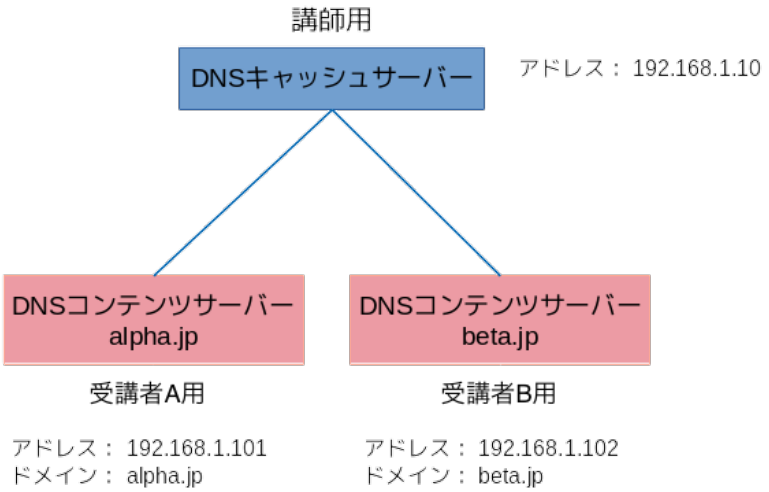


図 7　演習で使うアドレスとドメイン

表 2: 演習で使うドメインと IP アドレス

	講師	受講生 A	受講生 B
ドメイン名		alpha.jp.	beta.jp.
IP アドレス	192.168.1.10	192.168.1.101	192.168.1.102

3.5.1 アドレス解決の流れ

受講生 A のマシン (192.168.1.101) がホスト www.beta.jp を解決するときの動きを追ってみましょう。

Web ブラウザーで Web ページを表示させるとき、DNS キャッシュサーバーのことは特に意識せず Web サイトのアドレスを入力しています。ここでは Web アドレスを入力してリクエストしてページが表示されるまでの流れを例に、DNS がどのように動くのか簡単に説明します。

1. 受講生 A は、受講生 A のマシンの Web ブラウザーにアドレスとして www.beta.jp を入力します
2. Web ブラウザーは、Linux のリゾルバに問い合わせします
3. リゾルバは、/etc/resolv.conf ファイルで指定されている DNS キャッシュサーバー（この場合は講師用サーバー:192.168.1.10）へ問い合わせます
4. 講師のマシンは、講師の DNS キャッシュサーバーを参照します
5. 講師の DNS キャッシュサーバーは、受講生 B の DNS サーバーに問い合わせします
6. 受講生 B の DNS サーバーは、www.beta.jp ホストの IP アドレス（www.beta.jp → 192.168.1.102）を講師のマシンに返します
7. 講師のマシンは、結果を受講生 A のマシンへ返します
8. 受講生 A のマシンは、www.beta.jp に HTTP でアクセスし、Web ページを受け取って表示します

3.6 講師マシンへの DNS キャッシュサーバーの設定

最初に講師マシンを DNS キャッシュサーバーとして用意します。

3.6.1 必要なパッケージを確認

DNS キャッシュサーバーの構築に必要なパッケージは、unbound です。また、DNS サーバーの動作確認には、bind-utils パッケージが必要です。rpm コマンドに-q オプションとパッケージ名を指定し、パッケージがインストールされているか確認します。パッケージがインストールされている場合には、次のように表示されます。

unbound パッケージの確認

```
# rpm -q unbound bind-utils
unbound-1.6.6-1.el7.x86_64
bind-utils-9.9.4-72.el7.x86_64
```

パッケージがインストールされていない場合には、次のように表示されます。

unbound パッケージがない場合

```
# rpm -q unbound bind-utils
パッケージ unbound はインストールされていません。
パッケージ bind-utils はインストールされていません。
```

3.6.2 unbound のインストール

unbound パッケージがインストールされていない時には、yum コマンドでインストールします。インターネットに接続できない環境では、GUI から admin でログインし、インストールメディアが自動マウントされた状態でインストール作業を進めます。

unbound のインストール

```
# yum install unbound bind-utils
読み込んだプラグイン:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.jaist.ac.jp
 * updates: ftp.jaist.ac.jp
 * extras: ftp.jaist.ac.jp
依存性の解決をしています
```

```
--> トランザクションの確認を実行しています。
---> パッケージ bind-utils.x86_64 32:9.9.4-72.el7 を インストール
---> パッケージ unbound.x86_64 0:1.6.6-1.el7 を インストール
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package            アーキテクチャー
                                  バージョン                 リポジトリー   容量
================================================================================
インストール中:
 bind-utils         x86_64         32:9.9.4-72.el7           base         206 k
 unbound            x86_64         1.6.6-1.el7               base         673 k

トランザクションの要約
================================================================================
インストール  2 パッケージ

総ダウンロード容量: 879 k
インストール容量: 2.8 M
Is this ok [y/d/N]: y   ← yを入力
Downloading packages:
(1/2): bind-utils-9.9.4-72.el7.x86_64.rpm                  | 206 kB   00:00
(2/2): unbound-1.6.6-1.el7.x86_64.rpm                      | 673 kB   00:00
--------------------------------------------------------------------------------
合計                                              1.5 MB/s | 879 kB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中          : 32:bind-utils-9.9.4-72.el7.x86_64              1/2
  インストール中          : unbound-1.6.6-1.el7.x86_64                     2/2
  検証中                  : unbound-1.6.6-1.el7.x86_64                     1/2
  検証中                  : 32:bind-utils-9.9.4-72.el7.x86_64              2/2

インストール:
  bind-utils.x86_64 32:9.9.4-72.el7         unbound.x86_64 0:1.6.6-1.el7

完了しました!
```

3.6.3 リクエストを受け付けるIPアドレスの設定

/etc/unbound/local.d/interface.conf ファイルを作成し、DNS キャッシュサーバーがリクエストを受け付ける IP アドレスの設定をします。講師 PC の IP アドレスと、ループバックアドレスを許可します。

/etc/unbound/local.d/interface.conf

```
interface: 192.168.1.10
interface: 127.0.0.1
```

3.6.4 利用可能なクライアントの設定

DNS キャッシュサーバーは、不正に利用されないように必ず利用者を制限して使います。/etc/unbound/local.d/client.conf ファイルを作成し、この DNS キャッシュサーバーを利用できるクライアントの設定を行います。ここでは、192.168.1.0/24 のネットワーク全体を許可します。

/etc/unbound/local.d/client.conf

```
access-control: 192.168.1.0/24 allow
```

3.6.5 受講者用ドメインの設定

演習の環境では、jp ドメインの管理サーバーに alpha.jp、beta.jp という受講者用のドメインを登録することができません。そのため、DNS キャッシュサーバーに特別にスタブゾーンの設定を行います（この設定は、正式にドメイン名を取得した場合には不要な設定です）。設定のために、次のような/etc/unbound/conf.d/stub.conf ファイルを作成します。

/etc/unbound/conf.d/stub.conf

```
stub-zone:
    name: alpha.jp.
    stub-addr: 192.168.1.101
    stub-prime: no
    stub-first: no
stub-zone:
    name: beta.jp.
    stub-addr: 192.168.1.102
    stub-prime: no
    stub-first: no
```

3.6.6 unbound-keygen の起動

設定が終わったら、unbound の制御に使う鍵の生成を行います。次のように、unbound-keygen を起動すると、鍵が自動的に生成されます。

unbound-keygen の起動

```
# systemctl start unbound-keygen
```

3.6.7 unbound の設定確認

設定を行ったら、書式のチェックを行っておきましょう。

unbound の設定の確認

```
# unbound-checkconf
unbound-checkconf: no errors in /etc/unbound/unbound.conf
```

「no errors」と表示されていれば、書式エラーがないということです。エラーがある場合には、次のようにエラーの内容が表示されます。

unbound の設定にエラーがあるとき

```
# unbound-checkconf
[1550047518] unbound-checkconf[27152:0] fatal error: cannot parse interface specified
     as '127.0.0.1.'
```

エラーの内容を確認して、修正します。

3.6.8 ファイアウォールの設定

次に、DNS キャッシュサーバーへの問い合わせができるようにファイアウォールのサービス許可設定を行います。

DNS アクセスの許可

```
# firewall-cmd --add-service=dns
```

さらに、設定を保存しておきます。

Firewall ルールの保存

```
# firewall-cmd --runtime-to-permanent
```

3.6.9 unbound の起動と確認

設定が終わったら、unbound を起動しましょう。systemctl で、unbound ユニットを起動します.

unbound の起動

```
# systemctl start unbound
```

起動ができたら、念のため確認します。

unbound の起動確認

```
# systemctl status unbound
● unbound.service - Unbound recursive Domain Name Server
   Loaded: loaded (/usr/lib/systemd/system/unbound.service; disabled; vendor preset:
       disabled)
   Active: active (running) since 水 2019-02-13 17:46:55 JST; 4s ago
  Process: 28377 ExecStartPre=/usr/sbin/unbound-anchor -a /var/lib/unbound/root.key -c
      /etc/unbound/icannbundle.pem (code=exited, status=0/SUCCESS)
  Process: 28373 ExecStartPre=/usr/sbin/unbound-checkconf (code=exited,
      status=0/SUCCESS)
 Main PID: 28383 (unbound)
    Tasks: 4
   CGroup: /system.slice/unbound.service
           └─28383 /usr/sbin/unbound -d

 2月 13 17:46:55 host0 systemd[1]: Starting Unbound recursive Domain Nam.....
 2月 13 17:46:55 host0 unbound-checkconf[28373]: unbound-checkconf: no err...
 2月 13 17:46:55 host0 systemd[1]: Started Unbound recursive Domain Name...r.
 2月 13 17:46:55 host0 unbound[28383]: [28383:0] notice: init module 0: i...d
 2月 13 17:46:55 host0 unbound[28383]: [28383:0] notice: init module 1: v...r
 2月 13 17:46:55 host0 unbound[28383]: [28383:0] notice: init module 2: i...r
 2月 13 17:46:55 host0 unbound[28383]: [28383:0] info: start of service (....
Hint: Some lines were ellipsized, use -l to show in full.
```

Active の欄に「active (running)」と表示されていることを確認します。また、下の方にはログが表示されます。ここでも、「info: start of service」と表示されていて、unbound のサービスが起動していることが確認できます。

unbound の再起動

既に unbound を起動している状態で、設定変更などをした場合には、次のように unbound を再起動します。

unbound の再起動

```
# systemctl restart unbound
```

3.6.10 自動起動の設定

Linux 起動時に unbound が必ず起動されるように設定しておきましょう。自動起動になっているかは、次のように確認できます。

unbound の自動起動の確認

```
# systemctl is-enabled unbound
disabled
```

自動起動の設定がされている場合には、「enabled」と表示されます。この例のように「disabled」と表示される場合には、自動起動設定が行われていません。次のようにして、自動起動設定を行います。

unbound の自動起動の設定

```
# systemctl enable unbound
Created symlink from /etc/systemd/system/multi-user.target.wants/unbound.service to
     /usr/lib/systemd/system/unbound.service.
```

3.6.11 名前解決の確認

講師 PC がインターネットと通信できる環境の場合には、この時点でインターネットの様々なサイトの名前解決ができるはずです。まずは、それを確認しておきましょう。

名前解決の確認

```
# host www.yahoo.co.jp 192.168.1.10
Using domain server:
Name: 192.168.1.10
Address: 192.168.1.10#53
Aliases:

www.yahoo.co.jp is an alias for edge12.g.yimg.jp.
edge12.g.yimg.jp has address 182.22.25.252  ← IPアドレスが表示される
```

3.7 受講者マシンへの DNS コンテンツサーバーの設定

受講者マシンは、DNS コンテンツサーバーとして設定します。DNS コンテンツサーバーのソフトウェアとして BIND をインストールして設定します。

3.7.1 chroot 機能を利用した BIND のセキュリティ

chroot 機能はプログラムに対して特定のディレクトリ以外にはアクセスできないようにするための機能です。

chroot 機能を使って BIND を実行すると、bind プロセスは/var/named/chroot ディレクトリを/（ルート）ディレクトリとして動作します。たとえば、bind プロセスが/etc ディレクトリにアクセスしても、実際にアクセスされるのは/var/named/chroot/etc ディレクトリになります。

DNS というサービスを提供している関係上、BIND はインターネット上の数多くのサーバーで実行されており、セキュリティの攻撃を受けやすくなっています。万が一、BIND がセキュリティ攻撃を受けて乗っ取られてしまったとしても、chroot 機能のおかげで bind プロセスがアクセスできるディレクトリを限定することができるので、システムのその他のファイルへのアクセスを妨げ、被害を最小限に食い止めることができます。

CentOS7 では、シンボリックリンクやマウントなどの Linux の機能を使って、chroot 機能を使った場合でも、ほとんど管理方法が変わらないように工夫されています。そのため、本書でも chroot 機能を有効にします。

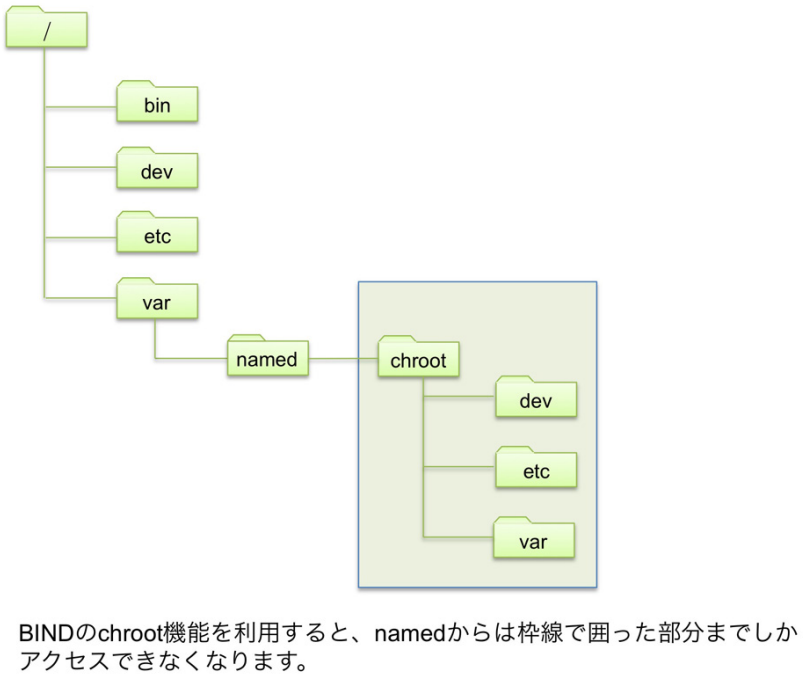


図 8　chroot 利用時のディレクトリイメージ

3.7.2　BIND パッケージの確認

DNS サーバーの構築に必要なパッケージを確認します。DNS の機能を提供するプログラムとして BIND があり、bind パッケージと bind-chroot パッケージが必要です。また、動作確認をするには bind-utils パッケージも必要です。rpm コマンドに-q オプションとパッケージ名を指定し、2 つのパッケージがインストールされているか確認します。パッケージがインストールされている場合は次のように表示されます。

bind パッケージの確認

```
# rpm -q bind bind-chroot bind-utils
bind-9.9.4-72.el7.x86_64
bind-chroot-9.9.4-72.el7.x86_64
bind-utils-9.9.4-72.el7.x86_64
```

パッケージがインストールされていない場合は次のように表示されます。

bind パッケージがない場合

```
# rpm -q bind bind-chroot
パッケージ bind はインストールされていません。
パッケージ bind-chroot はインストールされていません。
パッケージ bind-utils はインストールされていません。
```

3.7.3　BIND のインストール

DNS サーバーの構築に必要なパッケージがインストールされていないときは、yum コマンドでインストールをします。インターネットに接続できない環境では、GUI から admin でログインし、インストールメディアが自動マウントされた状態でインストール作業を進めます。

bind のインストール

```
# yum install bind bind-chroot bind-utils
```

```
読み込んだプラグイン:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.jaist.ac.jp
 * updates: ftp.jaist.ac.jp
 * extras: ftp.jaist.ac.jp
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> パッケージ bind.x86_64 32:9.9.4-72.el7 を インストール
---> パッケージ bind-chroot.x86_64 32:9.9.4-72.el7 を インストール
---> パッケージ bind-utils.x86_64 32:9.9.4-72.el7 を インストール
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package             アーキテクチャー
                                  バージョン                リポジトリー   容量
================================================================================
インストール中:
 bind                x86_64         32:9.9.4-72.el7          base          1.8 M
 bind-chroot         x86_64         32:9.9.4-72.el7          base           88 k
 bind-utils          x86_64         32:9.9.4-72.el7          base          206 k

トランザクションの要約
================================================================================
インストール  3 パッケージ

総ダウンロード容量: 2.1 M
インストール容量: 5.0 M
Is this ok [y/d/N]: y   ← yを入力
Downloading packages:
(1/3): bind-chroot-9.9.4-72.el7.x86_64.rpm                 |  88 kB   00:00
(2/3): bind-utils-9.9.4-72.el7.x86_64.rpm                  | 206 kB   00:00
(3/3): bind-9.9.4-72.el7.x86_64.rpm                        | 1.8 MB   00:00
--------------------------------------------------------------------------------
合計                                              3.4 MB/s | 2.1 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中          : 32:bind-9.9.4-72.el7.x86_64                     1/3
  インストール中          : 32:bind-chroot-9.9.4-72.el7.x86_64              2/3
  インストール中          : 32:bind-utils-9.9.4-72.el7.x86_64               3/3
  検証中                  : 32:bind-9.9.4-72.el7.x86_64                     1/3
  検証中                  : 32:bind-chroot-9.9.4-72.el7.x86_64              2/3
  検証中                  : 32:bind-utils-9.9.4-72.el7.x86_64               3/3

インストール:
  bind.x86_64 32:9.9.4-72.el7            bind-chroot.x86_64 32:9.9.4-72.el7
  bind-utils.x86_64 32:9.9.4-72.el7

完了しました!
```

3.7.4 ゾーンを設定する流れ

ゾーンを追加するために必要な作業は次となります。

- named.conf ファイルにゾーンを追加
- ゾーンファイルを記述

ゾーンを DNS サーバーである BIND で取り扱うために、BIND の基本的な設定ファイルである/etc/named.conf ファイルがあります。/etc/named.conf に基本的な設定とゾーンの定義を追加したら、ゾーンの詳細を定義するゾーンファイルを/var/named ディレクトリに作ります。

3.7.5 /etc/named.conf の基本設定

まず最初に、/etc/named.conf ファイルに DNS コンテンツサーバーとして動作する場合の、基本設定を行います。

/etc/named.conf の編集

```
# vi /etc/named.conf
```

受け付ける IP アドレスの設定（/etc/named.conf）

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.1.101; };    ←  修正
    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";
    recursing-file  "/var/named/data/named.recursing";
    secroots-file   "/var/named/data/named.secroots";
    allow-query     { any; };   ← 修正

    recursion no;   ←  修正

    dnssec-enable yes;
    dnssec-validation yes;

    /* Path to ISC DLV key */
    bindkeys-file "/etc/named.iscdlv.key";

    managed-keys-directory "/var/named/dynamic";

    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
    type hint;
    file "named.ca";
```

```
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

zone "alpha.jp" IN {     ←  正引きゾーンの設定を追加
    type master;
    file "alpha.jp.zone";
    allow-update { none; };
};
```

問い合わせを受け付けるアドレスの設定

デフォルトの named.conf ファイルは、127.0.0.1(ローカルループバックインターフェース) への問い合わせにしか返答しない設定なので、外部からの問い合わせを受けられるように、「192.168.1.101;」を listen-on に追加します。

問い合わせを許可するアドレスの設定

allow-query にはデフォルトでは「localhost;」と設定されていて、ローカルからしか DNS 問い合わせができないようになっています。DNS コンテンツサーバーは、インターネット上のすべての人から参照できなければならないので、この設定を「any」に変更します。

DNS コンテンツサーバーとしての設定

DNS コンテンツサーバーでは、recursion（再帰問合せ）を禁止にしておく必要があります。そのため、recursion に「no」を設定します。

正引きゾーンの設定

alpha.jp の正引きゾーンの設定を追加します。指定している「alpha.jp.zone」という名前のファイルにゾーンの設定を行います。

3.7.6 ゾーンファイルの作成

named.conf で定義したゾーンの内容を記述するゾーンファイルの作成を行います。

ゾーンファイルの準備

ゾーンファイルのお手本となる/var/named/named.empty ファイルをコピーします。

ゾーンファイルのコピー

```
# cd /var/named
# cp -p named.empty alpha.jp.zone
# ls -l alpha.jp.zone
-rw-r-----. 1 root named 152 12月 15  2009 alpha.jp.zone
```

ゾーンファイルの修正

コピーした/var/named/alpha.jp.zone ファイルを修正します。

ゾーンファイルの修正

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

alpha.jp.zone

```
$TTL 3H
$ORIGIN alpha.jp.
```

```
@       IN SOA  host1 root (
                                    2019021401      ; serial
                                       1D      ; refresh
                                       1H      ; retry
                                       1W      ; expire
                                       3H )    ; minimum
        NS      host1.alpha.jp.
        MX 10   mail.alpha.jp.

host1   A       192.168.1.101
www     A       192.168.1.101
mail    A       192.168.1.101
vhost1  A       192.168.1.101
vhost2  A       192.168.1.101
```

@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、このゾーンの管理ポリシーについて設定します。シリアルナンバー (serial) は、西暦 (4 桁の年) と月日 (2 桁ずつ) の後に 01 から 99 までの数字 (2 桁) が付いた 10 桁の数字で指定します。SOA レコードの先頭には@がありますが、これは$ORIGIN で指定したゾーン（ここでは alpha.jp.）に置き換えられます。

MX レコードは受講生ドメインのメールサーバーを定義します。

NS レコードや MX レコードの定義では、右側に FQDN を入れるので、最後に必ず「.」を付けてください。また、先頭が空白になっていますが、これは前の行と同じ対象（この場合には@）が省略されていることを示しています。

A レコードで名前と IP アドレスの対応を定義する箇所は、左側にホスト名、右側に IP アドレスが入ります。受講生マシンのホスト名である host1 や、以降の章で使うサーバーの名前である www や mail、vhost1、vhost2 と IP アドレスへの対応を記述しました。最後に「.」が付かない名前には、$ORIGIN で定義しているゾーン名 (ここでは alpha.jp.) が自動的に追加されます。

ゾーンファイルの書式確認

ゾーンファイルを編集時、よくあるミスとしては、括弧の不足、セミコロンの不足などがあります。編集後、BIND を起動する前に編集したゾーンファイルに間違いがないかよく確認しましょう。

zone ファイルの確認

```
# named-checkzone alpha.jp. /var/named/alpha.jp.zone
zone alpha.jp/IN: loaded serial 2019021401
OK
```

named-checkzone の引数は、$ORIGIN に指定したドメイン名と、ゾーンファイル名です。書式に問題がなければ、この例のように設定したシリアルナンバーが表示され、OK と表示されます。設定が間違っている場合には、次のように問題のある行番号が表示されます。

zone ファイルの書式が不正な場合

```
# named-checkzone alpha.jp. /var/named/alpha.jp.zone
alpha.jp.zone:9: unknown RR type 'host1.alpha.jp.'
zone alpha.jp/IN: loading from master file alpha.jp.zone failed: unknown class/type
zone alpha.jp/IN: not loaded due to errors.
```

3.7.7 設定ファイルの書式確認と注意点

/etc/named.conf ファイルでも、括弧の不足やセミコロンの不足などは良くあるミスです。一通りの設定ができたら、/etc/named.conf の初期確認をしておきましょう。次のように、named-checkconf を実行します。

/etc/named.conf の確認

```
# named-checkconf
```

この例のように、何も表示されなければ書式に問題がないということです。問題がある場合には、次のように問題がありそうな行番号が表示されます。

/etc/named.conf の書式が不正な場合

```
# named-checkconf
/etc/named.conf:68: missing ';' before end of file
```

listen-on port の{〜}にアドレスを追加するときに、IP アドレスの後にセミコロンを入れ忘れたり、allow-query { any; }; のように、{ }の中と外にセミコロンを入れ忘れたりといったミスが起こりがちです。エラー表示を見ながら、こうした問題を取り除きましょう。

3.7.8 ファイアウォールの設定

次に、DNS コンテンツサーバーへの問い合わせができるようにファイアウォールのサービス許可設定を行います。

DNS アクセスの許可

```
# firewall-cmd --add-service=dns
```

さらに、設定を保存しておきます。

Firewall ルールの保存

```
# firewall-cmd --runtime-to-permanent
```

3.7.9 BIND の起動と確認

各自のドメインを定義したら BIND を起動してみましょう。BIND の起動は、systemctl コマンドで named-chroot を使います。

BIND の起動

```
# systemctl start named-chroot
```

起動ができたら、念のため確認します。

BIND の起動確認

```
# systemctl status named-chroot
 named-chroot.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named-chroot.service; disabled; vendor
        preset: disabled)
   Active: active (running) since 木 2019-02-14 16:00:12 JST; 5s ago
  Process: 32681 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} -t
       /var/named/chroot $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 32678 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ];
       then /usr/sbin/named-checkconf -t /var/named/chroot -z "$NAMEDCONF"; else echo
       "Checking of zone files is disabled"; fi (code=exited, status=0/SUCCESS)
 Main PID: 32683 (named)
    Tasks: 4
   CGroup: /system.slice/named-chroot.service
           └─32683 /usr/sbin/named -u named -c /etc/named.conf -t /var/named/...

 2月 14 16:00:12 centos7 named[32683]: managed-keys-zone: loaded serial 57
 2月 14 16:00:12 centos7 named[32683]: zone 0.in-addr.arpa/IN: loaded serial 0
 2月 14 16:00:12 centos7 named[32683]: zone alpha.jp/IN: loaded serial 2019...1
 2月 14 16:00:12 centos7 named[32683]: zone 1.0.0.127.in-addr.arpa/IN: load...0
 2月 14 16:00:12 centos7 named[32683]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0...0
 2月 14 16:00:12 centos7 named[32683]: zone localhost/IN: loaded serial 0
```

```
 2月 14 16:00:12 centos7 named[32683]: zone localhost.localdomain/IN: loade...0
 2月 14 16:00:12 centos7 named[32683]: all zones loaded
 2月 14 16:00:12 centos7 named[32683]: running
 2月 14 16:00:12 centos7 systemd[1]: Started Berkeley Internet Name Domain...).
Hint: Some lines were ellipsized, use -l to show in full.
```

Active の欄に「active (running)」と表示されていることを確認します。また、下の方にはログが表示されます。ここでも、「Started Berkeley Internet Name Domain...」と表示されていて、BIND のサービスが起動していることが確認できます。

BIND の再起動

既に BIND を起動している状態で、設定変更などをした場合には、次のように BIND を再起動します。

BIND の再起動

```
# systemctl restart named-chroot
```

3.7.10 自動起動の設定

Linux 起動時に BIND が必ず起動されるように設定しておきましょう。自動起動になっているかは、次のように確認できます。

BIND の自動起動の確認

```
# systemctl is-enabled named-chroot
disabled
```

自動起動の設定がされている場合には、「enabled」と表示されます。この例のように「disabled」と表示される場合には、自動起動設定が行われていません。次のようにして、自動起動設定を行います。

BIND の自動起動の設定

```
# systemctl enable named-chroot
Created symlink from /etc/systemd/system/multi-user.target.wants/named-chroot.service
     to /usr/lib/systemd/system/named-chroot.service.
```

3.7.11 名前解決の確認

BIND が起動したら、名前解決が正常に行われるかを確認します。名前解決の確認には、host コマンドと dig コマンドが使用できます。

host コマンドで名前を確認

host コマンドで名前から IP アドレスを確認します。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ーのアドレスです。さきほど設定した DNS サーバーの IP アドレスを指定します。

host コマンドでの確認

```
# host host1.alpha.jp 192.168.1.101
Name: 192.168.1.101
Address: 192.168.1.101#53
Aliases:

host1.alpha.jp has address 192.168.1.101
# host www.alpha.jp 192.168.1.101
Using domain server:
Name: 192.168.1.101
Address: 192.168.1.101#53
Aliases:
```

```
www.alpha.jp has address 192.168.1.101
# host mail.alpha.jp 192.168.1.101
Using domain server:
Name: 192.168.1.101
Address: 192.168.1.101#53
Aliases:

mail.alpha.jp has address 192.168.1.101
```

-t オプションで ns を指定すると、ドメインに登録されている NS レコード (ネームサーバーの情報) が表示されます。

host コマンドでの NS レコードの動作確認

```
# host -t ns alpha.jp 192.168.1.100
Name: 192.168.1.101
Address: 192.168.1.101#53
Aliases:

alpha.jp name server host1.alpha.jp.
```

-t オプションで mx を指定するとドメインに登録されている MX レコード (メールサーバーの情報) が表示されます。

host コマンドでの MX レコードの動作確認

```
# host -t mx alpha.jp
Name: 192.168.1.101
Address: 192.168.1.101#53
Aliases:

alpha.jp mail is handled by 10 mail.alpha.jp.
```

dig コマンドでドメインを確認

dig コマンドでは、ゾーン全体の情報も確認できます。次のように、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全ての情報が表示されます。問い合わせをするサーバーは、[@をつけて指定します]。

dig コマンドでの確認

```
# dig alpha.jp axfr @192.168.1.101

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7_6 <<>> alpha.jp axfr @192.168.1.101
;; global options: +cmd
alpha.jp.       10800   IN  SOA host1.alpha.jp. root.alpha.jp. 2019021401 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   192.168.1.101
mail.alpha.jp.      10800   IN  A   192.168.1.101
vhost1.alpha.jp.    10800   IN  A   192.168.1.101
vhost2.alpha.jp.    10800   IN  A   192.168.1.101
www.alpha.jp.       10800   IN  A   192.168.1.101
alpha.jp.       10800   IN  SOA host1.alpha.jp. root.alpha.jp. 2019021401 86400 3600
    604800 10800
;; Query time: 6 msec
;; SERVER: 192.168.1.101#53(192.168.1.101)
```

```
;; WHEN: 木  2月 14 16:28:03 JST 2019
;; XFR size: 9 records (messages 1, bytes 242)
```

3.8 リゾルバの変更

講師マシンの DNS キャッシュサーバーの設定と、受講生マシンの DNS コンテンツサーバーの設定が完了したので、受講生マシンから講師マシンを経由すればすべての下位ドメインを問合せできます。講師マシンがインターネットと通信できる環境の場合には、すべてのドメインを問合せすることができます。受講生マシンと講師マシンの DNS サーバーの設定を変更しておきましょう。

GNOME のデスクトップのアプリケーションメニューから「システムツール」→「設定」を選択します。表示された設定画面の左側のメニューから「ネットワーク」を選択します。

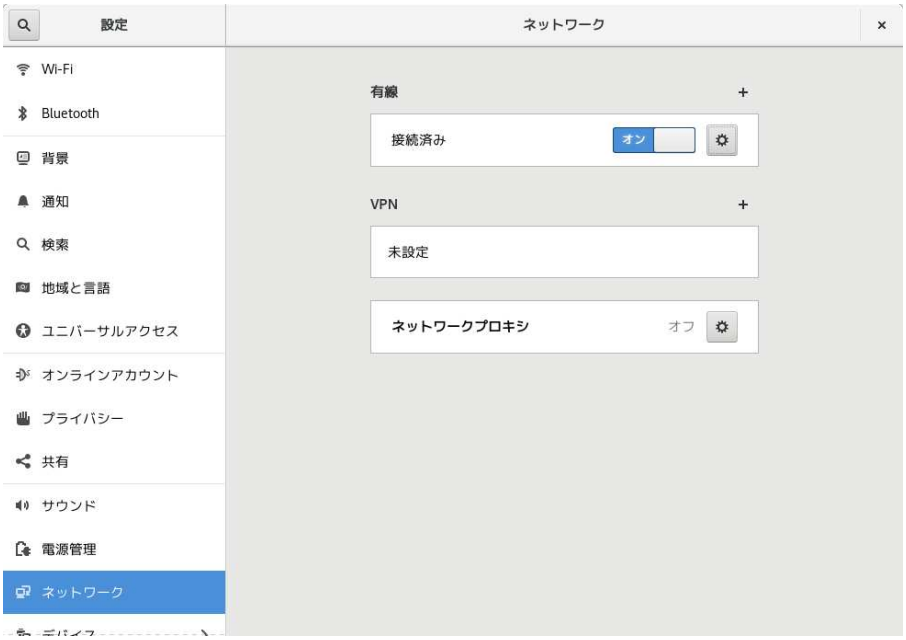


図 9 設定画面

「有線」の欄にある歯車のボタンをクリックすると、接続プロファイルの設定画面が表示されます。「IPv4」のタブをクリックすると、次のような画面になります。

DNS サーバーのアドレスを講師マシンの IP アドレス「192.168.1.10」に変更します。変更したら、「適用」ボタンを押して元の画面に戻ります。「有線」の項目にあるスイッチを、一旦「オフ」に変えます。再度、「オン」に変えると DNS 設定が変更されます。

3.8.1 名前解決の確認

リゾルバの変更ができたら、DNS サーバーを指定しなくても DNS の問い合わせができるようになります。host コマンドで、自分のドメインの NS レコードや MX レコードを問い合わせてみてください。host コマンドでは、-t オプションを使うことで、問い合わせるレコードを指定することができます。

host コマンドでの動作確認

```
# host -t ns alpha.jp
alpha.jp name server host1.alpha.jp.
# host -t mx alpha.jp
alpha.jp mail is handled by 10 mail.alpha.jp.
```

また、他の受講者マシンの設定も問合せができることを確認してみましょう。

host コマンドでの動作確認

図 10　ネットワーク詳細設定画面

```
# host -t ns beta.jp
beta.jp name server host1.beta.jp.
# host -t mx beta.jp
beta.jp mail is handled by 10 mail.beta.jp.
```

3.9　DNS コンテンツサーバーのセキュリティ

動作確認のため、dig の axfr を使ってゾーンを転送する例を紹介しました。しかし、インターネット上の見知らぬサイトに、すべてのゾーンデータを教えるのは懸命ではありません。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ます。zone ステートに allow-transfer オプションが記述された場合、options ステートメントの設定を上書きできます。下記のように設定することで、ゾーン転送を localhost とネットワークアドレス 192.168.1.0 以下にある端末からのみ許可できます。

/etc/named.conf への allow-transfer の設定

```
options {
 (略)
        allow-transfer  { localhost; 192.168.1.0/24; };
 (略)
};
```

4 Web サーバーの構築

ホームページや Web システムを公開するための Web サービスを設定します。基本としては、アクセス制限をかける設定と動作を確認します。応用としては、サーバーとして広く使われているバーチャルホスト機能にも触れてもらいます。

4.1 用語集

HTML(HyperText Markup Language)

Web サーバー用のドキュメントを書くためのタグを使って文章を構造的に記述できるマークアップ言語です。他ドキュメントへのハイパーリンクを書いたり、画像利用したり、リストや表などの高度な表現も可能です。

HTTP(HyperText Transfer Protocol)

Web ブラウザーと Web サーバーの間で HTML などのコンテンツ (データ) 送受信に使われる通信手順です。ファイルのリクエスト (要求) とファイルのレスポンス (返送) が組でセッションになります。

Apache Web サーバー

世界中でもっとも使われている Web サーバーであり、大規模な商用サイトから自宅サーバーまで幅広く利用されています。Apache ソフトウェア財団の Apache HTTP サーバープロジェクトで行われている、オープンソースソフトウェアです。

セッション

通信の接続を確立してから切断するまでを一つのセッションといいます。

ディレクティブ

Apache の設定ファイルで Apache の動作を設定する項目名です。

BASIC 認証

ユーザー名とパスワードを使い、HTTP で定義された認証方式です。ユーザー名とパスワードの組を: でつなぎ、Base64 でエンコードして送信します。盗聴や改竄が簡単であるという欠点がありますが、ほぼ全ての Web サーバーおよび Web ブラウザーが対応しており、広く使われています。

DIGEST 認証

ユーザー名とパスワードを使う認証方式です。ユーザー名とパスワードを MD5 でハッシュ化して送るため、盗聴や改竄を防げるため推奨される認証方式ですが、Web ブラウザーや端末によって対応していないものがあるために注意が必要です。

URL(Uniform Resource Locator)

インターネット上のリソースを指定するための記述方法で、ホームページのアドレスやメールのアドレスなどを指定できます。リソースを特定するスキーム名とアドレスを://でつないで書きます。

4.2　Web サーバーの仕組み

Web システムとは、インターネット環境で最も代表的なクライアントサーバーシステムで、クライアントの Web ブラウザーと Web サーバーから構成されます。Web サーバーは要求されたファイルを Web クライアントに提供し、クライアントは受け取ったファイルを表示します。提供される情報はテキストから画像や動画と幅広く、クライアントが対応しているデータならば広く扱えます。Web システムの文章データとしては XML ベースの規格である XHTML や従来の HTML などが一般的に使われています。Web サーバーとして Apache が使われている割合はかなり高いでしょう。

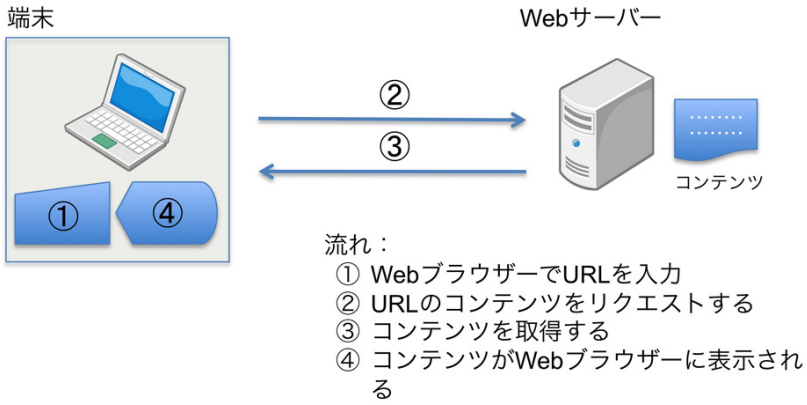


図 11　Web サーバーの動作の仕組み

Web サーバーと Web ブラウザーは HTTP 形式の通信手順でデータを転送します。転送されるデータはテキストや画像などのファイルで、HTML 形式 (フォーマット) のファイルが標準形式として使われています。HTML 形式のファイル内には、他のページやファイル、他の Web サーバーへのリンクが含まれていて、HTML 形式のデータはクモの巣 (web) 状にデータがつながるのも特徴です。

4.3　これから構築する Web サーバーの概略

各自が受講生マシンで Web サーバーを起ち上げ、Web ブラウザーから隣の受講生マシンの Web サーバーへのアクセスを試します。名前の解決には 3 章で構築した講師の DNS サーバーと隣の受講生マシンの DNS サーバーを利用します。Web システムはクライアントとサーバーが同一のマシンに混在しても、2 台以上で構成されても動作に違いはありませんが、テスト確認を曖昧としないために 2 台ペアでの演習を行いましょう。まず、実習を始める前に、Web サーバーのデフォルト設定に問題がなく、Web サーバーとして着実に動くかを確認します。必要なパッケージが導入されているか、設定ファイルが正しく設定されているかなどを確認するため、Web サーバーを起動して Web ブラウザーでアクセスしたり、Apache のログファイルを確認するなどして、Web サーバーが正常に起動していることを確認します。

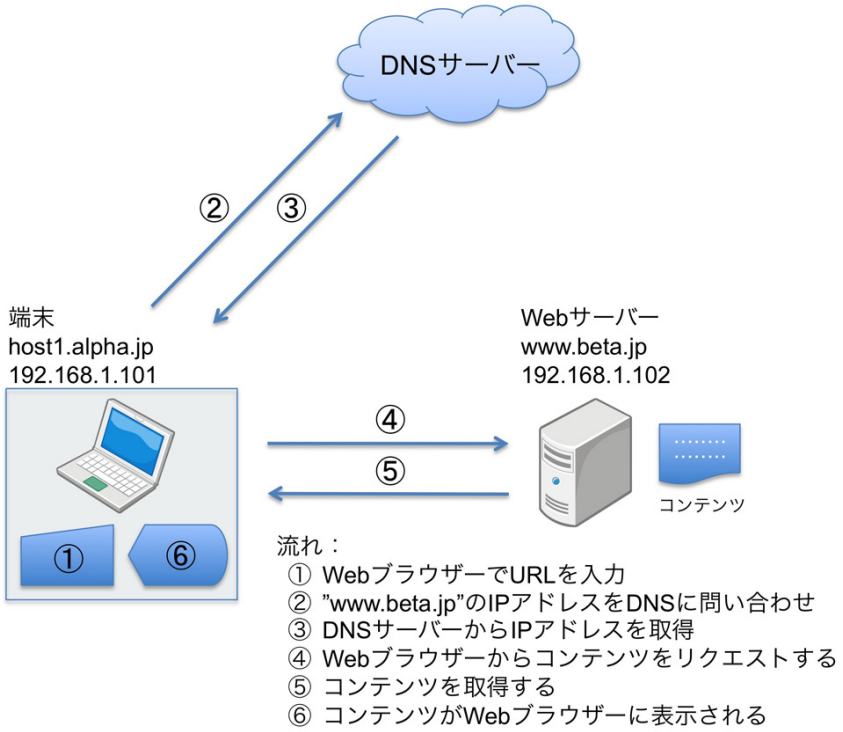


図 12　実習の環境

表 3: 講師

項目	設定値
ホスト名	host0.jp
IP アドレス	192.168.1.10

表 4: 受講生 A

項目	設定値
役割	Web サーバー
ホスト名	host1.alpha.jp
IP アドレス	192.168.1.101

表 5: 受講生 B

項目	設定値
役割	Web ブラウザー
ホスト名	host2.beta.jp
IP アドレス	192.168.1.102

4.4 Web サーバーの設定

Web サーバーの動作に必要なパッケージおよび設定ファイルを確認し、サービスを起動して動作を確認します。

4.4.1 必要なパッケージを確認

rpm コマンドで必要なパッケージがインストールされているかを確認します。

httpd パッケージの確認

```
# rpm -q httpd
httpd-2.4.6-88.el7.centos.x86_64
```

パッケージがインストールされていない場合は次のように表示されます。

httpd パッケージがない場合

```
# rpm -q httpd
パッケージ httpd はインストールされていません。
```

4.4.2 必要なパッケージをインストール

Web サーバーの構築に必要なパッケージがインストールされていないときは、yum コマンドでインストールをします。インターネットに接続できない環境では、GUI から admin でログインし、インストールメディアが自動マウントされた状態でインストール作業を進めます。

httpd のインストール

```
# yum install httpd
読み込んだプラグイン:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.jaist.ac.jp
 * extras: ftp.jaist.ac.jp
 * updates: ftp.jaist.ac.jp
base                                                     | 3.6 kB     00:00
extras                                                   | 3.4 kB     00:00
updates                                                  | 3.4 kB     00:00
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> パッケージ httpd.x86_64 0:2.4.6-88.el7.centos を インストール
--> 依存性の処理をしています: httpd-tools = 2.4.6-88.el7.centos のパッケージ:
     httpd-2.4.6-88.el7.centos.x86_64
--> 依存性の処理をしています: /etc/mime.types のパッケージ:
     httpd-2.4.6-88.el7.centos.x86_64
--> トランザクションの確認を実行しています。
---> パッケージ httpd-tools.x86_64 0:2.4.6-88.el7.centos を インストール
---> パッケージ mailcap.noarch 0:2.1.41-2.el7 を インストール
--> 依存性解決を終了しました。
```

```
依存性を解決しました

================================================================================
 Package            アーキテクチャー
                                  バージョン                    リポジトリー
                                                                          容量
================================================================================
インストール中:
 httpd              x86_64        2.4.6-88.el7.centos         base        2.7 M
依存性関連でのインストールをします:
 httpd-tools        x86_64        2.4.6-88.el7.centos         base         90 k
 mailcap            noarch        2.1.41-2.el7                base         31 k

トランザクションの要約
================================================================================
インストール  1 パッケージ (+2 個の依存関係のパッケージ)

総ダウンロード容量: 2.8 M
インストール容量: 9.6 M
Is this ok [y/d/N]: y   ← 確認してyを入力
Downloading packages:
(1/3): httpd-tools-2.4.6-88.el7.centos.x86_64.rpm          |  90 kB   00:00
(2/3): httpd-2.4.6-88.el7.centos.x86_64.rpm                | 2.7 MB   00:00
(3/3): mailcap-2.1.41-2.el7.noarch.rpm                     |  31 kB   00:00
--------------------------------------------------------------------------------
合計                                              2.6 MB/s | 2.8 MB  00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中          : mailcap-2.1.41-2.el7.noarch                    1/3
  インストール中          : httpd-tools-2.4.6-88.el7.centos.x86_64         2/3
  インストール中          : httpd-2.4.6-88.el7.centos.x86_64               3/3
  検証中                  : httpd-tools-2.4.6-88.el7.centos.x86_64         1/3
  検証中                  : mailcap-2.1.41-2.el7.noarch                    2/3
  検証中                  : httpd-2.4.6-88.el7.centos.x86_64               3/3

インストール:
  httpd.x86_64 0:2.4.6-88.el7.centos

依存性関連をインストールしました:
  httpd-tools.x86_64 0:2.4.6-88.el7.centos     mailcap.noarch 0:2.1.41-2.el7

完了しました!
```

4.4.3 設定ファイルの修正

Apache の設定ファイルは/etc/httpd/conf/httpd.conf ファイルと/etc/httpd/conf.d ディレクトリにある拡張子が conf のファイルです。設定ファイルで、先頭が#の行はコメント、ディレクティブの後にはスペースやタブで区切った設定内容の文字列を書きます。httpd.conf ファイルを見てみてください。

httpd のインストール

```
# cat /etc/httpd/conf/httpd.conf
 (略)
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
Include conf.modules.d/*.conf
 (略)
```

ディレクティブでデフォルトの設定と同じ内容の多くはコメントになっています。いくつかの基本的なディレクティブを次の表にまとめました。このデフォルトで設定されているディレクティブは最小限必要な項目なので、これらのディレクティブの設定値があれば (ディレクトリなどがあれば)、Apache は動作します。

表 6: 基本的なディレクティブ

ディレクティブ名	内容	設定例
Listen	サービスを受けるポート番号	80
DocumentRoot	公開するディレクトリ	/var/www/html
ServerName	サーバーの名前	www.alpha.jp
DirectoryIndex	/をアクセスした時にアクセスするファイル	index.html
AddDefaultCharset	レスポンスに使われる文字コード	off...なし、UTF-8

4.4.4 テストファイルの作成

Web サーバーの機能を一文に要約すると、クライアントが要求するドキュメントや画像などのデータを転送する働きをするのが Web サーバーです。Web サーバーが動作するかチェックするためには/var/www/html ディレクトリに index.html ファイルを作成します。

テストファイルの作成

```
# echo -n 'This is TEST Page on ' > /var/www/html/index.html
# hostname >> /var/www/html/index.html
# cat /var/www/html/index.html
This is TEST Page on host1.alpha.jp
```

4.4.5 ファイアウォールの設定

Apache の起動の前に、Web サーバーへの問い合わせができるようにファイアウォールのサービス許可設定を行います。

HTTP アクセスの許可

```
# firewall-cmd --add-service=http
```

さらに、設定を保存しておきます。

Firewall ルールの保存

```
# firewall-cmd --runtime-to-permanent
```

4.4.6 Apache を起動

設定とテスト用ファイルができたので Apache を起動してみましょう。systemctl コマンドで、httpd を起動します。

Apache の起動

```
# systemctl start httpd
```

Apache が起動しているかを、systemctl の status サブコマンドで確認します。Apache が起動している場合、下記のようにステータスが表示されます。

Apache の起動

```
# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset:
       disabled)
   Active: active (running) since 月 2019-02-18 15:39:15 JST; 4s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Process: 27591 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
  Process: 28290 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited,
      status=0/SUCCESS)
 Main PID: 27616 (httpd)
   Status: "Processing requests..."
    Tasks: 6
   CGroup: /system.slice/httpd.service
           ├─27616 /usr/sbin/httpd -DFOREGROUND
           ├─27617 /usr/sbin/httpd -DFOREGROUND
           ├─27618 /usr/sbin/httpd -DFOREGROUND
           ├─27619 /usr/sbin/httpd -DFOREGROUND
```

```
           ├─27620 /usr/sbin/httpd -DFOREGROUND
           └─27621 /usr/sbin/httpd -DFOREGROUND

 2月 18 15:39:15 centos7 systemd[1]: Starting The Apache HTTP Server...
 2月 18 15:39:15 centos7 systemd[1]: Started The Apache HTTP Server.
```

Active の欄に「active (running)」と表示されていることを確認します。また、下の方にはログが表示されます。ここでも、「Started The Apache HTTP Server.」と表示されていて、Apache のサービスが起動していることが確認できます。

4.4.7 自動起動の設定

Linux 起動時に Apache が必ず起動されるように設定しておきましょう。自動起動になっているかは、次のように確認できます。

Apache の自動起動の確認

```
# systemctl is-enabled httpd
disabled
```

自動起動の設定がされている場合には、「enabled」と表示されます。この例のように「disabled」と表示される場合には、自動起動設定が行われていません。次のようにして、自動起動設定を行います。

Apache の自動起動の設定

```
# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
    /usr/lib/systemd/system/httpd.service.
```

4.4.8 Web ブラウザーで自分のアドレスを確認

Web システムはコマンドで確認する代わりに、Web ブラウザーを使いグラフィカルに結果を確認できます。Web ブラウザーである Firefox から自分のアドレスを確認します。

```
http://www.alpha.jp/
```

図 13 ブラウザーの表示

自分のサーバーのアクセスができたら、隣の受講生マシンの Apache へアクセスします。

```
http://www.beta.jp/
```

4.5 ページが見つからないとき

テストファイルを作る前に Apache を起動し、Web ブラウザーである Firefox から用意されていないページをアクセスすると、どんな結果が出るでしょうか？ 実際に無いページにアクセスしてみてください。ファイルが無い旨のエラー (Not Found) が出ます。エラーページが表示されるのは Apache で対応するページが用意されているからです。エラーページは/usr/share/httpd/error ディレクトリにあり、トップページが無い場合は特別に用意されたテストページが表示されます。

図 14　隣の受講生のブラウザー表示

図 15　テストページ

エラーページのファイル群

```
# ls /usr/share/httpd/error/
HTTP_BAD_GATEWAY.html.var             HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
HTTP_BAD_REQUEST.html.var             HTTP_REQUEST_TIME_OUT.html.var
HTTP_FORBIDDEN.html.var               HTTP_REQUEST_URI_TOO_LARGE.html.var
HTTP_GONE.html.var                    HTTP_SERVICE_UNAVAILABLE.html.var
HTTP_INTERNAL_SERVER_ERROR.html.var   HTTP_UNAUTHORIZED.html.var
HTTP_LENGTH_REQUIRED.html.var         HTTP_UNSUPPORTED_MEDIA_TYPE.html.var
HTTP_METHOD_NOT_ALLOWED.html.var      HTTP_VARIANT_ALSO_VARIES.html.var
HTTP_NOT_FOUND.html.var               README
HTTP_NOT_IMPLEMENTED.html.var         contact.html.var
HTTP_PRECONDITION_FAILED.html.var     include
```

「アクセスしたページが見つからない」などの理由によって、それに対応するエラーページが表示されます。

4.5.1　Apache のエラーコードについて

Apache はアクセス情報とエラー情報をログに記録しています。また、エラーメッセージに対応するエラーページを表示します。ここでは Aapche がエラーを表示したときのエラーコードに対する意味を説明します。

表 7: Apache のエラーコード

番号	理由	意味
400	BAD_REQUEST	要求されたコードを理解できません
401	UNAUTHORIZED	アクセスする権利があることを確証できませんでした
403	FORBIDDEN	要求するディレクトリにアクセスする為の許可がありません
404	NOT_FOUND	要求されたファイルが見つかりません
405	METHOD_NOT_ALLOWED	許可されていないメソッドを受け取りました
408	REQUEST_TIME_OUT	指定時間内にリクエストを終えなかったのでネットワーク接続を閉じました
410	GONE	要求された URL はサーバー利用できず、転送先アドレスも理解できません
411	LENGTH REQUIRED	Content-Length メソッドが不正です
412	PRECONDITION_FAILED	URL に対してリクエストの必要条件は、明確な評価に失敗しました
413	REQUEST_ENTITY_TOO_LARGE	要求されたデータ量が容量限度を超えました
414	REQUEST_URI_TOO_LARGE	要求された URL 長は、このサーバーの容量限度を超えた為、処理できません
415	UNSUPPORTED_MEDIA_TYPE	サーバーは、メディアタイプがリクエストで送ったことをサポートしません
500	INTERNAL_SERVER_ERROR	サーバーは内部エラーの為、要求を完了することができませんでした
501	NOT_IMPLEMENTED	サーバーはリクエストを実行する為の必要な機能をサポートしていません
502	BAD_GATEWAY	ゲートウェイやプロキシが、無効な応答を上位のサーバーから受け取りました
503	SERVICE_UNAVAILABLE	サーバーが一時的な過負荷または保守時間のため、要求を受け付けられませんでした
506	VARIANT_ALSO_VARIES	要求するエンティティの値は、そのもの自体で交渉できるリソースです

4.5.2 ログファイルの確認

Apache が起動すると、アクセスされたファイルの履歴が/var/log/httpd/access_log ログファイルに記録され、エラーメッセージが/var/log/httpd/error_log ログファイルに記録されます。ログファイルを確認してみてください。

ログの確認

```
# tail /var/log/httpd/access_log
 （略）
192.168.1.101 - - [18/Feb/2019:15:51:28 +0900] "GET /images/poweredby.png HTTP/1.1" 200
    3956 "http://www.alpha.jp/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
    Gecko/20100101 Firefox/60.0"
192.168.1.101 - - [18/Feb/2019:15:52:44 +0900] "GET / HTTP/1.1" 200 34 "-" "Mozilla/5.0
    (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
192.168.1.101 - - [18/Feb/2019:15:56:32 +0900] "GET / HTTP/1.1" 200 33 "-" "Mozilla/5.0
    (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

# tail /var/log/httpd/error_log
 （略）
[Mon Feb 18 15:51:27.787711 2019] [autoindex:error] [pid 2493] [client
    192.168.1.101:46718] AH01276: Cannot serve directory /var/www/html/: No matching
    DirectoryIndex (index.html) found, and server-generated directory index forbidden
    by Options directive
[Mon Feb 18 15:52:22.858248 2019] [autoindex:error] [pid 2491] [client
    192.168.1.101:46726] AH01276: Cannot serve directory /var/www/html/: No matching
    DirectoryIndex (index.html) found, and server-generated directory index forbidden
    by Options directive
[Mon Feb 18 15:59:30.202894 2019] [autoindex:error] [pid 2494] [client
    192.168.1.101:46766] AH01276: Cannot serve directory /var/www/html/: No matching
    DirectoryIndex (index.html) found, and server-generated directory index forbidden
    by Options directive
```

ログファイルの名前は httpd.conf 設定ファイルで設定されています。エラーログの出力ファイル名は ErrorLog ディレクティブで、アクセスログの出力ファイル名は CustomLog ディレクティブです。ログファイルに記録されるクライアント情報はデフォルトが IP アドレスとなり、HostnameLookups ディレクティブを on にすると IP アドレスから調べたクライアントの名前を保存することも可能です。DNS サーバーから名前を引くと時間がかかるので、デフォルトでは off となっています。

表 8: ログ関係のディレクティブ

ディレクティブ名	内容	設定例
ErrorLog	エラーのログファイル	/var/log/httpd/error_log
CustomLog	アクセスのログファイルと形式	/var/log/httpd/access_log
HostnameLookups	DNS への問い合わせ	on...問い合わせ、off...問い合わせない

4.6 アクセス制御

通常、多くの人から自由に見てもらいたい情報を Web サービスで公開します。その一方で、特定の権限がある人だけに見せる会員制のホームページを作りたいときがあると思います。特定の個人だけアクセスできるように限定する手段として Apache には BASIC 認証と DIGEST 認証という機能が備えられています。この節では DocumentRoot ディレクティブで指定されている/var/www/html ディレクトリにある secret ディレクトリを特定の個人だけアクセスできるよう設定するために、セキュリティ上で安心な DIGEST 認証を設定してみましょう。

4.6.1 テキストファイルを作成

認証をかけたディレクトリとファイルを作成します。

認証を掛けるディレクトリとページの作成

```
# mkdir /var/www/html/secret
# echo -n 'This is Secret Page on ' > /var/www/html/secret/index.html
# hostname >> /var/www/html/secret/index.html
```

アクセス制御を設定する前であれば、制限をかけたいディレクトリやファイルであっても見えてしまいます。Web ブラウザーで以下のアドレスにアクセスできることを確認します。

```
http://www.alpha.jp/secret/
```

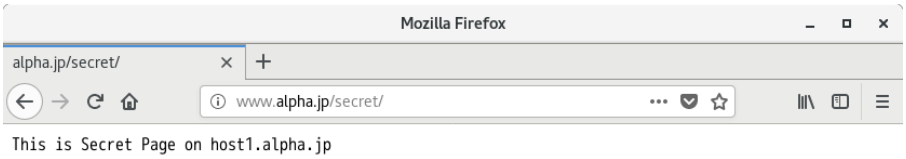


図 16 認証を掛ける前のページ

4.6.2 アクセス制御を設定

Apache でアクセス制御を設定するには設定ファイルにアクセス制御の設定を追加し、コマンドでパスワードファイルを作ります。/etc/httpd/conf/httpd.conf 設定ファイルに/var/www/html/secret ディレクトリの設定を追記します。

/etc/httpd/conf/httpd.conf ファイルの最下行に追記

```
<Directory /var/www/html/secret>
AuthType Digest
AuthName "Secret Zone"

AuthUserFile /etc/httpd/.htdigest
Require user linuc
</Directory>
```

Directory ディレクティブ (<Directory >～</Directory >) は/var/www/html/secret ディレクトリに認証をかける複数のディレクティブを囲みます。AuthType は認証の種類を Digest とし、AuthName は認証のダイアログに表示される認証名を “Secret Zone” とし、AuthUserFile でパスワードファイルを指定し、Require user でユーザー名を linuc としました。

表 9: アクセス制御関係のディレクティブ

ディレクティブ名	内容	設定例
Directory	ディレクトリ	/var/www/html/secret
AuthType	認証タイプ	Digest
AuthName	認証ドメイン	Secret Zone
AuthUserFile	パスワードファイル	/etc/httpd/.htdigest
Require user	ユーザー指定	linuc

Require user では、認証を許可するユーザーを指定します。認証にはユーザーとパスワードの情報が必要です。

次にパスワードファイルを作成します。パスワードファイルは重要なファイルなので、他のユーザーから見えないように設定します。ファイル作成後、Apache を動かしている apache ユーザーからのみ読みこめるように、ファイルの所有者とモードを変更します。

パスワードファイルを作成し、ユーザ linuc を登録

```
# htdigest -c /etc/httpd/.htdigest 'Secret Zone' linuc
Adding password for linuc in realm Secret Zone.
New password: linuc ← 実際には表示されない
Re-type new password: linuc ← 実際には表示されない
# ls -l /etc/httpd/.htdigest
-rw-r--r--. 1 root root 51  2月 18 16:26 /etc/httpd/.htdigest
# chown apache.apache /etc/httpd/.htdigest
# chmod 400 /etc/httpd/.htdigest
# ls -l /etc/httpd/.htdigest
-r--------. 1 apache apache 51  2月 18 16:26 /etc/httpd/.htdigest
```

4.6.3 設定の再読み込み

設定を変更したので、Apache に設定を再読み込みさせます。

設定の再読み込み

```
# systemctl reload httpd
```

4.6.4 Web ブラウザーで自分のアドレスを確認

Web ブラウザーで自分のアドレス（http://www.alpha.jp/secret）を確認します。認証のダイアログが表示されるので、初めに登録していない適当なユーザー名と適当なパスワードを入力すると何度も認証のダイアログが表示されること (認証エラー) を確認します。次に登録した linuc ユーザーとパスワードを入力すると作成したファイルが表示されます。

正しいユーザー名、パスワードを入力した場合は、secret フォルダの中の index.html ファイルにアクセスできるようになります。認証をかけたディレクトリにあるディレクトリの中もリカーシブ (再帰的) にユーザー名とパスワードの組み合わせを知っているユーザーのみがアクセスできます。

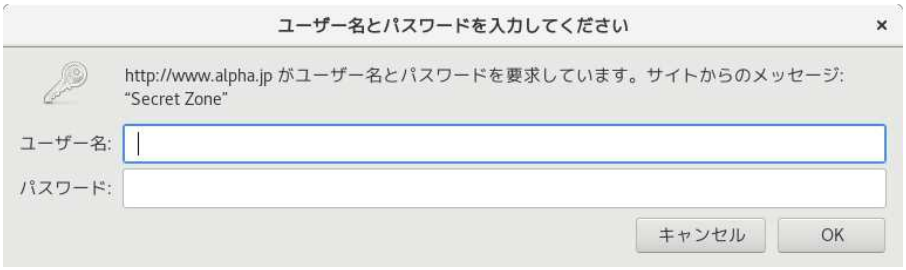


図 17 認証を掛けたページへのアクセス

4.6.5 ログファイルの確認

認証時のログファイルを確認します。

認証時のログの確認

```
# tail /var/log/httpd/access_log
 （略）
192.168.1.101 - - [18/Feb/2019:16:55:01 +0900] "GET /secret/ HTTP/1.1" 401 381 "-"
    "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

```
192.168.1.101 - linuc [18/Feb/2019:16:55:08 +0900] "GET /secret/ HTTP/1.1" 304 - "-"
    "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

# tail /var/log/httpd/error_log
(略)
[Mon Feb 18 16:58:56.588983 2019] [auth_digest:error] [pid 12524] [client
    192.168.1.101:46884] AH01790: user `lpic' in realm `Secret Zone' not found:
    /secret/
```

認証に失敗するとエラーログファイルにユーザーがみつからないというエラーメッセージが残ります。

4.7 バーチャルホストを作成する

バーチャルホスト機能を利用すると、1 台の Web サーバーで別々のホームページを見せることができます。バーチャルホスト機能を利用するには、以下の設定が必要です。

- DNS で 1 つの IP アドレスに対して別々のホスト名を割り当てる
- Apache でバーチャルホストの設定を行う
- Linux にそれぞれのバーチャルホスト用のディレクトリを作成する

ここでは、vhost1.alpha.jp と vhost2.alpha.jp という 2 つのバーチャルホストを作成します。それぞれの名前で Web ブラウザーからアクセスした時に、別々のホームページが見えるように設定を行います。

4.7.1 IP アドレスと名前の確認

2 つの名前を 1 つの IP アドレスに割り振っている設定を、host コマンドで確認します。

DNS を確認

```
# host vhost1.alpha.jp
vhost1.alpha.jp has address 192.168.1.101
# host vhost2.alpha.jp
vhost2.alpha.jp has address 192.168.1.101
```

4.7.2 バーチャルホストの設定

バーチャルホストの設定には、VirtualHost ディレクティブを使います。vhost1.alpha.jp と vhost2.alpha.jp という 2 つの URL アドレスを定義します。

表 10: バーチャルホスト関係のディレクティブ

ディレクティブ	内容	具体例
VirtualHost	指定した IP アドレスとポートに対するバーチャルホストの設定をする	*:80

/etc/httpd/conf.d/virtualhost.conf ファイルを作成し、次のように設定を行います。

/etc/httpd/conf.d/virtualhost.conf を作成

```
<VirtualHost *:80>
  ServerName www.alpha.jp
  DocumentRoot /var/www/html
  ServerAdmin webmaster@www.alpha.jp
  ErrorLog /var/log/httpd/error_log
  CustomLog /var/log/httpd/access_log common
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerName vhost1.alpha.jp
  DocumentRoot /var/www/vhost1.alpha.jp
  ServerAdmin webmaster@vhost1.alpha.jp
  ErrorLog /var/log/httpd/vhost1.alpha.jp-error_log
  CustomLog /var/log/httpd/vhost1.alpha.jp-access_log common
</VirtualHost>

<VirtualHost *:80>
  ServerName vhost2.alpha.jp
  DocumentRoot /var/www/vhost2.alpha.jp
  ServerAdmin webmaster@vhost2.alpha.jp
  ErrorLog /var/log/httpd/vhost2.alpha.jp-error_log
  CustomLog /var/log/httpd/vhost2.alpha.jp-access_log common
</VirtualHost>
```

VirtualHost のブロック内では、ドメイン名、ドメインにアクセスしたときに表示するコンテンツの参照先、サーバーの管理者のメールアドレス、ログの出力先を定義します。上記のように設定した場合は、www.alpha.jp にアクセスした場合は /var/www/html 配下のコンテンツが表示され、ログが/var/log/httpd/access_log に保存されます。エラーがあった場合はエラーログが/var/log/httpd/error_log に保存されます。

4.7.3 テストファイルを作成

同じマシンで 2 つの URL に対応するので、全く違った内容を表示するようにテストファイルを作成します。2 カ所の DocumentRoot ディレクティブで指定した 2 つのディレクトリを作成し、各ディレクトリにサンプルファイルを用意します。

テスト用ページの作成

```
# mkdir /var/www/vhost1.alpha.jp
# mkdir /var/www/vhost2.alpha.jp
# echo -n 'This is Virtual Page 1 on ' > /var/www/vhost1.alpha.jp/index.html
# hostname >> /var/www/vhost1.alpha.jp/index.html
# echo -n 'This is Virtual Page 2 on ' > /var/www/vhost2.alpha.jp/index.html
# hostname >> /var/www/vhost2.alpha.jp/index.html
```

4.7.4 設定ファイルの再読み込み

バーチャルホストの定義を書き加えて設定ファイルを変更したので、Apache に設定ファイルを再読み込みさせます。

設定ファイルの再読み込み

```
# systemctl reload httpd
```

4.7.5 Web ブラウザーで自分のアドレスを確認

それでは、vhost1.alpha.jp や vhost2.alpha.jp にアクセスして別々のコンテンツが表示されるか早速確認してみましょう。

```
http://vhost1.alpha.jp/
http://vhost2.alpha.jp/
```

2 つの URL が表示されないときは、DNS の設定が正しいかどうかを host コマンド等で再確認します。

4.7.6 ログファイルの確認

正しく起動したかをログファイルが作られているかで確認してみてください。

図 18　vhost1.alpha.jp の表示

図 19　vhost2.alpha.jp の表示

バーチャルホストログの確認

```
# ls -l /var/log/httpd
合計 20
-rw-r--r--. 1 root root 8482  2月 18 17:13 access_log
-rw-r--r--. 1 root root 7075  2月 18 17:26 error_log
-rw-r--r--. 1 root root  144  2月 18 17:28 vhost1.alpha.jp-access_log
-rw-r--r--. 1 root root    0  2月 18 17:26 vhost1.alpha.jp-error_log
-rw-r--r--. 1 root root  210  2月 18 17:33 vhost2.alpha.jp-access_log
-rw-r--r--. 1 root root    0  2月 18 17:26 vhost2.alpha.jp-error_log
```

5 メールサーバーの構築

メールのやり取りが行えるよう、メールサーバーを設定します。まずは Postfix を使って、メールサーバー同士でメールのやり取りが行えるように設定します。さらに POP/IMAP サーバーとメールクライアントを使って、より実践的なメール環境を構築します。

5.1 用語集

メールサーバー

電子メールのサービスを行います。クライアントよりメールを受け取り、バケツリレーの方式で相手先のメールサーバーまで送ります。また、受信用のメールサーバーでは、送ってきたメールを蓄積しておいて、クライアントの要求に応じて応答します。

MTA（Mail Transfer Agent）

メールの転送を行うプログラムです。Sendmail や Postfix などが代表例です。

SMTP(Simple Mail Transfer Protocol)

電子メールの送信、転送のときに利用されるプロトコルのことです。

SMTP 認証

SMTP でのメールを送信する際に認証を行う機構です。迷惑メール対策としてのメール中継の制限を、この認証機能で許可する、といった利用方法があります。

POP3(Post Office Protocol version3)

クライアントが電子メールを取り寄せるときに利用されるプロトコルです。シンプルな設計で、IMAP4 と比べて機能が少ないです。

IMAP4（Internet Message Access Protocol 4）

クライアントが電子メールを取り寄せるときに利用されるプロトコルです。メールのフォルダ機能サポート等、多機能です。

Postfix

MTA として動作するサーバープログラムです。Linux や Unix のシステムで古くから使われてきた Sendmail よりもセキュリティが高く、高速に動作すると言われています。

Dovecot

POP3 や IMAP4 のサーバー機能を提供するプログラムです。

Thunderbird

Mozilla Project が配布している、高機能なメールクライアントソフトウェアです。

5.2 メールサーバー実習の説明

メールサーバーの設定と動作確認を行います。メールは、インターネットにおいて、Web に並んで重要なサービスです。メールサーバーを設定し、実際にメールをやり取りすることで、動作原理を確認してみましょう。

5.2.1 メールとメールサーバー

メールは、メールサーバーを介してやり取りが行われます。

メールサーバーがメールを受け取ると、宛先のメールアドレスを担当しているメールサーバーまでバケツリレー方式で送られます。これは、Thunderbird/Outlook のようなメールクライアントソフトからのメールでも、Gmail/Hotmail/Yahoo メールのような Web メールからのメールでも、動作原理は同じです。

MTA(Mail Transfer Agent)

メールがバケツリレー方式で運ばれることは、前述の通りです。このバケツリレーをするプログラムのことを MTA といいます。本教科書では Postfix という MTA を利用します。他に有名な MTA として Postfix などがあります。

SMTP(Simple Mail Transfer Protocol)

メールサーバー間は、SMTP というプロトコルでやり取りされています。SMTP はかなり昔に設計、定義されたプロトコルのため、認証やアクセス制限などが無く、勝手にメールサーバーを利用されて迷惑メールを送られてしまうなどの問題がありました。そこでこのような問題を解決するために、ESMTP（拡張 SMTP）が定義されました。SMTP と呼ぶ場合、この ESMTP で定義された機能も含んでいることがあります。

SMTP 認証 (SMTP Authentication)

正規の SMTP 接続では、メールの中継を行います。ところが前述の通り、SMTP には認証機能が無いため、多くの場合、特定の場所以外からの中継を拒否します。SMTP 認証は、SMTP 上に認証 (Authentication) 機能を加え、その中継要求が正規のものか不正なものかを判断します。SMTP 認証は ESMTP の機能のうちの 1 つです。

POP3(Post Office Protocol version 3)

電子メールは、送受信でプロトコルが異なります。POP3 は電子メールを受信するときに利用するプロトコルです。非常にシンプルなプロトコルで、ユーザー名、パスワードを利用して接続し、メールの内容を受信します。

IMAP4

IMAP4 も POP3 同様、メールを受信するときに利用するプロトコルです。IMAP4 は POP3 に比べて機能が豊富で、大きな特徴としてフォルダ機能をサポートしていることが挙げられます。そのため、メール管理が非常に楽になります。

5.2.2 メールのやり取り

インターネット上で、沢山の人が電子メールを利用しています。電子メールは以下の手順でやり取りされます。

1. 送信側のメールクライアントからメールを送信します
2. メールは送信用メールサーバーを経由して相手のメールサーバーに配信されます
3. 相手の受信側のメールサーバーにメールが届きます
4. 受信側のメールクライアントで受信側のメールサーバーに接続します
5. メールが受信され、メールを見ることができます

前述の構成で実習を行う場合、サーバー、クライアントをそれぞれ 2 台ずつ、計 4 台必要になります。本実習では二人一組のペアを組んで 2 台で実習を行うため、以下のような構成を取ります。

- 1 台のマシンで、メールサーバーとメールクライアントの 1 台 2 役とします
- 自分のメールサーバーに、自分のメールアカウントを作成します
- 相手のメールサーバーに、相手のメールアカウントが作成されます
- 自分のメールクライアントは、自分のメールサーバーを送受信用サーバーとして設定します

ポイントは、自分のマシンは 1 台ですが、メールサーバーとメールクライアントの 1 台 2 役であることです。

5.3　実習の進め方

本章では、次の手順で実習を行います。実習でのメールの送受信は、大きく分けて 2 回あります。

mail コマンドを利用する

端末で mail コマンドを使って相手にメール送信します。以下の手順に従ってください。

1. 二人一組になります。それぞれ host1.alpha.jp を利用する A さん (usera@alpha.jp) と、host2.beta.jp を利用する B さん (userb@beta.jp) とします。
2. Postfix の設定ファイルを作成し、Postfix を再起動します。
3. 送受信テスト用の自分のアカウント（usera@alpha.jp、userb@beta.jp）を、それぞれのホストに作成します。
4. A さんが host1.alpha.jp から mail コマンドを使って、B さんにメールを送ります。
5. B さんは mail コマンドを使って、到着を確認します。
6. 逆に B さんから A さんにメールを送り、確認します。

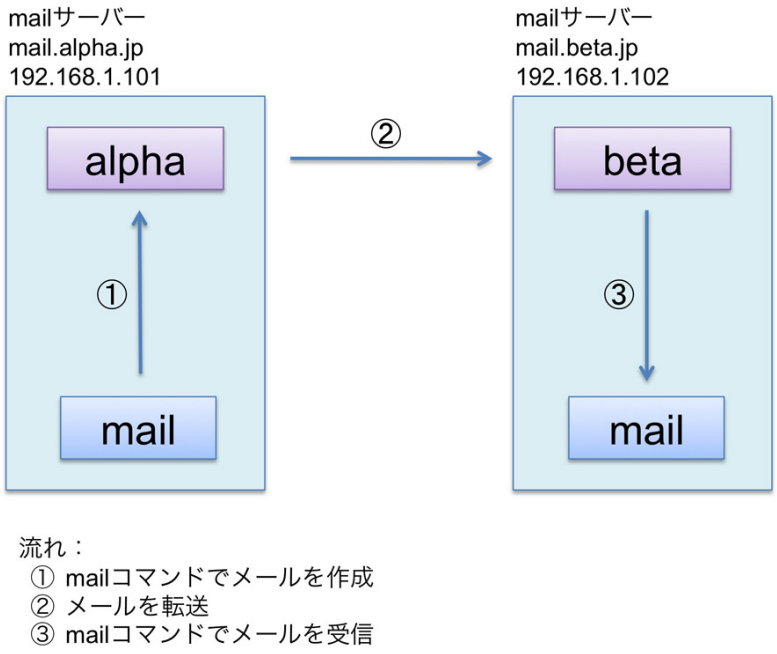


図 20　実習の流れ

メールクライアントソフトを利用する

メールクライアントを使って相手にメール送信します。本実習では Thunderbird を使います。以下の手順に従ってください。

1. POP/IMAP サーバーとして Dovecot の設定を行い、POP/IMAP サーバーを起動します。
2. メールクライアントとして Thunderbird を設定します。
3. A さんが host1.alpha.jp からメールクライアントを使って、B さんにメールを送ります。
4. B さんはメールクライアントを使って、到着を確認します。
5. 逆に B さんから A さんにメールを送り、確認します。

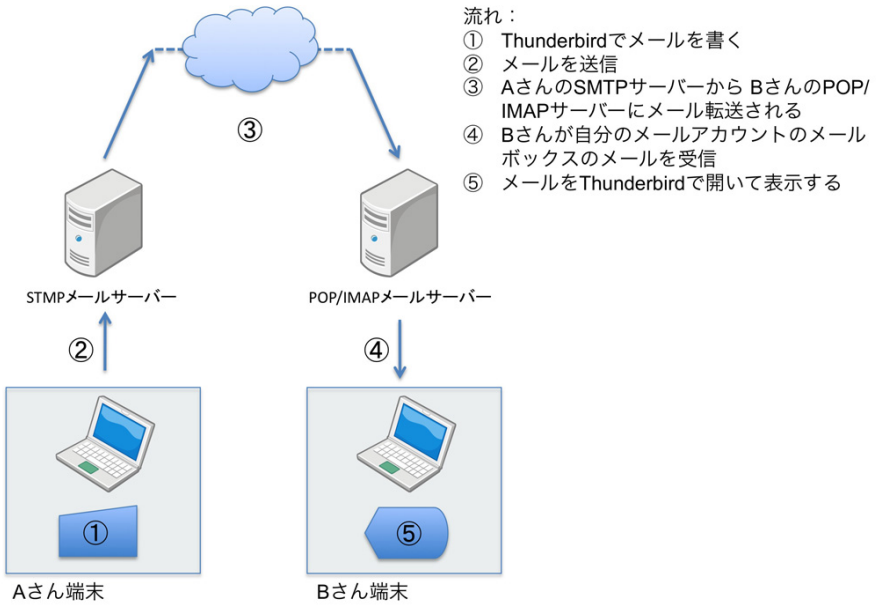


図 21　メールクライアントを使った実習の流れ

5.3.1　実習後の注意点

設定したメールサーバーは、あくまで実習用にメールのやり取りをするために設定されています。ですが、セキュリティ等決して頑丈に設定してある訳ではないので、以下の点に留意します。

- 決してグローバル環境には接続しない。
- 実習後はメールサーバーを止める、もしくは本体自体を止める。LAN の中に不正中継を探すマシンがあった場合、それに利用される可能性があるからです。

5.3.2　実習で使用するソフトウェアについて

Postfix

今回の実習では、MTA として Postfix を利用します。

mail コマンド

mail コマンドは、標準でインストールされている、メールを操作するコマンドです。mail コマンドには、メールを送る以外に、届いたメールを読む機能もあります。

本実習では、メールで送るときと届いたメールを読むときに利用します。

Dovecot

Dovecot は、POP3 や IMAP4 を機能させるサーバーのソフトウェアです。実習では、メールのクライアントソフトから IMAP4 でメールを受信します。そのときに IMAP4 のサーバーとして動作させます。標準ではインストールされていないので、実習時

にパッケージを追加し、設定ファイルを書き換えます。

Thunderbird

Mozilla Project により開発されている、フリーのメールクライアントソフトがこの Thunderbird です。Windows, Mac OS X, Linux 等と、動作環境は多岐に渡っており、各国語版も用意されております。機能も必要十分な内容がそろっています。本実習では、メールのクライアントソフトとして Thunderbird を使って実習を行います。実際に Thunderbird をインストールしメールの送受信を行うことで、メール設定が正しいか確認を行います。

saslauthd

saslauthd は、SMTP 認証の認証機構です。Postfix は設定ファイルで SMTP 認証の機能を有効にできますが、Postfix 自体は認証の機能を持っていません。設定ファイルで SMTP 認証の機能を有効にすると、Postfix は saslauthd に認証を依頼してその結果を受け取ります。

5.3.3　実習環境

実習では、受講生二人一組で実習を行います。本章では、それを便宜的に「A さん」「B さん」と呼びます。特に明示がない場合は、両者とも作業を行います。どちらか片方の方が作業をするときは、「次は A さんの作業です」といったように、作業する方を明示します。

設定は、alpha.jp のマシン上で行うものとします。ホスト名は、各自読み替えてください。

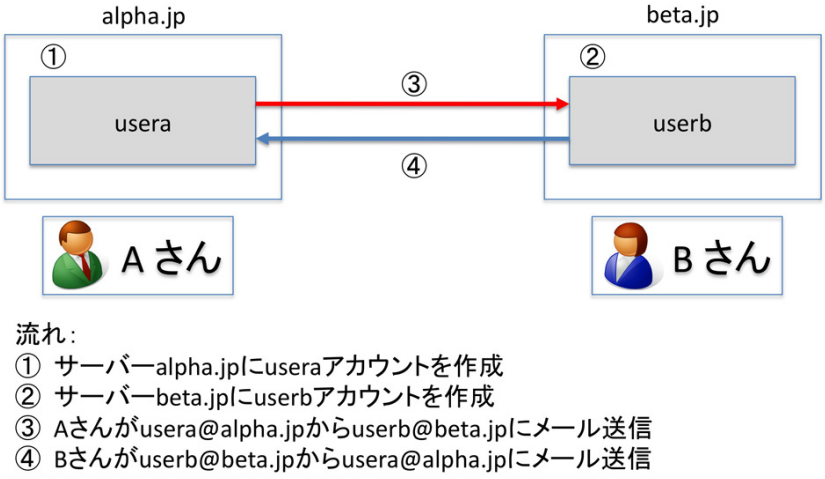


図 22　実習環境

5.4　Postfix のインストール

Postfix は、CentOS7 では標準でインストールされています。念のため、rpm コマンドでパッケージがインストールされているかを確認します。

postfix パッケージの確認

```
# rpm -q postfix
postfix-2.10.1-7.el7.x86_64
```

パッケージがインストールされていない場合は次の様に表示されます。

postfix パッケージの確認

```
# rpm -q postfix
パッケージ postfix はインストールされていません。
```

5.4.1　必要なパッケージをインストール

postfix パッケージがインストールされていないときは、yum コマンドでインストールをします。インターネットに接続できない環境では、GUI から admin でログインし、インストールメディアが自動マウントされた状態でインストール作業を進めます。

postfix のインストール

```
# yum install postfix
読み込んだプラグイン:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp-srv2.kddilabs.jp
 * extras: ftp-srv2.kddilabs.jp
 * updates: ftp-srv2.kddilabs.jp
依存性の解決をしています
```

```
--> トランザクションの確認を実行しています。
---> パッケージ postfix.x86_64 2:2.10.1-7.el7 を インストール
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package          アーキテクチャー
                                  バージョン                  リポジトリー    容量
================================================================================
インストール中:
 postfix          x86_64          2:2.10.1-7.el7             base          2.4 M

トランザクションの要約
================================================================================
インストール  1 パッケージ

総ダウンロード容量: 2.4 M
インストール容量: 12 M
Is this ok [y/d/N]: y   ← 確認してyを入力
Downloading packages:
postfix-2.10.1-7.el7.x86_64.rpm                                | 2.4 MB   00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中          : 2:postfix-2.10.1-7.el7.x86_64                    1/1
  検証中                  : 2:postfix-2.10.1-7.el7.x86_64                    1/1

インストール:
  postfix.x86_64 2:2.10.1-7.el7

完了しました!
```

5.4.2 main.cf の設定

Postfix の設定ファイルは、/etc/postfix/main.cf です。このファイルを確認してみると、次のような書式で設定されています。

/etc/postfix/main.cf の確認

```
# cat /etc/postfix/main.cf
(略)
# LOCAL PATHNAME INFORMATION
#
# The queue_directory specifies the location of the Postfix queue.
# This is also the root directory of Postfix daemons that run chrooted.
# See the files in examples/chroot-setup for setting up Postfix chroot
# environments on different UNIX systems.
#
queue_directory = /var/spool/postfix

# The command_directory parameter specifies the location of all
# postXXX commands.
#
```

```
command_directory = /usr/sbin
 （略）
# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
#myhostname = virtual.domain.tld
```

queue_directory や command_directory は、設定パラメータです。Postfix では、設定パラメータに値を指定することで、設定を行います。先頭が#で始まる行はコメント行です。myhostname のように、設定例がコメントで記載されていますので、それを参考に設定を行います。また、設定パラメーターは、$myhostname のようにして、変数のように参照して使うこともできます。

/etc/postfix/main.cf の編集

```
# vi /etc/postfix/main.cf
```

次のパラメータを探して設定します。

表 11: 設定が必要な項目

パラメータ	内容	設定例
myhostname	メールサーバーのホスト名	mail.alpha.jp
mydomain	メールサーバーのドメイン名	alpha.jp
inet_interfaces	メールを受け付ける IP アドレス	localhost, 192.168.1.101
mydestination	受信するメールドメイン	alpha.jp
mynetworks	クライアントの IP またはネットワーク	192.168.1.101
smtpd_sasl_auth_enable	SMTP 認証用の sasl を有効にします	yes
smtpd_recipient_restrictions	SMTP 認証を有効にします	permit_mynetworks, permit_sasl_authenticated, reject_unauth_destinaition

自ホスト名とドメインの設定

myhostname と mydomain に、自ホスト名とドメインを設定します。

自ホスト名とドメインの設定（/etc/postfix/main.cf）

```
# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
#myhostname = virtual.domain.tld
myhostname = mail.alpha.jp  ← 設定を追加

# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
```

```
# $mydomain is used as a default value for many other configuration
# parameters.
#
#mydomain = domain.tld
mydomain = alpha.jp ← 設定を追加
```

メールを受け付ける IP アドレスの設定

inet_interfaces に、メールを受け付ける IP アドレスを設定します。

メールを受け付ける IP アドレスの設定（/etc/postfix/main.cf）

```
# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on.  By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
#inet_interfaces = all
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
inet_interfaces = localhost, 192.168.1.101  ←  設定を追加
```

受信するメールドメインの設定

mydestination には、このメールサーバが受信するドメインを設定します。

受信するメールドメインの設定（/etc/postfix/main.cf）

```
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mydestination = alpha.jp    ← 設定を追加
```

クライアントの設定

mynetworks には、このメールサーバを使ってメールを送信するクライアントのアドレスを設定します。192.168.1.0/24 のようにネットワークで指定することもできます。ですが、この実習では自分の IP アドレスだけで充分です。

クライアントの設定（/etc/postfix/main.cf）

```
# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
#mynetworks = 168.100.189.0/28, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table
mynetworks = 192.168.1.101  ←  設定を追加
```

SMTP 認証の設定

SMTP 認証の設定として、smtpd_sasl_auth_enable と、smtpd_recipient_restrictions を設定します。コメントが用意されていませんので、ファイルの最後に追加しておきます。

SMTP 認証の設定（/etc/postfix/main.cf）

```
smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated,
    reject_unauth_destination
```

5.4.3　書式のチェック

/etc/postfix/main.cf の修正ができたら、書式チェックを行っておきます。

main.cf の書式チェック

```
# postfix check
```

書式が正しい場合には、何も表示されません。エラーが表示された場合には、エラー内容をよく見て修正します。

5.4.4　ファイアウォールの設定

Postfix の再起動の前に、メールサーバーでメールの受信ができるようにファイアウォールのサービス許可設定を行います。

SMTP アクセスの許可

```
# firewall-cmd --add-service=smtp
```

さらに、設定を保存しておきます。

Firewall ルールの保存

```
# firewall-cmd --runtime-to-permanent
```

5.4.5　Postfix の再起動

postfix サービスを再起動します。

postfix サービスの再起動

```
# systemctl restart postfix
```

postfix サービスの自動起動を確認します。

自動起動設定の確認

```
# systemctl is-enabled postfix
enabled
```

5.4.6 saslauthd サービスの起動

SMTP 認証用の saslauthd サービスを起動します。

saslauthd の起動

```
# systemctl start saslauthd
```

sasluauthd の自動起動設定も行っておきます。

saslauthd の自動起動設定

```
# systemctl enable saslauthd
Created symlink from /etc/systemd/system/multi-user.target.wants/saslauthd.service to
    /usr/lib/systemd/system/saslauthd.service.
```

5.5 アカウントの作成

それでは、実際にメールを送信する前に、宛先となるアカウントを作成します。

5.5.1 host1.alpha.jp に usera を作成

host1.alpha.jp で usera というアカウントを作成します。このアカウントは usera@alpha.jp というメールアドレスになります。

usera の作成

```
[root@host1 postfix]# useradd usera
[root@host1 postfix]# passwd usera
ユーザー usera のパスワードを変更。
新しいパスワード: userapass ← 入力文字は非表示
新しいパスワードを再入力してください: userapass   ← 入力文字は非表示
passwd: すべての認証トークンが正しく更新できました。
```

5.5.2 host2.beta.jp に userb を作成

host2.beta.jp で userb というアカウントを作成します。このアカウントは userb@beta.jp というメールアドレスになります。

userb の作成

```
[root@host2 postfix]# useradd userb
[root@host2 postfix]# passwd userb
ユーザー userb のパスワードを変更。
新しいパスワード: userbpass ← 入力文字は非表示
新しいパスワードを再入力してください: userbpass   ← 入力文字は非表示
passwd: すべての認証トークンが正しく更新できました。
```

5.6 メールの送受信

次にメールを送信します。メールの送受信は作成した一般ユーザーで行います。一般ユーザーで操作できるよう別の端末を起動し、su コマンドを使ってユーザーを切り替えます。メールの送信は mail コマンドを使用します。

5.6.1 ログの確認用端末の設定

1.「端末」を起動します
2. tail コマンドを実行して、/var/log/maillog を表示します。-f オプションを付けて実行すると、ログが書き込まれる毎に再読み込みされて最新のログを閲覧できます。

メールログの確認

```
# tail -f /var/log/maillog
```

5.6.2 メール送受信用端末の起動とユーザー切り替え

メール送受信用の端末を起動し、su コマンドでユーザーの切り替えを行います。

1.「端末」を起動します
2. su コマンドでユーザーを切り替えます

host1.alpha.jp で usera に切り替え

usera に切り替え

```
[root@host1 ~]# su - usera
[usera@host1 ~]$  id
uid=1003(usera) gid=1003(usera) groups=1003(usera),12(mail)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

host2.beta.jp で userb に切り替え

userb に切り替え

```
# su - userb
[userb@host2 ~]$  id
uid=1001(userb) gid=1001(userb) groups=1001(userb),12(mail)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

5.6.3 usera@alpha.jp から userb@beta.jp へメール送信

mail コマンドを使って、host1.alpha.jp の usera から userb@beta.jp へメールを送信します。

メール送信

```
[usera@host1 ~]$ mail userb@beta.jp ← mailコマンドの引数に宛先のアドレスを指定
Subject: Test mail from usera       ← Subjectを入力
This is Test Mail from usera        ← メッセージ本文を入力
.   ← メッセージ本文の入力が終わったらピリオドを入力
EOT
```

5.6.4 userb のメール着信確認

mail コマンドを使って、host2.beta.jp の userb にメールが届いているかを確認します。

メールの確認

```
[userb@host2 ~]$ mail
Heirloom Mail version 12.5 7/5/10.  Type ? for help.
"/var/spool/mail/userb": 1 message 1 new
>N  1 usera@mail.alpha.jp   Tue Feb 19 13:38  21/751   "Test mail from usera"
```

```
& 1 ← 1を入力
Message  1:
From usera@mail.alpha.jp  Tue Feb 19 13:38:31 2019
Return-Path: <usera@mail.alpha.jp>
X-Original-To: userb@beta.jp
Delivered-To: userb@beta.jp
Date: Tue, 19 Feb 2019 13:38:31 +0900
To: userb@beta.jp
Subject: Test mail from usera
User-Agent: Heirloom mailx 12.5 7/5/10
Content-Type: text/plain; charset=us-ascii
From: usera@mail.alpha.jp
Status: R

This is Test Mail from usera

& q ← qを入力
Held 1 message in /var/spool/mail/userb
```

このように、host1.alpha.jp から host2.beta.jp にメールが送られていることがわかります。以上で、A さんによる実習が終了です。次に、今度は B さんが A さんに対してメールを送ってみましょう。

5.7　メールクライアントソフトでのメールの送受信

通常のメールサーバーの運用では、メールの利用者はメールクライアントを使用してメールの送受信を行います。送信は SMTP、受信は IMAP や POP3 をプロトコルとして使用します。IMAP サーバーを利用してメールを受信できるよう、IMAP サーバーである Devecot と、メールクライアントとして Thunderbird をインストールしてメールを送受信してみます。

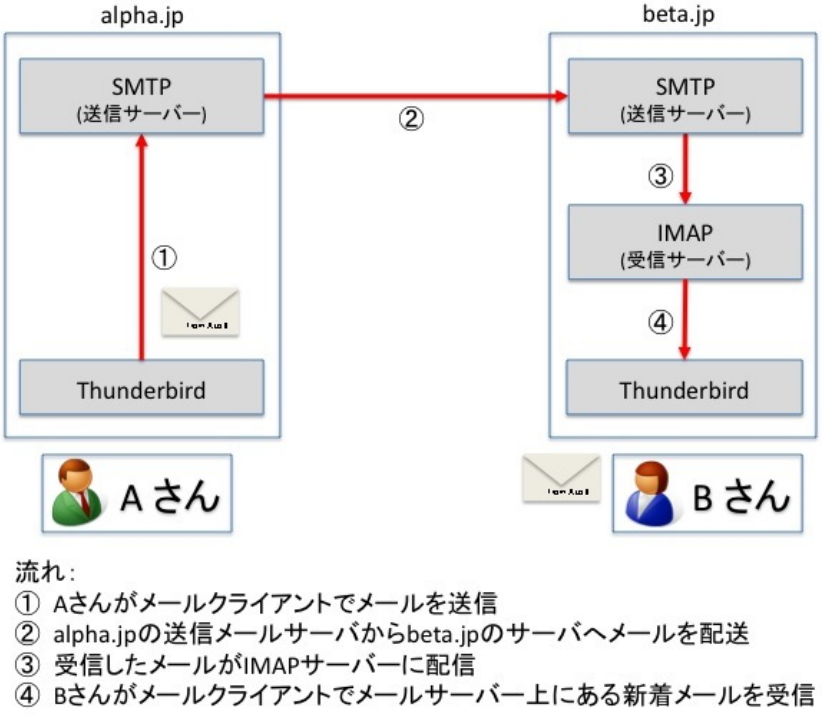


図 23　メールクライアントソフトでのメールの流れ

5.7.1 Dovecot パッケージの追加

それでは早速、必要なパッケージを追加して、クライアントでメールを送受信できるように設定してみましょう。まずは IMAP サーバーである Dovecot をインストールします。インターネットに接続できない環境では、GUI から admin でログインし、インストールメディアが自動マウントされた状態でインストール作業を進めます。

dovecot パッケージのインストール

```
# yum install dovecot
読み込んだプラグイン:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.riken.jp
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> パッケージ dovecot.x86_64 1:2.2.36-3.el7 を インストール
--> 依存性の処理をしています: libclucene-shared.so.1()(64bit) のパッケージ:
    1:dovecot-2.2.36-3.el7.x86_64
--> 依存性の処理をしています: libclucene-core.so.1()(64bit) のパッケージ:
    1:dovecot-2.2.36-3.el7.x86_64
--> トランザクションの確認を実行しています。
---> パッケージ clucene-core.x86_64 0:2.3.3.4-11.el7 を インストール
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package              アーキテクチャー
                                  バージョン               リポジトリー   容量
================================================================================
インストール中:
 dovecot              x86_64         1:2.2.36-3.el7          base          4.4 M
依存性関連でのインストールをします:
 clucene-core         x86_64         2.3.3.4-11.el7          base          528 k

トランザクションの要約
================================================================================
インストール  1 パッケージ (+1 個の依存関係のパッケージ)

総ダウンロード容量: 4.9 M
インストール容量: 16 M
Is this ok [y/d/N]: y   ← 確認してyを入力
Downloading packages:
(1/2): clucene-core-2.3.3.4-11.el7.x86_64.rpm              | 528 kB   00:15
(2/2): dovecot-2.2.36-3.el7.x86_64.rpm                     | 4.4 MB   00:15
--------------------------------------------------------------------------------
合計                                                317 kB/s | 4.9 MB  00:15
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中          : clucene-core-2.3.3.4-11.el7.x86_64              1/2
  インストール中          : 1:dovecot-2.2.36-3.el7.x86_64                   2/2
  検証中                  : clucene-core-2.3.3.4-11.el7.x86_64              1/2
  検証中                  : 1:dovecot-2.2.36-3.el7.x86_64                   2/2
```

```
インストール:
  dovecot.x86_64 1:2.2.36-3.el7

依存性関連をインストールしました:
  clucene-core.x86_64 0:2.3.3.4-11.el7

完了しました!
```

5.7.2　Dovecot の設定

次に、IMAP サーバーである Dovecot の設定を行います。設定ファイルは/etc/dovecot/dovecot.conf と/etc/dovecot/conf.d ディレクトリ以下に分かれています。

/etc/dovecot/dovecot.conf

全体的な設定ファイルです。デフォルトの設定がコメントアウトで記述されています。特に変更は必要ありません。

dovecot.conf を開く

```
# vi /etc/dovecot/dovecot.conf
```

/etc/dovecot/dovecot.conf の確認

```
（略）
# Protocols we want to be serving.
#protocols = imap pop3 lmtp ← IMAP/POP3/LMTPが使用可能

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
#listen = *, :: ← ホストのすべてのIPアドレスで接続を受け付ける
```

/etc/dovecot/conf.d/10-mail.conf

メールボックスの位置やアクセス権などを設定するファイルです。今回は mbox 形式のメールボックスを指定し、mail グループにて管理ができるようにします。

10-mail.conf を編集

```
# vi /etc/dovecot/conf.d/10-mail.conf
```

/etc/dovecot/10-mail.conf（メールボックスの設定）

```
（略）
#   mail_location = maildir:~/Maildir
#   mail_location = mbox:~/mail:INBOX=/var/mail/%u
#   mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
#
# <doc/wiki/MailLocation.txt>
#
mail_location = mbox:~/mail:INBOX=/var/mail/%u  ←  修正
```

次に、同じファイルの中のメールを管理するためのグループの設定を追加します。

/etc/dovecot/10-mail.conf（グループ設定）

```
# Group to enable temporarily for privileged operations. Currently this is
# used only with INBOX when either its initial creation or dotlocking fails.
# Typically this is set to "mail" to give access to /var/mail.
mail_privileged_group = mail      ← 修正

# Grant access to these supplementary groups for mail processes. Typically
# these are used to set up access to shared mailboxes. Note that it may be
# dangerous to set these if users can create symlinks (e.g. if "mail" group is
# set here, ln -s /var/mail ~/mail/var could allow a user to delete others'
# mailboxes, or ln -s /secret/shared/box ~/mail/mybox would allow reading it).
mail_access_groups = mail     ← 修正
```

/etc/dovecot/conf.d/10-auth.conf

認証を設定するファイルです。今回は暗号化していない平文での認証を許可し、Linux のログイン情報を認証に利用できるように設定します。

10-auth.conf を編集

```
# vi /etc/dovecot/conf.d/10-auth.conf
```

/etc/dovecot/10-auth.conf

```
##
## Authentication processes
##

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
#disable_plaintext_auth = yes
disable_plaintext_auth = no ← 追加
```

/etc/dovecot/conf.d/10-ssl.conf

SSL/TLS を設定するファイルです。今回は SSL/TLS 暗号化をしませんので、SSL の利用を停止しておきます。

10-ssl.conf の編集

```
# vi /etc/dovecot/conf.d/10-ssl.conf
```

/etc/dovecot/10-ssl.conf

```
##
## SSL settings
##

# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
# disable plain pop3 and imap, allowed are only pop3+TLS, pop3s, imap+TLS and imaps
# plain imap and pop3 are still allowed for local connections
#ssl = required ← コメントアウト
ssl = no    ← 追加
```

5.7.3 ファイアウォールの設定

Dovecot の起動の前に、POP3 と IMAP4 でメールの受信ができるようにファイアウォールのサービス許可設定を行います。

POP3, IMAP4 アクセスの許可

```
# firewall-cmd --add-service=pop3
# firewall-cmd --add-service=imap
```

さらに、設定を保存しておきます。

Firewall ルールの保存

```
# firewall-cmd --runtime-to-permanent
```

5.7.4 Dovecot の再起動

dovecot サービスを再起動します。

dovecot サービスの起動

```
# systemctl start dovecot
```

自動起動設定も行っておきます。

dovecot の自動起動設定

```
# systemctl enable dovecot
Created symlink from /etc/systemd/system/multi-user.target.wants/dovecot.service to
    /usr/lib/systemd/system/dovecot.service.
```

5.7.5 Thunderbird のインストール

メールクライアントとして Thunderbird をインストールします。インターネットに接続できない環境では、GUI から admin でログインし、インストールメディアが自動マウントされた状態でインストール作業を進めます。

```
# yum install thunderbird
読み込んだプラグイン:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.cat.net
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> パッケージ thunderbird.x86_64 0:52.9.1-1.el7.centos を インストール
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package            アーキテクチャー
                                 バージョン                   リポジトリー
                                                                          容量
================================================================================
インストール中:
 thunderbird        x86_64        52.9.1-1.el7.centos         base         76 M

トランザクションの要約
================================================================================
```

```
インストール  1 パッケージ

総ダウンロード容量: 76 M
インストール容量: 144 M
Is this ok [y/d/N]: y    ← 確認してyを入力
Downloading packages:
thunderbird-52.9.1-1.el7.centos.x86_64.rpm                 |  76 MB    00:22
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中          : thunderbird-52.9.1-1.el7.centos.x86_64          1/1
  検証中                  : thunderbird-52.9.1-1.el7.centos.x86_64          1/1

インストール:
  thunderbird.x86_64 0:52.9.1-1.el7.centos

完了しました!
```

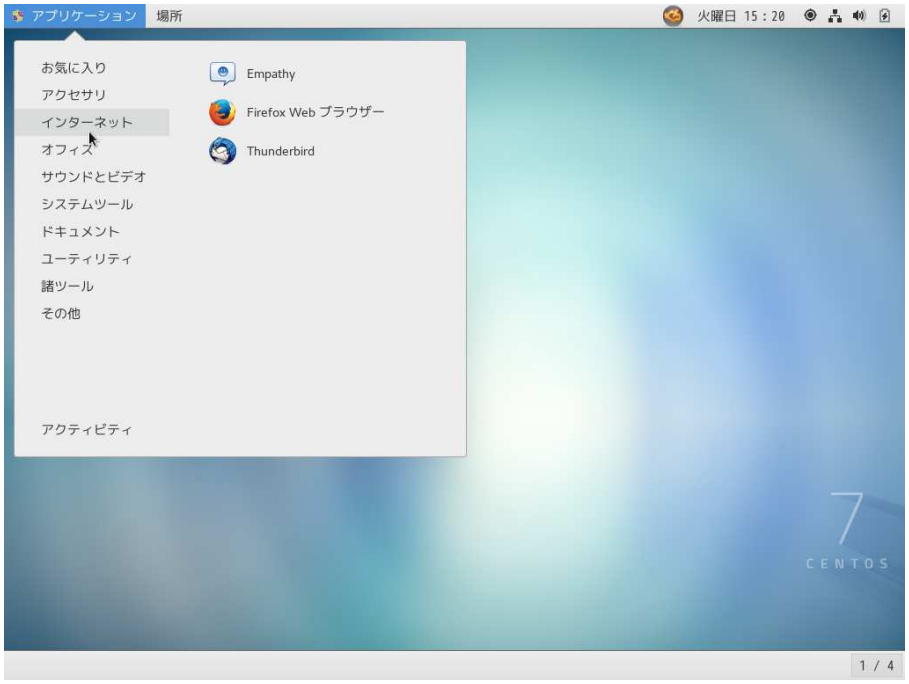
5.7.6　Thunderbird の起動

次に Thunderbird の設定を行います。以下の手順は受講者 A の場合です。

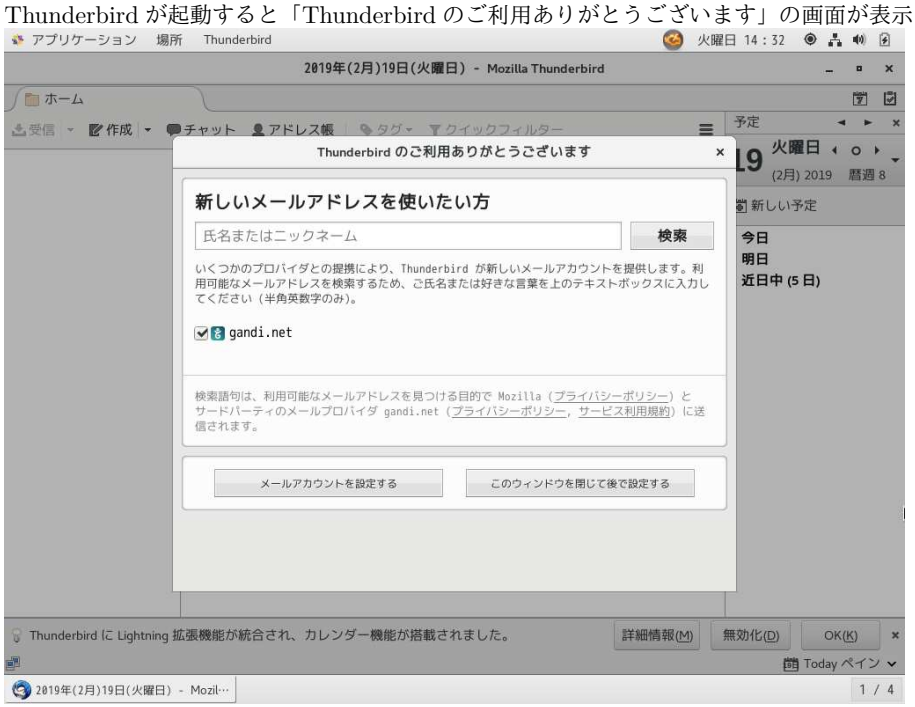
1. admin でログインしている場合にはログアウトします。ログアウトは、GNOME メニューバーの電源アイコン付近をクリックし、「admin」をクリックすると表示される「ログアウト」をクリックします。

2. メールの送受信テスト用に作成したユーザーアカウント usera でログインします。パスワードは userapass です。正しく設定されていない場合には、再度 admin でログインし、root ユーザになって passwd コマンドで設定し直して下さい。このパスワードがメールの送受信にも使用されます。

3.「アプリケーション」メニューから「インターネット」→「Thunderbird」を選択します。



4. Thunderbird が起動すると「Thunderbird のご利用ありがとうございます」の画面が表示されます。



5.「メールアカウントを設定する」をクリックすると、「メールアカウント設定」ダイアログが表示されます。

6. 以下のように設定して「続ける」をクリックします。

表 12: アカウント設定の設定値

設定項目	値
あなたの名前	UserA
メールアドレス	usera@alpha.jp
パスワード	userapass
パスワードを記憶する	チェックしておく

すると、「アカウント設定を Mozilla ISP データベースから検索しています。」と表示されます。検索はしばらく時間がかかります。

7.「アカウント設定が、一般的なサーバー名で検索したことにより見つかりました。」と表示されます。

図 24　メールアカウント画面

もし、アカウントの検索時「Thunderbird はあなたのアカウント設定を見つけられませんでした。」のように表示された場合には、次のような設定になるように手動設定をし、「再テスト」ボタンをクリックします。

表 13: メールアカウント設定の設定値

カテゴリ	項目	設定値
ユーザ名		usera
受信サーバー	サーバーのホスト名	mail.alpha.jp
	プロトコル	IMAP
	受信ポート番号	143
	SSL	接続の保護なし
	認証方式	通常のパスワード
送信サーバー	サーバーのホスト名	mail.alpha.jp
	送信ポート番号	25
	SSL	接続の保護なし
	認証方式	通常のパスワード

8.「完了」をクリックします。すると、接続が暗号化されないため、警告が表示されます。

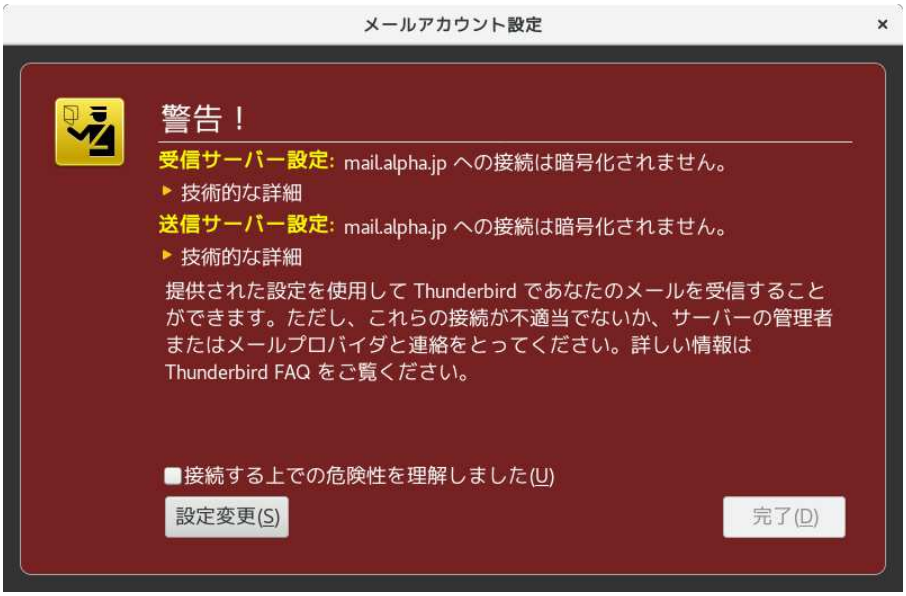


図 25　警告画面

「接続する上での危険性を理解しました」をチェックし、「完了」ボタンをクリックします。

5.7.7　メールの送信

メールを送信するには、「作成」ボタンをクリックしてメール作成ウインドウを呼び出します。

1. 宛先に自分のメールアドレス（usera@alpha.jp）を指定して、メールを作成、送信してみます。
2.「受信」ボタンをクリックして、メールが受信できることを確認します。
3. 宛先に他の受講生のメールアドレス（userb@beta.jp）を指定して、メールを作成、送信してみます。
4. 相手がメールを受信できたこと、相手からのメールを受信できることを確認します。

5.7.8　起動時のスタートページの設定

インターネットに接続できない環境で実習をしている場合には、起動時に「サーバーが見つかりませんでした」のエラーが表示されることがあります。エラーが表示されないようにするには、以下の手順で設定を修正します。

1. 三本線のボタンからメニューを表示し、「設定」→「設定」を選択します。

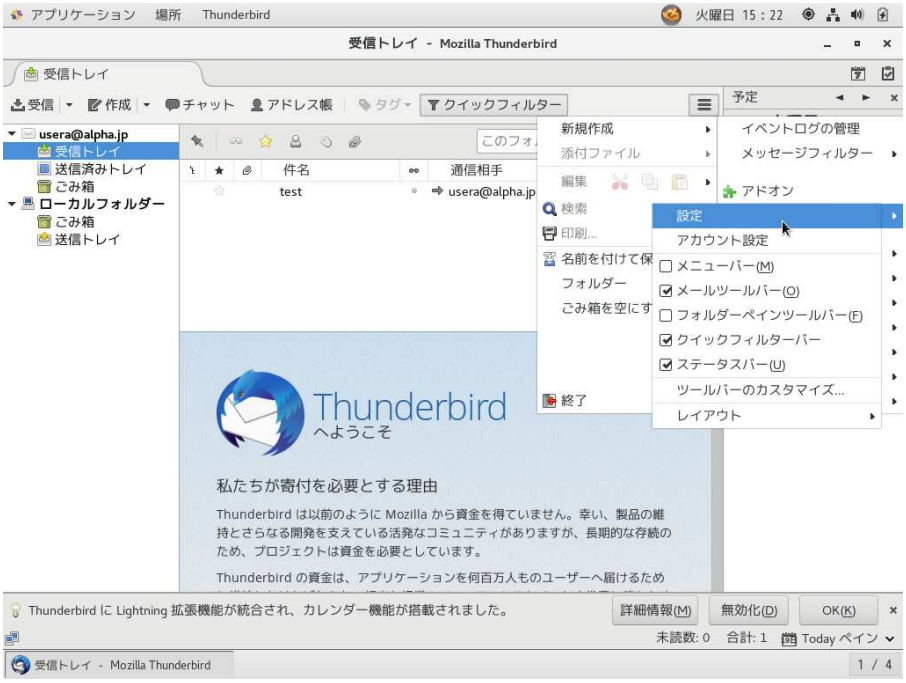


図 26　メニューから設定を選ぶ

1.「Thunderbird スタートページ」の「起動時にメッセージペインにスタートページを表示する」のチェックを外して、「閉じる」をクリックします。

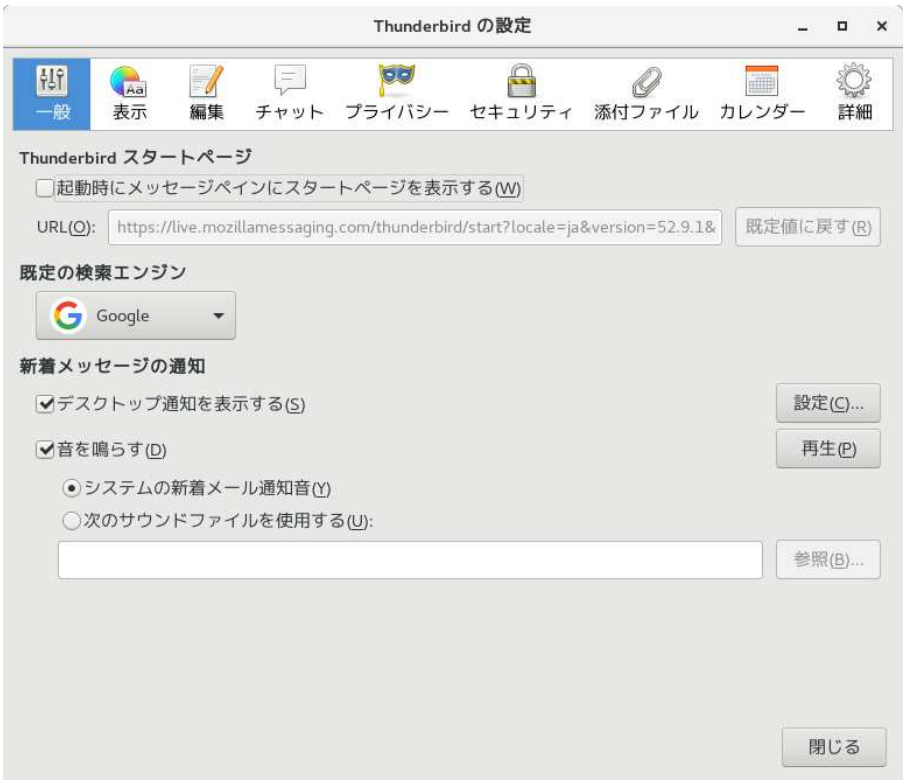


図 27　メニューから設定を選ぶ

5.8　まとめ

本章では、電子メールに関する学習を行いました。また、実際にメールサーバーを設定し、mail コマンドや Thunderbird を利用してメールの送受信の確認を行いました。メールサーバーの設定は、メールサーバーが正しく設定され起動していたとしても、DNS サーバーが正しく動いていなければ利用できないなどの理由から難しかったと思います。設定ファイルの記述に問題がない

のに、メールがどうしても送れない、受信できない場合は、まず DNS が正しく動いているか、host コマンドや dig コマンドを実行して確認します。また、ログ（/var/log/mail）を見て、エラーが出ていないか確認することも大切です。

6 ネットワークとセキュリティの管理

CentOS では、基本的なネットワークやセキュリティの設定はインストール時に行われます。ここでは、これらの設定を管理するための方法について説明します。

6.1 用語集

ネットワークインターフェース

LAN ケーブルを接続して、外部のマシンとの間でデータをやり取りするための物理的なインターフェースです。

ループバックインターフェース

マシン内部でデータをやり取りするための仮想的なインターフェースです。

IP(Internet Protocol)

IP は、ネットワークに接続したコンピューター間でデータをやり取りするためのプロトコルです。

IP アドレス

IP アドレスは、IP 通信で各コンピューターに割り当てられる値です。データの送り先として IP アドレスを指定すると、その IP アドレスが割り当てられたコンピューターに送信されたデータが届きます。

IPv4(Internet Protocol version 4)

現在のインターネットで利用されている通信プロトコルです。IPv4 では、IP アドレスを 4 バイト（32 ビット）で表します。本来は 32 個の 2 進数（0 と 1）の羅列ですが、人間に分かりやすくするために 1 バイトごとに 10 進数に変換して. (ドット) で区切って「192.168.1.1」の様に表記します。次世代の IP である IPv6 では、IP アドレスを 128 ビットで表します。

ネットワークアドレス

ホストが属しているネットワーク自体を指し示す IP アドレスです。

ブロードキャスト

ホストが属しているネットワーク全体を指し示す IP アドレスです。

ネットマスク

IP アドレスのうち、どこまでがネットワーク部で、どこまでがホスト部かを示すための値です。IP アドレスとネットマスクの 2 つの値から、ネットワークアドレス、ブロードキャストアドレスを割り出すことができます。

デフォルトゲートウェイ

インターネットは、小さなネットワークが相互に接続したネットワークです。小さなネットワーク間を接続する機器としてルーター (ゲートウェイ) が使われます。ゲートウェイは 1 つのネットワークに複数設置することができますが、特に指定が無い場合にはデフォルトゲートウェイを使って外部のネットワークとの通信を行います。

DHCP(Dynamic Host Configuration Protocol)

IP アドレスなどのネットワーク設定を自動的に割り当てるプロトコルです。

TCP(Transmission Control Protocol)

TCP は、コネクション方式で通信するプロトコルです。IP と組み合わせた TCP/IP がインターネットの標準的な通信プロトコルです。TCP の特長として、届かなかった通信パケットを再送信して確実に通信を行う仕組みがあります。

UDP(User Datagram Protocol)

UDP は、コネクションレス方式で通信するプロトコルです。TCP とは異なり、データの再送信が行われないので通信の確実性は劣りますが、セッションを確立するための「3 ウェイハンドシェイク」の手間が不要なためシンプルな通信に適しています。たとえば、大量に問い合わせが行われる DNS への名前解決の問い合わせは UDP となっています。

ポート番号

ポート番号は、TCP と UDP が通信する際に使用する値です。たとえば Web サーバーはポート番号 80 番を使用して動作しているので、Web ブラウザーは目的の Web サーバーのポート番号 80 番に接続します。ポート番号は 0 番から 65535 番まで使用できますが、0〜1023 番は WELL KNOWN PORT、1024〜49151 番は REGISTERED PORT として予約されています。

ICMP(Internet Control Message Protocol)

データの転送エラーやデータ転送量などの情報を通知するためのプロトコルです。

ping コマンド

ping コマンドは、ICMP を使って宛先に指定したホストに到達することができるかどうかを確認するコマンドです。

6.2 ネットワーク管理

ネットワークが上手く使えない場合には、確認のためにネットワークインターフェースが正しく設定されているかを調べる必要があります。また、設定が間違っている場合には、インターフェースの設定を変更する必要があります。ここでは、ネットワークインターフェースの確認と設定の方法について解説します。

6.2.1 ネットワークインターフェースの確認

Linux をインストールしたマシンが正常にネットワークに接続できるかどうか、設定を確認します。ip コマンドで確認します。

ネットワークインターフェースの確認

```
# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
     1000
    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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
     default qlen 1000
    link/ether 08:00:27:cb:f6:31 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 brd 192.168.1.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 2001:db8::10/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::40f4:1400:1f0d:132b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group
     default qlen 1000
    link/ether 52:54:00:96:78:2c brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN
     group default qlen 1000
    link/ether 52:54:00:96:78:2c brd ff:ff:ff:ff:ff:ff
```

ip コマンドで表示された lo は仮想的なループバックインターフェースです。また、この例では enp0s3 が物理的なインターフェースです。この名称は、インストールした PC によって変わります。enoXX、ensXX、enpXsX、ethX、enxXX などの名称になる場合もあります。

6.2.2 ネットワークインターフェースの再設定

インストール時に IP アドレスの設定を間違えた時などは、ネットワークインターフェースを再設定します。設定は、GNOME の管理画面から行うことができます。

GNOME のデスクトップのアプリケーションメニューから「システムツール」→「設定」を選択します。表示された設定画面の左側のメニューから「ネットワーク」を選択します。

「有線」の欄にある歯車のボタンをクリックすると、接続プロファイルの設定画面が表示されます。「IPv4」のタブをクリックすると、次のような画面になります。

この画面で、設定を変更することで、ネットワークインターフェースの設定を変更することができます。変更したら、「適用」ボタンを押して元の画面に戻ります。「有線」の項目にあるスイッチを、一旦「オフ」に変えます。再度、「オン」に変えるとネットワークインターフェース設定が変更されます。

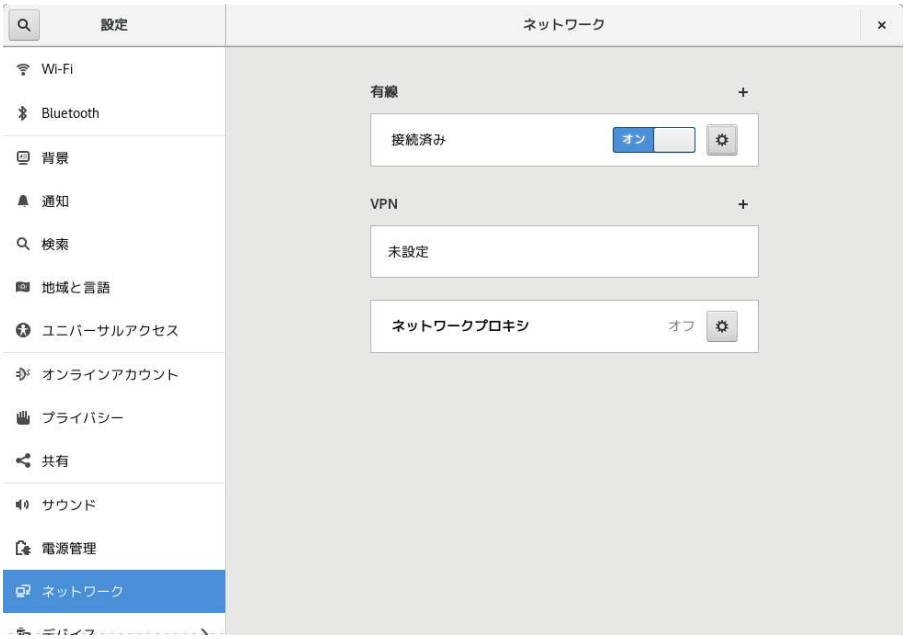


図 28　設定画面

図 29　ネットワーク詳細設定画面

6.2.3 ネットワークインターフェースの動作確認

ネットワークインターフェースが動作しているかは ping コマンドで確認します。ping コマンドで確認する IP アドレスとして自分の物理ネットワークインターフェースの IP アドレス、講師のマシンの IP アドレス (192.168.1.10) やその他のマシンの IP アドレスなどを指定します。ping コマンドは [Control]+[c] で中止できます。

ping コマンドによる確認

```
# ping 192.168.1.101    ← 自分のIPアドレス
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.209 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.151 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.174 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=0.144 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=64 time=0.144 ms
^C
--- 192.168.1.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.144/0.164/0.209/0.027 ms
```

6.2.4 サービスのポート番号を確認

どんなネットワークサービスが自分の PC で動いているかを、ss コマンドと lsof コマンドで確認します。ss -at コマンドを実行すると、現在の TCP 通信の状態をすべて表示します。

ss コマンドによる確認

```
# ss -at
State      Recv-Q Send-Q Local Address:Port                 Peer Address:Port
LISTEN     0      50         *:microsoft-ds             *:*
LISTEN     0      50         *:netbios-ssn              *:*
LISTEN     0      100        *:pop3                     *:*
LISTEN     0      100        *:imap                     *:*
LISTEN     0      128        *:sunrpc                   *:*
LISTEN     0      5      192.168.122.1:domain                 *:*
LISTEN     0      10     192.168.1.101:domain                 *:*
LISTEN     0      10     127.0.0.1:domain                 *:*
LISTEN     0      128        *:ssh                      *:*
 （略）
```

lsof -i コマンドを実行すると現在開かれているすべてのポートを表示します。

lsof コマンドによる確認

```
# lsof -i
COMMAND    PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
rpcbind    655    rpc    6u  IPv4  15791      0t0  UDP *:sunrpc
rpcbind    655    rpc    7u  IPv4  15855      0t0  UDP *:821
rpcbind    655    rpc    8u  IPv4  15856      0t0  TCP *:sunrpc (LISTEN)
rpcbind    655    rpc    9u  IPv6  15857      0t0  UDP *:sunrpc
rpcbind    655    rpc   10u  IPv6  15858      0t0  UDP *:821
rpcbind    655    rpc   11u  IPv6  15859      0t0  TCP *:sunrpc (LISTEN)
avahi-dae  672  avahi   12u  IPv4  17393      0t0  UDP *:mdns
avahi-dae  672  avahi   13u  IPv4  17394      0t0  UDP *:37814
chronyd    707 chrony    1u  IPv4  17271      0t0  UDP localhost:323
chronyd    707 chrony    2u  IPv6  17272      0t0  UDP localhost:323
```

```
dhclient   943   root    6u  IPv4  20446      0t0  UDP *:bootpc
sshd      1155   root    3u  IPv4  22236      0t0  TCP *:ssh (LISTEN)
sshd      1155   root    4u  IPv6  22370      0t0  TCP *:ssh (LISTEN)
（略）
```

ss コマンドは-a オプションでサービスの状態をすべて表示、-t オプションで TCP(Transmission Control Protocol) のサービスが使うポートなどの情報のみを表示します。lsof コマンドは-i オプションでサービスを受けているポートと対応するプログラムの情報を表示します。ポート番号とサービスの対応 (WELL KNOWN PORT NUMBERS:0〜1023 や REGISTERED PORT NUMBERS:1024〜49151) が定義されている/etc/services ファイルも確認してみてください。

lsof コマンドによる確認

```
# cat /etc/services
（略）
tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
rje             5/tcp                           # Remote Job Entry
rje             5/udp                           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
```

6.3 SSH によるリモートログイン

SSH はネットワーク経由でリモートにある Linux サーバーにログインするために使用するプロトコルです。通信が暗号化されているため、覗き見されてもパスワードや作業内容が分からない他、公開鍵を使った認証を行うことでパスワードをネットワークに流すことなくログインすることができます。Linux では、OpenSSH のサーバーとクライアントが用意されています。

6.3.1 TELNET との違い

SSH と同様のリモートログインに TELNET が使用できますが、TELNET は通信が暗号化されていないためパスワードや作業内容などを覗き見することができてしまうという問題があります。SSH は特別な設定をしなくても TELNET と同様のパスワードによるリモートログインが行えるので、通信が暗号化されている SSH が標準的に使われており、現在では TELNET を使用する必要は無くなっています。CentOS 7 では、デフォルトでは telnet コマンドはインストールされていません。

6.3.2 パスワードによる認証

ssh コマンドは特別な設定を行わなくても、パスワード認証でリモートログインすることができます。

以下のようにして、自分自身に SSH で接続してみます。初めての接続の場合には、SSH サーバーの電子証明書が送られてきて接続してもよいか訪ねられるので「yes」と入力します。パスワード認証が可能だと、パスワードの入力が要求されます。

ssh による接続

```
[root@host1 admin]# ssh usera@localhost ← usera として localhost に接続
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:qeRiiKeZpaNMdtkClnl4n0iRsjGPrnvGcLpcbhwXH8g.
ECDSA key fingerprint is MD5:25:d1:b0:2e:b8:f8:19:fb:f7:e0:a7:a6:19:06:b7:96.
Are you sure you want to continue connecting (yes/no)? yes  ← yes を入力
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
usera@localhost's password: userapass   ← 実際には非表示
Last failed login: Tue Feb 19 15:23:31 JST 2019 from 192.168.7.1 on ssh:notty
There were 5 failed login attempts since the last successful login.
Last login: Tue Feb 19 14:36:42 2019
```

```
[usera@host1 ~]$ exit    ← リモートログインを修了
ログアウト
Connection to localhost closed.
[root@host1 admin]#      ← 元のユーザーrootに復帰
```

6.3.3 公開鍵による認証

パスワード認証は、通信経路が SSH で暗号化されているといっても、パスワードがネットワークを流れていること、またパスワードを自動的に生成して順番に試していく「総当たり攻撃」を受けた場合不正にログインされてしまう可能性があるので、インターネット上に公開されているサーバーで使用するには相応しくありません。

公開鍵認証は、あらかじめサーバーに設置した公開鍵と対になっている秘密鍵を持っているユーザーしかリモートログインできない認証方法です。

以下の手順で公開鍵認証を設定します。

1. 公開鍵と秘密鍵を生成する
   ssh-keygen コマンドを使用して一対の公開鍵 (id_dsa.pub) と秘密鍵 (id_dsa) を生成します。鍵のファイルはホームディレクトリに作られた.ssh ディレクトリに保存されます。
   秘密鍵には不正利用を防止するためのパスフレーズを設定します。接続時にパスフレーズを正しく入力できないと、秘密鍵は利用できないので、公開鍵認証による接続はできません。このパスフレーズは SSH クライアント側で秘密鍵に対して処理されるので、ネットワーク上には情報は流れません。

公開鍵と秘密鍵の作成

```
[root@host1 ~]# su - usera    ← ユーザーuseraにユーザを切り替え
[usera@host1 ~]$  ssh-keygen -t dsa  ← DSA暗号形式で鍵を生成
Generating public/private dsa key pair.
Enter file in which to save the key (/home/usera/.ssh/id_dsa): ← Enterキーを入力
Created directory '/home/usera/.ssh'.
Enter passphrase (empty for no passphrase):  ← パスフレーズを入力（非表示）
Enter same passphrase again:     ← パスフレーズを入力（非表示）
Your identification has been saved in /home/usera/.ssh/id_dsa.
Your public key has been saved in /home/usera/.ssh/id_dsa.pub.
The key fingerprint is:  ↓ 鍵についた指紋。鍵の称号に使用可能
SHA256:Y5FJTNhlaIA7z3dT/bCTf0br+X4ZjIQCFFUKDSmtttE usera@host1.alpha.jp
The key's randomart image is:
+---[DSA 1024]----+
|      .+@O++.     |
|     ...=*=.      |
|      .+.=.  ..   |
|     o+ E o ...o  |
|     .+o S ... o= |
|      .o...o  .+oo|
|        . . .   ++|
|                o*|
|               .=*|
+----[SHA256]-----+
[usera@host1 ~]$
```

2. 接続先に~/.ssh/authorized_keys を作成する
   ユーザーに SSH での接続を許可するには、ユーザーアカウントを作成し、そのユーザーのホームディレクトリに~/.ssh/authorized_keys ファイルを作成しておきます。~/.ssh/のパーミッションは 700(drwx——)、authorized_keys ファイルのパーミッションは 600(-rwx——) に設定する必要があります。

認証用ファイルの作成

```
[usera@host1 ~]$ ls -ld .ssh
drwx------. 2 usera usera 38  2月 19 17:28 .ssh
[usera@host1 ~]$ cd .ssh
[usera@host1 .ssh]$ cat id_dsa.pub >> authorized_keys
[usera@host1 .ssh]$ chmod 600 authorized_keys
[usera@host1 .ssh]$ ls -l authorized_keys
-rw-------. 1 usera usera 610  2月 19 17:34 authorized_keys
```

3. 公開鍵認証で接続する
   公開鍵認証で接続します。ssh コマンドの使用法自体はパスワード認証と同じですが、パスワードの代わりに秘密鍵に設定したパスフレーズの入力が必要です。

公開鍵での接続

```
[usera@host1 ~]$ ssh usera@localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:qeRiiKeZpaNMdtkClnl4n0iRsjGPrnvGcLpcbhwXH8g.
ECDSA key fingerprint is MD5:25:d1:b0:2e:b8:f8:19:fb:f7:e0:a7:a6:19:06:b7:96.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Enter passphrase for key '/home/usera/.ssh/id_dsa':  ←  パスフレーズを入力
Last login: Tue Feb 19 17:34:19 2019
[usera@host1 ~]$ exit
ログアウト
Connection to localhost closed.
```

6.3.4 パスワード認証の禁止

パスワード認証が有効になっていると、パスワードの総当たり攻撃により不正にリモートログインできてしまいます。公開鍵認証で接続できるようになった後には、OpenSSH サーバーの設定を変更してパスワード認証を禁止しておきます。

1. パスワード認証で接続できることを確認します。

パスワード認証での確認

```
# ssh localhost
root@localhost's password:
Last login: Tue Feb 19 17:13:43 2019
[root@host1 ~]# exit
ログアウト
Connection to localhost closed.
```

2. 設定ファイル/etc/ssh/sshd_config を修正します。

パスワード認証の禁止

```
（略）
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no    ←  変更
（略）
```

3. 設定を変更後、sshd 設定の再読み込み

sshd のリロード

```
# systemctl reload sshd
```

4. 公開鍵認証を設定していないユーザーで OpenSSH サーバーに接続します。

接続の確認

```
# ssh localhost
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

6.4 ファイアウォールの設定

ファイアウォールはネットワークにおいて様々なアクセス制限を行い、ネットワークからの攻撃や不正なアクセス等を防ぐ機能です。CentOS7 のファイアウォール機能は、firewalld によって管理されています。firewalld では、ネットワークインターフェースへのパケットの受信の許可、拒否のルールを管理しています。firewalld の設定は firewall-cmd というコマンドで、設定を行います。

6.4.1 ファイアウォール設定の確認

許可されているサービスを調べるには、–list-services オプションを使います。

許可サービスの確認

```
# firewall-cmd --list-services
ssh dhcpv6-client dns smtp pop3  http imap
```

6.4.2 許可サービスの追加

許可サービスを追加するには、次のように–add-service オプションを使います。

許可サービスの追加

```
# firewall-cmd --add-service=imap
success
```

この例では、imap サービスを許可しています。設定可能なサービスについては、次のようにして–get-services オプションで調べることができます。

利用可能なサービスの調査

```
# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bgp bitcoin
    bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine
    condor-collector ctdb dhcp dhcpv6 dhcpv6-client dns docker-registry docker-swarm
    dropbox-lansync elasticsearch freeipa-ldap freeipa-ldaps freeipa-replication
    freeipa-trust ftp ganglia-client ganglia-master git gre high-availability http
    https imap imaps ipp ipp-client ipsec irc ircs iscsi-target jenkins kadmin
    kerberos kibana klogin kpasswd kprop kshell ldap ldaps libvirt libvirt-tls
    managesieve mdns minidlna mongodb mosh mountd ms-wbt mssql murmur mysql nfs nfs3
    nmea-0183 nrpe ntp openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole pmcd
    pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy proxy-dhcp ptp pulseaudio
    puppetmaster quassel radius redis rpc-bind rsh rsyncd samba samba-client sane sip
    sips smtp smtp-submission smtps snmp snmptrap spideroak-lansync squid ssh
    syncthing syncthing-gui synergy syslog syslog-tls telnet tftp tftp-client tinc
    tor-socks transmission-client upnp-client vdsm vnc-server wbem-https xmpp-bosh
    xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server
```

6.4.3　許可サービスの取り消し

許可されているサービスを停止するには、–remove-service オプションを使います。

許可サービスの追加

```
# firewall-cmd --remove-service=imap
success
```

6.4.4　ファイアウォール設定の保存

–add-service、–remove-service などで行ったファイアウォールルールの変更は、一時的なものです。そのため、再起動をすると失われてしまいます。再起動後も設定をに有効にするには、次のように–runtime-to-permanent オプションを使って、設定を保存します。

許可サービスの追加

```
# firewall-cmd --runtime-to-permanent
```

Linux サーバー構築標準教科書

2012 年 6 月 1 日 V2.0.0 発行

2012 年 6 月 20 日 V2.0.1 発行

2019 年 10 月 1 日 V3.0.0 発行

2022 年 5 月 1 日 V3.0.1 発行

2022 年 8 月 9 日 V3.0.2 発行

www.lpi.or.jp