サーバー構築を実習形式で学べる決定版

Linux
サーバー構築標準教科書

Ver. 4.0.1

LinuC
Linux技術者認定
https://linuc.org

open your NEXT future
LPI-JAPAN
Linux Professional Certification
https://lpi.or.jp

Linuxサーバー構築標準教科書

2025年4月1日　Ver.4.0.1
LPI-Japan　発行

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」（以下、本教科書）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。

本教科書は、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、本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。

本教科書の最新情報は以下の Web ページをご参照ください。

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

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ができます。

本教科書でカバーされない範囲

実習手順の記述を簡潔にするため、LinuC レベル 1 の学習範囲に含まれる Linux の基本的な操作やシステム管理についての記述は必要最低限になっています。また、自分で調べることも学習において重要であるため、あえて調べてみる余地を残しています。分からない部分は別途調べて理解を深めてください。

想定している実習環境

本教科書は一人で独学自習できることを想定しています。実習環境として、以下の環境を構築しています。

仮想マシンを利用

仮想マシンを利用して学習環境を構築します。仮想マシンを利用すると、Windows や Linux、macOS 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、複数のサーバーをネットワークで接続して行う DNS やメールの実習も 1 台の物理マシンで行えます。

仮想マシン環境を実現するソフトウェアとして、以下のようなものがあげられます。

- VirtualBox（Windows、Linux、macOS）
- VMware Workstation（Windows）
- VMware Fusion（macOS）
- Parallels Desktop（macOS）
- UTM（macOS）
- Linux KVM（Linux）

本教科書では、VirtualBox を Windows 上で実行して解説を進めます。

OS

本教科書では、Linux ディストリビューションとして AlmaLinux のバージョン 9.3 を利用します。

実習例では Intel/AMD x86_64 アーキテクチャに対応したバージョンを利用していますが、ARM 版などその他のアーキテクチャに対応したバージョンでも実習を行うことができます。

ネットワーク

実習を行うネットワークはインターネットに接続できることを前提としています。

インターネットに接続できない場合には、OS のインストール時に実習で必要となるソフトウェアをあらかじめインストールすることで実習を進めることができます。OS インストール手順解説の指示に従って事前にソフトウェアをインストールしてください。

教室での実習

教室などで複数の受講者が実習を行う場合でも、基本的に各受講者がそれぞれ自身の仮想マシンを使って個別に実習を行うことを想定しています。

受講者同士がネットワークを経由して相互に接続する実習を行いたい場合、講師が仮想マシンや OS の設定変更を個別に指示するか、各受講者が教科書の内容を一通り実習した後に、受講者相互で接続する実習を行うようにしてください。その際には特に以下の点について留意してください。

- VirtualBox で作成する仮想マシンのネットワークをブリッジ接続して相互接続できるようにする
- 各受講者に異なる IP アドレスとホスト名・ドメイン名を割り当てる
- DNS による名前解決の設定を変更する

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 サーバー構築の概要

本教科書で行う実習の全体像や事前に説明しておくべき事項を解説しています。

2 章仮想マシン環境の準備

仮想マシンについての解説と、VirtualBox のインストール、仮想マシンの作成を行います。

3 章 Linux のインストールと設定

仮想マシンに Linux をインストールします。

4 章 Web サーバーのインストールと設定

Linux に Web サーバーとして Apache HTTP サーバーをインストールします。

5 章 DNS サーバーのインストールと設定

Linux に DNS サーバーとして BIND をインストールして、ドメインを設定し、名前解決が行えるようにします。複数の仮想マシンを用意し、相互に名前解決で接続できるようにします。

6 章メールサーバーのインストールと設定

Linux にメールサーバーとして Postfix や Dovecot をインストールし、メールの送受信が行えるように設定します。

7 章ネットワークとセキュリティの設定

Linux のネットワークやセキュリティを設定します。

執筆者・制作者紹介

本教科書は、オープンなプロジェクト形式で開発を行っています。企画段階から意見交換を行い、事前の技術的な調査、執筆、レビューなどをプロジェクトのメンバーで分担して行っています。

宮原徹（バージョン 4 執筆担当）

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 4 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だけでなく、より分かりやすい実習になるように調整を行ってみました。

塚田美香（バージョン 4 執筆担当）

今回私は、第 2 章から第 4 章の実習部分を担当させてもらいました。このようなプロジェクトに関わるのは初めてで、大変良い経験となりました。本教科書がこれから Linux を学ばれる多くの皆様の助けになれば幸いです。

バージョン 4 の開発にご協力をいただいた方々（50 音順）

- 板橋 章夫
- 川西 康友（理化学研究所）
- 鯨井 貴博（株式会社ゼウス・エンタープライズ）
- 小林 充雄（株式会社クロテック）
- 坂本 直志 (東京電機大学)
- 竹本 季史（インターノウス株式会社）
- 谷口 厚志（株式会社メンバーズ）
- 富樫 晃（神奈川県立西部総合職業技術校）
- 畑屋 歩
- 福永 昭臣（株式会社ボールド）
- 水澤 泰敬

また、これまでバージョン 1 からバージョン 3 まで、沢山の執筆者、レビュアー、そして利用者の皆様からフィードバックをいただきました。厚く御礼申し上げます。

著作権

使用に関する権利

本教科書は、クリエイティブ・コモンズ・ライセンスの「表示 - 非営利 - 改変禁止 4.0 国際 (CC BY-NC-ND 4.0)」によってライセンスされています。

図 1: CC BY-NC-ND 4.0

表示

本教科書は、特定非営利活動法人エルピーアイジャパンに著作権が帰属するものであることを表示してください。

非営利

本教科書は、非営利目的で教材として自由に利用することができます。

商業上の利得や金銭的報酬を主な目的とした営利目的での利用は、特定非営利活動法人エルピーアイジャパンによる許諾が必要です。ただし、本教科書を利用した教育において、本教科書自体の対価を請求しない場合は、営利目的の教育であっても基本的に使用できます。その場合も含め、LPI-Japan 事務局までお気軽にお問い合わせください。

＊営利目的の利用とは以下のとおり規定しております。営利企業または非営利団体において、商業上の利得や金銭的報酬を目的に本教科書の印刷実費以上の対価を受講生に請求して本教科書の複製を用いた研修や講義を行うこと。

改変禁止

本教科書は、改変せず使用してください。本教科書に対する改変は、特定非営利活動法人エルピーアイジャパンまたは特定非営利活動法人エルピーアイジャパンが認める団体により行われています。

フィードバック

フィードバックは誰でも参加できる Slack で受け付けていますので、積極的にご参加ください。Slack 参加の詳細は以下の本教科書の Web ページを参照してください。

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

図 2: https://linuc.org/textbooks/server/

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

Linux 技術者認定「LinuC（リナック）」のご紹介

Linux 技術者認定「LinuC（リナック）」とは、クラウド／ DX 時代の IT エンジニアに求められるシステム構築から運用管理に必要なスキルを証明できる技術者認定です。アーキテクチャ設計からシステム構築、運用管理までの技術領域を広くカバーしており、４つのレベルの認定取得を通じて一歩ずつ確実に求められるスキルを習得し、それを証明することができます。

LinuC の出題範囲策定や試験開発は、実際に現場で活躍しているハイレベルな IT エンジニアが参加するコミュニティによって行われています。そのため、グローバルで業界標準として利用されている技術領域をカバーし、システム開発や運用管理の現場で本当に必要とされる知識や実践的なスキルを問う内容になっています。その結果として従来型の Linux 領域にとどまった技術認定とは異なり、国内・海外を問わず活躍を目指す IT エンジニアにとっても十分役立つ技術者認定となりました。

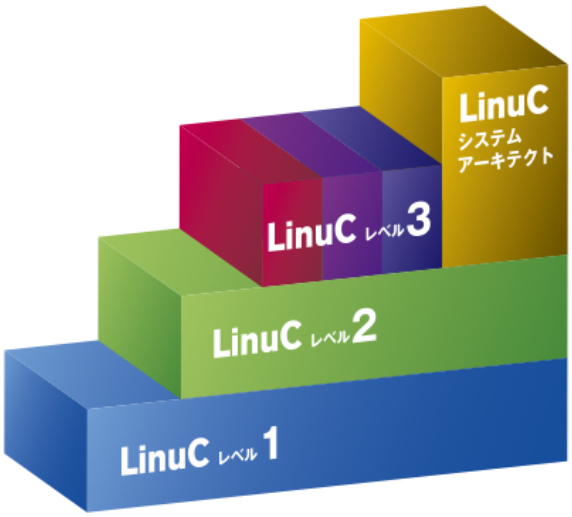

図 3: LinuC の体系図

LinuC レベル 1

コンピュータシステムを理解し、仮想環境を含む Linux システムの基本操作とシステム管理が行える即戦力エンジニアの証明（ITSS レベル 1）

LinuC レベル 2

仮想環境を含む Linux のシステム設計・ネットワーク構築において、アーキテクチャに基づいた設計・導入・保守・問題解決ができるエンジニアの証明（ITSS レベル 2）

LinuC レベル 3

異種混在環境の運用スキル、Linux ベースのセキュアなシステム設計・構築スキル、大規模な仮想化システムや高可用性システムの構築スキルといったスペシャリストの証明（ITSS レベル 3）

LinuC システムアーキテクト

オンプレ／クラウド、物理／仮想化を含むシステムのライフサイクル全体を俯瞰して最適なアーキテクチャを設計・構築ができる上級エンジニアの証明（ITSS レベル 4 相当のスキルレベル）

LinuC の詳細については、以下の Web サイトを参照してください。

https://linuc.org/about/01.html

図 4: https://linuc.org/about/01.html

目次

1 Linux サーバーの概要 **1**
1.1 用語集 . . . 1
1.2 実習で利用するハードウェア . . . 2
1.2.1 マシン本体 . . . 2
1.2.2 CPU . . . 2
1.2.3 メモリ . . . 2
1.2.4 ストレージ . . . 3
1.2.5 ネットワーク . . . 3
1.3 利用する Linux のディストリビューション . . . 3
1.3.1 インストール用 ISO イメージの入手 . . . 3
1.4 ISO イメージのファイル名 . . . 4
1.5 バージョン . . . 4
1.6 アーキテクチャ . . . 5
1.6.1 x86_64 . . . 5
1.6.2 aarch64 . . . 5
1.7 ISO イメージの種類 . . . 5
1.8 ネットワーク環境について . . . 5
1.8.1 インターネットへの接続 . . . 5
1.8.2 仮想マシン間の接続 . . . 6
1.9 ネットワークの設定項目 . . . 6
1.9.1 ドメイン名 . . . 6
1.9.2 ホスト名 . . . 6
1.9.3 IP アドレス . . . 6
1.9.4 サブネットマスク . . . 6
1.9.5 ネットワークアドレス . . . 7
1.9.6 デフォルトゲートウエイ . . . 7
1.9.7 DNS サーバーアドレス . . . 7
1.10 ネットワーク設定まとめ . . . 7
1.10.1 1 台目の仮想マシン . . . 7
1.10.2 2 台目の仮想マシン . . . 8
1.10.3 3 台目の仮想マシン . . . 8

2 VirtualBox のインストールと仮想マシンの作成 **9**
2.1 用語集 . . . 9
2.2 仮想化環境とは . . . 9
2.2.1 仮想化技術とクラウドの関係 . . . 9
2.3 仮想マシンとは . . . 9
2.3.1 ホスト OS 型とハイパーバイザー型 . . . 9
2.4 仮想マシンのメリット . . . 9
2.4.1 ホスト OS と共存できる . . . 9
2.4.2 ハードウェアを用意する必要がない . . . 10
2.4.3 柔軟性 . . . 10
2.4.4 OS 選択 . . . 10
2.5 仮想マシンのデメリット . . . 10
2.5.1 速度 . . . 10
2.5.2 リソースの消費 . . . 10
2.6 仮想マシンのリソース . . . 10
2.6.1 CPU . . . 10
2.6.2 メモリ . . . 10
2.6.3 ディスク . . . 10
2.6.4 光学ドライブ . . . 11
2.6.5 ネットワーク . . . 11
2.7 VirtualBox のインストール . . . 11
2.7.1 使用するコンピュータの仮想化支援技術の有効化 . . . 11
2.7.2 VirtualBox のダウンロード . . . 11
2.7.3 VirtualBox のインストーラの実行 . . . 13
2.8 VirtualBox の起動 . . . 13
2.8.1 VirtualBox マネージャー . . . 14
2.9 ホストキーによるホスト OS の操作への復帰 . . . 14

2.9.1 ホストキーが押せない場合のホスト OS の操作への復帰 . . . 14
2.9.2 ホストキーの変更 . . . 14
2.10 仮想マシンの作成 . . . 15
2.10.1 仮想マシンの名前と OS の設定 . . . 15
2.10.2 仮想マシンのハードウェアの設定 . . . 16
2.10.3 仮想ハードディスクの設定 . . . 17
2.10.4 仮想マシンの設定を確認 . . . 18
2.11 仮想マシンの設定方法 . . . 19
2.11.1 EFI の有効化を確認 . . . 19
2.12 ネットワークの追加 . . . 20
2.12.1 仮想ネットワークの設定を確認 . . . 21
2.12.2 仮想マシンにネットワークアダプターを追加 . . . 22
2.13 ISO イメージを仮想光学ドライブで読み込む設定 . . . 23

3 Linux のインストールと設定 24
3.1 用語集 . . . 24
3.2 仮想マシンの起動 . . . 24
3.2.1 ISO イメージファイルのマウント . . . 24
3.3 OS のインストール . . . 25
3.3.1 インストーラ起動オプションの選択 . . . 25
3.3.2 言語選択 . . . 26
3.3.3 インストール概要 . . . 26
3.3.4 インストール先の設定 . . . 27
3.3.5 ネットワークとホスト名の設定 . . . 28
3.3.6 インターネットへの接続を設定（NAT） . . . 28
3.3.7 ホスト OS との接続を設定（ホストオンリー） . . . 29
3.3.8 ホスト名の設定 . . . 30
3.3.9 ソフトウェアの追加インストール（インターネット接続ができない場合のみ） . . . 31
3.3.10 root パスワードの設定 . . . 32
3.3.11 インストールの開始 . . . 33
3.4 インストール後の初期設定 . . . 34
3.4.1 プライバシー . . . 34
3.4.2 オンラインアカウント . . . 35
3.4.3 ユーザー情報 . . . 36
3.5 ログインとログアウト . . . 38
3.5.1 ログアウトする方法 . . . 38
3.5.2 ログインする方法 . . . 38
3.6 コマンド実行のための端末を起動 . . . 39
3.7 ネットワーク接続の確認 . . . 39
3.7.1 名前解決の確認 . . . 39
3.7.2 インターネットへの接続の確認 . . . 40
3.7.3 ゲスト OS からホスト OS への接続の確認 . . . 40
3.7.4 ホスト OS からゲスト OS への接続の確認 . . . 40

4 Web サーバーの構築 41
4.1 用語集 . . . 41
4.2 Web サーバーの仕組み . . . 42
4.3 パッケージのインストール . . . 42
4.3.1 dnf コマンド . . . 42
4.3.2 dnf コマンドと yum コマンド . . . 42
4.3.3 sudo コマンドによる root 権限の取得 . . . 42
4.3.4 dnf コマンドが参照するパッケージリポジトリ . . . 43
4.3.5 プロキシーが必要な場合には . . . 43
4.3.6 インターネットにアクセスできない環境でのパッケージ管理 . . . 43
4.3.7 依存関係の解消 . . . 43
4.4 dnf install コマンドによるパッケージのインストール . . . 43
4.5 Web サーバーの起動 . . . 44
4.5.1 systemctl コマンド . . . 44
4.5.2 systemctl start コマンドによる Web サーバーの起動 . . . 44
4.6 systemctl status コマンドによる Web サーバーの動作確認 . . . 45
4.6.1 systemctl status コマンドによる動作状態の確認 . . . 45

4.6.2 Loaded の意味 . . . 45
4.6.3 Active の意味 . . . 45
4.7 Web サーバーへの接続の確認 . . . 45
4.7.1 curl コマンドによるローカル接続確認 . . . 45
4.7.2 ゲスト OS の Web ブラウザによるローカル接続確認 . . . 46
4.8 Web サーバーの停止 . . . 46
4.8.1 systemctl status コマンドによる動作状態の確認 . . . 47
4.8.2 curl コマンドによる接続確認 . . . 47
4.8.3 Web サーバーを再度起動 . . . 47
4.9 ファイアウォールの設定とリモート接続確認 . . . 47
4.9.1 仮想マシンでのリモート接続の確認方法 . . . 47
4.9.2 デフォルト状態では接続できないことを確認する . . . 47
4.9.3 ファイアウォールの設定を変更して接続を許可する . . . 48
4.10 システム起動時の Web サーバー自動起動とファイアウォールの設定 . . . 49
4.10.1 OS 再起動 . . . 49
4.10.2 再起動後の動作確認 . . . 49
4.10.3 Web サーバーの自動起動設定 . . . 49
4.10.4 ファイアウォールの自動設定 . . . 49
4.10.5 OS 再起動と確認 . . . 50
4.11 ログの確認 . . . 50
4.11.1 Web サーバーのログファイルの確認 . . . 50
4.11.2 アクセスログの確認 . . . 50
4.11.3 エラー番号 . . . 50
4.11.4 エラーログの確認 . . . 51
4.11.5 index.html を配置する . . . 51
4.11.6 Web ブラウザからのアクセスとログの確認 . . . 52

5 DNS サーバーの構築 53
5.1 用語集 . . . 53
5.2 DNS の仕組み . . . 54
5.2.1 HOSTS ファイルと DNS . . . 54
5.2.2 DNS コンテンツサーバーによるドメイン名の管理 . . . 54
5.2.3 DNS キャッシュサーバーによる名前解決 . . . 54
5.3 ドメインの構造 . . . 55
5.3.1 ルートドメイン . . . 55
5.3.2 トップレベルドメイン . . . 55
5.3.3 ドメイン名の記述 . . . 55
5.3.4 ドメイン名の記述例 . . . 55
5.3.5 サブドメイン . . . 55
5.3.6 ドメイン名の取得 . . . 56
5.4 DNS を使った名前解決 . . . 57
5.5 演習で構築する DNS の概略 . . . 58
5.5.1 jp ゾーンのマシン . . . 58
5.5.2 example1.jp ゾーンのマシン . . . 58
5.5.3 example2.jp ゾーンのマシン . . . 58
5.6 演習の手順 . . . 58
5.7 演習で使うアドレスとドメイン . . . 58
5.8 アドレス解決の流れ . . . 58
5.9 DNS コンテンツサーバーの設定 . . . 60
5.9.1 chroot 機能を利用した BIND のセキュリティ . . . 60
5.9.2 ゾーンを設定する流れ . . . 60
5.10 BIND のインストール . . . 61
5.11 /etc/named.conf の基本設定 . . . 61
5.11.1 問い合わせを受け付けるアドレスの設定 . . . 62
5.11.2 問い合わせを許可するアドレスの設定 . . . 62
5.11.3 DNS コンテンツサーバーとしての設定 . . . 62
5.11.4 正引きゾーンの追加 . . . 62
5.12 ゾーンファイルの準備 . . . 62
5.13 ゾーンファイルの修正 . . . 63
5.13.1 $TTL . . . 63
5.13.2 $ORIGIN . . . 63

5.13.3 SOA レコード . . . 63
5.13.4 NS レコード . . . 63
5.13.5 MX レコード . . . 63
5.13.6 A レコード . . . 63
5.14 設定ファイルとゾーンファイルの書式確認 . . . 64
5.14.1 設定ファイルの書式確認と注意点 . . . 64
5.14.2 ゾーンファイルの書式確認 . . . 64
5.15 BIND の起動と確認 . . . 64
5.16 自動起動の設定 . . . 65
5.17 ファイアウォールの設定 . . . 65
5.18 参照する DNS サーバーの設定と NAT ネットワークの停止 . . . 65
5.18.1 /etc/resolv.conf による参照 DNS サーバーの設定 . . . 66
5.18.2 エディタで/etc/resolv.conf を修正すると？ . . . 66
5.18.3 ネットワークインターフェースの名前を確認 . . . 66
5.18.4 NAT ネットワークを停止する . . . 66
5.18.5 /etc/resolv.conf の変更の確認 . . . 66
5.18.6 NAT ネットワークの接続の有効化 . . . 67
5.19 名前解決の確認 . . . 67
5.19.1 dig コマンドでホストの名前解決を確認 . . . 67
5.19.2 正しく名前解決が行えない場合 . . . 68
5.19.3 その他の A レコードを名前解決する . . . 68
5.19.4 dig コマンドで NS レコードを確認 . . . 68
5.19.5 dig コマンドで MX レコードを確認 . . . 68
5.19.6 参照する DNS サーバーを指定した dig コマンドの実行 . . . 69
5.20 example2.jp サーバーと jp サーバーの追加 . . . 69
5.20.1 仮想マシン作成の留意事項 . . . 69
5.20.2 ネットワークアダプターの追加を忘れずに . . . 69
5.20.3 OS のインストール . . . 69
5.20.4 相互通信の確認 . . . 69
5.21 example2.jp ゾーンの設定 . . . 70
5.21.1 named.conf の設定 . . . 70
5.21.2 ゾーン定義ファイルの作成 . . . 71
5.21.3 ゾーンファイルの修正 . . . 71
5.21.4 BIND の起動 . . . 71
5.21.5 自動起動とファイアウォールの設定 . . . 71
5.21.6 NAT ネットワークを停止する . . . 71
5.21.7 名前解決の確認 . . . 72
5.22 上位 jp ゾーンの DNS コンテンツサーバーの設定 . . . 72
5.22.1 named.conf の設定 . . . 72
5.22.2 ゾーン定義ファイルの作成 . . . 73
5.22.3 ゾーンファイルの修正 . . . 73
5.22.4 BIND の起動 . . . 74
5.22.5 自動起動とファイアウォールの設定 . . . 74
5.22.6 NAT ネットワークを停止する . . . 74
5.22.7 名前解決の確認 . . . 74
5.23 相互に名前解決できることを確認 . . . 74
5.23.1 参照する DNS サーバーの変更 . . . 74
5.23.2 名前解決の確認 . . . 75
5.24 うまく名前解決が行えない場合には . . . 76

6 メールサーバーの構築 77
6.1 用語集 . . . 77
6.2 メールのやり取りの仕組み . . . 77
6.3 メールの送信 . . . 78
6.3.1 MTA（Mail Transfer Agent） . . . 78
6.3.2 MDA（Mail Delivery Agent） . . . 78
6.3.3 MUA（Mail User Agent） . . . 78
6.3.4 SMTP（Simple Mail Transfer Protocol）の拡張 . . . 78
6.3.5 SMTP 認証（SMTP Authentication）とリレー . . . 78
6.4 メールの受信 . . . 78
6.4.1 メールの配送 . . . 78

6.4.2 ローカル配送 . . . 78
6.4.3 POP3（Post Office Protocol version 3） . . . 79
6.4.4 IMAP4（Internet Message Access Protocol 4） . . . 79
6.5 メールサーバーの構築 . . . 79
6.5.1 mail コマンドを利用したメールの送受信 . . . 79
6.5.2 Thunderbird を利用したメールの送受信 . . . 80
6.6 Postfix のインストール . . . 80
6.6.1 NAT ネットワークの有効化 . . . 80
6.6.2 パッケージのインストール . . . 80
6.6.3 NAT ネットワークの無効化 . . . 80
6.6.4 インターネット接続できない環境でのインストール . . . 81
6.7 Postfix の設定ファイル main.cf の設定 . . . 81
6.7.1 myhostname . . . 81
6.7.2 mydomain . . . 82
6.7.3 inet_interfaces . . . 82
6.7.4 mydestination . . . 82
6.7.5 smtpd_sasl_auth_enable . . . 82
6.7.6 smtpd_recipient_restrictions . . . 82
6.8 書式のチェック . . . 83
6.9 Postfix の起動 . . . 83
6.10 自動起動とファイアウォールの設定 . . . 83
6.10.1 Postfix の起動順の設定（任意） . . . 83
6.10.2 systemctl edit コマンドで修正 . . . 83
6.10.3 vim で設定ファイルを修正 . . . 83
6.11 SMTP 認証（SASL 認証連携）の設定 . . . 84
6.11.1 Postfix の設定の確認 . . . 84
6.11.2 saslauthd サービスの起動 . . . 84
6.12 アカウントの作成 . . . 84
6.12.1 host1 に user1 を作成 . . . 84
6.12.2 host2 に user2 を作成 . . . 84
6.13 mail コマンドを使ったメール送受信のテスト . . . 85
6.13.1 ログの確認用端末の設定（任意） . . . 85
6.13.2 メール送受信用端末の起動とユーザー切り替え . . . 85
6.13.3 host1 で user1 に切り替え . . . 85
6.13.4 host2 で user2 に切り替え . . . 85
6.13.5 user1@example1.jp から user2@example2.jp へメール送信 . . . 85
6.13.6 user2 のメール着信確認 . . . 86
6.14 メールクライアントソフトでのメールの送受信 . . . 86
6.15 Dovecot の設定 . . . 86
6.15.1 /etc/dovecot/dovecot.conf（変更不要） . . . 86
6.15.2 /etc/dovecot/conf.d/10-mail.conf . . . 87
6.15.3 /etc/dovecot/conf.d/10-auth.conf . . . 87
6.15.4 /etc/dovecot/conf.d/10-ssl.conf . . . 88
6.16 Dovecot の起動 . . . 88
6.16.1 自動起動とファイアウォールの設定 . . . 88
6.17 Thunderbird の設定 . . . 88
6.17.1 ユーザーを切り替えてログイン . . . 88
6.17.2 Thunderbird の起動 . . . 89
6.17.3 Thunderbird の基本設定 . . . 90
6.17.4 Thunderbird の送受信メールサーバーの設定 . . . 91
6.18 メールの送信 . . . 93
6.18.1 自分宛のメール送信 . . . 93
6.18.2 別サーバー宛のメール送信 . . . 93
6.19 うまく動作しない場合には . . . 93
6.20 ISO イメージからのパッケージのインストール . . . 94
6.20.1 ISO イメージのマウントポイントの確認 . . . 94
6.20.2 パッケージのインストール . . . 94

7 ネットワークとセキュリティの設定 95
7.1 用語集 . . . 95
7.2 ネットワーク管理 . . . 96

7.2.1 ネットワークインターフェースの確認 96
7.2.2 ネットワークインターフェースの再設定 97
7.2.3 ネットワークインターフェースの動作確認 98
7.3 サービスのポート番号を確認 98
7.3.1 ss コマンドを使ったポート使用状況の確認 98
7.3.2 lsof コマンドを使ったポート使用状況の確認 98
7.3.3 services ファイルによるポート番号の確認 99
7.4 SSH によるリモートログイン 99
7.4.1 パスワードによる認証 99
7.4.2 公開鍵による認証 100
7.4.3 パスワード認証の禁止 101
7.5 ファイアウォールの設定 102
7.5.1 ファイアウォール設定の確認 102
7.5.2 許可サービスの追加 102
7.5.3 設定可能なサービスの確認 102
7.5.4 許可サービスの取り消し 103
7.5.5 ファイアウォール設定の保存 103

1 Linux サーバーの概要

本教科書では、Linux をインストールしてサーバー環境を構築する実習をしながら、LinuC レベル 1 からレベル 2 の出題範囲に含まれる主要な技術の理解を目指します。

第 1 章では Linux サーバーの概要について解説し、2 章以降に行う実習に必要な環境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。

Linux 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 Linux ディストリビューションです。Linux 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、リリースが行われています。

AlmaLinux

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 Linux 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 Linux ディストリビューションで、AlmaLinux コミュニティによって開発されています。

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、ホスト一つ一つに 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ホストの所在地を示す「住所」にあたります。現在では IPv4 と IPv6 の 2 種類がありますが、本教科書では IPv4 を使います。IPv4 では 32 ビットの IP アドレスを、8 ビット毎に.(ドット) で区切って 10 進数で表記します。

ネットワークアドレス

IP ネットワークは 1 つ以上のホストの集まりであるネットワークを形成し、そのネットワーク内の通信と、ネットワーク間の通信で構成されています。このネットワークの識別に利用されるのが、ネットワークアドレスです。IP アドレスは、所属しているネットワークを表すネットワーク部と、ホスト個別に割り当てられるホスト部に分けることができます。

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を識別するための数値のことをいいます。32 ビットのうち、先頭からネットワーク部となる部分のビットを 1 として表します。たとえば先頭から 24 ビットがネットワーク部になる場合、IP アドレスの後ろに「/24」と表記したり、10 進数に変換して「255.255.255.0」と表記したりします。

ホスト名

ネットワークに接続されたコンピュータに割り当てられた名称のことをいいます。ドメイン名を省略した短い名前を指す場合と、ドメイン名を含んだ FQDN(Fully Qualified Domain Name) の形式のものがあります。

ドメイン名

インターネット上に存在するコンピュータやネットワークを識別するために付けられている名前です。会社などの組織毎に独自のドメイン名を取得して利用します。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。

DNS サーバーアドレス

FQDN（＝ホスト名＋ドメイン名）と IP アドレスの変換を「名前解決」と呼びます。この名前解決を行うのが DNS（Domain Name System）です。ホストは設定した DNS サーバーアドレスに対して名前解決を依頼します。

ハードディスク（HDD）

磁気を用いた記憶媒体であり、パソコンの記憶媒体の他、ビデオレコーダー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。

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。

USB メモリ

USB に接続して利用する外部ストレージです。OS のインストールディスクとして利用できます。

1.2 実習で利用するハードウェア

本教科書の実習では、仮想マシンを使って実習環境を構築します。

仮想マシンの詳細については第 2 章で解説しますが、ここでは実習用に用意するハードウェアの仕様などについて解説します。

1.2.1 マシン本体

Windows や Linux、macOS が動作する、いわゆる「パソコン」（PC）を想定しています。用意した PC に「VirtualBox」のような仮想マシンソフトウェアをインストールして実習環境を構築します。

1.2.2 CPU

実習では CPU 負荷の高い処理は行わないため、高性能な CPU は必要ありません。ただし、仮想マシンを実行するために仮想化支援技術が必要となります。古い CPU でない限り仮想化支援技術を備えていますが、BIOS（UEFI）で有効にする必要があります。

本教科書では VirtualBox を利用するため、CPU アーキテクチャは IA（Intel Architecture）の CPU を想定しています。

ARM アーキテクチャの CPU に対応した仮想マシンソフトウェアを利用し、ARM 対応の Linux ディストリビューションをインストールすれば同様の実習を行うこともできます。仮想マシンの設定などが異なりますので、適宜読み替えて実習を行ってください。

1.2.3 メモリ

AlmaLinux 9.3 では 1.5GB 以上のメモリが推奨されています。

DNS やメールの実習では 3 台の仮想マシンを使って実習を行うので、3 台分で 4.5GB のメモリが必要です。メモリは仮想マシンの他に、OS（ホスト OS と呼ぶ）やその他のアプリケーションを同時に動作させるために必要となるので、最低でも 8GB、快適に実習を行うには 16GB 以上のメモリが搭載されていることが望ましいでしょう。

メモリの搭載量が少ない場合には、仮想マシンへのメモリの割り当てを減らす、余計なアプリケーションを動作させないようにするなどのメモリ効率改善や、メモリスワップ発生による速度低下を許容する必要があります。

1.2.4　ストレージ

仮想マシンが利用する仮想ハードディスクはファイルとして PC に接続されたハードディスクや SSD などのストレージに保管されます。仮想ハードディスクファイルは利用した分だけストレージを消費します。

実習で利用する環境はおよそ 7GB の容量を消費します。VirtualBox のデフォルトでは最大 20GB の仮想ハードディスクファイルが作られるので、3 台の仮想マシンで少なくとも 21GB、最大 60GB の容量を用意しておく必要があります。

仮想ハードディスクファイルの読み書きは頻繁に行われるため、ハードディスクよりも高速な SSD を利用するのが望ましいでしょう。

1.2.5　ネットワーク

実習ではインターネットに接続できる必要があります。実習用に用意する PC は有線 LAN、無線 LAN、どちらの方式で接続されていても構いません。

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、AlmaLinux 9.3 の Intel/AMD x86_64 アーキテクチャに対応したバージョンを利用します。

```
https://almalinux.org/ja/
```

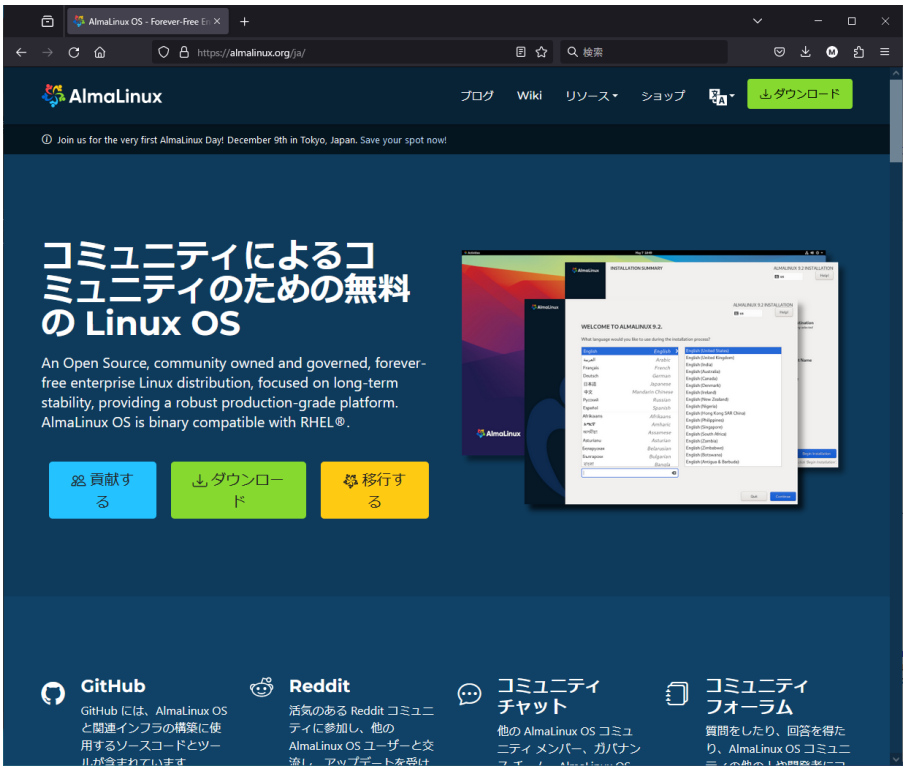

図 5: AlmaLinux の Web サイト

AlmaLinux は、商用ディストリビューションである Red Hat Enterprise Linux をベースにしたディストリビューションとして提供されています。利用に際し費用が発生しない、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　インストール用 ISO イメージの入手

AlmaLinux が配布している ISO イメージを、ダウンロードします。仮想マシンは、ISO イメージを仮想光学ドライブにセットすることでインストールが行えるので、インストール用の DVD/USB メモリを作成する必要がありません。

ダウンロード方法　ISO イメージをダウンロードするには、以下の URL にアクセスします。

```
https://mirrors.almalinux.org/isos/x86_64/9.3.html
```

この URL にアクセスすると、多くのミラーサイトが表示されます。例えば、IIJ 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。

```
http://ftp.iij.ad.jp/pub/linux/almalinux/9.3/isos/x86_64/
```

URL が利用できない場合　URL はサイト構造の変更によって変わっている場合があります。その場合には、AlmaLinux の Web サイトからリンクを辿ってダウンロードサイトを探してください。

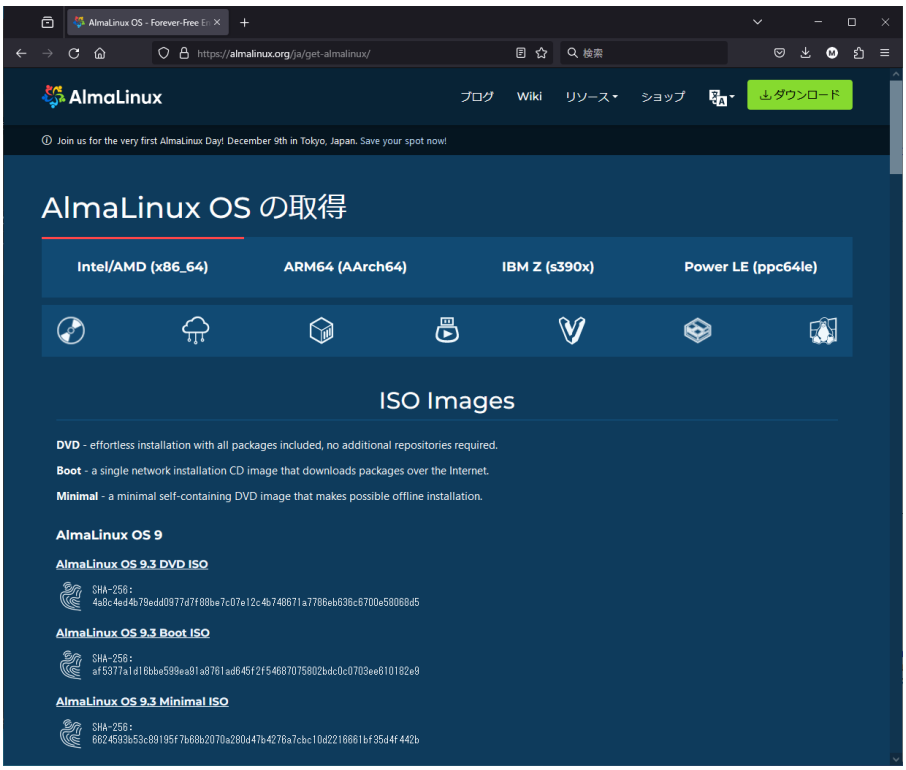

図 6: AlmaLinux のダウンロードサイト

1.4　ISO イメージのファイル名

ISO イメージは以下のようなファイル名になっています。

```
AlmaLinux-バージョン-アーキテクチャ-イメージの種類.iso
```

1.5　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 AlmaLinux 9.3 を利用した構築方法について解説しています。

今後のバージョンアップで、より新しいバージョンの AlmaLinux が入手可能になっているかもしれません。バージョン 9 系であればマイナーバージョンによる大きな差は無いので同様の手順でサーバーの構築ができると推測されますが、セキュリティ対応などの関係でデフォルトの設定が変更されることで動作が変わる場合があります。

初めて本教科書の内容を学習する際には、まずはバージョンを合わせて動作を確認し、その後異なるバージョンで同様に動作するか確認してみてください。バージョンによる動作の違いについては、本教科書の情報交換を行う Slack で情報共有を行い、本教科書の今後のバージョンアップで対応していく予定です。

1.6 アーキテクチャ

Linux カーネルは様々な種類の CPU アーキテクチャに対応しています。アーキテクチャによってバイナリが異なるため、アーキテクチャに合わせた ISO イメージを選択する必要があります。主なアーキテクチャには以下のものがあります。

1.6.1 x86_64

Intel や AMD の CPU アーキテクチャです。64 ビット版になります。

1.6.2 aarch64

ARM の CPU アーキテクチャです。64 ビット版になります。

この他に PowerPC(ppc64le) や、IBM S/390(s390x) などのアーキテクチャに対応した ISO イメージが用意されています。

1.7 ISO イメージの種類

ISO イメージには、いくつかの種類があります。インストールの目的によってイメージを選択します。AlmaLinux には以下の 3 種類の ISO イメージが用意されています。

- 起動のみ（boot.iso）
- 最低限（minimal.iso）
- フルパッケージ（dvd.iso）

本教科書の演習では、GUI 環境も使用するため、フルインストールが可能な ISO イメージ（dvd.iso）を使ってインストールします。AlmaLinux-9.3-x86_64-dvd.iso をダウンロードしてください。

実務では最小限のインストールを行い、必要なパッケージを追加してサーバーを構築することが多いので、慣れてきたら最低限の OS インストールを行う ISO イメージ（minimal.iso）からインストールを行う方法にも挑戦してみてください。

boot.iso

インストーラの起動のみを行う ISO イメージです。インストールするファイルをネットワーク経由で取得するネットワークインストールを行います。システムに障害が発生した際に復旧を行うレスキューモードで起動するためにも使用できます。起動可能な最小限の構成になっているので、ISO イメージのサイズは小さくなっています。

minimal.iso

最低限の OS インストールを行う ISO イメージです。DVD メディアの容量制限に引っかからないため、DVD メディアを作成してインストールを行うにはこちらを選択します。

dvd.iso

フルインストールが可能な ISO イメージです。サイズが大きいため、DVD メディアを作成することができませんが、仮想マシンにインストールする場合にはそのまま利用可能です。

物理マシンに直接インストールしたい場合には、USB メモリに書き込んで起動用 USB メモリを作成します。

1.8 ネットワーク環境について

本教科書では仮想マシンを利用して実習を行うため、特別なネットワークは必要ありませんが、以下の点について確認しておいてください。

1.8.1 インターネットへの接続

ソフトウェアのインストールを行うため、インターネット上に用意されているリポジトリサーバーへのアクセスが必要になります。実習で利用する VirtualBox では、NAT ネットワークで外部ネットワークに接続します。VirtualBox の NAT ネットワークは、実習用 PC がインターネットに接続できていれば特別な設定は必要ありません。

プロキシーを経由して接続する必要がある場合には、接続のための設定情報を入手するか、ネットワーク管理者に相談してください。

1.8.2 仮想マシン間の接続

仮想マシン間の接続は仮想ネットワークを経由して行うため、利用している仮想マシンソフトウェアの仕様によって異なります。本教科書で利用する VirtualBox はホストオンリーネットワーク接続を行うため、仮想マシン同士とホスト OS のみしか相互に接続できず、外部のネットワークからは隔離されているため、IP アドレスなどは独自に設定できます。

1.9 ネットワークの設定項目

サーバーを構築していくにあたり、以下のような設定項目をどう設定するか決める必要があります。

1.9.1 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。このドメイン名は、あくまでこのネットワーク内だけ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、2 台の仮想マシンのために example1.jp と example2.jp の 2 つのドメイン名を使用します。また、DNS による名前解決を行うためにもう 1 台の仮想マシンを作成し、jp ドメインとして設定します。

マシン	設定値
1 台目	example1.jp.
2 台目	example2.jp.
3 台目	jp.

1.9.2 ホスト名

自分の PC に設定するホスト名です。ドメイン名に合わせて host1 と host2 を設定します。jp ドメイン用には host0 を設定します。

ホスト名は、ドメイン名と合わせて FQDN(Fully Qualified Domain Name) で記述する場合もあります。FQDN 表記の場合には、それぞれ以下のようになります。

マシン	設定値
1 台目	host1.example1.jp.
2 台目	host2.example2.jp.
3 台目	host0.jp.

1.9.3 IP アドレス

IP アドレスは、VirtualBox のホストオンリーネットワーク接続のデフォルトに合わせて、それぞれ以下のように設定しています。

マシン	設定値
1 台目	192.168.56.101
2 台目	192.168.56.102
3 台目	192.168.56.100

1.9.4 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。

1.9.5　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 192.168.56.0 となります。

1.9.6　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。

ホストオンリーネットワーク接続では外部への接続を行わないため、デフォルトゲートウェイは設定しません。

実際のサーバー構築においては、ネットワークの設定を確認の上、適切なデフォルトゲートウェイのアドレスを確認、設定してください。

1.9.7　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。

本教科書の第 5 章で実際に DNS サーバーを設定し動作させます。実習では、まず自分自身（ローカル）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自身の IP アドレスとします。

マシン	設定値
1 台目	192.168.56.101
2 台目	192.168.56.102
3 台目	192.168.56.100

1.10　ネットワーク設定まとめ

各仮想マシン毎のネットワークの設定は以下のようになります。

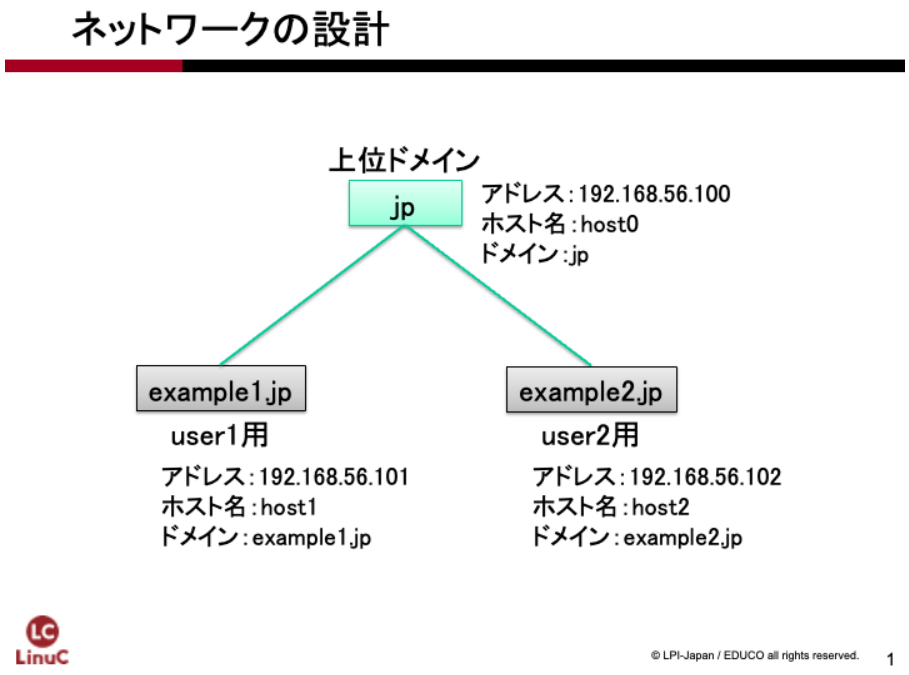

図 7: ネットワークの設計

1.10.1　1 台目の仮想マシン

設定項目	設定値
ホスト名	host1.example1.jp.
IP アドレス	192.168.56.101
サブネットマスク	255.255.255.0(/24)
ネットワークアドレス	192.168.56.0
デフォルトゲートウエイ	設定不要
DNS サーバーアドレス	192.168.56.101

1.10.2 2 台目の仮想マシン

設定項目	設定値
ホスト名	host2.example2.jp.
IP アドレス	192.168.56.102
サブネットマスク	255.255.255.0(/24)
ネットワークアドレス	192.168.56.0
デフォルトゲートウエイ	設定不要
DNS サーバーアドレス	192.168.56.102

1.10.3 3 台目の仮想マシン

設定項目	設定値
ホスト名	host0.jp.
IP アドレス	192.168.56.100
サブネットマスク	255.255.255.0(/24)
ネットワークアドレス	192.168.56.0
デフォルトゲートウエイ	設定不要
DNS サーバーアドレス	192.168.56.100

2 VirtualBox のインストールと仮想マシンの作成

第 2 章では、実習環境の準備として VirtualBox をインストールして仮想マシンを作成します。

2.1 用語集

仮想化技術

ハードウェアの機能をソフトウェアで実現する技術。仮想マシンや仮想ネットワーク、仮想ストレージなど様々な種類の技術が存在しています。

仮想マシン

仮想化技術を利用して、コンピュータのハードウェアをソフトウェアで実現したもの。OS をインストールし、様々なアプリケーションを実行できます。

仮想ホスト

仮想マシンを実行するマシンのこと。仮想ホストでは通常の OS が動作しており、仮想マシンは通常のアプリケーションのように動作します。

ゲスト OS

仮想マシンにインストールする OS のこと。仮想ホストで動作している OS とは異なる種類の OS をインストールして実行できます。

2.2 仮想化環境とは

仮想化環境とは、仮想化技術を活用して構築、動作させるシステムの環境のことです。仮想化技術には、仮想マシン、仮想ネットワーク、仮想ストレージ、コンテナなどがあります。

2.2.1 仮想化技術とクラウドの関係

現在では、クラウドを活用してシステムを構築、運用することが増えています。多くのクラウドサービスが仮想化技術を活用する事で、効率良くシステムリソースをユーザーに対して提供することでそのサービスを実現しています。クラウドを理解し、活用するためにも、仮想化技術の基礎を理解しておく必要があります。

2.3 仮想マシンとは

仮想マシンは、ソフトウェアで仮想的なマシンを実行する仕組みです。仮想マシンにはゲスト OS をインストールし、アプリケーションを実行できます。

2.3.1 ホスト OS 型とハイパーバイザー型

仮想マシンの実行環境には、ホスト OS 上でアプリケーションのように実行される「ホスト OS 型」と、仮想マシン実行専用のソフトウェアであるハイパーバイザーが動作する「ハイパーバイザー型」に大別されます。

本教科書では、簡単に実行できるホスト OS 型を取り上げています。

2.4 仮想マシンのメリット

仮想マシンの利用には、いくつかのメリットがあります。

2.4.1 ホスト OS と共存できる

仮想マシンはホスト OS 上でアプリケーションのように実行されます。そして同時にその他のアプリケーションも実行できるので、たとえば Web ブラウザで何かを調べながら、ターミナルで仮想マシン上の Linux にログインして操作する、ということが簡単に行えます。

2.4.2 ハードウェアを用意する必要がない

Linux を使ったサーバー構築を学習する際に大きな障害となるのが学習用ハードウェア環境の準備です。ハードウェアを用意できたとしても、Linux をインストールするには設定を変更するなどのハードウェアの知識が要求されます。メーカーの違いによって設定方法も異なるため、学習の助けを得るのも難しくなります。

仮想マシンではそのような前提知識が要求されることが少ないので、容易に Linux をインストールして学習環境を構築できます。

2.4.3 柔軟性

仮想マシンは必要に応じて簡単に CPU 数やメモリ容量、ディスク容量などを変更できる柔軟性を備えています。

2.4.4 OS 選択

仮想マシンにはそれぞれ別々の OS をインストールできます。たとえば Linux と Windows がそれぞれインストールされた仮想マシンを同時並行で実行できます。

2.5 仮想マシンのデメリット

仮想マシンの利用には、メリットだけでなくデメリットもあります。

2.5.1 速度

仮想マシンは、コンピュータの動作をソフトウェアで実現しているので、ハードウェアそのもので OS やアプリケーションを動作させるよりも動作が遅くなります。

2.5.2 リソースの消費

仮想マシンを複数同時に実行することができますが、その分だけ CPU やメモリ、ストレージなどのリソースを消費します。ホスト OS を快適に動かすのにもリソースが必要となるため、使用するハードウェアにはより多くのリソースを用意する必要があります。

2.6 仮想マシンのリソース

仮想マシンを作成する際に、CPU やメモリ、ネットワークインターフェースなどのリソースを利用者自らが調整する必要があります。物理マシンを利用する場合にも同様の作業が必要ですが、仮想マシンの場合、ある程度後から変更できるメリットがあります。まずはデフォルト設定のまま動かしてみて、後から必要に応じて変更しても構いません。

2.6.1 CPU

仮想マシンには必要なだけの数の CPU を割り当てられます。ただし、CPU をホスト OS やその他の仮想マシンとの間で取り合うことになるため、沢山割り当てればよいというわけではありません。

2.6.2 メモリ

仮想マシンに設定したメモリ容量は、その分だけホストマシンから仮想マシンに対して割り当てられます。ホストマシンにはホスト OS だけでなく、仮想マシンに割り当てられるだけの容量のメモリを搭載しておく必要があります。

2.6.3 ディスク

HDD や SSD にあたる仮想ディスクは、仮想ディスクファイルという形でホスト OS 上に作成されます。可変式ファイルの場合には、仮想マシンが使用した分だけのサイズのファイルになります。固定式ファイルの場合には、作成時に設定しただけの容量のファイルが作成されます。可変式にすることで、ホストマシンのディスク使用を節約できます。

2.6.4 光学ドライブ

仮想マシンに割り当てられた仮想光学ドライブは、物理的な光学ドライブにセットされた CD/DVD メディアを割り当てる他、ISO イメージを割り当てることができます。仮想マシンに OS をインストールするのが容易な理由の一つです。

2.6.5 ネットワーク

仮想マシンに割り当てられるネットワークインターフェースを割り当てることができます。仮想マシンソフトウェアによって機能が異なるため、それぞれの機能に応じた利用が必要になります。

2.7 VirtualBox のインストール

本教科書の実習では、仮想マシンソフトウェアである「VirtualBox」上に 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。

2.7.1 使用するコンピュータの仮想化支援技術の有効化

VirtualBox を実行するには、使用するコンピュータが搭載しているプロセッサの仮想化支援技術が有効になっている必要があります。仮想化支援技術は、Intel の CPU では Intel VT、AMD の CPU では AMD-V と呼ばれます。

仮想化支援技術を有効にするには、コンピュータの UEFI 設定画面で設定を行います。設定方法の詳細は、使用するコンピュータの説明書などを確認してください。

2.7.2 VirtualBox のダウンロード

VirutalBox は、次の URL からダウンロードできます。

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

図 8: VirtualBox の Web サイト

ホスト OS の種類に合わせてダウンロードを行えます。今回は Windows 環境にインストールします。左のメニューから Downloads をクリックし、VirtualBox binaries から Windows hosts をクリックして、インストーラをダウンロードします。

図 9: VirtualBox のダウンロードサイト

今回はバージョン 7.0.12 を使用しています。これ以降のバージョンでネットワーク機能（ホストオンリーアダプター）が変更になっている可能性があるため、新しいバージョンになっている場合には過去のビルド（VirtualBox older builds）からバージョン 7.0.12 をダウンロードしてください。

```
https://www.virtualbox.org/wiki/Download_Old_Builds_7_0
```

2.7.3 VirtualBox のインストーラの実行

VirtualBox のインストーラをダウンロードが完了したら、インストーラを実行してインストールします。

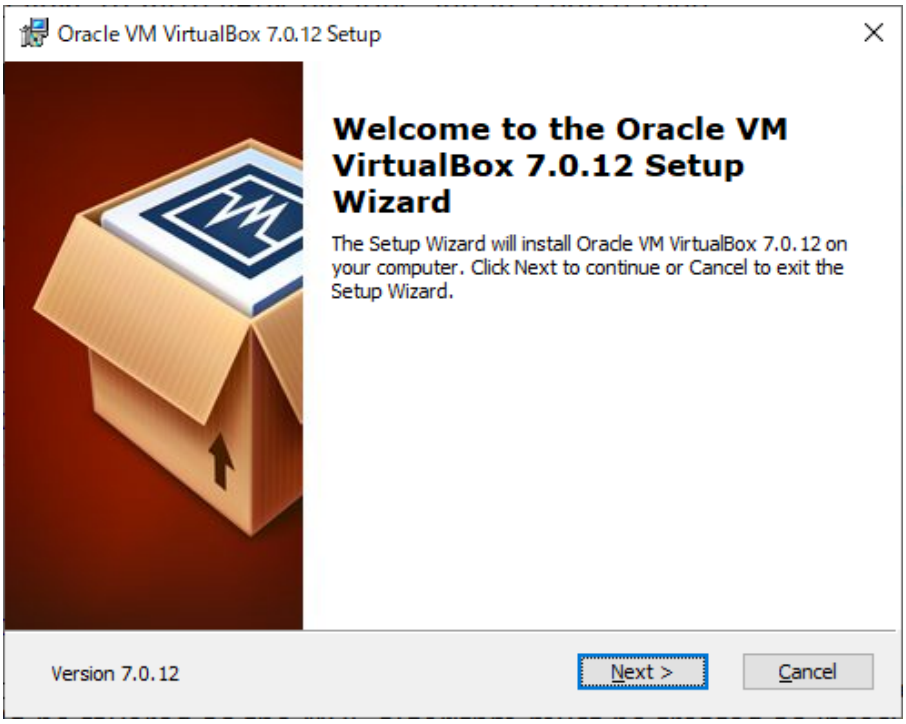

図 10: VirtualBox のインストーラ画面

インストール時の特別な設定は行いませんので、インストーラの指示に従ってインストールします。インストーラが仮想ネットワークのインストールについて、また Python との連携について確認を取ってきますが、そのまま進めても問題ありません。

インストール終了後、Finish ボタンをクリックするとインストーラが終了し、VirtualBox が起動します。

2.8 VirtualBox の起動

VirtualBox を起動します。

インストーラが自動で起動した状態ではない場合には、デスクトップ上に追加された VirtualBox をダブルクリックして起動します。

VirtualBox が起動すると、VirtualBox マネージャーが表示されます。

図 11: VirtualBox マネージャーの画面

2.8.1 VirtualBox マネージャー

VirtualBox マネージャーは、VirtualBox による仮想マシン実行環境全体を管理するツールです。主に以下のことが行えます。

- VirtualBox 自身の各種設定
- 仮想マシンの管理
- メディア（ISO イメージ等）の管理
- 仮想ネットワークの管理

2.9 ホストキーによるホスト OS の操作への復帰

仮想マシンをマウスで操作している際に、ホスト OS にマウス操作を戻したくなった場合にはホストキーを押します。デフォルトではキーボード右側の Ctrl キーがホストキーに設定されています。ホストキーの設定は、仮想マシンのウインドウの右下にも表示されています。

2.9.1 ホストキーが押せない場合のホスト OS の操作への復帰

キーボードによっては右 Ctrl キーが無いため、仮想マシンの操作から抜け出せなくなります。そのような場合になった場合には、OS をシャットダウンして仮想マシンを停止すれば操作をホスト OS 側に戻すことができます。

2.9.2 ホストキーの変更

VirtualBox マネージャーの環境設定から、ホストキーを変更できます。

環境設定は、VirtualBox マネージャーの「ファイル (F)」メニューから「環境設定 (P)」を選択して呼び出します。「環境設定」ウインドウが表示されたら、左側の設定項目から「入力」を選択し、「仮想マシン (M)」タブを選択します。一番最初にある「ホストキーの組み合わせ」の「ショートカット」部分をクリックします。新たにホストキーにしたいキーを押すことで、ホストキーがそのキーに切り替わります。

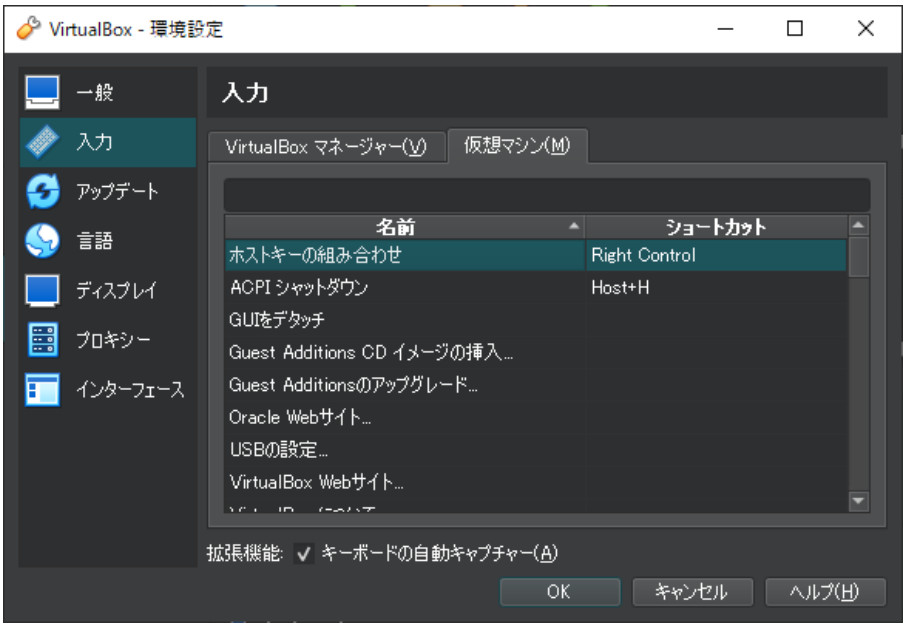

図 12: ホストキーの設定画面

ホストキーに設定できるのは Ctrl キー、Alt キー、Windows キー、Shift キー（単独設定不可）です。左の Ctrl キーをホストキーに設定してしまうと、コマンドライン操作などの際に重複してしまうので、避けた方がよいでしょう。複数のキーを同時に押す組み合わせをホストキーにすることもできるので、2 つないし 3 つのキーの組み合わせをホストキーとして設定するとよいでしょう。

2.10 仮想マシンの作成

仮想マシンを作成します。

VirtualBox マネージャーで「新規 (N)」ボタンをクリックし、新しい仮想マシンの作成を開始します。各種設定を対話形式で設定していくガイド付きモードと、1 つの画面で設定できるエキスパートモードが選択できます。

ここではデフォルトのガイド付きモードで作成します。

2.10.1 仮想マシンの名前と OS の設定

まず最初に、仮想マシンの名前と OS の種類などを設定します。これらの設定は後から変更することもできるので、その他の項目で必要なものは後ほど設定を行います。

設定する項目と、設定する内容は以下の通りです。

項目	設定値
名前	host1.example1.jp
タイプ	Linux
バージョン	Red Hat 9.x (64-bit)

設定したら、「次へ」ボタンをクリックします。

図 13: 仮想マシンの名前と OS 選択の画面

2.10.2 仮想マシンのハードウェアの設定

仮想マシンのハードウェアの設定として、メインメモリーの容量とプロセッサの数を設定します。また、EFI の設定も行います。

項目	設定値
メインメモリー	2048MB
Processors	1
Enable EFI(special OSes only)	チェックする

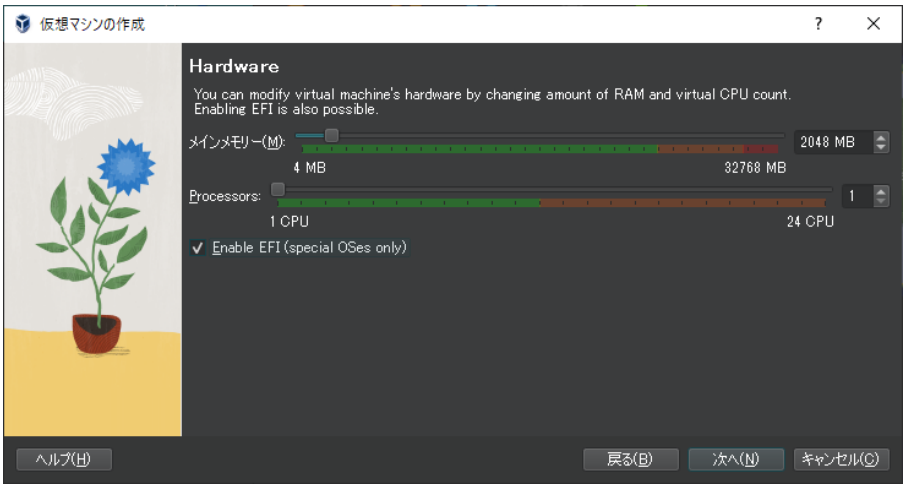

図 14: 仮想マシンのメモリや CPU 設定の画面

メインメモリーの容量は、使用しているコンピュータが搭載しているメモリ容量以下に制限されます。仮想マシンにある程度多めのメインメモリーを割り当てることで快適に動作させることができます。デフォルトの 2048MB でも動作しますが、余裕があれば 4096MB（4GB）割り当ててもよいでしょう。

プロセッサの数は、使用しているコンピュータが搭載しているプロセッサのコア数以下に制限されます。また、プロセッサによっては仮想 CPU 機能により見かけ上コア数が 2 倍になっている場合もあります。デフォルトの 1 つでも動作しますが、余裕があれば 2 つ割り当ててもよいでしょう。

Enable EFI は、デフォルトではチェックされていません。OS のインストール画面のサイズが合わず一部の画面が見えなくなる問題が発生するため、チェックしておきます。

設定しましたら、「次へ」ボタンをクリックします。

2.10.3 仮想ハードディスクの設定

OS のインストール先となる仮想ハードディスクの設定をします。

項目	設定値
Disk Size	20.00GB

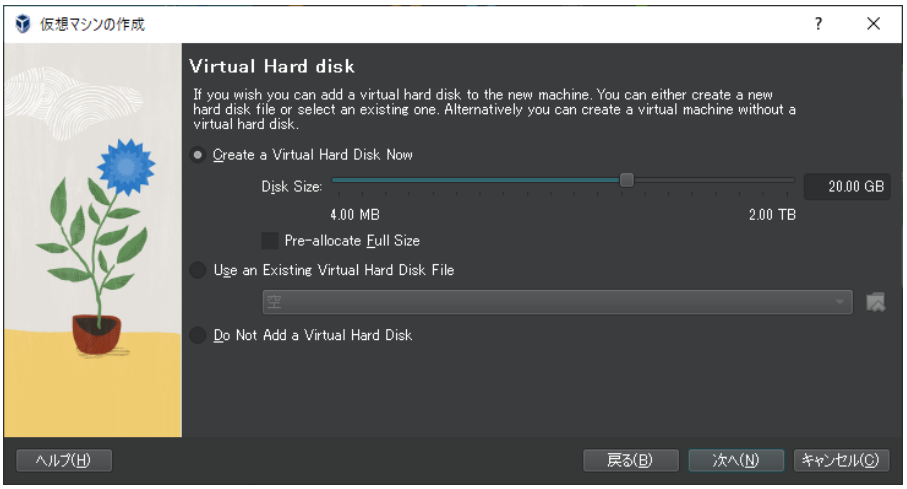

図 15: 仮想ハードディスクの設定画面

仮想ハードディスクの容量は、使用しているコンピュータが搭載しているストレージの容量以下に制限されます。また、「Pre-allocate Full Size」をチェックしない限り、仮想ハードディスクの容量はゲスト OS が使用した分だけしか消費しませんので、設定は最大消費容量を指定することになります。

OS のインストールとサーバーの設定だけではそれほど沢山の容量を必要としませんが、実際に運用するサーバーを作る場合には必要な容量を見積もって設定する必要があります。また、後から仮想ハードディスクを追加することもできるので、追加方法について学習してもよいでしょう。

設定しましたら、「次へ」ボタンをクリックします。

2.10.4 仮想マシンの設定を確認

最後に仮想マシンの設定を確認します。ここまでの手順で設定した内容が反映されているかどうか、あらためて確認します。

確認後、「完了」ボタンをクリックします。

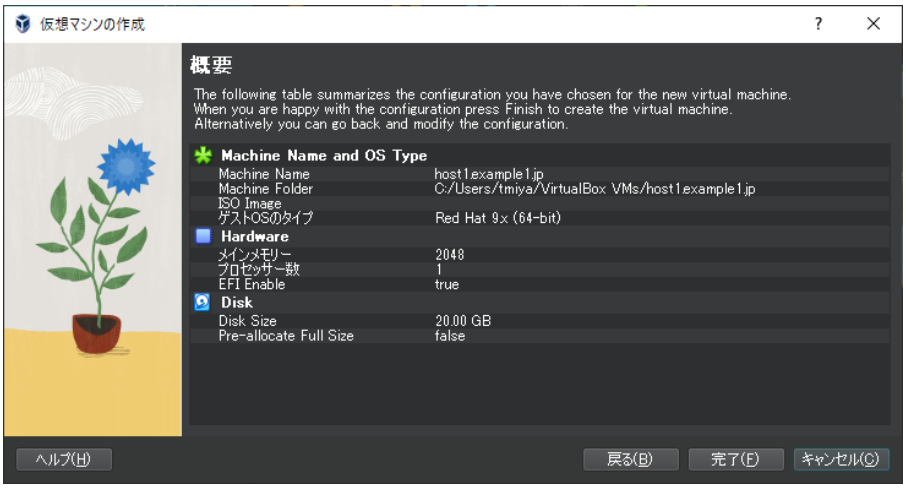

図 16: 仮想マシンの設定確認画面

VirtualBox マネージャーに、新しく作成した仮想マシンが追加されたことを確認します。

2.11 仮想マシンの設定方法

仮想マシンの設定を確認したり、変更するには、VirtualBox マネージャーで仮想マシンを選択し、右側に表示される各種設定情報から変更が行えます。設定項目の見出し部分（「一般」や「システム」など）をクリックすると、その項目の設定画面が表示されます。

図 17: 仮想マシンの設定画面

試しに、仮想マシン作成時に設定した EFI の有効化について確認してみます。

2.11.1 EFI の有効化を確認

仮想マシン作成時に EFI の有効化を行っていないと、OS のインストーラを起動した際に画面のサイズが合わず一部の画面が見えなくなる問題が発生します。

「システム」をクリックし、以下の設定にチェックが入っていることを確認します。チェックが入っていない場合にはチェックします。

確認、あるいは修正ができたら、「OK」ボタンをクリックします。

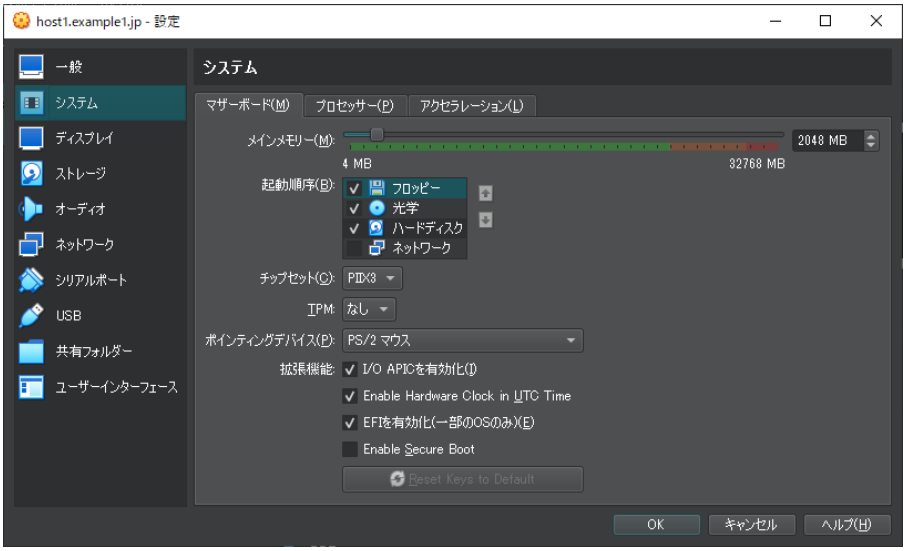

図 18: 仮想マシンのシステム設定の画面

2.12　ネットワークの追加

ホストマシンと仮想マシンが通信できるようにするため、以下の手順で「ホストオンリーアダプター」を追加します。

デフォルトでは NAT を使った外部ネットワークに接続するネットワークのみ設定されています。

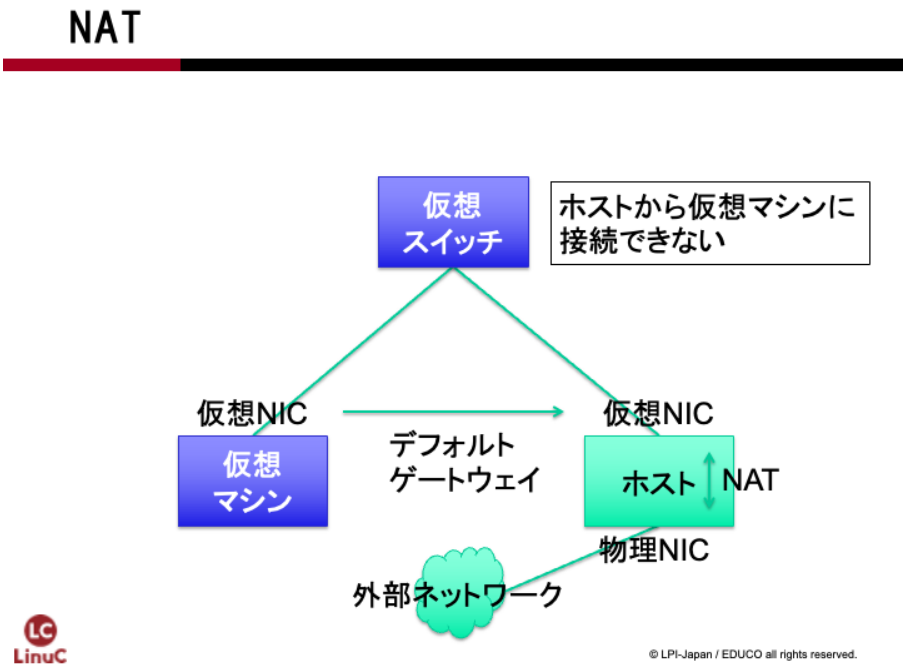

図 19: NAT

2.12.1 仮想ネットワークの設定を確認

VirtualBox の仮想ネットワークの設定を確認します。

VirtualBox マネージャーの「ツール」右側のリストボタン（横 3 本線）をクリックし、「ネットワーク」を選択します。

「Host-only Networks」タブを選択し、「VirtualBox Host-Only Ethernet Adapter」が存在していることを確認します。IP アドレスは、インストール時に自動的に作成された場合には 192.168.56.1/24 が設定されています。この IP アドレスは、ホスト OS に作成された仮想ネットワークアダプターに割り当てられている IP アドレスです。

DHCP サーバーもデフォルトで有効になっていますが、本教科書の演習では DHCP サーバーは使用しません。

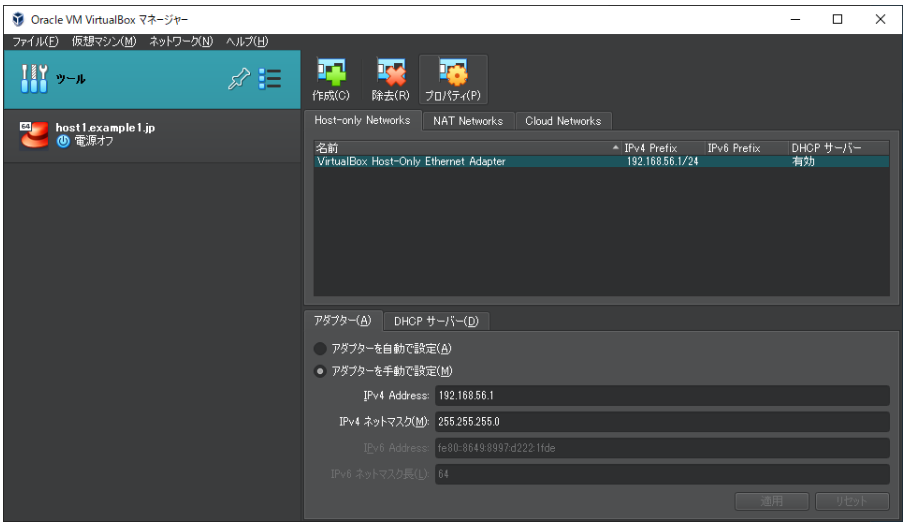

図 20: 仮想ネットワークの設定画面

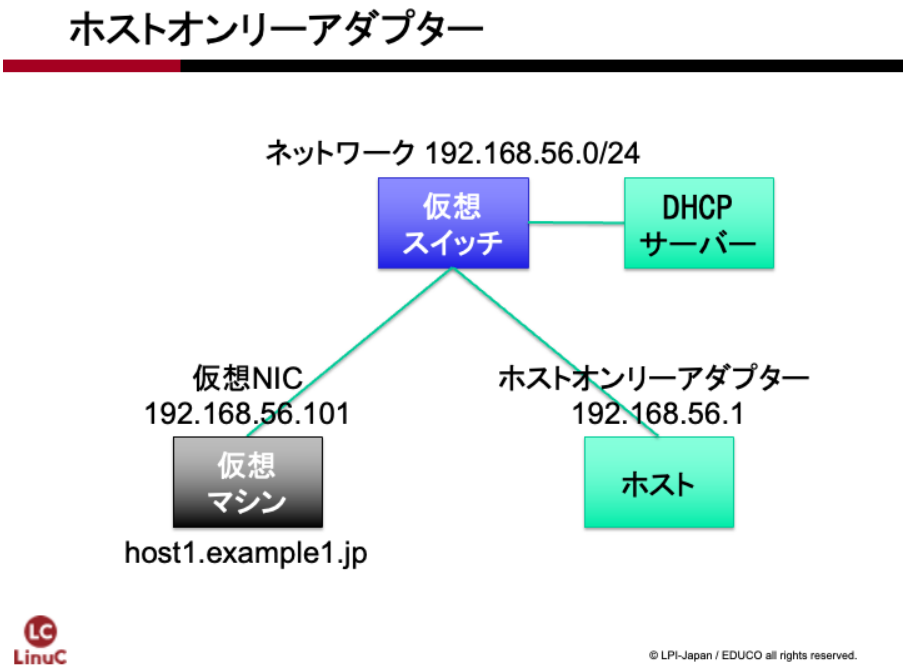

図 21: ホストオンリーアダプター

2.12.2 仮想マシンにネットワークアダプターを追加

仮想マシンにネットワークアダプターを追加します。

VirtualBox マネージャーから仮想マシンを選択し、右側の各種設定情報から「ネットワーク」を選択します。

仮想マシンには最大 4 つのネットワークアダプターを設定できます。アダプター 1 は NAT が設定されており、外部のネットワークやその先にあるインターネットに接続できます。

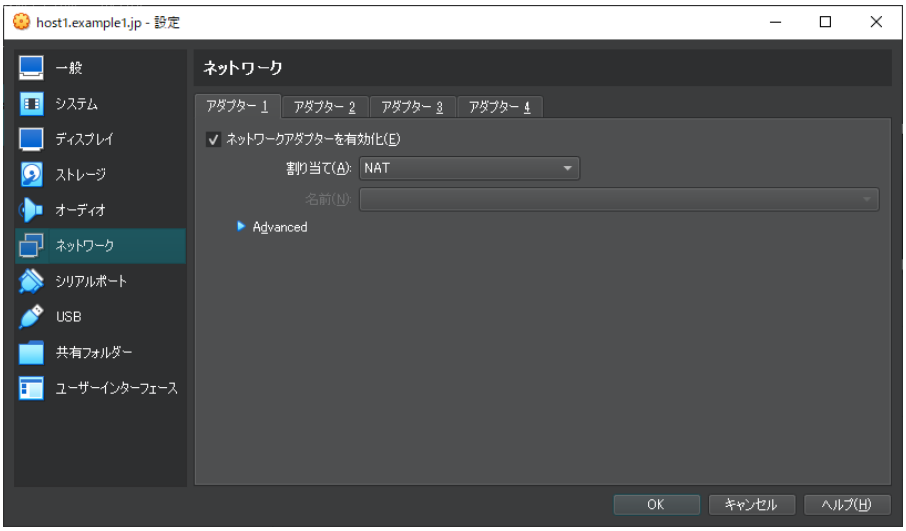

図 22: 仮想マシンのネットワークアダプター 1 の設定画面

アダプター 2 のタブを選択し、「ネットワークアダプターを有効化」をチェックします。「割り当て」を「ホストオンリーアダプター」に設定すると、「名前」に「VirtualBox Host-Only Ethernet Adapter」が設定されます。これによって、この仮想マシンはホストマシンと通信が行えるようになります。

設定したら、「OK」をクリックします。

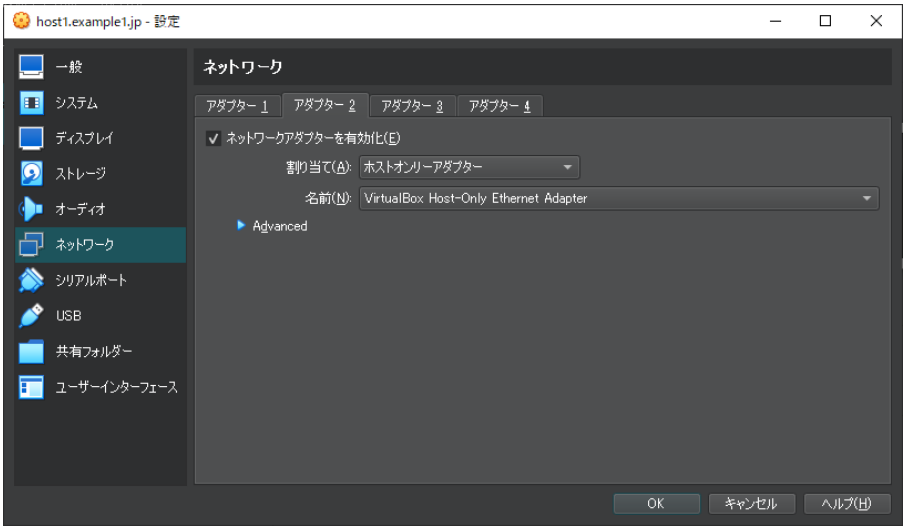

図 23: 仮想マシンのネットワークアダプター 2 の設定画面

2.13　ISO イメージを仮想光学ドライブで読み込む設定

第 1 章でダウンロードした OS インストール用の ISO イメージを、仮想マシンの仮想光学ドライブで読み込む設定をします。

VirtualBox マネージャーから仮想マシンを選択し、右側の各種設定情報から「ストレージ」を選択します。

仮想マシンには、IDE と SATA の 2 種類のストレージデバイスが接続されています。仮想光学ドライブは IDE に接続されています。読み込む設定が何もされていない場合、「空」と表示されているのが仮想光学ドライブですので、選択します。

「光学ドライブ:IDE セカンダリデバイス 0」の右側にある円形のボタンをクリックし、「ディスクファイルを選択」を選択すると、ファイルダイアログが開きます。準備しておいた OS インストール用の ISO イメージを選択し、「開く」ボタンをクリックします。「空」の表示がファイル名に変わります。

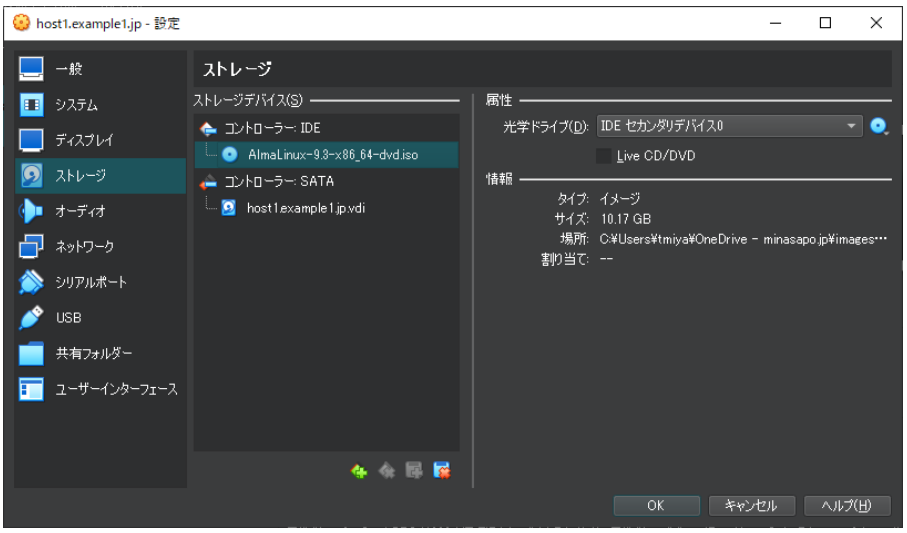

図 24: 仮想マシンの仮想光学ドライブの設定画面

以上で仮想マシンの作成と、OS インストールの準備は完了です。

3　Linux のインストールと設定

第 3 章では、VirtualBox で作成した仮想マシンに AlmaLinux をインストールします。インストール後、初期設定とネットワーク接続の確認を行います。

3.1　用語集

ISO イメージ

CD-ROM や DVD-ROM は ISO9660 という形式でファイルを保存しています。この ISO9660 形式のディスクをそのまま 1 つのファイルに保存したものを ISO イメージと呼びます。

ブートローダー

OS の起動時に最初に読み込まれるプログラムです。ブートローダーが Linux カーネルを読み込んでシステムが起動します。

パーティション

Linux はディスクを複数の区画（パーティション）に分割して利用します。データを保存するパーティションの他、仮想メモリが利用するスワップ用のパーティションなどがあります。

NAT

NAT（Network Address Translation）は IP 通信時に IP アドレスを変換して通信を行う仕組みです。IPv4 ではインターネットに直接接続できるグローバル IP アドレスが枯渇しているため、1 つのグローバル IP アドレスを複数のホストで共有して利用できるように NAT を活用しています。

root

Linux のシステム管理者のユーザー名です。

3.2　仮想マシンの起動

VirtualBox マネージャーで仮想マシンを選択し、「起動」をクリックします。別ウインドウで仮想マシンが起動します。

3.2.1　ISO イメージファイルのマウント

あらかじめ仮想マシンの仮想光学ドライブに ISO イメージを読み込ませる設定をしていない場合、起動用の ISO イメージを設定するダイアログが表示されます。

画面の指示に従って、ISO イメージのファイルを指定し、「マウントとブートのリトライ」ボタンをクリックして起動します。

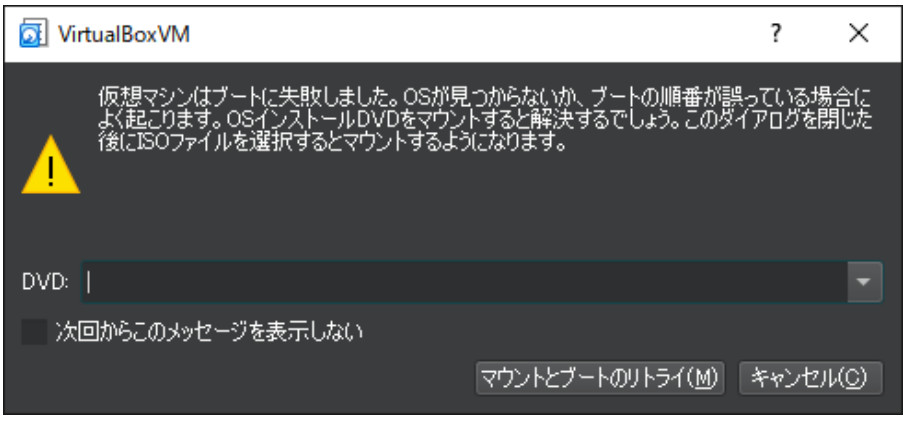

図 25: マウントとブートのリトライダイアログ画面

3.3 OS のインストール

以下の手順に従って、OS のインストールを行います。

3.3.1 インストーラ起動オプションの選択

OS インストール用の ISO イメージを読み込んで起動すると、最初に GRUB というブートローダーが起動します。ここでインストーラの起動オプションを選択できます。

デフォルトの「Test this media & Install AlmaLinux 9.3」で起動し、ISO イメージが壊れていないかテストを行った上でインストールを行います。すでに選択されているので、Enter キーを押します。

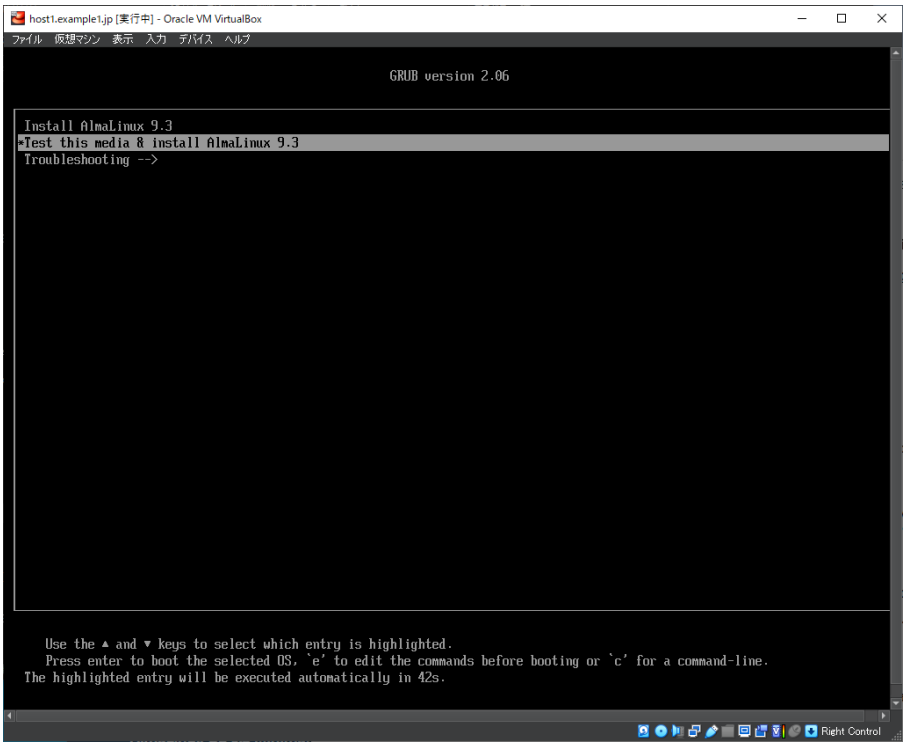

図 26: GRUB 画面

3.3.2　言語選択

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）」が表示されます。

「続行」ボタンをクリックします。

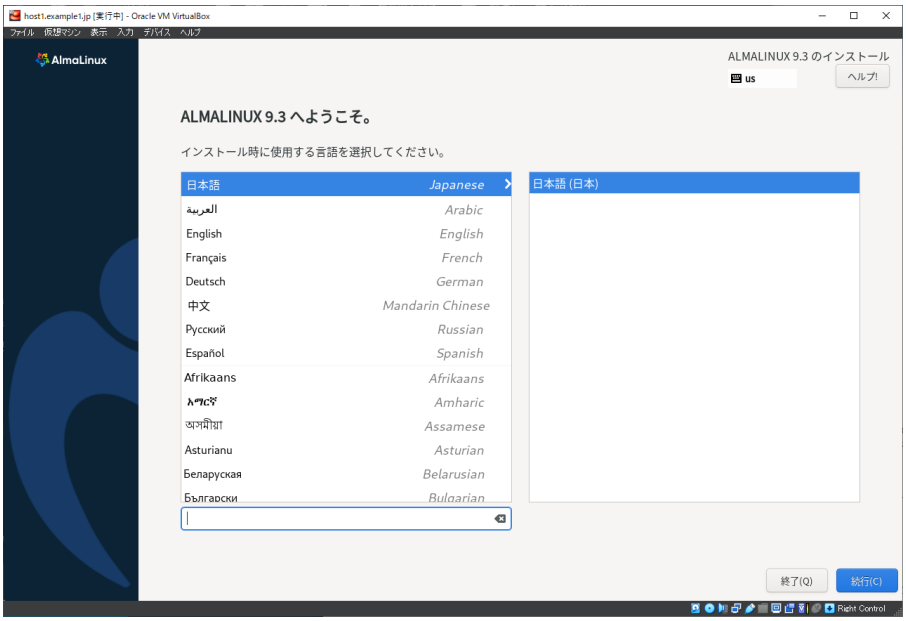

図 27: 言語選択画面

3.3.3　インストール概要

インストール概要の画面では、各種設定がまとめて表示されます。

「!」が付いた項目は、必ず設定が必要な項目です。

図 28: インストール概要画面

3.3.4　インストール先の設定

「インストール先」をクリックします。画面が切り替わり、「デバイスの選択」や「ストレージの設定」が行えます。

デフォルトで、仮想マシン作成時に新規作成した仮想ハードディスクがローカル標準ディスクとして選択されています。また、そのデバイスのパーティション設定は自動構成が選択されています。

今回はデフォルトのままインストール先を設定するので、そのまま「完了」ボタンをクリックします。

図 29: インストール先の設定画面

3.3.5　ネットワークとホスト名の設定

「ネットワークとホスト名」をクリックします。画面が切り替わり、ネットワークインターフェースの設定と、ホスト名の設定が行えます。

仮想マシンの設定で、ネットワークアダプターを 2 つ設定したので、ネットワークインターフェースが 2 つ認識されています。

3.3.6　インターネットへの接続を設定（NAT）

「Ethernet (enp0s3)」は、NAT を設定したアダプター 1 にあたります。こちらのネットワークインターフェースを経由して、外部ネットワークやインターネットに接続します。

VirtualBox の NAT は DHCP で自動的に設定されるので、以下のように設定されていることを確認します。

項目	設定値
IP アドレス	10.0.2.15/24
デフォルトルート	10.0.2.2
DNS	環境によって異なる

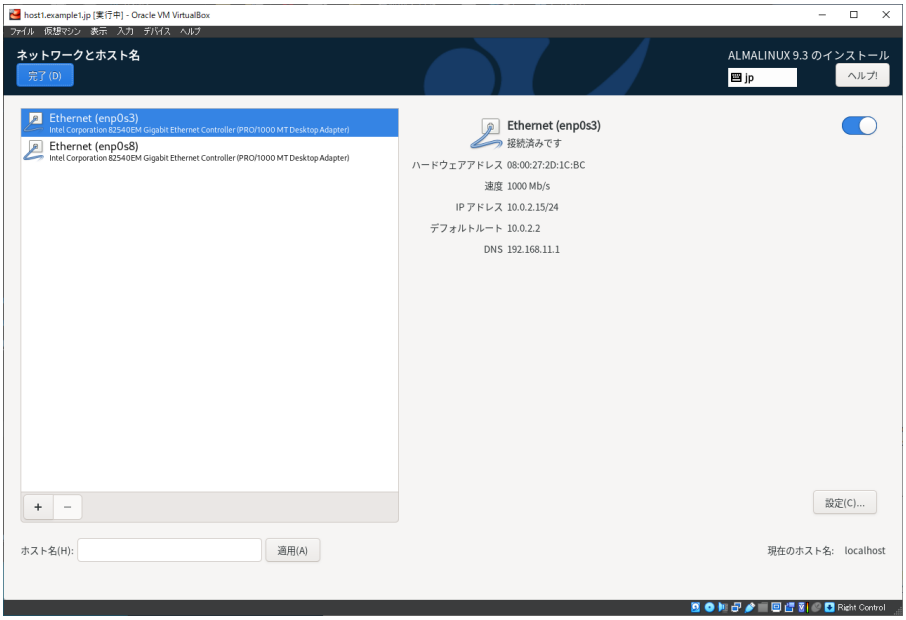

図 30: ネットワークのインターフェース画面

DNS は環境によって異なりますが、ホスト OS 側のネットワーク設定と同じになるので、ホスト OS のネットワーク設定を確認して適切な参照先 DNS が設定されていることを確認します。

3.3.7　ホスト OS との接続を設定（ホストオンリー）

「Ethernet (enp0s8)」は、ホストオンリーを設定したアダプター 2 にあたります。ホスト OS から仮想マシン上で動作するゲスト OS に接続します。

VirtualBox のネットワーク設定でホストオンリーアダプターの設定の際に DHCP サーバーを有効にしていれば、「192.168.56.3」などの IP アドレスが自動的に設定されていますが、サーバーとして使用するため固定 IP アドレスを設定します。

設定する値は以下の通りです。

項目	設定値
アドレス	192.168.56.101
ネットマスク	24（自動的に設定される）
ゲートウェイ	設定しない
DNS サーバー	192.168.56.101

手順は以下の通りです。

1. 「Ethernet (enp0s8)」を選択し、「設定 (C)…」をクリックします。
2. 「enp0s8 の編集」ダイアログが表示されるので、「IPv4 設定」タブを選択します。
3. 「メソッド (M)」を「手動」に変更します。
4. 「アドレス」で「追加」をクリックし、各項目を以下の値に設定します。DNS は自分自身を参照するように「192.168.56.101」を設定します。
5. 「保存 (S)」ボタンをクリックします。

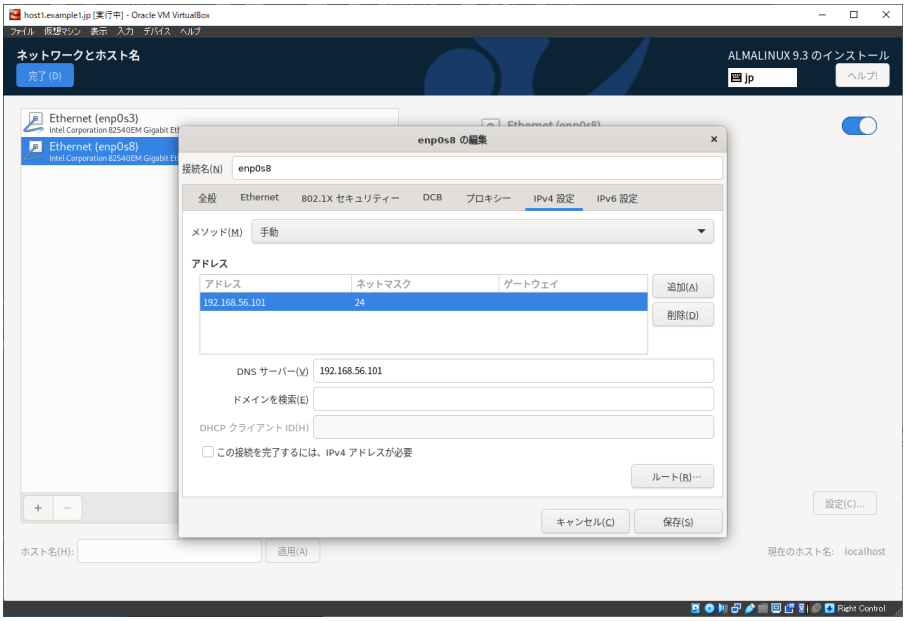

図 31: ホストオンリーネットワークのインターフェース設定画面

3.3.8　ホスト名の設定

ホスト名を設定します。ホスト名は以下の通り設定します。

項目	設定値
ホスト名	host1.example1.jp

入力後、「適用 (A)」ボタンをクリックします。

「現在のホスト名」の表示が変更されるのを確認します。

2 つのネットワークインターフェースおよびホスト名の設定が完了したら、「完了 (D)」ボタンをクリックします。

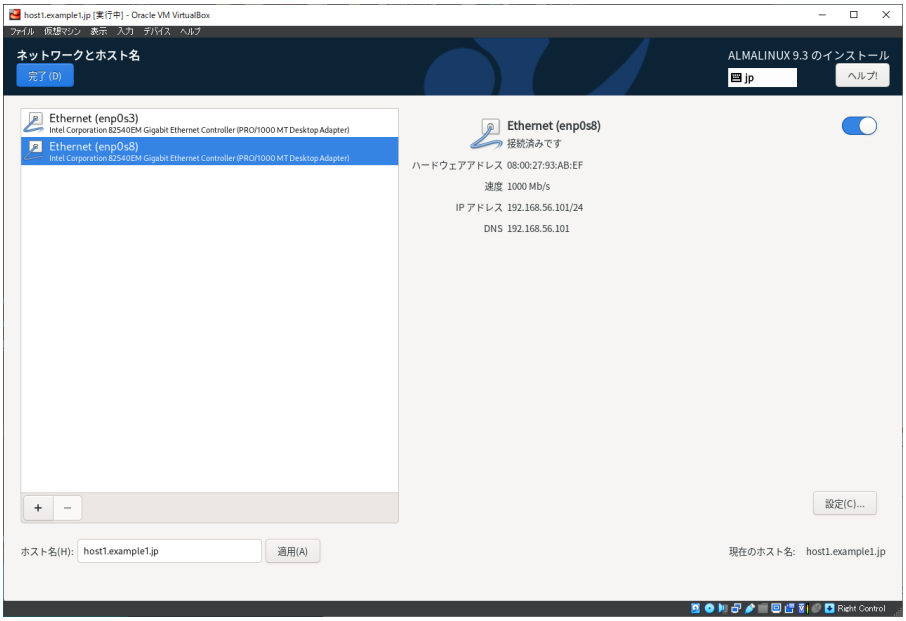

図 32: ホスト名の設定画面

3.3.9　ソフトウェアの追加インストール（インターネット接続ができない場合のみ）

インターネットに接続できない環境の場合、実習で利用するソフトウェアをあらかじめインストールしておきます。

上記「ソフトウェアの選択」画面で、右側の「選択した環境用のその他のソフトウェア」リストから、インストールしたいソフトウェアをチェックします。本教科書の実習のためには、以下のソフトウェアをチェックして追加でインストールします。

- DNS ネームサーバー
- メールサーバー
- ベーシック Web サーバー

ただし、これら 3 つを追加でインストールしても、メールを送受信するためのソフトウェアがインストールされません。それらは別途 ISO イメージから手動で追加インストールが必要となります。

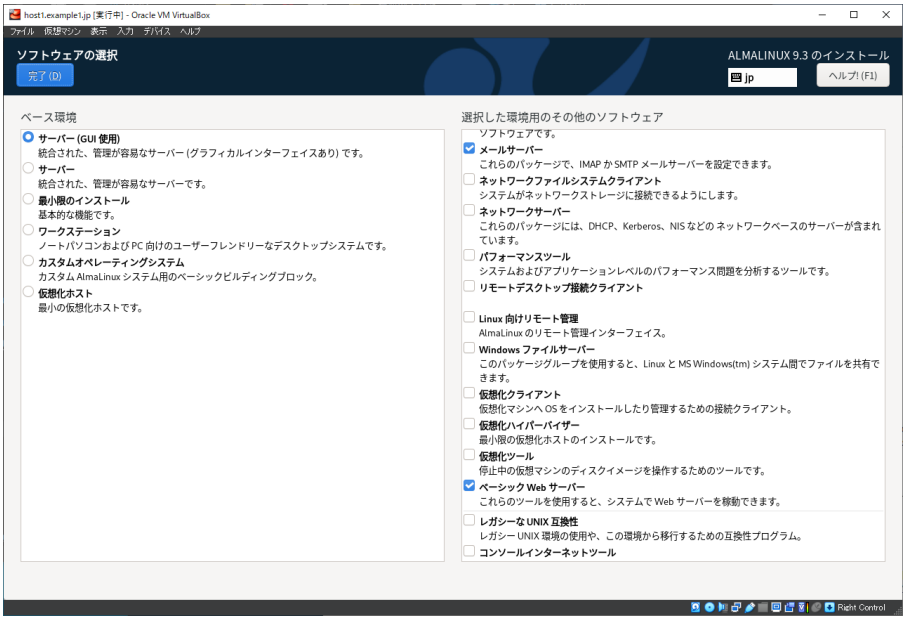

図 33: ソフトウェアの追加選択画面

3.3.10　root パスワードの設定

「root パスワード (R)」をクリックします。Linux のシステム管理権限を持つ root ユーザーのパスワードを設定します。

root パスワードは、入力間違えが無いように「root パスワード (R)」と「確認 (C)」に同じパスワードを 2 回入力します。長さや大文字小文字数字を混ぜるなど一定の要件を満たさないと脆弱なパスワードと判定されます。良好以上になるパスワードを入力するか、脆弱と判定された場合には「完了 (D)」ボタンを 2 回クリックする必要があります。

root パスワードの設定が完了したら、「完了 (D)」ボタンをクリックします。

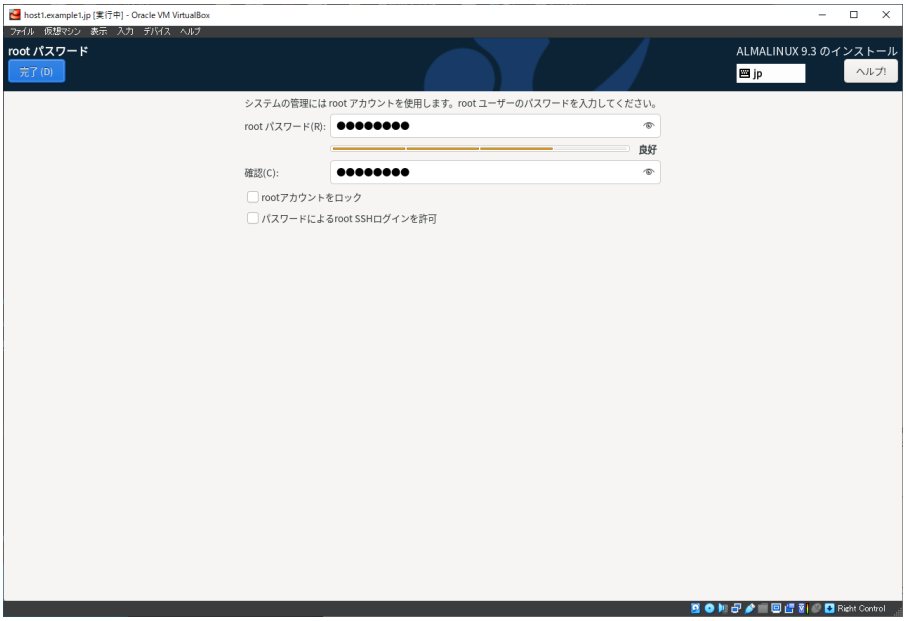

図 34: root パスワードの設定画面

3.3.11　インストールの開始

必要な設定が終わったら、インストールを開始します。「インストール開始」ボタンをクリックします。

図 35: インストール開始前の画面

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。

インストールが終了したら、「システムの再起動」ボタンをクリックして、再起動します。

図 36: インストール終了の画面

3.4　インストール後の初期設定

再起動ができたら、初期設定を行います。

「セットアップ開始 (S)」ボタンをクリックします。

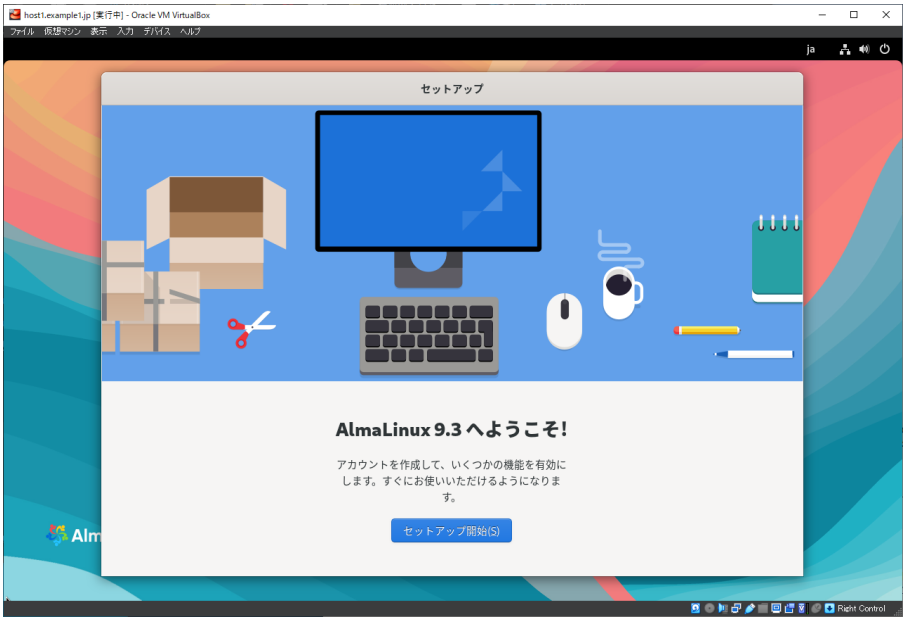

図 37: セットアップ開始の画面

3.4.1　プライバシー

位置情報サービスを使って情報を提供するかどうかを設定します。

システムの動作には影響がありませんので、設定はオンでもオフでも、どちらでも構いません。

設定したら、「次へ (N)」ボタンをクリックします。

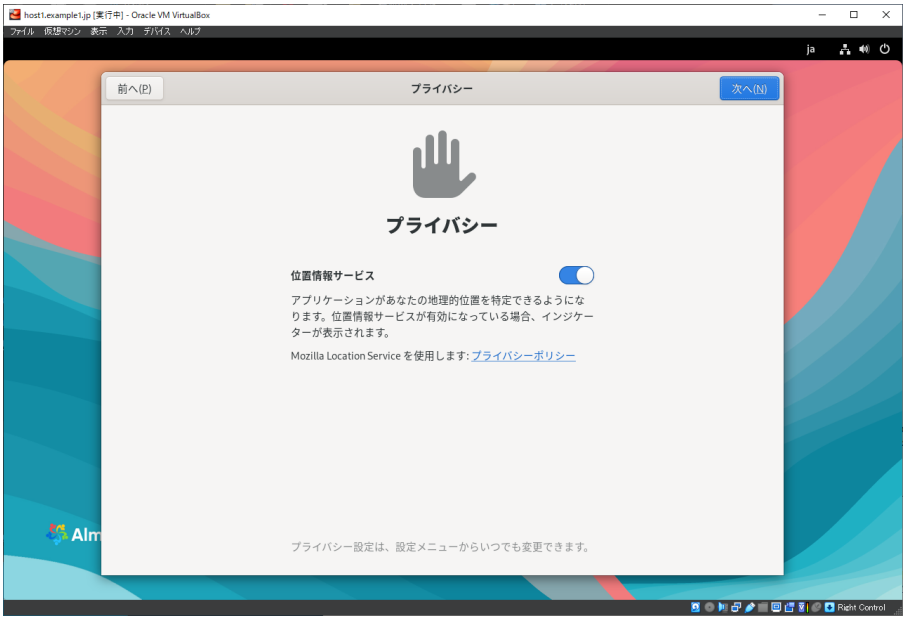

図 38: プライバシー設定の画面

3.4.2　オンラインアカウント

オンラインアカウントへの接続を設定します。

接続する必要はありませんので、「スキップ (S)」ボタンをクリックします。

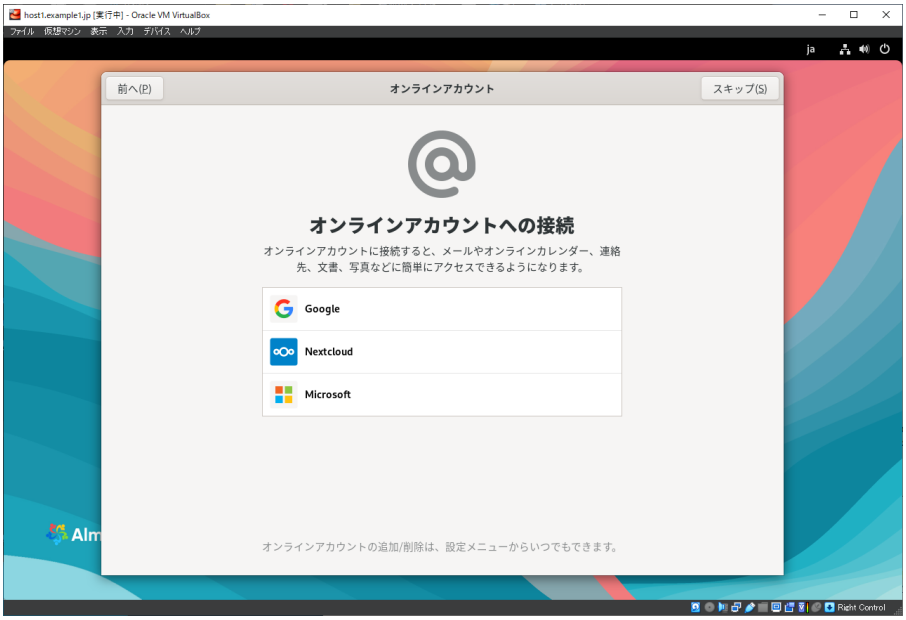

図 39: オンラインアカウント設定の画面

3.4.3　ユーザー情報

初期ユーザーを作成します。演習用のサーバー管理ユーザーとして、以下のように設定します。

項目	設定値
フルネーム	admin
ユーザー名	admin

設定したら、「次へ (N)」ボタンをクリックします。

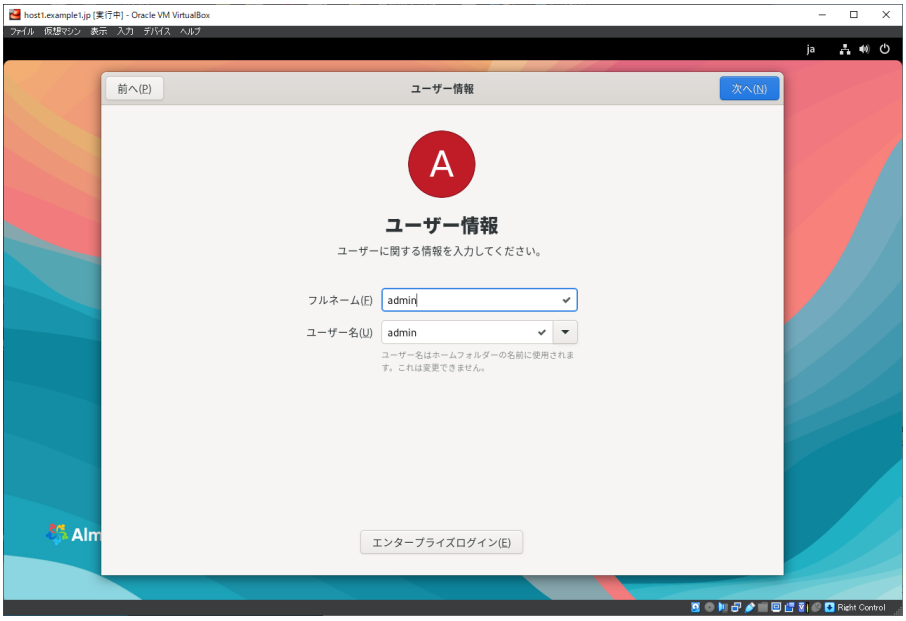

図 40: ユーザー名設定の画面

admin ユーザーのパスワードも設定します。root パスワード同様、「パスワード」と「確認」に同じパスワードを入力します。

設定したら、「次へ (N)」ボタンをクリックします。

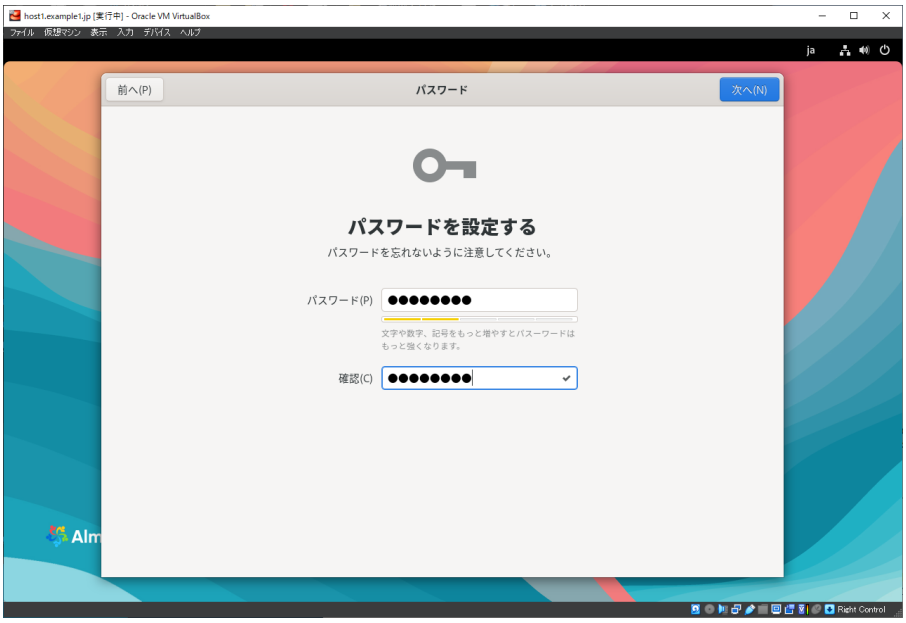

図 41: パスワード設定の画面

すべての設定を完了したら、「AlmaLinux を使い始める (S)」ボタンをクリックします。

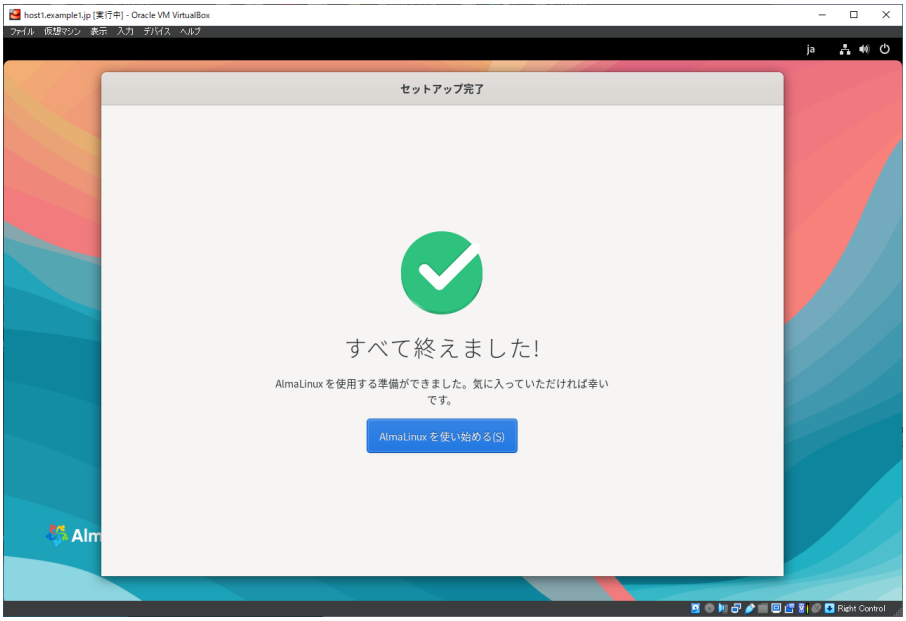

図 42: AlmaLinux を使い始めるの画面

初期設定画面から GUI 画面に切り替わります。「AlmaLinux へようこそ」と表示され、操作方法を確認するツアーを開始できます。ツアーを見たい場合には「ツアーをチェックする」ボタンをクリックします。はじめて AlmaLinux の GUI を操作するのであれば、短いツアーですのでチェックしてみてください。

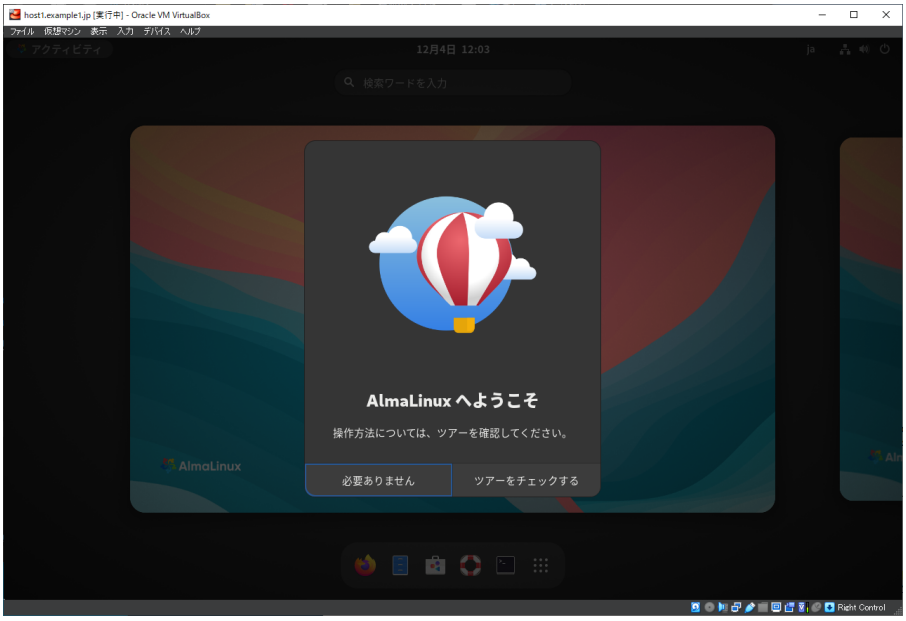

図 43: AlmaLinux へようこその画面

3.5　ログインとログアウト

Linux を使い始めるにはログインを、使い終わったらログアウトを行います。

現在は、初期ユーザーで自動的にログインした状態なので、一度ログアウトし、再度ログインしてみます。

3.5.1　ログアウトする方法

ログアウトするには、画面右上にあるメニューバーの電源アイコンをクリックし、「電源オフ/ログアウト」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。

ログアウトを確認するダイアログが表示されるので、「ログアウト」をクリックします。

図 44: ログアウト選択の画面

3.5.2　ログインする方法

ログインするには、ログインしたいユーザーをクリックして選択し、パスワードを入力します。

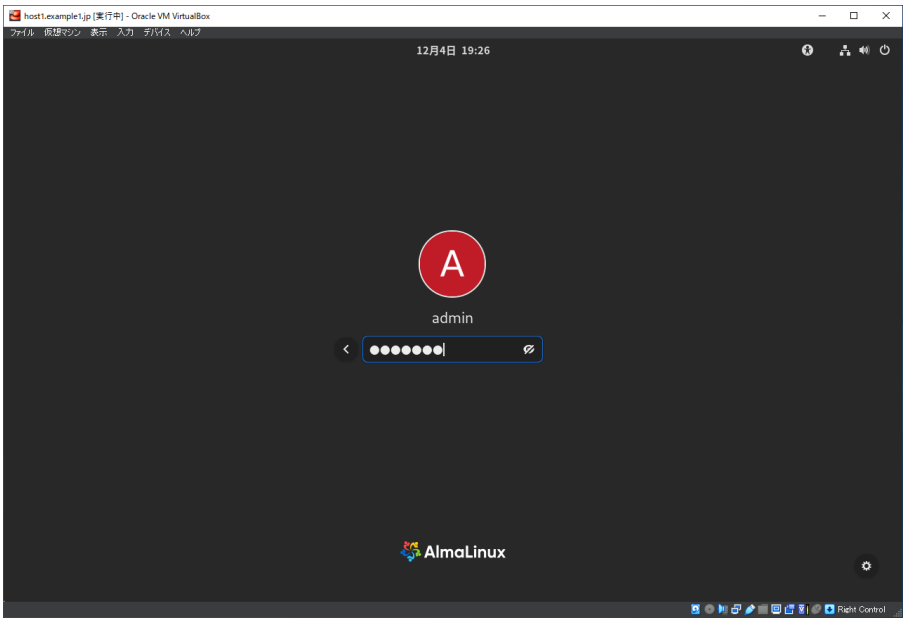

図 45: ログインの画面

3.6 コマンド実行のための端末を起動

コマンドを実行して Linux を操作するために「端末」アプリを起動します。

画面左上の「アクティビティ」をクリックし、画面下に表示されるアイコンから「端末」アプリのアイコンをクリックします。

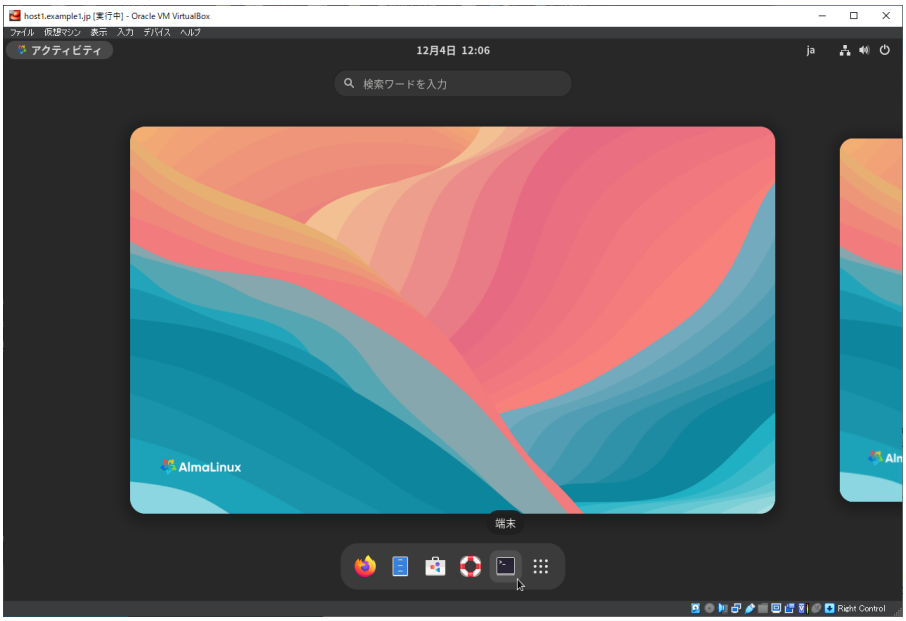

図 46: 端末起動の画面

また、仮想マシンで直接操作するのではなく、ホスト OS から SSH を使ってリモート接続して操作することもできます。方法については第 7 章の解説を参考にしてください。

3.7 ネットワーク接続の確認

ネットワークに正しく接続されているかの確認を行います。ネットワークは NAT を経由した外部ネットワークやインターネットの接続と、ホストオンリーネットワークを経由したホスト OS との接続があります。それぞれについてテストします。

3.7.1 名前解決の確認

dig コマンドを使って、DNS による名前解決を確認します。

```
$ dig lpi.or.jp
 （略）
;; QUESTION SECTION:
;lpi.or.jp.          IN   A

;; ANSWER SECTION:
lpi.or.jp.       5    IN   A    219.94.215.12
 （略）
```

きちんと DNS の名前解決が行えていることが分かります。

3.7.2 インターネットへの接続の確認

ping コマンドを使って、インターネット上のサーバーへの接続を確認します。

```
$ ping lpi.or.jp
PING lpi.or.jp (219.94.215.12) 56(84) bytes of data.
64 バイト応答 送信元 12.215.94.219.static.www232b.sakura.ne.jp (219.94.215.12):
    icmp_seq=1 ttl=128 時間=15.8ミリ秒
^C
--- lpi.or.jp ping 統計 ---
送信パケット数 1, 受信パケット数 1, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 15.780/15.780/15.780/0.000 ms
```

ping コマンドの実行を停止するには Ctrl + c を押します。正しく通信できていることがわかります。宛先の IP アドレスが dig コマンドで調べた IP アドレスと同じになっていることも分かります。

ping コマンドでの接続に対する応答を返さないサーバーもあるので、返答が無い場合にはその他のサーバーへの接続も試してみるか、Linux の GUI で Web ブラウザを起動して Web サイトへ接続するなども試してみるとよいでしょう。

3.7.3 ゲスト OS からホスト OS への接続の確認

ping コマンドで、ホスト OS への接続を確認します。ホスト OS 側の IP アドレスは、VirtualBox のネットワーク設定で確認した「192.168.56.1」になります。

```
$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 バイト応答 送信元 192.168.56.1: icmp_seq=1 ttl=64 時間=6.59ミリ秒
^C
--- 192.168.56.1 ping 統計 ---
送信パケット数 1, 受信パケット数 1, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.592/6.592/6.592/0.000 ms
```

3.7.4 ホスト OS からゲスト OS への接続の確認

ホスト OS 側でコマンドプロンプトを起動して、ping コマンドでゲスト OS への接続を確認します。ゲスト OS 側の IP アドレスは「192.168.56.101」になります。

コマンドプロンプトを起動するには、画面左下の「ここに入力して検索」に「cmd」と入力します。「コマンドプロンプト」が検索されるので、クリックして起動します。

ping コマンドを実行します。

```
> ping 192.168.56.101

192.168.56.101 に ping を送信しています 32 バイトのデータ:
192.168.56.101 からの応答: バイト数 =32 時間 <1ms TTL=64

192.168.56.101 の ping 統計:
    パケット数: 送信 = 1、受信 = 1、損失 = 0 (0% の損失)、
ラウンド トリップの概算時間 (ミリ秒):
    最小 = 0ms、最大 = 0ms、平均 = 0ms
Ctrl+C
^C
```

確認が行えたら Ctrl+C を入力して動作を停止します。

4 Web サーバーの構築

第 4 章では、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。パッケージのインストールやシステム管理権限、セキュリティを考慮したネットワークのアクセス制限などについても触れます。

4.1 用語集

HTML(HyperText Markup Language)

Web ページ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、画像を利用したり、リストや表などの高度な表現も可能です。現在では、ページレイアウトなどを定義する CSS（Cascading Style Sheets）や、プログラムを記述できる JavaScript などと組み合わせて高度な Web ページを作成できます。

HTTP(HyperText Transfer Protocol)

Web ブラウザと 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。現在では、通信を暗号化するなどしてセキュリティを高めた HTTPS（Hypertext Transfer Protocol Secure）が使用されています。

Apache HTTP 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開発が行われているオープンソースソフトウェアです。

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。

パッケージ

プログラムの本体であるバイナリや設定ファイル、ドキュメントなどを一式まとめてインストール可能なようにまとめたものです。大きなプログラムの場合、本体とライブラリ、各種拡張機能などを別々のパッケージとしてまとめて選択的にインストールできるようにしていることがあります。Linux ディストリビューションは、このパッケージの集合体と考えることができます。

システム管理権限

Linux は複数のユーザーが同時に利用できるマルチユーザー型の OS です。システム管理権限は、システム全体に対する変更などが行える権限で、一般的なユーザーには与えられていません。Linux では root ユーザーか、sudo コマンドを使える権限が与えられたユーザーのみシステム管理権限を行使できます。

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型のシステムで、Web サーバーとクライアントの Web ブラウザとで構成されます。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。

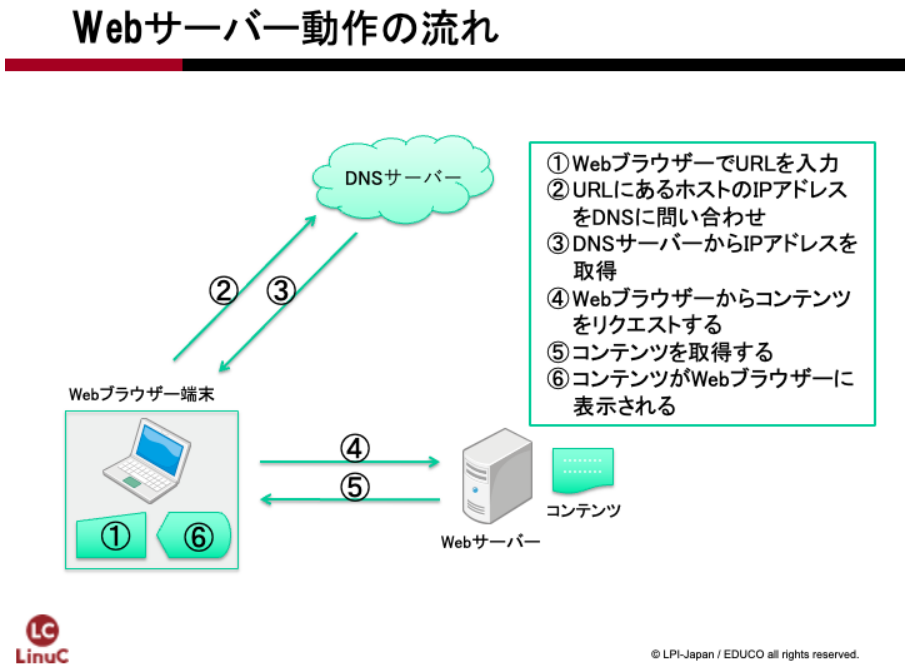

図 47: Web サーバー動作の流れ

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 HTML が一般的に使われています。

Web サーバーとして Apache HTTP サーバーが多く使われていますが、Nginx（エンジンエックス）や各種 Web アプリケーションを動作させるのに特化した Web アプリケーションサーバーなども使われています。

4.3　パッケージのインストール

Linux で Web サーバーを動作させるには、まず Web サーバーのインストールを行います。ソースコードをコンパイルして動かすこともできますが、パッケージを使えば簡単にインストールが行えます。実習で使用している AlmaLinux では RPM 形式のパッケージを使用しており、パッケージ管理ツールとして dnf コマンドが使用できます。

4.3.1　dnf コマンド

dnf コマンドを使うと、パッケージのインストールや削除、アップデートなどが行えます。

4.3.2　dnf コマンドと yum コマンド

dnf コマンドは、従来は yum コマンドが行っていたパッケージ管理を置き換えるコマンドです。dnf install コマンド等、使用しているサブコマンドはそのまま yum コマンドでも実行できます。

4.3.3　sudo コマンドによる root 権限の取得

dnf コマンドによるパッケージのインストールは、システムの変更を伴うため管理者である root ユーザーの権限が必要になります。コマンド実行時に root 権限を取得するには、sudo コマンドを使用します。OS インストール時、あるいは初期設定時に作成したユーザーには sudo コマンドを実行する権限が与えられています。

4.3.4 dnf コマンドが参照するパッケージリポジトリ

dnf コマンドは、インストールに使用するパッケージをリポジトリと呼ばれる場所から取得します。通常はインターネット上にリポジトリサーバーが用意されており、インターネット経由でパッケージをダウンロードします。今回はインターネットにアクセスできる前提で作業を行います。

4.3.5 プロキシーが必要な場合には

インターネットへのアクセスにプロキシーを経由する必要がある場合、dnf コマンドの設定でプロキシーを設定することでアクセスできるようになります。設定方法はマニュアル等を参照してください。

4.3.6 インターネットにアクセスできない環境でのパッケージ管理

インターネットにアクセスできない環境ではインターネット上のリポジトリにアクセスできないため、以下のような方法でインストールを行う必要があります。本教科書ではインストール時に必要なソフトウェアを選択してインストールする方式を採っていますが、その他の方法が必要となる場合もあります。

- OS インストール時にあらかじめ必要なソフトウェアを選択してインストールしておく
- ISO イメージに含まれている RPM パッケージファイルを dnf コマンドを使って手動でインストールする
- ISO イメージをリポジトリとして扱うように設定ファイルを作成する
- アクセス可能なローカルネットワーク上にリポジトリサーバーを用意する

ただし、リポジトリサーバーを用意する以外の方法ではセキュリティアップデートなどが行われた最新のパッケージをインストールするのが難しくなります。実際のシステムを運用するには、継続的にパッケージをアップデートできる方法を考えておくべきでしょう。

4.3.7 依存関係の解消

動作するために複数のパッケージが必要となる場合、dnf コマンドはインストールするパッケージが必要とするパッケージも同時にインストールを行います。必要となるパッケージがあることを依存関係と呼びます。

4.4 dnf install コマンドによるパッケージのインストール

Apache HTTP サーバーのパッケージ名は httpd です。インストールするには、sudo コマンドを頭に付けて dnf install コマンドを実行します。

```
$ sudo dnf install httpd
メタデータの期限切れの最終確認: 1:41:14 前の 2023年12月05日 08時09分38秒 に実施
    しました。
依存関係が解決しました。
================================================================================
 パッケージ                Arch       バージョン              リポジトリ   サイズ
================================================================================
インストール:
 httpd                     x86_64     2.4.57-5.el9            appstream      46 k
依存関係のインストール:
 almalinux-logos-httpd     noarch     90.5.1-1.1.el9          appstream      18 k
弱い依存関係のインストール:
 mod_http2                 x86_64     1.15.19-5.el9           appstream     148 k
 mod_lua                   x86_64     2.4.57-5.el9            appstream      60 k

トランザクションの概要
================================================================================
インストール  4 パッケージ

ダウンロードサイズの合計: 272 k
インストール後のサイズ: 601 k
これでよろしいですか? [y/N]: y ← yと入力
パッケージのダウンロード:
(1/4): almalinux-logos-httpd-90.5.1-1.1.el9.noa  33 kB/s |  18 kB     00:00
（略）
GPG 鍵 0xB86B3716 をインポート中:
```

```
 Userid     : "AlmaLinux OS 9 <packager@almalinux.org>"
 Fingerprint: BF18 AC28 7617 8908 D6E7 1267 D36C B86C B86B 3716
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-AlmaLinux-9
これでよろしいですか? [y/N]: y ← yと入力
鍵のインポートに成功しました
トランザクションの確認を実行中
トランザクションの確認に成功しました。
トランザクションのテストを実行中
トランザクションのテストに成功しました。
トランザクションを実行中
  準備              :                                                        1/1
  インストール中    : mod_lua-2.4.57-5.el9.x86_64                            1/4
  インストール中    : almalinux-logos-httpd-90.5.1-1.1.el9.noarch             2/4
  インストール中    : mod_http2-1.15.19-5.el9.x86_64                         3/4
  インストール中    : httpd-2.4.57-5.el9.x86_64                              4/4
  scriptletの実行中: httpd-2.4.57-5.el9.x86_64                              4/4
  検証              : almalinux-logos-httpd-90.5.1-1.1.el9.noarch            1/4
  検証              : httpd-2.4.57-5.el9.x86_64                              2/4
  検証              : mod_http2-1.15.19-5.el9.x86_64                         3/4
  検証              : mod_lua-2.4.57-5.el9.x86_64                            4/4

インストール済み:
  almalinux-logos-httpd-90.5.1-1.1.el9.noarch    httpd-2.4.57-5.el9.x86_64
  mod_http2-1.15.19-5.el9.x86_64                 mod_lua-2.4.57-5.el9.x86_64

完了しました!
```

sudo コマンドを初めて実行する際には、実行しているユーザーのパスワードが要求されます。sudo コマンド実行後、しばらくの間は再度実行する際にはパスワードが要求されませんが、一定時間経過後は再度要求されます。

dnf コマンドはリポジトリにアクセスし、新たに利用可能になったパッケージのリストを取得してパッケージデータベースを更新します。

インストールを行う httpd パッケージの依存関係を確認し、必要となる追加のパッケージも同時にインストールを行います。実行例では、依存関係にある 1 つのパッケージ、弱い依存関係にある 2 つのパッケージの合計 3 のパッケージも追加でインストールすることを提案しています。問題なければ y を入力してパッケージのダウンロードとインストールを行います。

4.5 Web サーバーの起動

Web サーバーを起動します。systemctl コマンドを使って、Web サーバーをバックグラウンドサービスとして起動します。

4.5.1 systemctl コマンド

systemctl コマンドは systemd を制御するためのコマンドです。systemd は Linux が OS として起動する際、Linux カーネルが起動した後に一番最初に実行されるプロセスで、OS 全体を制御します。systemctl コマンドは管理対象をユニットという単位で管理します。

4.5.2 systemctl start コマンドによる Web サーバーの起動

Web サーバーを起動するには、systemctl start コマンドを実行します。systemd は Web サーバーを httpd.service ユニットとして管理しています。ユニット名の.service は省略形の名称が重複していない限り省略できます。

```
$ sudo systemctl start httpd
```

エラーなどが発生しなければ、systemctl start コマンドの実行は何も表示されず終了します。

4.6 systemctl status コマンドによる Web サーバーの動作確認

Web サーバーが正しくバックグラウンドサービスとして実行されているかを確認します。システム的に確認する方法と、Web サーバーとして動作していることの確認の両方を実行します。

4.6.1 systemctl status コマンドによる動作状態の確認

systemctl status コマンドで、管理対象となるユニットの状態やログの一部などを確認できます。

表示を終了するには Q キーを入力します。

```
$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d>
     Active: active (running) since Tue 2023-12-05 09:52:55 JST; 1s ago
       Docs: man:httpd.service(8)
   Main PID: 37780 (httpd)
     Status: "Started, listening on: port 443, port 80"
      Tasks: 214 (limit: 10786)
     Memory: 30.3M
        CPU: 46ms
     CGroup: /system.slice/httpd.service
             tq37780 /usr/sbin/httpd -DFOREGROUND
             tq37781 /usr/sbin/httpd -DFOREGROUND
             tq37782 /usr/sbin/httpd -DFOREGROUND
             tq37783 /usr/sbin/httpd -DFOREGROUND
             tq37784 /usr/sbin/httpd -DFOREGROUND
             mq37785 /usr/sbin/httpd -DFOREGROUND

12月 05 09:52:55 host1.example1.jp systemd[1]: Starting The Apache HTTP Server.>
12月 05 09:52:55 host1.example1.jp systemd[1]: Started The Apache HTTP Server.
12月 05 09:52:55 host1.example1.jp httpd[37780]: Server configured, listening o>
```

4.6.2 Loaded の意味

ユニットの設定が systemd に読み込まれているかどうかを表しています。ユニットの定義ファイルの位置や、自動起動の設定がされているかが確認できます。

4.6.3 Active の意味

現在、ユニットがアクティブかどうかを表しています。また、プロセスが動作しているかどうかが確認できます。

4.7 Web サーバーへの接続の確認

Web サーバーが起動できたら、様々な方法で Web サーバーに接続して動作を確認します。

4.7.1 curl コマンドによるローカル接続確認

ローカルで Web サーバーが動作していることを確認します。curl コマンドを実行して、自分自身（ローカル）を指す localhost に接続してみます。

```
$ curl localhost
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
        <head>
                <title>Test Page for the HTTP Server on AlmaLinux</title>
                <meta http-equiv="Content-Type" content="text/html;
                    charset=UTF-8" />
```

```
                <div class="footer">
                        <a href="https://apache.org">Apache&trade;</a> is a
                            registered trademark of <a
                            href="https://apache.org">the Apache Software
                            Foundation</a> in the United States and/or other
                            countries.<br />
                        <a href="https://nginx.com">NGINX&trade;</a> is a
                            registered trademark of <a
                            href="https://www.f5.com">F5 Networks, Inc.</a>.
                </div>
        </body>
</html>
```

Web サーバーが実行されているので、サンプルページの HTML が返ってきます。

4.7.2　ゲスト OS の Web ブラウザによるローカル接続確認

仮想マシンの GUI を使って Web ブラウザを実行し、Web サーバーに接続してみます。

1. 画面左上の「アクティビティ」をクリック
2. 画面下に並んでいるアイコンから一番左の Firefox を実行
3. アドレスに「localhost」を入力し Enter キーを押す

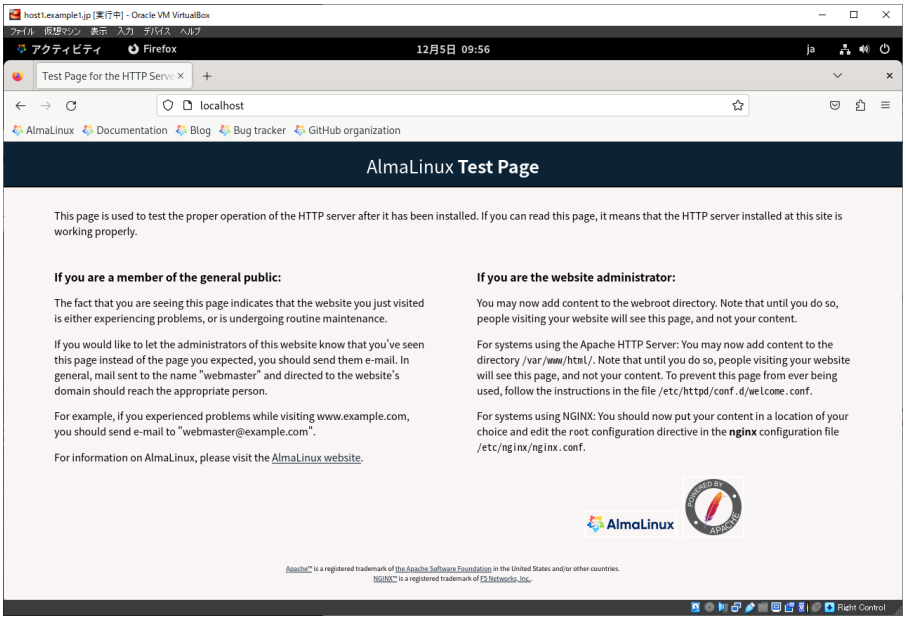

図 48: Firefox から localhost へ接続した画面

Web ブラウザの要求に対して、Web サーバーが HTML を返し、Web ブラウザがその HTML を解釈して Web ページを表示しているのが分かります。HTML が要求したことにより追加で取得した画像ファイルも埋め込まれています。1 つの Web ページを表示するのに、Web ブラウザの裏側で複数のセッションがやり取りされて画像ファイルなどが取得されるようになっています。

4.8　Web サーバーの停止

Web サーバーを停止してみます。停止には systemctl stop コマンドを実行します。

```
$ sudo systemctl stop httpd
```

4.8.1 systemctl status コマンドによる動作状態の確認

systemctl status コマンドを実行して、動作状態を確認します。

```
$ sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d>
     Active: inactive (dead)
       Docs: man:httpd.service(8)

12月 05 09:52:55 host1.example1.jp systemd[1]: Starting The Apache HTTP Server.>
12月 05 09:52:55 host1.example1.jp systemd[1]: Started The Apache HTTP Server.
12月 05 09:52:55 host1.example1.jp httpd[37780]: Server configured, listening o>
12月 05 09:57:37 host1.example1.jp systemd[1]: Stopping The Apache HTTP Server.>
12月 05 09:57:38 host1.example1.jp systemd[1]: httpd.service: Deactivated succe>
12月 05 09:57:38 host1.example1.jp systemd[1]: Stopped The Apache HTTP Server.
```

Active 行が inactive (dead) になっており、プロセス関係の情報が表示されなくなり、Web サーバーが停止しているのが分かります。

4.8.2 curl コマンドによる接続確認

Web サーバーが停止している状態で、curl コマンドを実行してみます。

```
$ curl localhost
curl: (7) Failed to connect to localhost port 80: 接続を拒否されました
```

即座にエラーが表示されて、Web サーバーにアクセスできないのが分かります。

4.8.3 Web サーバーを再度起動

Web サーバーを再度起動し、正常にアクセスできるように復旧したことを確認してください。

```
$ sudo systemctl start httpd
```

```
$ curl localhost
```

4.9 ファイアウォールの設定とリモート接続確認

セキュリティを強化するため、デフォルトでは外部からは必要最低限のネットワーク通信しか受け付けないようにファイアウォールが設定されています。そのため、ローカルでは Web サーバーに接続できても、リモートからは接続できません。ファイアウォールの設定を変更して、リモートからの接続を許可します。

4.9.1 仮想マシンでのリモート接続の確認方法

リモート接続の確認は、ホスト OS 上で起動した Web ブラウザからゲスト OS 上の Web サーバーに接続します。同一の物理マシンですが、論理的には別々の OS、別々の IP アドレスを持っており、ネットワークを経由してリモートにある Web サーバーに接続していることになります。

4.9.2 デフォルト状態では接続できないことを確認する

まず、現在のファイアウォールの設定を確認します。ファイアウォールの操作には、firewall-cmd コマンドを使用します。

```
$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
```

```
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

services 行に http が含まれていないので、接続が許可されていません。

ホスト OS 上で Web ブラウザを起動したら、アドレスに「192.168.56.101」を入力し、Enter キーを押します。

図 49: ホスト OS から接続できずタイムアウトした画面

Web サーバーからの結果を待ち続けており、テストページは表示されません。Web サーバーは動作していますが、ファイアウォールが手前で遮っているため Web サーバーに到達できないためです。そのまま待っていると、Web ブラウザがタイムアウトしてエラーが表示されます。

4.9.3　ファイアウォールの設定を変更して接続を許可する

firewall-cmd コマンドで http サービスの接続を許可します。

```
$ sudo firewall-cmd --add-service=http --zone=public
success
```

再度ファイアウォールの設定を確認します。

```
$ sudo firewall-cmd --list-all
public (active)
  target: default
 （略）
  services: cockpit dhcpv6-client http ssh
 （略）
```

services 行に http が追加され、接続が許可されました。

再度、ホスト OS 上の Web ブラウザから「192.168.56.101」に接続します。

テストページが表示されたら、ホスト OS からのアクセスはファイアウォールを通過して Web サーバーに接続できています。

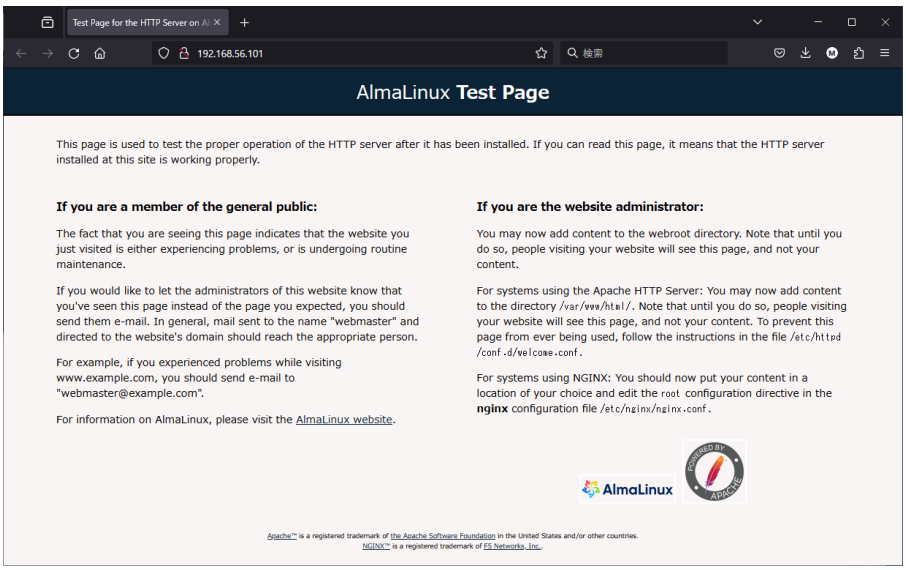

図 50: ホスト OS から接続できた画面

4.10 システム起動時の Web サーバー自動起動とファイアウォールの設定

Web サーバーの起動や、ファイアウォールの設定を行いましたが、これらの設定は OS を再起動すると無効になってしまいます。OS を再起動しても Web サーバーが自動的に起動し、ファイアウォールの設定が行われて HTTP によるアクセスが可能になるように設定しましょう。

4.10.1 OS 再起動

まず、OS を再起動し、Web サーバーが自動起動せず、ファイアウォールの設定が行われないことを確認します。OS を再起動するには reboot コマンドを実行します。reboot コマンドは sudo コマンドを付けて実行しないでも実行できます。

```
$ reboot
```

4.10.2 再起動後の動作確認

Web サーバーが自動的に起動していないこと、ファイアウォールの設定で HTTP が許可されていないことを確認します。

```
$ sudo systemctl status httpd
```

```
$ sudo firewall-cmd --list-all
```

4.10.3 Web サーバーの自動起動設定

Web サーバーを OS 起動時に自動的に起動するようにするには systemctl enable コマンドを実行します。

```
$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
    /usr/lib/systemd/system/httpd.service.
```

4.10.4 ファイアウォールの自動設定

OS 起動時にファイアウォールを自動的に設定するようにするには firewall-cmd に permanent オプションを付けて実行します。permanent オプションを付けて実行した場合、ファイアウォールの設定に即座には反映されないので、firewall-cmd コマンドを reload オプションをつけて実行し、変更を反映させる必要があります。

```
$ sudo firewall-cmd --add-service=http --zone=public --permanent
$ sudo firewall-cmd --reload
$ sudo firewall-cmd --list-all
```

4.10.5 OS 再起動と確認

再度、OS を再起動します。再起動後、Web サーバーが自動起動していること、ファイアウォールの設定が行われていることを確認します。

```
$ sudo systemctl status httpd
```

```
$ sudo firewall-cmd --list-all
```

4.11 ログの確認

Web サーバーへのアクセスや発生したエラーはログファイルに記録されています。アクセスログはどのような人がどのようなページにアクセスしているかの分析に、エラーログは発生した障害の解決に利用されます。

それぞれ、どのような情報が記録されているか確認してみましょう。

4.11.1 Web サーバーのログファイルの確認

Web サーバーのログは/var/log/httpd ディレクトリに記録されています。このディレクトリの中にあるファイルを確認するには、管理者権限が必要です。

```
$ sudo ls /var/log/httpd
access_log  error_log
```

access_log と error_log の 2 つのログファイルが確認できます。

4.11.2 アクセスログの確認

アクセスログに記録されている内容を確認します。

```
$ sudo cat /var/log/httpd/access_log
::1 - - [05/Dec/2023:10:12:37 +0900] "GET / HTTP/1.1" 403 4681 "-" "curl/7.76.1"
192.168.56.1 - - [05/Dec/2023:10:12:45 +0900] "GET / HTTP/1.1" 403 4681 "-"
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101
    Firefox/120.0"
```

アクセスログには、以下のような情報が記録されています。

- アクセス元の IP アドレス
- アクセスした時間
- アクセスの内容
- アクセスの結果（エラー番号）
- アクセスした際に入力された URL
- アクセスに利用されたブラウザの種類

4.11.3 エラー番号

アクセスログの中で分かりにくいのが、アクセスの結果を示すエラー番号です。エラー番号は様々な種類が定義されていますが、現時点で記録されているのは主に以下の 2 つです。

- 200：アクセスが成功した
- 404：アクセスしたファイルなどが見つからず失敗した

この 2 つ以外にも、304 や 403 などが記録されている場合があります。

4.11.4 エラーログの確認

エラーログに記録されている内容を確認します。

```
$ sudo cat /var/log/httpd/error_log
[Tue Dec 05 10:10:15.601798 2023] [core:notice] [pid 38902:tid 38902] SELinux
    policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Tue Dec 05 10:10:15.602587 2023] [suexec:notice] [pid 38902:tid 38902]
    AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Tue Dec 05 10:10:15.620801 2023] [lbmethod_heartbeat:notice] [pid 38902:tid
    38902] AH02282: No slotmem from mod_heartmonitor
[Tue Dec 05 10:10:15.627884 2023] [mpm_event:notice] [pid 38902:tid 38902]
    AH00489: Apache/2.4.57 (AlmaLinux) OpenSSL/3.0.7 mod_fcgid/2.3.9 configured
    -- resuming normal operations
[Tue Dec 05 10:10:15.627895 2023] [core:notice] [pid 38902:tid 38902] AH00094:
    Command line: '/usr/sbin/httpd -D FOREGROUND'
[Tue Dec 05 10:11:02.247923 2023] [autoindex:error] [pid 38905:tid 39025]
    [client 192.168.56.1:49370] AH01276: Cannot serve directory /var/www/html/:
    No matching DirectoryIndex (index.html) found, and server-generated
    directory index forbidden by Options directive
[Tue Dec 05 10:12:37.353480 2023] [autoindex:error] [pid 38906:tid 39093]
    [client ::1:33426] AH01276: Cannot serve directory /var/www/html/: No
    matching DirectoryIndex (index.html) found, and server-generated directory
    index forbidden by Options directive
[Tue Dec 05 10:12:45.191945 2023] [autoindex:error] [pid 38906:tid 39095]
    [client 192.168.56.1:49378] AH01276: Cannot serve directory /var/www/html/:
    No matching DirectoryIndex (index.html) found, and server-generated
    directory index forbidden by Options directive
```

エラーログには、Web サーバーの起動や停止の際に出力されたログや、アクセスログにも記録されているエラーの詳細などが記録されています。

エラーログの中に以下のようなエラーがあります。

```
AH01276: Cannot serve directory /var/www/html/: No matching DirectoryIndex
    (index.html) found, and server-generated directory index forbidden by
    Options directive
```

これは、/var/www/html ディレクトリに index.html が存在しない、というエラーです。実はテストページとして表示されていたのは、エラー 404 が発生した時に Web サーバーが返す HTML の内容がテストページのように見えていただけで、実際には必要なファイルが見つからないエラーが発生していたわけです。このエラーを解決してみましょう。

4.11.5 index.html を配置する

Web サーバーは、Web ブラウザからのリクエストに応じて表示する HTML ファイルなどを/var/www/html ディレクトリ以下に配置するように設定されています。現時点ではこのディレクトリには何も配置されていないため、404 エラーが発生していました。このエラーを解消するため、index.html を作成してみます。index.html は、Web ブラウザがファイル名を指定しなかった場合にデフォルトで参照される HTML ファイルの名前です。

以下のコマンドを実行して、/var/www/html ディレクトリに index.html ファイルを作成します。

```
$ sudo sh -c "echo 'Hello,World' > /var/www/html/index.html"
```

書き込みには管理者権限が必要ですが、リダイレクトによってファイルを作成するにはコマンド全体を sudo コマンドで呼び出した sh -c コマンドで実行する必要があります。また、コマンド全体を「"」(ダブルクォーテーション) で囲い、書き込み文字列を「'」(シングルクォーテーション) で囲っています。この違いにも注意してコマンドを入力してください。

4.11.6 Web ブラウザからのアクセスとログの確認

Web ブラウザから再度 Web サーバーにアクセスしてみます。今度はテストページではなく、「Hello,World」の文字列が表示されたのではないでしょうか。

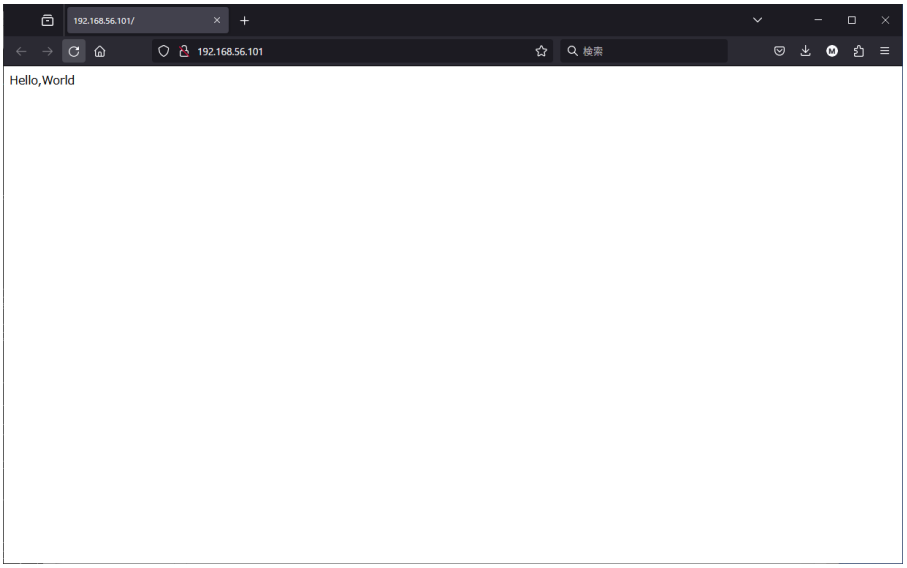

図 51: 「Hello,World」に表示が変わった画面

また、アクセスログにどのように記録されているか、エラーログにエラーが記録されていないことも確認してみてください。

5 DNS サーバーの構築

第 5 章では、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。

5.1 用語集

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。

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。

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。

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 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。クライアントのネットワーク設定で 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。取得したドメイン名を利用するには、DNS コンテンツサーバーを用意して設定する必要があります。

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。

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。

5.2 DNS の仕組み

インターネットでのコンピュータ同士の通信は、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。

5.2.1 HOSTS ファイルと DNS

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。

5.2.2 DNS コンテンツサーバーによるドメイン名の管理

ドメイン名を割り当てられた組織毎に DNS コンテンツサーバーを用意します。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。

あるドメインに所属しているホストにアクセスしたいユーザーは、そのドメインの 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。

DNS の仕組みでは、ドメイン（ゾーン）の管理権限がそれぞれの DNS 管理者に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。

5.2.3 DNS キャッシュサーバーによる名前解決

ユーザーの端末が名前解決を必要とする度に、アクセス先の DNS コンテンツサーバーを探して問い合わせをするのは非常に煩雑になります。そこで端末は DNS キャッシュサーバーに名前解決を依頼し、DNS キャッシュサーバーが後述する再帰問い合わせという方法で DNS コンテンツサーバーに対して問い合わせを行います。名前解決が完了すると、その結果だけを端末に返します。

このような仕組みになっているので、PC やスマートフォンでは IP アドレス設定の際に検索に使用する DNS サーバーの設定が必要になります。この設定が、名前解決を依頼する DNS キャッシュサーバーの IP アドレスとなります。

また、名前解決の結果は一定期間キャッシュされています。同じホストの名前解決が依頼されるとキャッシュから返答するようにしているので、毎回調べる必要がありません。

5.3　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。

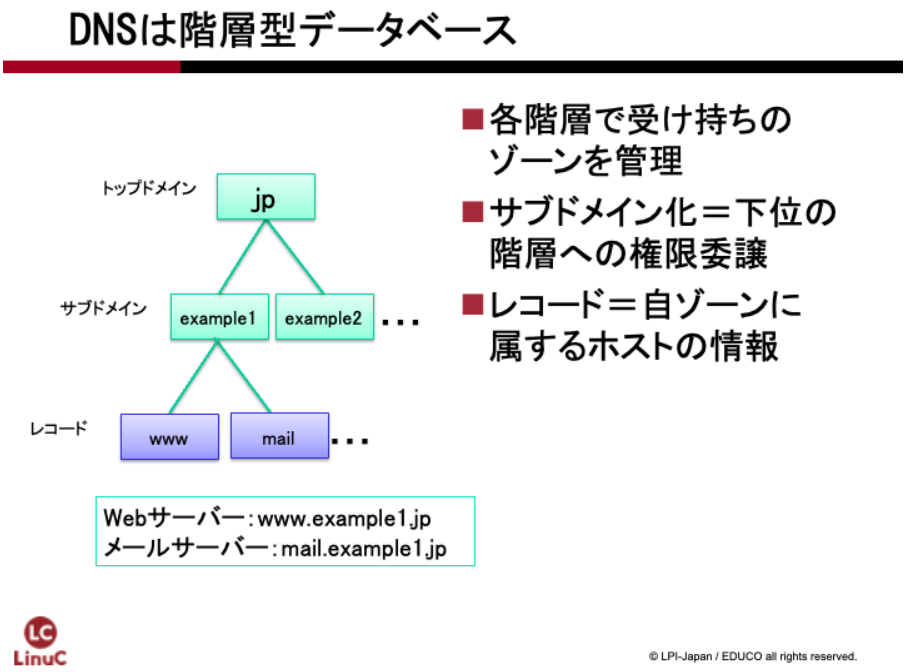

図 52: DNS は階層型データベース

5.3.1　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、ドメイン名として記述する際には「.」（ドット）で表されます。

5.3.2　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 example.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。

5.3.3　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。

5.3.4　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。

5.3.5　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、上位のドメイン（表記上の右側）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

5.3.6 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。

独自の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、登録料などの費用がかかります。

短いドメイン名にこだわらない、費用をかけたくない場合には、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。

5.4　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. 端末は、自分の組織やプロバイダーの用意した 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「www.example1.jp」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 IP 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問い合わせます。「jp」を管理する DNS コンテンツサーバーは、サブドメイン「example1.jp」を管理する DNS コンテンツサーバーの IP 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、「example1.jp」を管理する DNS コンテンツサーバーへ問い合わせます。「example1.jp」を管理する DNS コンテンツサーバーは、「www.example1.jp」の IP 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、クライアントに「www.example1.jp」の IP アドレスを返します。

このように、DNS キャッシュサーバーがルートサーバーから順番に問い合わせを行い、最終的に目的のドメインを管理する DNS コンテンツサーバーまで問い合わせをしていくことを「反復問い合わせ」と呼びます。

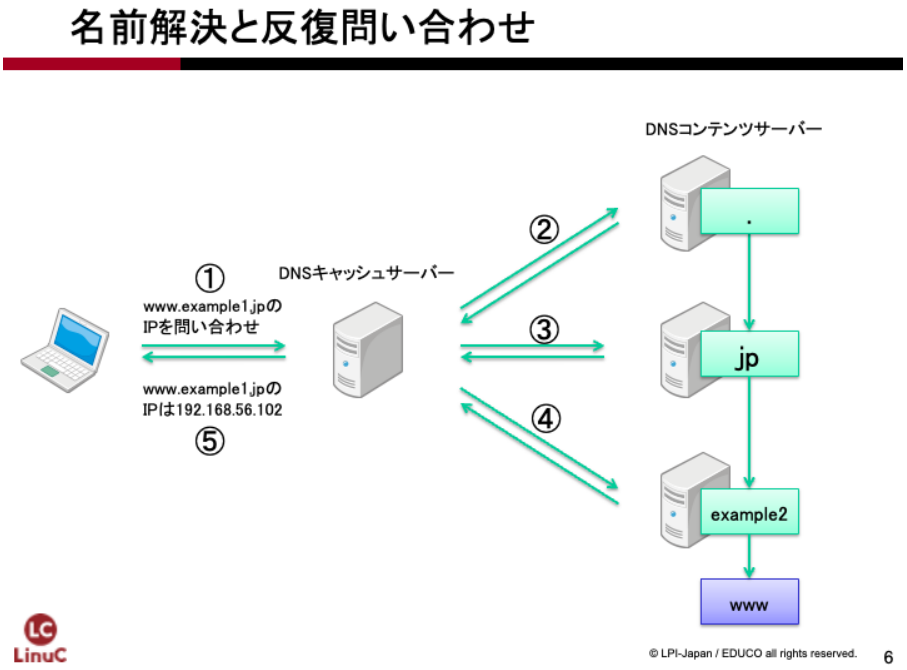

図 53: 名前解決と反復問い合わせ

5.5 演習で構築する DNS の概略

2 つのドメイン名「example1.jp」と「example2.jp」を設定し、相互に名前解決ができるように DNS コンテンツサーバーを構築します。また、jp ドメインからサブドメイン化してゾーンの管理権限を委譲する設定も行います。以下の 3 つの DNS サーバーを構築します。

5.5.1 jp ゾーンのマシン

example1.jp と example2.jp をサブドメイン化し、ゾーン管理権限の委譲を設定します。また DNS キャッシュサーバーとしても動作します。

5.5.2 example1.jp ゾーンのマシン

example1.jp ゾーンを管理する DNS コンテンツサーバーとして設定します。

5.5.3 example2.jp ゾーンのマシン

example2.jp ゾーンを管理する DNS コンテンツサーバーとして設定します。

5.6 演習の手順

3 台のマシンを相互に接続し、相互に DNS を参照できるように設定します。追加でマシン 2 台分の仮想マシンの作成や OS のインストールなどが必要となります。以下の作業を行っていきます。

1. DNS サーバーソフトウェアである BIND を使って、example1.jp ゾーンを管理する DNS コンテンツサーバーを設定します。
2. jp ゾーンおよび example2.jp ゾーンの仮想マシンを作成し、OS をインストールします。それぞれ IP アドレスなどの設定が異なる点に注意してください。
3. example2.jp ゾーンを管理する DNS コンテンツサーバーを設定します。
4. jp ゾーンを管理する DNS コンテンツサーバーを設定します。example1.jp ゾーンおよび example2.jp ゾーンに対する権限委譲を設定します。
5. example1.jp ゾーンと example2.jp ゾーンの DNS を相互に参照できることを確認します。

次章のメールサーバー構築では 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。

5.7 演習で使うアドレスとドメイン

演習で使用する 3 台のサーバーのドメイン名、ホスト名、IP アドレスは以下の通りです。

ドメイン名	ホスト名	IP アドレス
jp.	host0.jp.	192.168.56.100
example1.jp.	host1.example1.jp.	192.168.56.101
example2.jp.	host2.example2.jp.	192.168.56.102

5.8 アドレス解決の流れ

host1.example1.jp のマシン (192.168.56.101) がホスト www.example2.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. host1.example1.jp マシンは、Web ブラウザにアドレスとして www.example2.jp を入力します。
2. Web ブラウザは、Linux のリゾルバに問い合わせます。
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（192.168.56.100）へ問い合わせます。
4. DNS キャッシュサーバーは、jp ゾーンの DNS コンテンツサーバーを参照し、example2.jp ゾーンの DNS コンテンツサーバーの IP アドレス（192.168.56.102）を返します。
5. DNS キャッシュサーバーは、example2.jp ゾーンの DNS コンテンツサーバーに問い合わせします。

6. example2.jp ゾーンの DNS コンテンツサーバーは、www.example2.jp ホストの IP アドレス（192.168.56.102）を返します。
7. DNS キャッシュサーバーは、結果を example1.jp マシンへ返します。
8. Web ブラウザは、www.example2.jp に HTTP でアクセスし、Web ページを受け取って表示します。

実際のインターネットでの DNS の名前解決ではルートゾーンから順番に再帰問い合わせを行いますが、演習環境では DNS キャッシュサーバー自身が jp ゾーンの DNS コンテンツサーバーでもあるため、すぐに example2.jp の DNS コンテンツサーバーに関する情報を返す点が異なります。

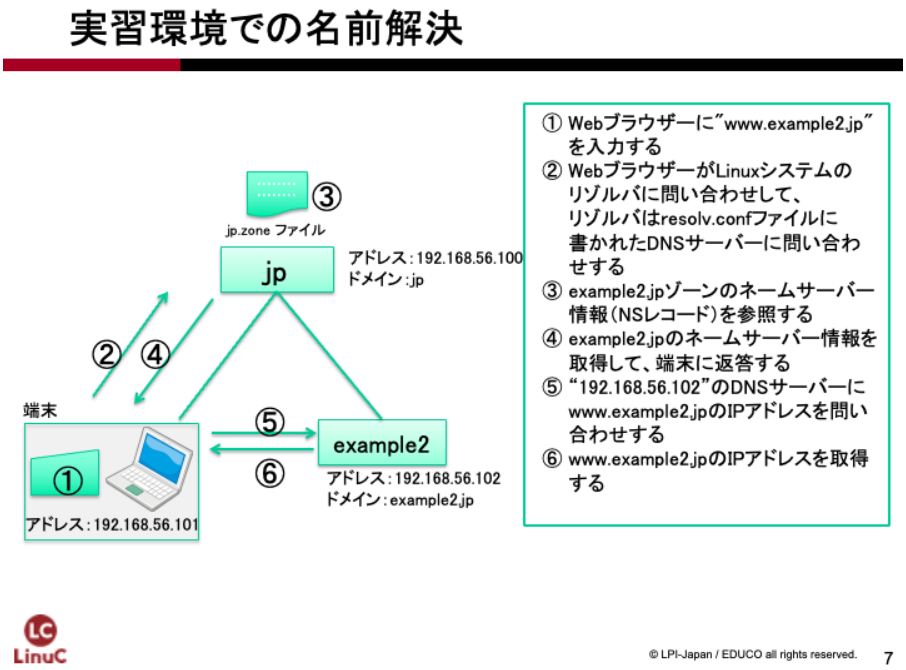

図 54: 実習環境での名前解決

5.9 DNS コンテンツサーバーの設定

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、各ゾーンの設定を行います。

5.9.1 chroot 機能を利用した BIND のセキュリティ

chroot 機能はセキュリティを高めるために、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 ディレクトリになるので、パスワードその他のセキュリティに関する情報にアクセスできません。

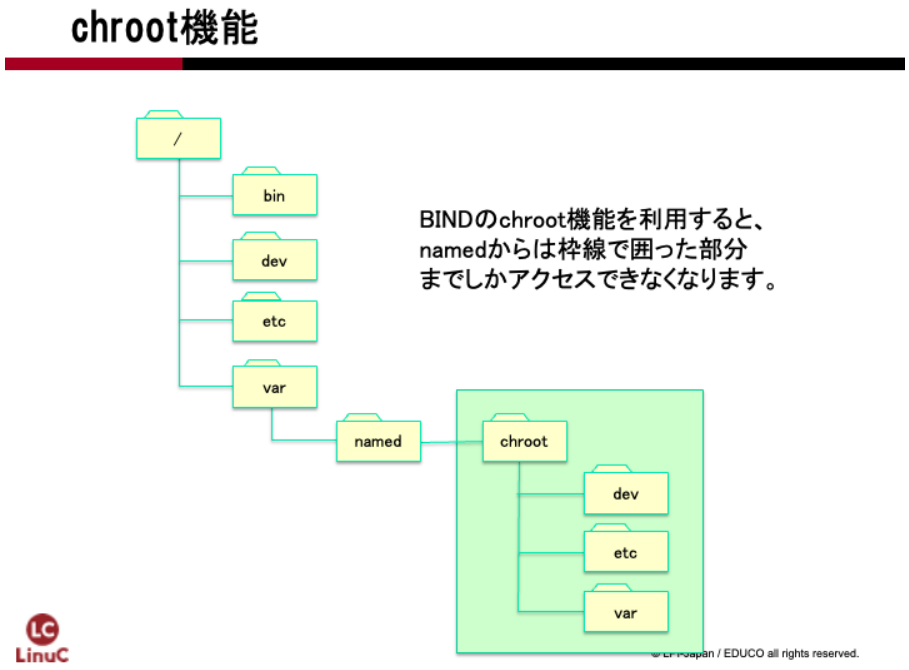

図 55: chroot 機能

DNS という重要な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。

AlmaLinux 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。追加で bind-chroot パッケージのインストールが必要だったり、systemctl コマンドで扱うユニット名（named-chroot）が異なる点に注意してください。

5.9.2 ゾーンを設定する流れ

ゾーンを設定するために必要な作業は以下の手順となります。

1. BIND のインストールを行います。
2. named.conf ファイルにゾーンを追加します。
3. ゾーンファイルを作成し、レコードなどを記述します。
4. 設定ファイルに書式間違いが無いか確認します。
5. BIND の起動を行います。
6. BIND の自動起動の設定を行います。
7. ファイアウォールの設定を変更します。
8. NAT ネットワークを無効化します。
9. /etc/resolv.conf の参照する DNS サーバーの設定を確認します。
10. 名前解決の確認を行います。

BIND の基本的な設定ファイルとして/etc/named.conf ファイルがあります。/etc/named.conf に BIND の基本的な設定とゾーンの定義を追加します。さらに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。

5.10 BIND のインストール

BIND のインストールには、bind パッケージと bind-chroot パッケージが必要です。必要なパッケージがインストールされていないときには、dnf コマンドでインストールします。

```
[admin@host1 ~]$ sudo dnf install bind bind-chroot
```

5.11 /etc/named.conf の基本設定

/etc/named.conf ファイルに BIND を DNS コンテンツサーバーとして動作させる基本設定を行います。

```
[admin@host1 ~]$ sudo vi /etc/named.conf
```

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.56.101; }; ←
        サーバー自身のIPアドレスを追加
    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; }; ← localhostをanyに変更

    recursion no; ← yesをnoに変更

    dnssec-validation yes;

    managed-keys-directory "/var/named/dynamic";
    geoip-directory "/usr/share/GeoIP";

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

    include "/etc/crypto-policies/back-ends/bind.config";
};

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

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

zone "example1.jp" IN { ← example1.jpゾーンを指定
    type master;
    file "example1.jp.zone"; ← example1.jp用のゾーン定義ファイル名を指定
    allow-update { none; };
};
```

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

設定の内容は以下の通りです。

5.11.1 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サーバー自身の IP アドレスである「192.168.56.101;」を listen-on に追加します。後ほど設定する example2.jp サーバーや jp サーバーの場合にはそれぞれのサーバーの IP アドレスを記述します。

5.11.2 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。

5.11.3 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。ただし、jp サーバーの場合には DNS キャッシュサーバーとしても動作させるので yes のままにしておく必要があります。

5.11.4 正引きゾーンの追加

/etc/named.conf にゾーン定義を追加します。上記例では以下のように設定しています。

```
zone "example1.jp" IN {
    type master;
    file "example1.jp.zone";
    allow-update { none; };
};
```

5.12 ゾーンファイルの準備

ゾーンファイルのテンプレートとなる/var/named/named.empty ファイルをコピーします。新たに作成するファイルのファイル名はゾーン定義の file 句で指定したファイル名を指定します。example1.jp ゾーンであれば example1.jp.zone となります。また、コピー元と同じ所有権（root:named）、パーミッション（640）にするため、cp コマンドに-p オプションを付けて実行します。

```
[admin@host1 ~]$ sudo cp -p /var/named/named.empty /var/named/example1.jp.zone
[admin@host1 ~]$ sudo ls -l /var/named/example1.jp.zone
-rw-r-----. 1 root named 152 11月  8 01:28 /var/named/example1.jp.zone
```

5.13 ゾーンファイルの修正

コピーした/var/named/example1.jp.zone ファイルを修正します。

```
[admin@host1 ~]$ sudo vi /var/named/example1.jp.zone
```

```
$TTL 3H
$ORIGIN example1.jp.
@       IN SOA  host1 root (
                                        2023100901       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      host1.example1.jp.
        MX 10   mail.example1.jp.

host1   A       192.168.56.101
www     A       192.168.56.101
mail    A       192.168.56.101
```

5.13.1 $TTL

このゾーン定義ファイル内の記述の TTL（Time to Live・生存時間・有効期間）が 3 時間であることをデフォルト指定しています。

5.13.2 $ORIGIN

このゾーン定義が対象としているゾーン名を指定します。ゾーン定義内のホスト名はすべて FQDN で最後が「.」で終わる必要がありますが、省略された場合には$ORIGIN で指定されたゾーン名で補完されます。たとえば、「www」という記述は「www.example1.jp.」と補完されて扱われます。

5.13.3 SOA レコード

@から始まるゾーンファイルの最初のレコードは SOA レコードです。このゾーンの管理ポリシーについて設定します。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 example1.jp.）に置き換えられます。host1 はこの DNS コンテンツサーバーのホスト名、root は管理者ユーザーです。どちらもゾーン名が補完されて、「host1.example1.jp.」「root.example1.jp.」となりますが、ユーザー名は最初の「.」を「@」にしてメールアドレスとして読み替えます。これらは単なる文字列なので、BIND の動作には影響を与えません。シリアルナンバー (serial) は、ゾーン定義を変更する毎に必ず変更する必要があります。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。日が異なる場合には日付を、同じ日に変更が複数回あった場合には最後の 2 桁を変更するのを忘れないようにしてください。

5.13.4 NS レコード

NS レコードは、このゾーンの DNS コンテンツサーバーである自分自身を定義します。

5.13.5 MX レコード

MX レコードは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。

5.13.6 A レコード

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。マシンのホスト名である host1 や、その他のサービスで使う名前である www や mail 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 example1.jp.) が自動的に追加されます。

5.14 設定ファイルとゾーンファイルの書式確認

設定ファイルとゾーンファイルの作成が終わったら、書式の確認を行います。設定が多岐に渡るため、間違いがないかを確認するためのコマンドを使って確認します。

5.14.1 設定ファイルの書式確認と注意点

/etc/named.conf ファイルの編集時、括弧やセミコロンの不足などは良くある設定ミスです。named-checkconf コマンドで/etc/named.conf に間違いがないか確認しましょう。

```
[admin@host1 ~]$ sudo 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。

```
[admin@host1 ~]$ sudo named-checkconf
/etc/named.conf:11: missing ';' before '}'
```

これは、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記述するのを忘れたというエラーです。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。

5.14.2 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、FQDN で記述すべきところを最後の. が抜けているなどがあります。named-checkzone コマンドを使ってゾーンファイルに間違いがないか確認しましょう。引数は$ORIGIN に指定したゾーン名と、確認を行うゾーンファイル名です。

```
[admin@host1 ~]$ sudo named-checkzone example1.jp. /var/named/example1.jp.zone
zone example1.jp/IN: loaded serial 2023100901
OK
```

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。

```
[admin@host1 ~]$ sudo named-checkzone example1.jp. /var/named/example1.jp.zone
zone example1.jp/IN: NS 'host1.example1.jp.example1.jp' has no address records
    (A or AAAA)
zone example1.jp/IN: not loaded due to errors.
```

この例では、NS レコードの右側に書いたホスト名が FQDN になっていないため、「host1.example1.jp.example1.jp」となってしまい、対応する A レコードが見つからないというエラーが発生しています。

5.15 BIND の起動と確認

BIND を起動してみましょう。BIND の起動は、systemctl コマンドで named-chroot ユニットを使います。

```
[admin@host1 ~]$ sudo systemctl start named-chroot
```

起動できたら、状態を確認します。

```
[admin@host1 ~]$ sudo systemctl status named-chroot
● named-chroot.service - Berkeley Internet Name Domain (DNS)
     Loaded: loaded (/usr/lib/systemd/system/named-chroot.service; disabled;
         preset: disabled)
     Active: active (running) since Mon 2023-10-09 15:18:44 JST; 6s ago
    Process: 12416 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" ==
        "yes" ]; then /usr/sbin/named>
    Process: 12418 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} -t
        /var/named/chroot $OPTIONS (code=exi>
   Main PID: 12419 (named)
```

```
      Tasks: 6 (limit: 10696)
     Memory: 21.2M
        CPU: 32ms
     CGroup: /system.slice/named-chroot.service
             └─12419 /usr/sbin/named -u named -c /etc/named.conf -t
                 /var/named/chroot

Oct 09 15:18:44 localhost.localdomain named[12419]: network unreachable
    resolving './DNSKEY/IN': 2001:500:9f:>
Oct 09 15:18:44 localhost.localdomain named[12419]: network unreachable
    resolving './NS/IN': 2001:500:9f::42#>
Oct 09 15:18:44 localhost.localdomain named[12419]: network unreachable
    resolving './DNSKEY/IN': 2001:7fe::53>
Oct 09 15:18:44 localhost.localdomain named[12419]: network unreachable
    resolving './NS/IN': 2001:7fe::53#53
Oct 09 15:18:44 localhost.localdomain named[12419]: zone
    1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.>
Oct 09 15:18:44 localhost.localdomain named[12419]: all zones loaded
Oct 09 15:18:44 localhost.localdomain named[12419]: running
Oct 09 15:18:44 localhost.localdomain systemd[1]: Started Berkeley Internet
    Name Domain (DNS).
Oct 09 15:18:44 localhost.localdomain named[12419]: managed-keys-zone: Key
    20326 for zone . is now trusted (a>
Oct 09 15:18:44 localhost.localdomain named[12419]: resolver priming query
    complete
```

Active の欄に「active (running)」と表示されていることを確認します。また、下に表示されるログにも「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。

表示を終了するには Q キーを押します。

5.16 自動起動の設定

システム起動時に BIND が自動的に起動されるように設定しておきましょう。systemctl enable コマンドで設定します。

```
[admin@host1 ~]$ sudo systemctl enable named-chroot
Created symlink
    /etc/systemd/system/multi-user.target.wants/named-chroot.service →
    /usr/lib/systemd/system/named-chroot.service.
```

自動起動になっているかは、次のように確認できます。

```
[admin@host1 ~]$ sudo systemctl is-enabled named-chroot
enabled
```

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。

5.17 ファイアウォールの設定

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。以下の firewall-cmd コマンドを実行します。

```
[admin@host1 ~]$ sudo firewall-cmd --add-service=dns --zone=public --permanent
[admin@host1 ~]$ sudo firewall-cmd --reload
```

5.18 参照する DNS サーバーの設定と NAT ネットワークの停止

DNS サーバーによる名前解決を確認する前に、参照する DNS サーバーの設定について確認、変更します。また、変更にあたっては、実習環境で使用している仮想マシンの NAT ネットワークの停止を行います。

5.18.1 /etc/resolv.conf による参照 DNS サーバーの設定

Linux で名前解決を行うために参照する DNS サーバーは/etc/resolv.conf に記述されています。

```
[admin@host1 ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search flets-east.jp example1.jp
nameserver 192.168.11.1
nameserver 192.168.56.101
```

設定の内容は使用しているネットワーク環境によって異なりますが、参照する DNS サーバーを指定する nameserver の設定値として、インストール時に設定した「192.168.56.101」以外に設定されているのが NAT ネットワークの DHCP で設定された参照する DNS サーバーです。

5.18.2 エディタで/etc/resolv.conf を修正すると？

エディタで/etc/resolv.conf を修正すると、設定は即時有効となります。

ただし、ファイルの先頭行に「# Generated by NetworkManager」と書かれている通り、/etc/resolv.conf は NetworkManager が自動的に生成しています。そのため、システムの再起動などによって再生成されるとエディタでの変更は破棄されます。一時的に動作を変更したいような場合にはよいですが、継続して設定を適用したい場合には大元となるネットワークの設定を変更する必要があります。本章の最後では、ネットワークインターフェースの設定を変更する方法を解説しています。

以下の手順では NetworkManager を操作して設定ファイルを再生成しています。ただし、この方法も継続的な変更ではなく、システムが再起動されると NAT ネットワークのネットワークインターフェースが有効化され、変更は破棄され元に戻ります。

5.18.3 ネットワークインターフェースの名前を確認

まず、ネットワークインターフェースの名前を確認します。

```
[admin@host1 ~]$ ip -4 a
（略）
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85842sec preferred_lft 85842sec
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
```

この例では、IP アドレス「10.0.2.15」から NAT ネットワークのネットワークインターフェースは「enp0s3」と分かります。

5.18.4 NAT ネットワークを停止する

NAT ネットワークのネットワークインターフェース enp0s3 を停止します。NetworkManager をコマンドラインから操作する nmcli コマンドを使用します。

```
[admin@host1 ~]$ sudo nmcli connection down enp0s3
接続 'enp0s3' が正常に非アクティブ化されました (D-Bus アクティブパス:
    /org/freedesktop/NetworkManager/ActiveConnection/2)
```

5.18.5 /etc/resolv.conf の変更の確認

NetworkManager は設定が変更される度に/etc/resolv.conf を再生成します。変更されたことを確認します。

```
[admin@host1 ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search example1.jp
nameserver 192.168.56.101
```

このように、参照する DNS サーバーの設定が自分自身だけになりました。

5.18.6 NAT ネットワークの接続の有効化

再度インターネットなどに接続する必要が出た場合には、nmcli コマンドでネットワークインターフェースを有効化する必要があります。

```
[admin@host1 ~]$ sudo nmcli connection up enp0s3
接続が正常にアクティベートされました (D-Bus アクティブパス:
    /org/freedesktop/NetworkManager/ActiveConnection/6)
```

インターネットに接続できるように戻ったことを確認したら、再度 NAT ネットワークを無効化しておきます。

```
[admin@host1 ~]$ sudo nmcli connection down enp0s3
```

ただし、この無効化は OS が再起動されると NAT ネットワークが有効になり元に戻ることに注意してください。

5.19 名前解決の確認

BIND を起動し、NAT ネットワークを無効化したら、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 dig コマンドが使用できます。

5.19.1 dig コマンドでホストの名前解決を確認

dig コマンドでホスト名から IP アドレスが解決されることを確認します。dig コマンドの引数に名前解決するホスト名を指定して実行します。

```
[admin@host1 ~]$ dig host1.example1.jp

; <<>> DiG 9.16.23-RH <<>> host1.example1.jp @192.168.56.101
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32703
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 8375a6e1adca8c5501000000657d0ee8465100965cb0f7dc (good)
;; QUESTION SECTION:
;host1.example1.jp.     IN  A

;; ANSWER SECTION:
host1.example1.jp.  10800   IN  A   192.168.56.101

;; Query time: 0 msec
;; SERVER: 192.168.56.101#53(192.168.56.101) ← 参照したDNSサーバーのIPアドレス
;; WHEN: Sat Dec 16 11:43:52 JST 2023
;; MSG SIZE  rcvd: 90
```

host1.example1.jp の A レコードが正しく設定されていることが確認できます。

5.19.2 正しく名前解決が行えない場合

正しく動作しない場合、結果の下から 3 行にある「SERVER」の結果が「192.168.56.101」になっているかを確認してください。NAT ネットワークの無効化を忘れていると、参照する DNS サーバーが NAT ネットワーク側に設定された IP アドレスになっています。

5.19.3 その他の A レコードを名前解決する

同様に、www.example1.jp や mail.example1.jp も確認してみます。

```
[admin@host1 ~]$ dig www.example1.jp
 （略）
;; QUESTION SECTION:
;www.example1.jp.        IN   A

;; ANSWER SECTION:
www.example1.jp.    10800    IN   A    192.168.56.101
 （略）
```

```
[admin@host1 ~]$ dig mail.example1.jp
 （略）
;; QUESTION SECTION:
;mail.example1.jp.       IN   A

;; ANSWER SECTION:
mail.example1.jp.   10800    IN   A    192.168.56.101
 （略）
```

5.19.4 dig コマンドで NS レコードを確認

ドメイン名の後に 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) と、NS レコードで返されたホストの A レコードが表示されます。

```
[admin@host1 ~]$ dig example1.jp ns
 （略）
;; QUESTION SECTION:
;example1.jp.            IN   NS

;; ANSWER SECTION:
example1.jp.        10800    IN   NS   host1.example1.jp.

;; ADDITIONAL SECTION:
host1.example1.jp.  10800    IN   A    192.168.56.101
 （略）
```

5.19.5 dig コマンドで MX レコードを確認

ドメイン名の後に 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) と、MX レコードで返されたホストの A レコードが表示されます。

```
[admin@host1 ~]$ dig example1.jp mx
 （略）
;; QUESTION SECTION:
;example1.jp.            IN   MX

;; ANSWER SECTION:
example1.jp.        10800    IN   MX   10 mail.example1.jp.

;; ADDITIONAL SECTION:
```

```
mail.example1.jp.    10800    IN   A    192.168.56.101
（略）
```

5.19.6 参照する DNS サーバーを指定した dig コマンドの実行

dig コマンドの引数に「[@IP アドレス]」を指定することで、一時的に参照する DNS サーバーを変更することができます。DNS サーバーの動作が正しくないような場合の原因究明に利用できます。

```
[admin@host1 ~]$ dig www.example1.jp @192.168.56.101
```

5.20 example2.jp サーバーと jp サーバーの追加

example1.jp ドメインの設定ができたので、相互に名前解決ができるように仮想マシンを 2 台追加し、それぞれ example2.jp ドメイン、jp ドメインを管理する DNS サーバーとして設定します。

追加の手順は以下の情報を参考に、第 2 章および第 3 章の手順を繰り返し行ってください。

5.20.1 仮想マシン作成の留意事項

VirtualBox で仮想マシンを追加で 2 台作成します。

既に作成している仮想マシンと同じように作成しますが、区別が付くように仮想マシンの名前をホスト名に合わせてください。

名前
host2.example2.jp
host0.jp

5.20.2 ネットワークアダプターの追加を忘れずに

仮想マシンはホストオンリーネットワークで相互に通信を行いますので、ネットワークアダプターの追加を忘れないようにしてください。

5.20.3 OS のインストール

作成した仮想マシンに OS をインストールします。

ホスト名と IP アドレスをそれぞれのサーバーに合わせて設定します。

ホスト名	IP アドレス	DNS
host2.example2.jp	192.168.56.102	192.168.56.102
host0.jp	192.168.56.100	192.168.56.100

5.20.4 相互通信の確認

OS が起動したら、仮想マシン間で相互に通信ができることを ping コマンドを使って確認してください。

以下のように、各マシンから既に動作している host1.example1.jp に向けて ping コマンドを実行してみます。

```
$ ping 192.168.56.101
```

5.21 example2.jp ゾーンの設定

まず、example2.jp ゾーンの設定を行います。手順は example1.jp ゾーンを設定した以下の手順と同じです。

1. BIND のインストールを行います。
2. named.conf ファイルにゾーンを追加します。
3. ゾーンファイルを作成し、レコードなどを記述します。
4. 設定ファイルに書式間違いが無いか確認します。
5. BIND の起動を行います。
6. BIND の自動起動の設定を行います。
7. ファイアウォールの設定を変更します。
8. NAT ネットワークを無効化します。
9. /etc/resolv.conf の参照する DNS サーバーの設定を確認します。
10. 名前解決の確認を行います。

以下、example1.jp ゾーンの設定と異なるポイントです。

5.21.1 named.conf の設定

named.conf を設定します。listen-on に設定する IP アドレスが「192.168.56.102」になる点と、定義するゾーン名が「example2.jp」になる点が異なります。

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.56.102; }; ←
        ホストのIPアドレスを追加
    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; }; ← localhostをanyに変更

    recursion no; ← yesをnoに変更

    dnssec-validation yes;

    managed-keys-directory "/var/named/dynamic";
    geoip-directory "/usr/share/GeoIP";

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

    include "/etc/crypto-policies/back-ends/bind.config";
};

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

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

zone "example2.jp" IN { ← example2.jpゾーンを指定
    type master;
    file "example2.jp.zone"; ← example2.jp用のゾーン定義ファイル名を指定
```

```
    allow-update { none; };
};

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

5.21.2 ゾーン定義ファイルの作成

ゾーン定義ファイルを作成します。named.empty を example2.jp.zone としてコピーします。-p オプションを忘れないようにしてください。

```
[admin@host2 ~]$ sudo cp -p /var/named/named.empty /var/named/example2.jp.zone
[admin@host2 ~]$ sudo ls -l /var/named/example2.jp.zone
-rw-r-----. 1 root named 152 11月  8 01:28 /var/named/example2.jp.zone
```

5.21.3 ゾーンファイルの修正

コピーした/var/named/example2.jp.zone ファイルを修正します。ゾーン名が「example2.jp」、ホスト名が「host2」、IP アドレスが「192.168.56.102」になっている点に注意が必要です。

```
[admin@host2 ~]$ sudo vi /var/named/example2.jp.zone
```

```
$TTL 3H
$ORIGIN example2.jp.
@       IN SOA  host2 root (
                                        2023100901      ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      host2.example2.jp.
        MX 10   mail.example2.jp.

host2   A       192.168.56.102
www     A       192.168.56.102
mail    A       192.168.56.102
```

5.21.4 BIND の起動

example2.jp ゾーンの設定が完了したら、BIND を起動します。

```
[admin@host2 ~]$ sudo systemctl start named-chroot
```

5.21.5 自動起動とファイアウォールの設定

自動起動の設定や、ファイアウォールの設定を行います。

```
[admin@host2 ~]$ sudo systemctl enable named-chroot
[admin@host2 ~]$ sudo firewall-cmd --add-service=dns --zone=public --permanent
[admin@host2 ~]$ sudo firewall-cmd --reload
```

5.21.6 NAT ネットワークを停止する

NAT ネットワークのネットワークインターフェース enp0s3 を停止します。

```
[admin@host2 ~]$ sudo nmcli connection down enp0s3
```

/etc/resolv.conf が変更されていることを確認します。

```
[admin@host2 ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search example2.jp
nameserver 192.168.56.102
```

5.21.7 名前解決の確認

名前解決を確認します。確認方法は example1.jp ゾーンを設定した際に行った dig コマンドと同様です。

```
[admin@host2 ~]$ dig host2.example2.jp
```

```
[admin@host2 ~]$ dig example2.jp ns
```

```
[admin@host2 ~]$ dig example2.jp mx
```

5.22 上位 jp ゾーンの DNS コンテンツサーバーの設定

example1.jp ゾーン、example2.jp ゾーンの設定が完了したら、上位ゾーンにあたる jp ゾーンの設定を行います。手順は example1.jp ゾーンや example2.jp ゾーンを設定した以下の手順と同じです。

1. BIND のインストールを行います。
2. named.conf ファイルにゾーンを追加します。
3. ゾーンファイルを作成し、レコードなどを記述します。
4. 設定ファイルに書式間違いが無いか確認します。
5. BIND の起動を行います。
6. BIND の自動起動の設定を行います。
7. ファイアウォールの設定を変更します。
8. NAT ネットワークを無効化します。
9. /etc/resolv.conf の参照する DNS サーバーの設定を確認します。
10. 名前解決の確認を行います。

以下、example1.jp ゾーンの設定と異なるポイントです。

5.22.1 named.conf の設定

named.conf を設定します。listen-on に設定する IP アドレスが 192.168.56.100 になる点と、定義するゾーン名が.jp になる点が異なります。また、この DNS サーバーは DNS キャッシュサーバーとしても動作させるので、recursion の設定を「yes;」のままにしておきます。

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.56.100; }; ←
        ホストのIPアドレスを追加
    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; }; ← localhostをanyに変更

    recursion yes; ← yesのままにすること

    dnssec-validation yes;

    managed-keys-directory "/var/named/dynamic";
    geoip-directory "/usr/share/GeoIP";
```

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

    include "/etc/crypto-policies/back-ends/bind.config";
};

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

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

zone "jp" IN { ← jpゾーンを指定
    type master;
    file "jp.zone"; ← jp用のゾーン定義ファイル名を指定
    allow-update { none; };
};

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

5.22.2 ゾーン定義ファイルの作成

ゾーン定義ファイルを作成します。named.empty を jp.zone としてコピーします。-p オプションを忘れないようにしてください。

```
[admin@host0 ~]$ sudo cp -p /var/named/named.empty /var/named/jp.zone
[admin@host0 ~]$ sudo ls -l /var/named/jp.zone
-rw-r-----. 1 root named 152 11月  8 01:28 /var/named/jp.zone
```

5.22.3 ゾーンファイルの修正

コピーした/var/named/jp.zone ファイルを修正します。ゾーン名が「jp」、ホスト名が「host0」、IP アドレスが「192.168.56.100」になっている点に注意が必要です。メールや Web などには使用しないので、MX レコードや www、mail などの A レコードは作成しません。

サブドメインとして権限の委譲を行った example1.jp ゾーンと example2.jp ゾーンの NS レコード、それぞれのゾーンを管理する DNS コンテンツサーバーを指定する A レコードをグルーレコードとして記述します。

```
[admin@host0 ~]$ sudo vi /var/named/jp.zone
```

```
$TTL 3H
$ORIGIN jp.
@       IN SOA  host0 root (
                                        2023100901      ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      host0.jp.
example1.jp. NS      host1.example1.jp.
example2.jp. NS      host2.example2.jp.
```

```
host0   A       192.168.56.100
host1.example1.jp.      A       192.168.56.101
host2.example2.jp.      A       192.168.56.102
```

5.22.4 BIND の起動

jp ゾーンの設定が完了したら、BIND を起動します。

```
[admin@host0 ~]$ sudo systemctl start named-chroot
```

5.22.5 自動起動とファイアウォールの設定

自動起動の設定や、ファイアウォールの設定を行います。

```
[admin@host0 ~]$ sudo systemctl enable named-chroot
[admin@host0 ~]$ sudo firewall-cmd --add-service=dns --zone=public --permanent
[admin@host0 ~]$ sudo firewall-cmd --reload
```

5.22.6 NAT ネットワークを停止する

NAT ネットワークのネットワークインターフェース enp0s3 を停止します。

```
[admin@host0 ~]$ sudo nmcli connection down enp0s3
```

/etc/resolv.conf が変更されていることを確認します。

```
[admin@host0 ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search jp
nameserver 192.168.56.100
```

5.22.7 名前解決の確認

名前解決を確認します。example1.jp ゾーンと example2.jp ゾーンの DNS コンテンツサーバーを示すグルーレコードが名前解決できるかを確認するため、それぞれの NS レコードを問い合わせます。

```
[admin@host0 ~]$ dig example1.jp ns
```

```
[admin@host0 ~]$ dig example2.jp ns
```

5.23 相互に名前解決できることを確認

3 台の DNS サーバーが設定できたので、example1.jp と example2.jp を相互に名前解決できることを確認します。

5.23.1 参照する DNS サーバーの変更

example1.jp ゾーンと example2.jp ゾーンを相互に参照できるようにするには、各マシンが名前解決のために参照する DNS サーバーを host0.jp（192.168.56.100）に設定しておく必要があります。

以下の手順で、host1 と host2 の設定をそれぞれ変更します。

1. デスクトップを右クリックし、ポップアップメニューから「設定」を選択します。
2. 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。
3. 「Ethernet (enp0s8)」の欄にある歯車のボタンをクリックします。
4. 接続プロファイルの設定画面が表示されます。

5. 「IPv4」のタブをクリックします。
6. DNS サーバーのアドレスを host0.jp マシンの IP アドレス「192.168.56.100」に変更します。
7. 「適用」ボタンを押して元の画面に戻ります。
8. 「Ethernet (enp0s8)」の欄にあるスイッチを一旦「オフ」に変えます。
9. 再度「オン」に変えると DNS 設定が変更されます。

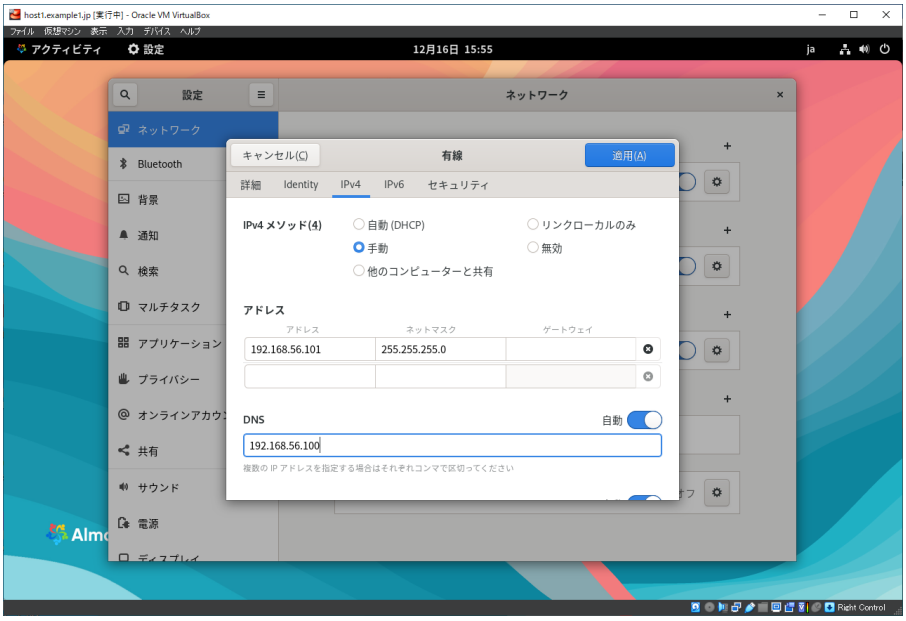

図 56: 参照する DNS サーバーのアドレス変更

設定は/etc/resolv.conf が再生成されることで適用されています。ファイルの内容を確認してみてください。

5.23.2　名前解決の確認

dig コマンドで、他のドメインの NS レコードや MX レコードを問い合わせてみます。

以下は、host1 上で実行して、example2.jp ドメインのレコードが名前解決できるかを確認しています。

```
[admin@host1 ~]$ dig www.example2.jp
 （略）
;; QUESTION SECTION:
;www.example2.jp.        IN   A

;; ANSWER SECTION:
www.example2.jp.     10716    IN   A    192.168.56.102

;; Query time: 0 msec
;; SERVER: 192.168.56.100#53(192.168.56.100)
 （略）
```

```
$ dig example2.jp mx
 （略）
;; QUESTION SECTION:
;example2.jp.            IN   MX

;; ANSWER SECTION:
example2.jp.         10704    IN   MX   10 mail.example2.jp.

;; Query time: 0 msec
;; SERVER: 192.168.56.100#53(192.168.56.100)
 （略）
```

MX レコードの名前解決は、次の章で行うメールサーバーを動かす上での前提条件となります。きちんと名前解決が行えていることを確認した上で次の章に進んでください。

5.24 うまく名前解決が行えない場合には

サーバー相互の名前解決がうまく動作しない場合には、以下の点を確認してください。

- DNS サーバーが起動している
- DNS サーバーが名前解決に応答している
- ping コマンドでお互いに IP 通信が行えている
- ファイアウォールの設定が変更されている
- 参照する DNS サーバーの設定が「192.168.56.100」に変更されている

6 メールサーバーの構築

第 6 章では、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 IMAP サーバーの Dovecot とメールクライアントの Thunderbird を使って、より実践的なメール環境を構築します。

6.1 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、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。

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。

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。

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。動作環境は Windows、macOS、Linux 等と多岐に渡っています。

6.2 メールのやり取りの仕組み

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が、電子メールは以下の手順でやり取りされています。括弧内はそれぞれの手順に関わる動作やプロトコルです。

1. 送信者のメールクライアントから送信用メールサーバーにメールを送信します（SMTP 認証）
2. 送信用メールサーバーは受信用メールサーバーにメールを転送します（SMTP）
3. 受信用メールサーバーは宛先アドレスのメールボックスにメールを配送します（メール配送）
4. 受信者のメールクライアントでメールボックスのあるメールサーバーに接続します（POP3 や IMAP4）

5. 受信者のメールクライアントがメールを受信して、メールを見ることができます

6.3 メールの送信

メールの送信は、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に転送され、最終的に宛先のメールボックスに入れられます。Thunderbird や Outlook のようなメールクライアントから送信したメールでも、Gmail のような Web メールから送信したメールでも、動作原理は同じです。

メールの送信に関わるプログラムやプロトコルについて解説します。

6.3.1 MTA（Mail Transfer Agent）

メールクライアントから送信されたメールは、送信用メールサーバーから宛先の受信用メールサーバーに転送されます。このメールの転送を行うプログラムを MTA と呼びます。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 Sendmail があります。

6.3.2 MDA（Mail Delivery Agent）

受信用メールサーバーが受信したメールを宛先アドレスのメールボックスに配送するプログラムを MDA と呼びます。MTA である Postfix が MDA の機能も受け持ちます。他に MDA として Procmail があります。

6.3.3 MUA（Mail User Agent）

メールの利用者が、メールの送信や受信を行うプログラムを MUA と呼びます。本教科書では Thunderbird を利用します。Web メールも MUA の一種となります。

6.3.4 SMTP（Simple Mail Transfer Protocol）の拡張

メールクライアントからのメール送信や、メールサーバー間のメールの転送は、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問題がありました。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。

6.3.5 SMTP 認証（SMTP Authentication）とリレー

SMTP 認証は ESMTP の機能のうちの 1 つです。通常の SMTP には認証機能が無いため、送信元の IP アドレスを制限するなど適切に設定されていないと迷惑メール送信の踏み台とされてしまう問題があります。SMTP 認証は、メールの送信時に認証を行い、認証が行われた場合のみメールの送信を受け付けます。受け付けたメールを宛先の受信用メールサーバーに転送することをリレーと呼びます。

6.4 メールの受信

送信されたメールが宛先のメールボックスに配送されると、受信者はメールを受信して読むことができます。メールの受信に関係するいくつかの事項について解説します。

6.4.1 メールの配送

受信用メールサーバーがメールを受け取り、宛先アドレスのメールボックスにメールを届けることを配送と呼びます。メールボックスが無い場合、宛先不明として送信用メールサーバーにエラーを返します。メールの配送を行うソフトウェアを MDA（Mail Delivery Agent）と呼びます。

6.4.2 ローカル配送

送信側と受信側が同じメールサーバーを使用している場合、メールは外部のメールサーバーに転送する必要がなく、すぐに宛先アドレスのメールボックスに配送されます。これをローカル配送と呼びます。

6.4.3　POP3（Post Office Protocol version 3）

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。

6.4.4　IMAP4（Internet Message Access Protocol 4）

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。メールを IMAP サーバーに残しておくことができるので、メールクライアントと Web メールを併用したりすることもできますが、その分だけメールサーバーのストレージを消費するので容量管理が必要になります。

6.5　メールサーバーの構築

メールサーバーの構築は、送信用メールサーバーと受信用メールサーバーの 2 台を構築します。それぞれのサーバーを以下のように設定します。

- 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。
- MTA として Postfix をインストールし、送信用メールサーバーおよび受信用メールサーバーとして設定します。
- IMAP サーバーとして Dovecot をインストールし、設定します。
- それぞれのメールサーバーにメールアカウントを作成します。
- メールクライアントとして Thunderbird をインストールし、サーバー自身を送受信用サーバーとして設定します。

通常、メールサーバーとメールクライアントは別々のマシンを用意し、その間を SMTP や POP3、IMAP4 などで接続しますが、実習では同じマシンで行います。

構築は、Postfix の設定と mail コマンドによるメールの送受信、Dovecot と Thunderbird の設定とメールの送受信の 2 段階に分けて行います。

6.5.1　mail コマンドを利用したメールの送受信

Postfix の設定が完了したら、動作確認として mail コマンドを使ってメールのやり取りを行います。メールクライアントの設定を必要としないので、MTA が正しく設定されていることを確認するのに適しています。

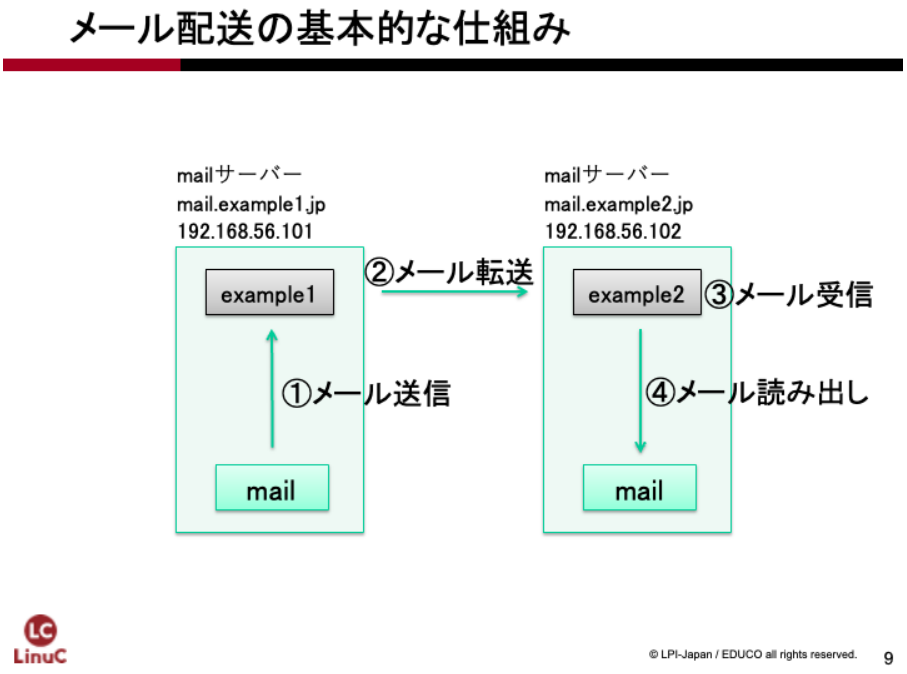

図 57: メール配送の基本的な仕組み

6.5.2　Thunderbird を利用したメールの送受信

Thunderbird をメールクライアントとして設定し、SMTP 認証によるメール送信や、Dovecot による IMAP サーバーからのメール受信を行います。

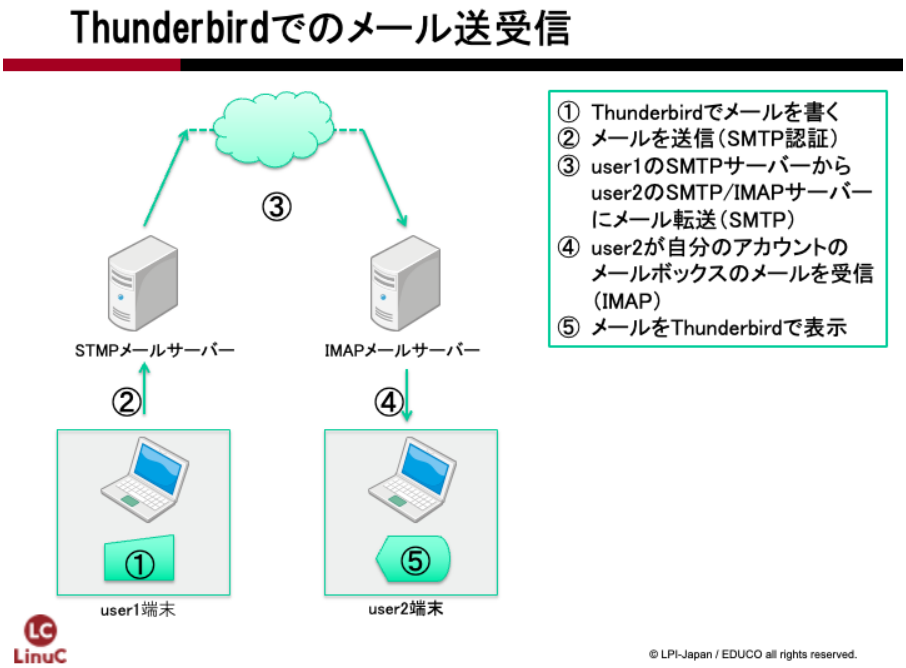

図 58: Thunderbird でのメール送受信

6.6　Postfix のインストール

Postfix を dnf コマンドでインストールをします。また、SMTP 認証で使用する Cyrus-SASL（cyrus-sasl パッケージ）と mail コマンドが含まれている s-nail パッケージも一緒にインストールしておきます。また、IMAP サーバーとして使用する dovecot パッケージ、メールクライアントとして使用する thunderbird パッケージも同時にインストールしておきます。

6.6.1　NAT ネットワークの有効化

DNS サーバーの設定で NAT ネットワークを無効化している場合には、インターネットに接続できるように有効化します。インストールが終わった後に再度無効化します。

```
$ sudo nmcli con up enp0s3
接続が正常にアクティベートされました (D-Bus アクティブパス:
    /org/freedesktop/NetworkManager/ActiveConnection/11)
```

6.6.2　パッケージのインストール

dnf コマンドで必要なパッケージをインストールします。

```
$ sudo dnf install postfix cyrus-sasl s-nail dovecot thunderbird
```

6.6.3　NAT ネットワークの無効化

NAT ネットワークを無効化します。

```
$ sudo nmcli con down enp0s3
接続 'enp0s3' が正常に非アクティブ化されました (D-Bus アクティブパス:
    /org/freedesktop/NetworkManager/ActiveConnection/11)
```

6.6.4 インターネット接続できない環境でのインストール

インターネット接続できない環境での実習の場合、OS インストール時にメールサーバーを追加でインストールしていますが、s-nail パッケージはインストールされていません。同様に Thunderbird（thunderbird パッケージ）もインストールされていません。

本章の最後で ISO イメージに含まれている RPM パッケージからインストールする方法を解説しているので、その手順に従って s-nail と thunderbird をインストールしてください。

6.7 Postfix の設定ファイル main.cf の設定

Postfix の設定ファイルは/etc/postfix/main.cf です。次のパラメータを探して設定します。

```
$ sudo vi /etc/postfix/main.cf
```

パラメータによっては、デフォルト値が設定されているので、その場合には書き換えます。コメントアウトされた形で記述されている場合には、コメントアウトを外して設定を有効にした上で値を記述します。

「smtpd_sasl_auth_enable」と「smtpd_recipient_restrictions」は、main.cf に記述されていないので、ファイルの最後に追加しておきます。

host1 と host2 に、それぞれ以下のように設定します。

host1 の設定

項目	設定値
myhostname	mail.example1.jp
mydomain	example1.jp
inet_interfaces	localhost, 192.168.56.101
mydestination	$mydomain
smtpd_sasl_auth_enable	yes
smtpd_recipient_restrictions	permit_mynetworks, permit_sasl_authenticated, reject_unauth_destination

host2 の設定

項目	設定値
myhostname	mail.example2.jp
mydomain	example2.jp
inet_interfaces	localhost, 192.168.56.102
mydestination	$mydomain
smtpd_sasl_auth_enable	yes
smtpd_recipient_restrictions	permit_mynetworks, permit_sasl_authenticated, reject_unauth_destination

それぞれのパラメータの意味と設定値は以下の通りです。

6.7.1 myhostname

メールサーバーのホスト名を設定します。

host1 の設定

```
myhostname = mail.example1.jp
```

host2 の設定

```
myhostname = mail.example2.jp
```

6.7.2 mydomain

メールサーバーのドメイン名を設定します。

host1 の設定

```
mydomain = example1.jp
```

host2 の設定

```
mydomain = example2.jp
```

6.7.3 inet_interfaces

メールを受け付けるネットワークインターフェースの IP アドレスを設定します。localhost の記述を忘れると、自分自身からのメールを受け付けなくなるので注意が必要です。

host1 の設定

```
inet_interfaces = localhost, 192.168.56.101
```

host2 の設定

```
inet_interfaces = localhost, 192.168.56.102
```

6.7.4 mydestination

メールを受信するドメイン名を設定します。宛先メールアドレス（アカウント名@ドメイン名）がこのドメイン名になっているメールのみ受け取ります。

host1 と host2 共通の設定

```
mydestination = $mydomain
```

6.7.5 smtpd_sasl_auth_enable

SMTP 認証用の SASL 認証連携を有効にします。

host1 と host2 共通の設定

```
smtpd_sasl_auth_enable = yes
```

6.7.6 smtpd_recipient_restrictions

SMTP 認証で SASL 認証されたクライアントからのメール送信を許可するように設定します。

host1 と host2 共通の設定

```
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated,
    reject_unauth_destination
```

これら 2 つの設定は後述する SMTP 認証（SASL 認証連携）に関係する設定ですが、先に設定しておきます。

6.8 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。

```
$ sudo 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。

6.9 Postfix の起動

postfix サービスを起動します。

```
$ sudo systemctl start postfix
```

6.10 自動起動とファイアウォールの設定

自動起動の設定や、ファイアウォールの設定を行います。

```
$ sudo systemctl enable postfix
$ sudo firewall-cmd --add-service=smtp --zone=public --permanent
$ sudo firewall-cmd --reload
```

6.10.1 Postfix の起動順の設定（任意）

上記 systemctl enable コマンドで Postfix の自動起動を設定しても、システムを再起動すると Postfix が起動していないことがあります。これはシステム起動時のサービスの起動順の設定の仕様によるものです。

具体的には、ネットワークサービスの起動完了を待たずに Postfix を起動しようとしているのが原因です。正しい動作をさせるためには systemd が参照している設定ファイルを修正する必要があります。

本教科書の実習では頻繁にシステム再起動を行うことはないので、起動していない場合にはその都度 Postfix を起動すればよいですが、連続稼働させるシステムで Postfix を動作させる場合には、以下の修正を行ってください。

6.10.2 systemctl edit コマンドで修正

systemctl edit コマンドを使うと、対象となるサービスの設定ファイルを編集することができます。ただし、起動するエディタが nano になるので、vim と使い勝手が若干異なる点に注意が必要です。

6.10.3 vim で設定ファイルを修正

vim で設定ファイルを直接開いて修正しても構いません。設定ファイルは/usr/lib/systemd/system/postfix.service です。

```
$ sudo vi /usr/lib/systemd/system/postfix.service
```

```
[Unit]
Description=Postfix Mail Transport Agent
After=syslog.target network-online.target ←
    network.target を network-online.target に変更
Conflicts=sendmail.service exim.service
（略）
```

After=network.target は、ネットワークサービスの起動の後に Postfix を起動しますが、ネットワークが使えるようになったことは確認しません。After=network-online.target にすることで、ネットワークが使えるようになったことを確認した後に Postfix を起動します。

変更が適用されたことを確認するため、システムを再起動し、Postfix が自動的に起動していることを確認してください。

6.11 SMTP 認証（SASL 認証連携）の設定

SMTP 認証は、メール送信時に認証を行う仕組みです。Postfix 自体は認証の機能を持っていませんので、SASL 認証との連携を行います。SASL 認証連携を行うには、Postfix への設定と saslauthd サービスの起動が必要です。

6.11.1 Postfix の設定の確認

Postfix の設定ファイルである main.cf で SASL 認証を有効にします。既に以下の 2 つのパラメータを追加で設定しています。

項目	設定値
smtpd_sasl_auth_enable	yes
smtpd_recipient_restrictions	permit_mynetworks, permit_sasl_authenticated, reject_unauth_destination

SMTP 認証の機能を有効にすると、Postfix は saslauthd に認証を依頼し、認証が成功するとメールを受け取ってリレーを行います。

6.11.2 saslauthd サービスの起動

SMTP 認証用の saslauthd サービスを起動します。

```
$ sudo systemctl start saslauthd
```

saslauthd の自動起動設定も行っておきます。

```
$ sudo systemctl enable saslauthd
```

これで Postfix の設定は完了です。

6.12 アカウントの作成

メールのやり取りを行うためのアカウントを作成します。アカウントは host1 と host2 の双方で行います。

6.12.1 host1 に user1 を作成

host1 で user1 というアカウントを作成します。このアカウントは user1@example1.jp というメールアドレスになります。passwd コマンドでパスワードの設定も行っておきます。

```
[admin@host1 ~]$ sudo useradd user1
[admin@host1 ~]$ sudo passwd user1
user1ユーザーのパスワードを変更。
新しいパスワード: userpass ← 入力文字は非表示
正しくないパスワード: このパスワードは辞書チェックに失敗しました -
    辞書の単語に基づいています
新しいパスワードを再入力してください: userpass ← 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。
```

6.12.2 host2 に user2 を作成

host2 で user2 というアカウントを作成します。このアカウントは user2@example2.jp というメールアドレスになります。passwd コマンドでパスワードの設定も行っておきます。

```
[admin@host2 ~]$ sudo useradd user2
[admin@host2 ~]$ sudo passwd user2
user2ユーザーのパスワードを変更。
新しいパスワード: userpass ← 入力文字は非表示
```

```
正しくないパスワード: このパスワードは辞書チェックに失敗しました -
    辞書の単語に基づいています
新しいパスワードを再入力してください: userpass ← 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。
```

6.13 mail コマンドを使ったメール送受信のテスト

メールの送受信が行えるかテストを行います。メールの送受信は作成したユーザー user1 と user2 で行います。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。メールの送受信には mail コマンドを使用します。

6.13.1 ログの確認用端末の設定（任意）

メールサーバーはバックグラウンドで動作するため、どのように動いているのか確認するためにはログを参照する必要があります。

tail コマンドに-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。

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。

```
$ sudo tail -f /var/log/maillog
```

6.13.2 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。ユーザーを完全に切り替えるために「su - ユーザー名」と「-」（ハイフン）を付けて実行してください。

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。

6.13.3 host1 で user1 に切り替え

host1 は user1 で操作を行います。

```
[admin@host1 ~]$ sudo su - user1
[user1@host1 ~]$
```

6.13.4 host2 で user2 に切り替え

host2 は user2 で操作を行います。

```
[admin@host2 ~]$ sudo su - user2
[user2@host2 ~]$
```

6.13.5 user1@example1.jp から user2@example2.jp へメール送信

mail コマンドを使って、host1 の user1 から user2@example2.jp へメールを送信します。

```
[user1@host1 ~]$ mail user2@example2.jp ←
    mailコマンドの引数に宛先のアドレスを指定
Subject: Test mail from user1 ← Subjectを入力
This is test mail from user1 ← メッセージ本文を入力
^D ← メッセージ本文の入力が終わったらCtrl+dを入力
-------
(Preliminary) Envelope contains:
To: user2@example2.jp
```

```
Subject: Test mail from user1
Send this message [yes/no, empty: recompose]? yes ←
    yesと入力するとメールが送信される
```

6.13.6 user2 のメール着信確認

mail コマンドを使って、host2.example2.jp の user2 にメールが届いているかを確認します。

```
[user2@host2 ~]$ mail
Heirloom Mail version 12.5 7/5/10.  Type ? for help.
"/var/spool/mail/user2": 1 message 1 new
>N  1 user1@mail.example1.jp   Tue Feb 19 13:38  21/751   "Test mail from user1"
& 1 ← 1を入力
Message  1:
From user1@mail.example1.jp  Tue Feb 19 13:38:31 2019
Return-Path: <user1@mail.example1.jp>
X-Original-To: user2@example2.jp
Delivered-To: user2@example2.jp
Date: Tue, 19 Feb 2019 13:38:31 +0900
To: user2@example2.jp
Subject: Test mail from user1
User-Agent: Heirloom mailx 12.5 7/5/10
Content-Type: text/plain; charset=us-ascii
From: user1@mail.example1.jp
Status: R

This is Test Mail from user1

& q ← qを入力
Held 1 message in /var/spool/mail/user2
```

このように、host1.example1.jp から host2.example2.jp にメールが送られていることがわかります。

反対に user2@example2.jp から user1@example1.jp へのメールも送信できることを確認してみましょう。

6.14 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 IMAP4 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 Dovecot 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。

以下の作業は host1 で行いますが、host2 でも設定して双方向でメールのやり取りができるようにしてもよいでしょう。

6.15 Dovecot の設定

IMAP サーバーとして Dovecot の設定を行います。

今回設定するファイルは/etc/dovecot/dovecot.conf と、/etc/dovecot/conf.d ディレクトリ以下に分かれている以下のファイルです。

- /etc/dovecot/dovecot.conf（変更不要）
- /etc/dovecot/conf.d/10-mail.conf
- /etc/dovecot/conf.d/10-auth.conf
- /etc/dovecot/conf.d/10-ssl.conf

6.15.1 /etc/dovecot/dovecot.conf（変更不要）

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。

```
$ sudo vi /etc/dovecot/dovecot.conf
 （略）
# Protocols we want to be serving.
#protocols = imap pop3 lmtp submission  ← IMAP/POP3/LMTP/SMTP
    submission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
```

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

メールボックスの位置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ます。また、メールボックスへのアクセス権限を設定します。

```
$ sudo vi /etc/dovecot/conf.d/10-mail.conf
```

```
 （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 =
mail_location = mbox:~/mail:INBOX=/var/mail/%u  ← 上にある2番目の例を参考に追加

 （略）

# 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_privileged_group = mail ←
    権限が必要な動作はmailグループとして行うように追加

# 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_access_groups = mail ← mailグループにアクセス権限を与えるように追加
 （略）
```

6.15.3 /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ます。

```
$ sudo vi /etc/dovecot/conf.d/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 ← noに変更
 （略）
```

6.15.4 /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/TLS 暗号化の利用を停止しておきます。

```
$ sudo vi /etc/dovecot/conf.d/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 = no    ← requiredをnoに変更
```

6.16 Dovecot の起動

dovecot サービスを起動します。

```
$ sudo systemctl start dovecot
```

6.16.1 自動起動とファイアウォールの設定

自動起動の設定や、ファイアウォールの設定を行います。

```
$ sudo systemctl enable dovecot
$ sudo firewall-cmd --add-service=imap --zone=public --permanent
$ sudo firewall-cmd --reload
```

6.17 Thunderbird の設定

次にメールクライアントとして Thunderbird の設定を行います。

6.17.1 ユーザーを切り替えてログイン

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 user1 でログインします。

その他のユーザーでログインしている場合にはログアウトします。

パスワードは userpass です。正しく設定されていない場合には、再度 admin ユーザーでログインし、passwd コマンドで設定し直してください。このパスワードが Thunderbird の設定にも使用されます。

6.17.2　Thunderbird の起動

Thunderbird を起動します。

画面左上にある「アクティビティ」をクリックし、画面下に表示されるアイコンドックから一番右にある「アプリケーションを表示する」をクリックします。表示されるアプリケーションアイコンから「Thunderbird」をクリックします。

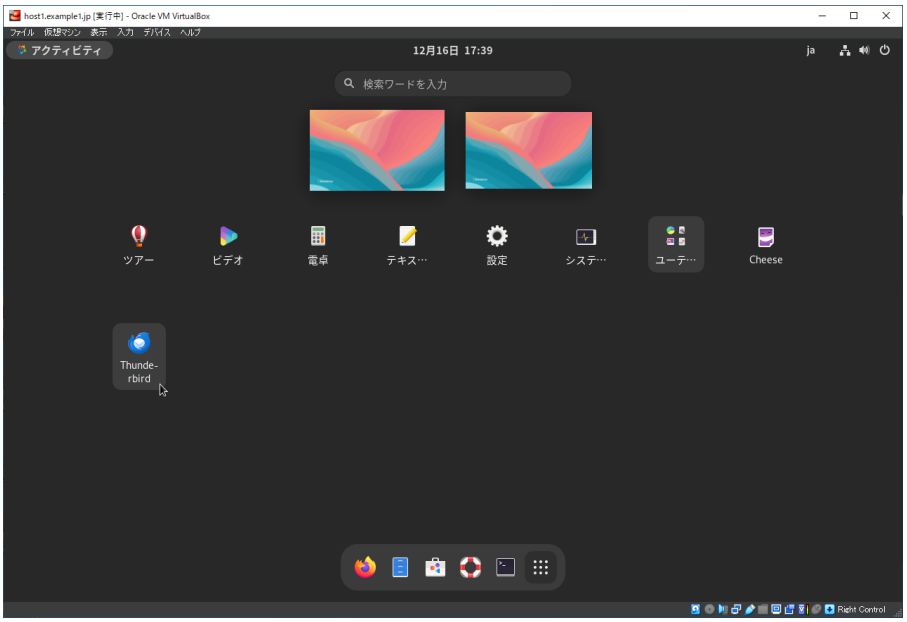

図 59: Thunderbird の起動

Thunderbird が起動すると別途 Web ブラウザが開いて Thunderbird の Web ページが表示されますが、Web ブラウザごと閉じて構いません。

6.17.3　Thunderbird の基本設定

Thunderbird のアプリケーションウインドウを表示し、「既存のメールアドレスのセットアップ」タブが表示されていることを確認します。

各設定項目の値を以下のように入力します。

設定項目	設定値
あなたの名前	user1
メールアドレス	user1@example1.jp
パスワード	userpass
パスワードを記憶する	チェックしておく

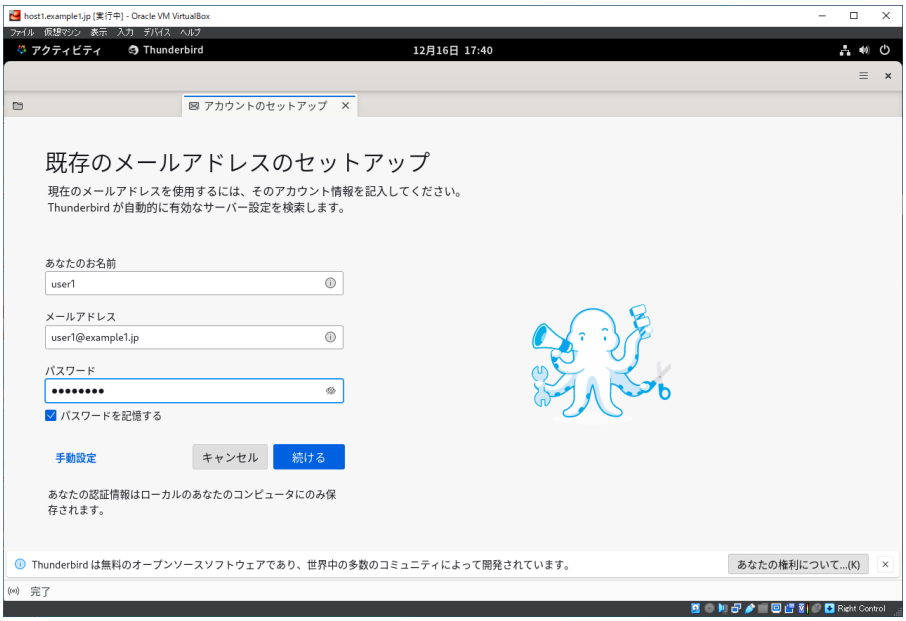

図 60: メールアドレスとパスワードの設定

6.17.4　Thunderbird の送受信メールサーバーの設定

「手動設定」をクリックして、Thunderbird の送受信メールサーバーの設定を行います。

「受信サーバー」と「送信サーバー」が表示されるので、各設定項目を以下のように入力します。

設定項目	設定値
プロトコル	IMAP
ホスト名	mail.example1.jp
ポート番号	143
接続の保護	なし
認証方式	通常のパスワード認証
ユーザ名	user1

設定項目	設定値
ホスト名	mail.example1.jp
ポート番号	25
接続の保護	なし
認証方式	通常のパスワード認証
ユーザ名	user1

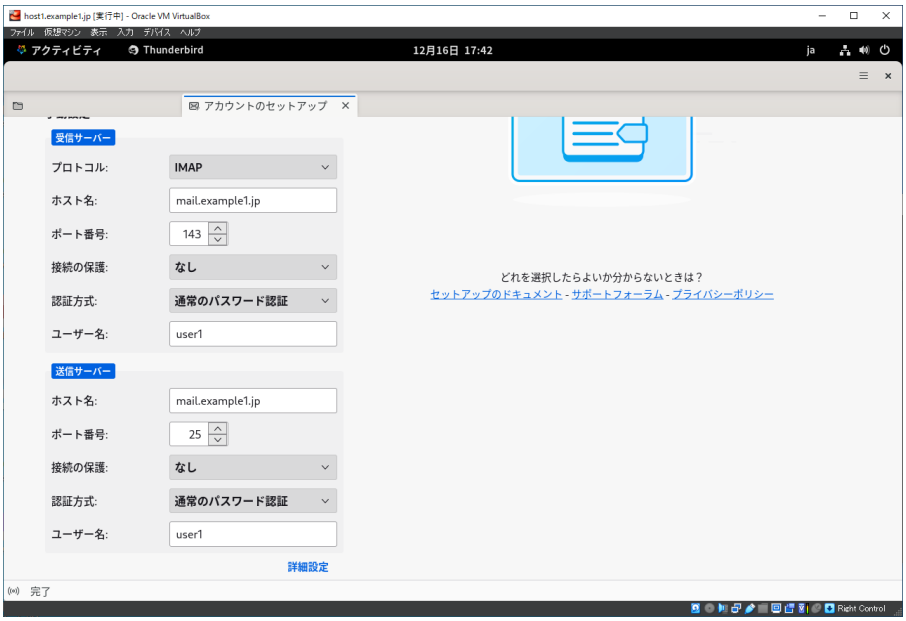

図 61: 受信サーバーと送信サーバーの設定

入力を終えたら「完了」ボタンをクリックします。

「警告！」ダイアログが表示されますが、左下の「接続する上での危険性を理解しました」をチェックし、「確認」ボタンをクリックします。

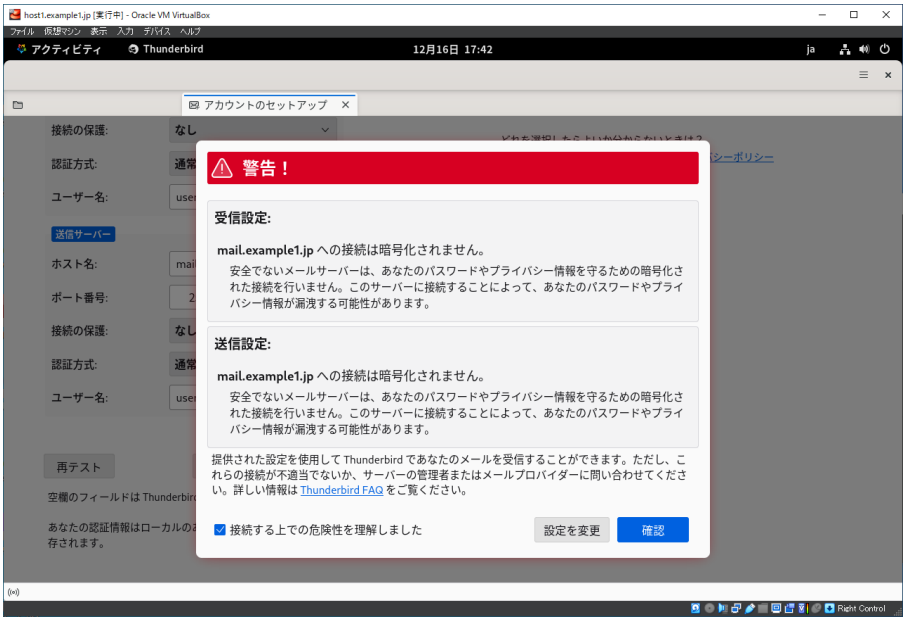

図 62: 警告ダイアログ

最後に「リンクしたサービスへの接続」の「完了」ボタンをクリックします。

6.18　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。

6.18.1　自分宛のメール送信

1. 「作成」ボタンをクリック
2. 宛先に自分のメールアドレス（user1@example1.jp）を指定して、メールを作成、送信してみます。
3. 「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。

6.18.2　別サーバー宛のメール送信

1. 「作成」ボタンをクリック
2. 宛先に host2 のメールアドレス（user2@example2.jp）を指定して、メールを作成、送信してみます。
3. host2 で mail コマンドを使ってメールを受信できたことを確認します。
4. mail コマンドで user1@example1.jp 宛にメールを送信し、host1 で受信できることを確認します。

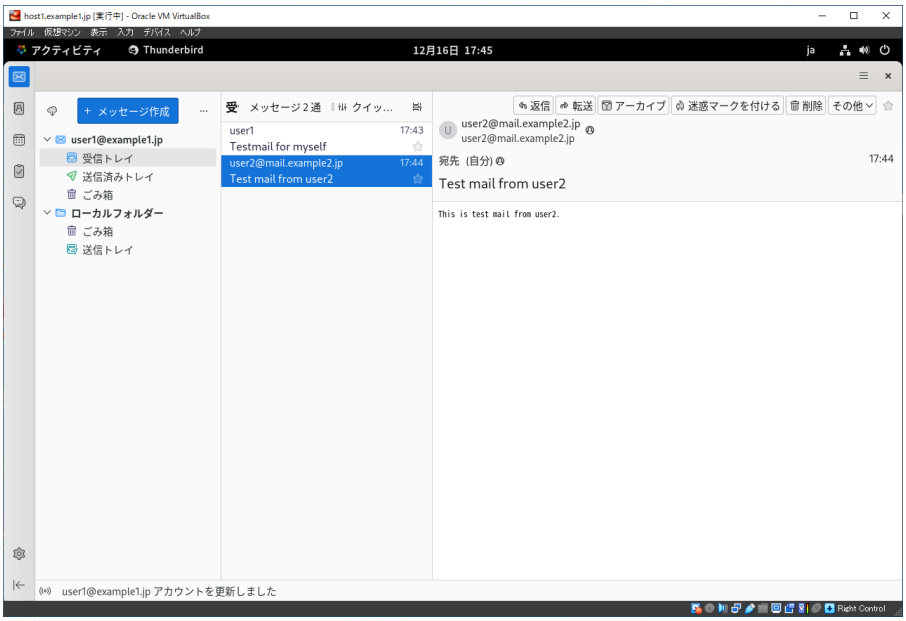

図 63: Thunderbird のメール送受信画面

6.19　うまく動作しない場合には

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場合があります。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 dig コマンドを実行して確認します。また、ログ（/var/log/maillog）を見て、エラーが出ていないかを確認してみてください。

6.20 ISO イメージからのパッケージのインストール

dnf コマンドはインターネット上にあるパッケージリポジトリからダウンロードしてインストールを行います。インターネット接続できない環境でパッケージをインストールするには、ISO イメージに入っている RPM パッケージからインストールを行います。

1. 仮想マシンに admin ユーザーでログインします
2. 仮想マシンの仮想光学ドライブに ISO イメージを読み込ませます
3. ISO イメージが自動マウントされます

6.20.1 ISO イメージのマウントポイントの確認

ISO イメージが自動マウントされたマウントポイント（ディレクトリ）を確認します。

マウントポイントは mount コマンドで確認できます。マウントポイントが多数あるため、grep コマンドで絞り込みます。

```
$ mount | grep dvd
/dev/sr0 on /run/media/admin/AlmaLinux-9-3-x86_64-dvd type iso9660
    (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,
    gid=1000,dmode=500,fmode=400,uhelper=udisks2)
```

ISO イメージは ISO9660 ファイル形式として「/run/media/ユーザー名」以下にマウントされていることが分かります。

ISO イメージの中身を確認してみます。

```
$ ls /run/media/admin/AlmaLinux-9-3-x86_64-dvd/
AppStream  EULA                    TRANS.TBL          images
BaseOS     LICENSE                 boot.catalog       media.repo
EFI        RPM-GPG-KEY-AlmaLinux-9 extra_files.json
```

6.20.2 パッケージのインストール

マウントされた ISO イメージからパッケージをインストールします。

OS インストール時にインストールできなかった s-nail と thunderbird をインストールします。これらは AppStream ディレクトリ内に保存されています。

```
$ cd /run/media/admin/AlmaLinux-9-3-x86_64-dvd/AppStream/Packages/
$ ls -l s-nail-14.9.22-6.el9.x86_64.rpm
-r--r--r--. 5 admin admin 635625  3月 10  2022 s-nail-14.9.22-6.el9.x86_64.rpm
$ ls -l thunderbird-115.4.1-1.el9_2.alma.x86_64.rpm
-r--r--r--. 5 admin admin 111843172 11月  3 00:37
    thunderbird-115.4.1-1.el9_2.alma.x86_64.rpm
```

パッケージの RPM ファイルが確認できたら、dnf コマンドの引数として指定してインストールします。

```
$ pwd
/run/media/admin/AlmaLinux-9-3-x86_64-dvd/AppStream/Packages
$ sudo dnf install ./s-nail-14.9.22-6.el9.x86_64.rpm
$ sudo dnf install ./thunderbird-115.4.1-1.el9_2.alma.x86_64.rpm
```

7 ネットワークとセキュリティの設定

第 7 章では、ネットワークとセキュリティの設定についてあらためて確認します。AlmaLinux 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、これらを OS インストール後に設定する方法について説明します。

7.1 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。

ブロードキャストアドレス

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。このアドレスに対して通信を行うことで、ネットワークに属しているホストすべてに対して通信が行えます。

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。

7.2　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。

7.2.1　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。

```
# 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 fq_codel state UP
    group default qlen 1000
    link/ether 08:00:27:2d:1c:bc brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 65668sec preferred_lft 65668sec
    inet6 fe80::a00:27ff:fe2d:1cbc/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    link/ether 08:00:27:93:ab:ef brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe93:abef/64 scope link noprefixroute
```

```
       valid_lft forever preferred_lft forever
```

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 NAT、enp0s8 がホストオンリーのインターフェースです。この名称は、enoXX、ensXX、ethX、enxXX などの名称になる場合もあります。

7.2.2　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。

デスクトップを右クリックし、ポップアップメニューから「設定」を選択します。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。

表示されるネットワークインターフェースから、変更したいネットワークインターフェースの欄にある歯車のボタンをクリックします。

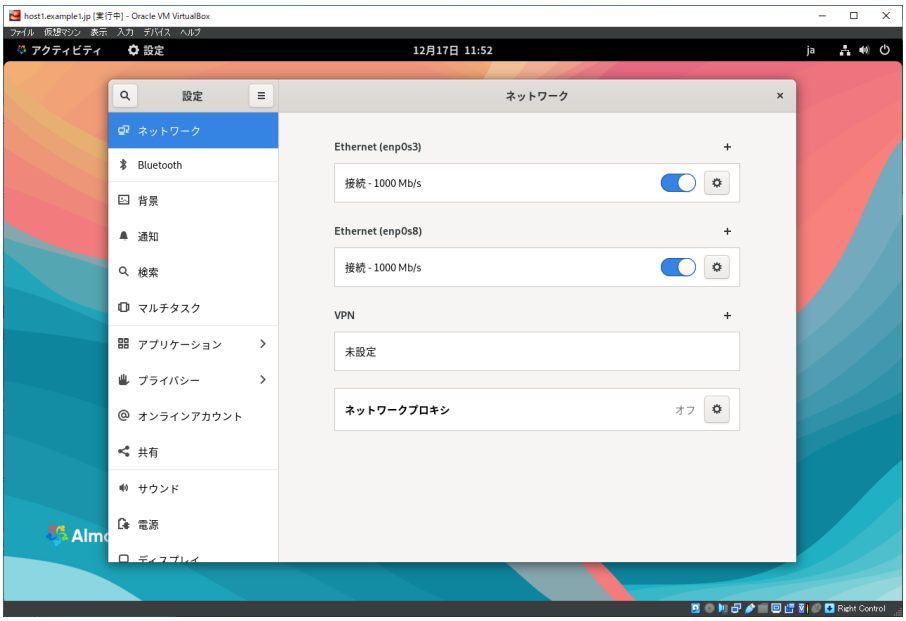

図 64: ネットワークの設定画面

接続プロファイルの設定画面が表示されるので、設定を変更したら「適用」ボタンを押して元の画面に戻ります。

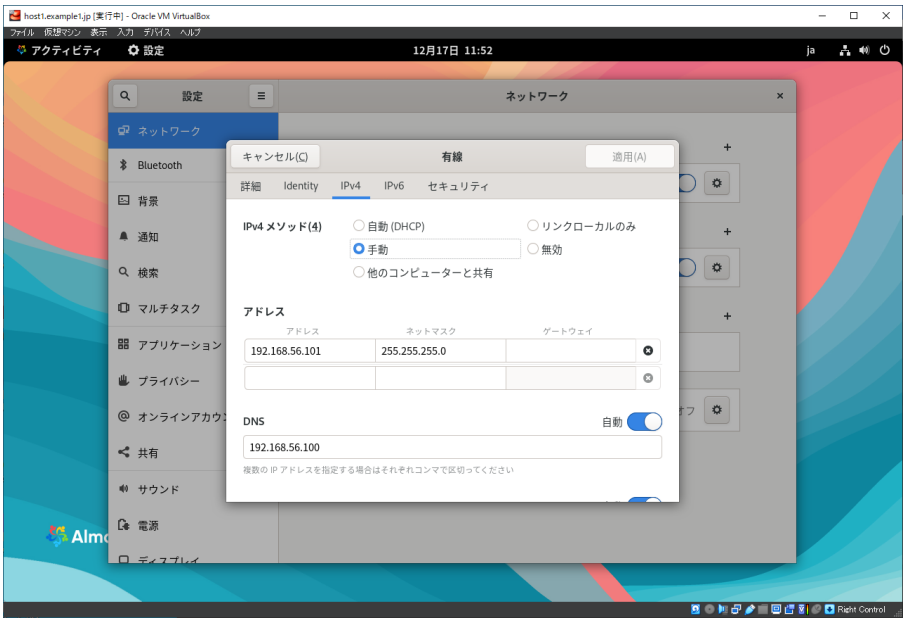

図 65: インターフェースの設定画面

変更した設定を適用するには、変更したネットワークインターフェースの欄にあるスイッチを一旦「オフ」に変えて再度「オン」にします。

7.2.3　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（192.168.56.100）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 Ctrl+C で中止できます。

```
$ ping 192.168.56.101 ← 自分のIPアドレス
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 バイト応答 送信元 192.168.56.101: icmp_seq=1 ttl=64 時間=0.148ミリ秒
64 バイト応答 送信元 192.168.56.101: icmp_seq=2 ttl=64 時間=0.038ミリ秒
64 バイト応答 送信元 192.168.56.101: icmp_seq=3 ttl=64 時間=0.040ミリ秒
^C
--- 192.168.56.101 ping 統計 ---
送信パケット数 3, 受信パケット数 3, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.038/0.075/0.148/0.051 ms
```

7.3　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。

7.3.1　ss コマンドを使ったポート使用状況の確認

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。

```
$ ss -at
LISTEN           0            100                        127.0.0.1:smtp
                           0.0.0.0:*
LISTEN           0             10                        127.0.0.1:domain
                         0.0.0.0:*
LISTEN           0             10                        127.0.0.1:domain
                         0.0.0.0:*
LISTEN           0            100                 192.168.56.101:smtp
                           0.0.0.0:*
LISTEN           0           4096                        127.0.0.1:ipp
                            0.0.0.0:*
LISTEN           0            100                          0.0.0.0:imap
                           0.0.0.0:*
LISTEN           0            100                          0.0.0.0:pop3
                           0.0.0.0:*
LISTEN           0            128                          0.0.0.0:ssh
                            0.0.0.0:*
LISTEN           0           4096                        127.0.0.1:rndc
                           0.0.0.0:*
 （略）
```

7.3.2　lsof コマンドを使ったポート使用状況の確認

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。

```
$ sudo lsof -i
COMMAND     PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
 （略）
sshd        997     root    3u  IPv4  21008      0t0  TCP *:ssh (LISTEN)
sshd        997     root    4u  IPv6  21010      0t0  TCP *:ssh (LISTEN)
```

```
dovecot    1292     root   21u  IPv4  22257      0t0  TCP *:pop3 (LISTEN)
dovecot    1292     root   22u  IPv6  22258      0t0  TCP *:pop3 (LISTEN)
dovecot    1292     root   37u  IPv4  22271      0t0  TCP *:imap (LISTEN)
dovecot    1292     root   38u  IPv6  22272      0t0  TCP *:imap (LISTEN)
master    22406     root   13u  IPv4 454695      0t0  TCP localhost:smtp
    (LISTEN)
master    22406     root   14u  IPv4 454696      0t0  TCP mail.example1.jp:smtp
    (LISTEN)
master    22406     root   15u  IPv6 454697      0t0  TCP localhost:smtp
    (LISTEN)
named     24939    named   23u  IPv4 516918      0t0  UDP localhost:domain
named     24939    named   24u  IPv4 516919      0t0  UDP localhost:domain
named     24939    named   25u  IPv4 516920      0t0  TCP localhost:domain
    (LISTEN)
named     24939    named   27u  IPv4 516921      0t0  TCP localhost:domain
    (LISTEN)
 （略）
```

7.3.3　services ファイルによるポート番号の確認

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。

```
$ 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
 （略）
```

7.4　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。

7.4.1　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 サーバーの公開鍵が送られてきて接続してもよいか確認されるので「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。

```
[admin@host1 ~]$ ssh user1@localhost← user1としてlocalhostに接続
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:7+us06xcMV24dGBfoGXIKCyiDWexydVXYlbYGMyV4Mk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes ←
    yesを入力
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
user1@localhost's password: userpass ← 実際には非表示
```

```
Last login: Sat Dec  2 11:44:55 2023
[use1@host1 ~]$ exit ← リモートログインを修了
ログアウト
Connection to localhost closed.
[admin@host1 ~]$ ← 元のadminユーザーに復帰
```

7.4.2 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_rsa.pub）と秘密鍵（id_rsa）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。

```
[admin@host1 ~]$ su - user1 ← user1ユーザーに切り替え
パスワード:userpass ← user1のパスワードを入力（非表示）
最終ログイン: 2023/12/05 (火) 11:46:38 JST 日時 pts/1
[user1@host1 ~]$ ssh-keygen ← 鍵形式を省略したのでRSA形式で鍵を生成
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user1/.ssh/id_rsa): ←
    Enterキーを入力
Created directory '/home/user1/.ssh'.
Enter passphrase (empty for no passphrase): userpass ←
    秘密鍵にパスフレーズを設定（非表示）
Enter same passphrase again: userpass ← パスフレーズを再入力（非表示）
Your identification has been saved in /home/user1/.ssh/id_rsa
Your public key has been saved in /home/user1/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:aiUB6c+AYV4K+8b6d3RgAYmYsoH3d2waUN0G3MmHNDk user1@host1.example1.jp
The key's randomart image is:
+---[RSA 3072]----+
|.o .o=.o.*o+     |
|B = =.. o E..    |
|.O B ..o . o     |
|o + + =.+        |
| o   *.*S        |
|  +   =+.        |
| o   .o.         |
|.   ...          |
| ... .           |
+----[SHA256]-----+
```

1. 接続先に authorized_keys を作成するユーザーに公開鍵認証による 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 ディレクトリ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。

```
[user1@host1 ~]$ ls -ld .ssh
drwx------. 2 user1 user1 38 12月  5 11:47 .ssh
[user1@host1 ~]$ cd .ssh
[user1@host1 .ssh]$ touch authorized_keys
[user1@host1 .ssh]$ chmod 600 authorized_keys
```

```
[user1@host1 .ssh]$ cat id_rsa.pub >> authorized_keys
[user1@host1 .ssh]$ ls -l authorized_keys
-rw-------. 1 user1 user1 577 12月  5 12:08 authorized_keys
```

1. 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。

```
[user1@host1 ~]$ ssh user1@localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:7+us06xcMV24dGBfoGXIKCyiDWexydVXYlbYGMyV4Mk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes ←
    yesと入力
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/user1/.ssh/id_rsa':userpass
    ←秘密鍵のパスフレーズを入力（非表示）
Last login: Tue Dec  5 11:46:46 2023
[user1@host1 ~]$ exit
ログアウト
Connection to localhost closed.
```

7.4.3 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、SSH 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. admin ユーザーでパスワード認証で接続できることを確認する user1 ユーザーになっている場合には exit して admin ユーザーに戻ります。

```
[user1@host1 ~]$ exit
ログアウト
[admin@host1 ~]$ ssh localhost
admin@localhost's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Dec  5 12:16:49 2023 from ::1
[admin@host1 ~]$ exit
ログアウト
Connection to localhost closed.
[admin@host1 ~]$
```

1. 設定ファイル/etc/ssh/sshd_config を修正する

```
[admin@host1 ~]$ sudo vi /etc/ssh/sshd_config
```

```
（略）
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
PasswordAuthentication no ← noに設定した行を追加
#PermitEmptyPasswords no
（略）
```

1. 設定を変更後、sshd 設定を再読み込みする

```
[admin@host1 ~]$ sudo systemctl reload sshd
```

1. 公開鍵認証を設定していないユーザーで SSH サーバーに接続して、接続できないことを確認する

```
[admin@host1 ~]$ ssh localhost
admin@localhost: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

この設定を行った後はパスワード認証でリモートログインできなくなるので、かならず事前に管理権限を持ったユーザーが公開鍵認証でリモートログインできるようにしておく必要があります。

7.5 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。

AlmaLinux 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 コマンドで行います。

7.5.1 ファイアウォール設定の確認

まず、許可されているサービスを調べます。

```
$ sudo firewall-cmd --list-services
cockpit dhcpv6-client http ssh
```

ここでは、HTTP や SSH などのプロトコルが使用するポートが受信を許可されています。

7.5.2 許可サービスの追加

サービスの許可を追加します。以下の例では、imap サービスを許可しています。

```
$ sudo firewall-cmd --add-service=imap
success
```

この設定は即座に有効になりますが、システムを再起動すると有効にはなりません。再起動後も有効にするには、これまでの実習で行った以下のような設定を行った後に再読み込みを行うか、後述する設定の保存を行います。

```
$ sudo firewall-cmd --add-service=imap --zone=public --permanent
$ sudo firewall-cmd --reload
```

7.5.3 設定可能なサービスの確認

設定可能なサービスはあらかじめ定義されているので、一覧を確認してみます。

```
$ sudo 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
```

プロトコル名の場合もあれば、使用したいソフトウェアの名称で定義されている場合もあります。

7.5.4 許可サービスの取り消し

許可されているサービスを取り消しすることもできます。

```
$ sudo firewall-cmd --remove-service=imap
success
```

この設定も一時的なもので、システムの再起動時に許可したくない場合には、次の設定の保存が必要です。

7.5.5 ファイアウォール設定の保存

上記の方法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。再起動後も設定を有効にするには、現在の設定を保存します。

```
$ sudo firewall-cmd --runtime-to-permanent
```

Linux サーバー構築標準教科書

2025 年 4 月 1 日　Ver.4.0.1

発行元　LPI-Japan