できる

dekiru PRO series

Red Hat Enterprise Linux 7

レッドハット株式会社
ソリューションアーキテクト 平 初
&できるシリーズ編集部

redhat

ハンズオン形式でらくらく入門!

基本操作・各種サーバー構築・運用管理手法を網羅

ファイアウォール・DNSサーバー・Webサーバー・ファイルサーバー・メールサーバー
データベースサーバー・仮想化・コンテナー etc.

インプレス

できる

dekiru PRO series

Red Hat Enterprise Linux 7

レッドハット株式会社
ソリューションアーキテクト 平 初
&できるシリーズ編集部

redhat

ハンズオン形式でらくらく入門!

基本操作・各種サーバー構築・運用管理手法を網羅

ファイアウォール・DNSサーバー・Webサーバー・ファイルサーバー・メールサーバー
データベースサーバー・仮想化・コンテナー etc.

インプレス

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。
Red Hat、Fedora は、Red Hat, Inc. の米国およびその他の国における商標です。
Microsoft、Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
その他本書に記載する製品名は、一般に各開発メーカーの商標または登録商標です。

なお、本文中には ™ および ® マークは明記していません。

まえがき

本書では、Red Hatが提供するエンタープライズ向けLinuxの「Red Hat Enterprise Linux（RHEL)」というディストリビューションを解説していきます。RHELは世界中で最も多く使われている商用Linuxディストリビューションです。政府、情報通信、金融、製造、流通、医療、電力、放送、教育など幅広く使われており、世界中のさまざまな仕組みを陰で支えています。私が本書の執筆のためによく利用していたカフェのコーヒー豆、このコーヒー豆の受発注システムにもRHELが採用されています。

ITの現場は目まぐるしく変化を遂げています。10年程前までは「Linuxなんて趣味のソフトウェアでしょう？」と言われていましたが、今ではメインフレームやUNIXの置き換え先として、世界中のミッションクリティカルな環境でLinuxが採用される時代です。身近なところでも、あなたのリビングにある液晶テレビ、きっと中身はLinuxです。

この書籍を読む前にRHELに対して詳しく知っている必要はありません。1つ1つ解説していきます。ただし、なにか起きたとしても慌てないでください。エラーメッセージをよく読むと対処方法がだいたい書かれています。徐々に分かるようになります。

本書は、RHEL 7を活用して組織内におけるイントラネットサーバーを構築したり、インターネット回線を使って外部向けサーバーを構築したりすることを手順を追って紹介しています。また、RHEL 7で新しく採用されたsystemdをはじめ、KVMを使った仮想化、Dockerを使ったコンテナー管理などの最新の技術についても触れています。

RHELの歴史から、インストール方法、具体的な使い方まで、一通り頭から読んでいただければ、それなりにRHEL 7を理解したつもりになれるでしょう。「理解したつもりになる」ということは小さな自信となります。新しい知識を学習する上で非常に重要なことです。RHEL 7は10年間の製品ライフサイクルで提供される製品なので、本書の知識は製品リリースの10年後である2024年まで生きます。少し時間をかけて勉強してみる価値は必ずあります。

本書は特定のサーバーアプリケーションについて詳細に解説しているわけではありませんので、この書籍をきっかけにRHEL 7に収録されているさまざまなソフトウェアにも、さらに興味を持っていただけたら幸いです。

最後に、本書を執筆している間、献身的に支えてくれた妻 愛美と長男。執筆に行き詰まったときに笑顔で癒やされました。また、執筆中に無事に誕生してくれた次男。オープンソースソフトウェアの開発を日々支えている方々、職場の皆様、そして、できるシリーズ編集部の高橋さんに感謝します。

2015年5月吉日

平 初

本書の読み方

本書では、新人のIT管理者向けにビジュアルを使い、Red Hat Enterprise Linux 7（RHEL 7）の基本操作やサーバー構築について、ハンズオン形式で解説します。

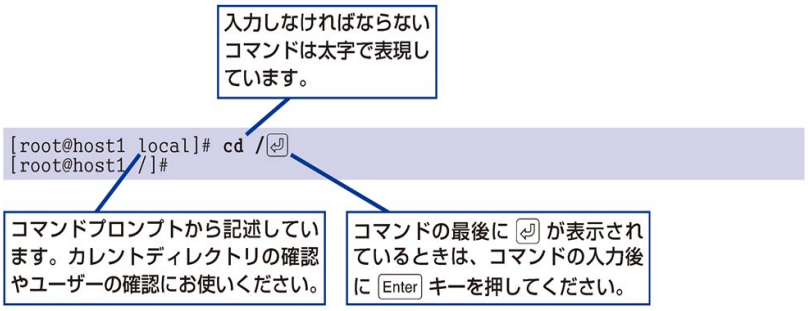
Red Hatログインのアカウントを作成し、RHEL 7の評価版のダウンロードから始め、OSのインストールと操作の基礎を学んだあと、企業内で利用する各種サーバーを構築します。

Red Hatログインのアカウントを持っていない場合は、必ず第2章を参考にアカウントを取得してください。また、RHEL 7のサブスクリプションを持っていない場合は、同じく第2章を参考に評価版サブスクリプションを取得し、RHEL 7のインストールと設定のあとで第4章を参考にシステムをRed Hatカスタマーポータルに登録してください。

RHELは、企業内の基幹業務システムの運用を目的としたOSであるため、システム管理、各種サーバーの構築など、運用管理の応用に重点を置いています。また、RHEL 7で一新された、新しい管理手法を学ぶ技術者にも役立つ内容です。

コマンドの表記

本書では、コマンドラインでの操作方法を次のように表現しています。



ライセンスについて

RHEL 7を利用するにはRed Hatのサブスクリプション登録が必要です。本書では評価版サブスクリプションを取得してRHEL 7を利用する手順を例に解説しています。なお、LinuxカーネルはGNU General Public License（GPL、GNU一般公有使用許諾契約書）によって保護されます。そのほか、RHEL 7に含まれるソフトウェアの多くにはオープンソースライセンスが適用されます。

他のディストリビューションへの対応について

本書はRHEL 7に合わせた設定を行なっているので、他のLinux（CentOSやFedora、Ubuntu、Debianなど）および以前のバージョンでの動作は保証いたしません。必ず、

本書に記述された手順にしたがって読み進めてください。

ご質問をお送りいただく前に

本書の内容についてご質問をお送りいただく前に、弊社の書籍紹介ページhttp://book.impress.co.jp/books/1114101057からたどれる「正誤表」をご確認ください。これまでに判明した正誤があれば、「お問い合わせ/正誤表」タブのページに正誤表が表示されます。

本書の内容に関するご質問方法と免責事項

ご質問は返信用切手を同封した封書もしくはメールにてお受けいたします（電話、FAXでのご質問には対応しておりません）。本文に、書籍名、ご質問のページ番号、ご質問内容、パソコンのメーカー名および機種名、増設した周辺機器、ネットワーク環境をできる限り詳細にお書きの上、お送りください。

ただし、お客様固有の環境に依存するご質問や、編集部で現象が確認できない場合、インターネット上の情報が更新され代替するものがない場合など、確実な解決方法をご提示できないこともあります。また、回答作成のための調査に時間がかかる場合もあり、回答期限のお約束はできません。本書の内容およびご質問の回答は、お客様の問題解決を保証するものではないことを、あらかじめご了承ください。

●読者の皆様のお問い合わせ先

インプレスカスタマーセンター
〒101-0051 東京都千代田区神田神保町一丁目105番地
info@impress.co.jp

第5章　サーバーを準備する　111

第1章 Red Hat Enterprise Linuxについて知る

現在、Linuxは、企業や政府、自治体、研究機関など、業種業態問わず広く利用されています。この章は、Linuxの生い立ちから、Red Hat Enterprise Linuxの歴史、その販売・開発形態、サポートライフサイクル、そして、最新バージョンRed Hat Enterprise Linux 7の特徴など、本書を読み進める上での前提となる部分についてご説明します。

●この章の内容

1-1 Linuxってなに？

Linuxの基礎知識

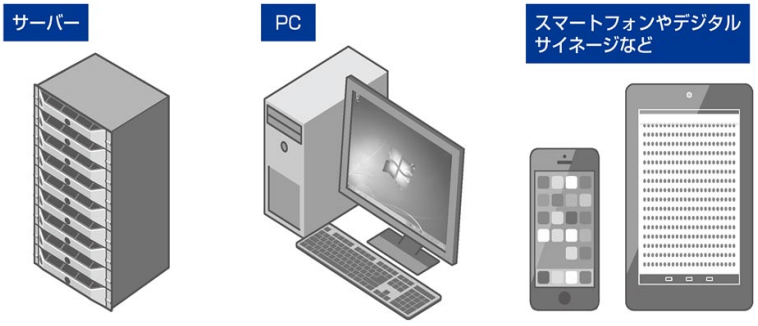
LinuxはオープンソースのOSです。Linuxはサーバーから、デスクトップ、組み込み用途まで幅広く使われています。とりわけ、UNIXというOSを置き換えるサーバー用途に多く使われています。TCP/IPが発明された後に誕生したLinuxは、ネットワークと親和性を最初から意識して作られており、さまざまなサーバー機能を提供することができます。1990年代のインターネット黎明期に急速に普及しました。本書では、Red Hat Enterprise Linux 7（RHEL 7）というLinuxディストリビューションを用いて、さまざまなサーバーとしての使い方を解説します。

Linuxの誕生と普及

Linuxは、Linus Torvalds氏がヘルシンキ大学の大学院生だったときに開発したのが始まりです。誰もが自由に改変・再配布できるGPLというライセンスに基づき、無償で公開されたことにより、多くの開発者の賛同を得ることに成功しました。その結果、Intelのx86系CPU以外にも、AlphaやPowerPC、ARMなど幅広くさまざまなプラットフォームに移植されました。

現在では、その適用範囲はサーバーだけではなく、街角にあるデジタルサイネージ（電子看板）や、いつも持ち歩くスマートフォンなどにも採用されています。Googleが開発しているAndroidや、Mozillaが開発しているFirefox OSなどスマートフォン向けのOSも、今回の主役であるRed Hat Enterprise Linuxも見た目は違えど、同じLinuxカーネルを採用しています。

さまざまなプラットフォームで使われるLinux



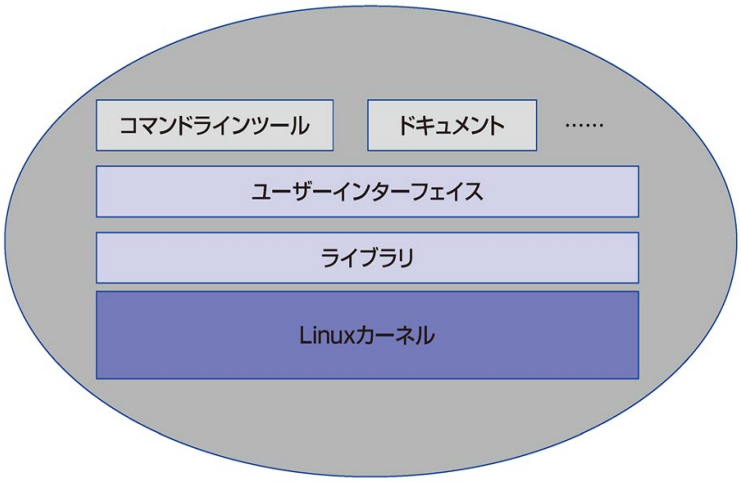
Linuxカーネルとディストリビューション

Linuxカーネルは、OSの中核となる部分でアプリケーションとハードウェアの橋渡しを行います。主な役割としては、プロセス管理やメモリー管理、デバイス管理などがあります。また、アプリケーションに対して、カーネル内部の機能を呼び出すためのシステムコールという特別な関数を提供します。

カーネルが各種ハードウェアを抽象化する仕組みを提供することで、異なるメーカーのコンピューターであっても、CPUのアーキテクチャーが一緒でありデバイスドライバーさえ用意されていれば同じアプリケーションを動かすことができます。また、CPUのアーキテクチャーが異なっていたとしても再コンパイルして大規模な移植作業なしに動かすことができます。

Linuxカーネルを主軸として、各種コマンドラインツール、ライブラリ、ユーザーインターフェイス、ドキュメントなどを同梱した配布形態を、ディストリビューションといいます。

Linuxディストリビューション



1-2 Red Hat Enterprise Linuxってなに？

RHELの概要

Red Hat Enterprise Linux（RHEL）は、Red Hatが提供する企業向けのLinuxディストリビューションです。Red Hatがソフトウェアをパッケージングし、品質管理（QA）プロセスを経て、各種メーカーのハードウェア上で動作認定した上で提供されます。さまざまなハードウェアプラットフォームで利用できます。また、契約ユーザーには、長期の製品ライフサイクルとテクニカルサポートが提供されます。このレッスンでは、RHELを使う上で知っておきたい販売形態や開発形態、サポートライフサイクルなどの仕組みについて解説します。

RHELの概要

RHELは、ノートPCからメインフレームまでを幅広くカバーする、クライアント向けおよびサーバー向けOSです。3000個以上のRPMパッケージで構成され、3000種類を超えるハードウェア認定、9000種類を超えるISVアプリケーション認定、また、現在ではIntel EM64T、IBM POWER、IBM System z向けアーキテクチャーをサポートしています。2014年6月にリリースしたRHEL 7からは64bit版のみを提供しています。

サブスクリプション契約のもとに、無制限回数のテクニカルサポートが提供されます。物理サーバー、仮想サーバー、パブリッククラウドと、幅広い環境で稼働します。

RHELは、企業や政府、自治体、研究機関など、業種業態問わず広く利用されています。日本市場では国内の商用Linuxディストリビューションのうち約85%のシェアを占めています（執筆時点）。

これまでのRHEL

バージョン	リリース
RHEL 2.1	2002年 3月
RHEL 3	2003年10月
RHEL 4	2005年 2月
RHEL 5	2007年 3月
RHEL 6	2010年11月
RHEL 7	2014年 6月

RHELの歴史

昔に遡ること1993年、Red Hat Enterprise Linuxの前身となるRed Hat Linuxがありました。Red Hat Linuxは、SlackwareやDebian GNU/Linuxなどと並ぶ最古参のLinuxディストリビューションです。

Red Hat Linuxは、RPMパッケージというパッケージ管理システムを採用し、AnacondaというGUIのインストーラーが搭載された、当時では画期的なLinuxディストリビューションでした。その当時は、FTP版と言われる無償提供版がFTPサーバーからダウンロードできたり、PC雑誌の付録とかについていたり、また、サポート付きRed Hat Linuxが箱に入った状態で量販店のソフトウェアコーナーで売られたりしていました。

1990年代後半になると、SAP社やOracle社などの商用のアプリケーションがRed Hat Linux上での動作を保証するようになりました。しかし、当時のRed Hat Linuxは、マイナーバージョンアップするだけでもカーネルやライブラリの互換性が失われてしまい、企業で利用するには少々難点がありました。

そこから方向転換をして、2002年に企業向けに長期間安定して提供するRed Hat Enterprise Linux（RHEL）2.1をリリースしました。それまでのFTP版にあたるものは、2003年にFedora Core（現在のFedora）としてリリースされるようになりました。The Fedora Projectで開発されるFedora Coreは、次世代RHEL開発用のLinuxディストリビューションとして定義されました。

2002年3月にRHEL 2.1をリリースした後、2年から3年おきにメジャーバージョンアップを繰り返し、2014年6月にRHEL 7をリリースしました。

次のページに続く

RHELの販売形態

RHELは、デスクトップ、ワークステーション、サーバー、メインフレーム向けに製品型番が用意されており、通常1年もしくは3年単位のサブスクリプション形式で提供されます。「サブスクリプション」という言葉に耳慣れない方もいるかもしれませんが、簡単に言うと保守サポート契約です。RHELのサブスクリプションは会計上、ソフトウェア資産には該当しません。

RHELには、Red Hatの直販営業もしくはRed Hat認定ディストリビューターから販売されRed Hatから1 ～ 3次サポートが提供されるリテール版RHEL（L1-L3サポート）と、サーバーベンダーから販売されサーバーベンダーから1 ～ 2次サポートが提供されるOEM版RHEL（L3サポート）が存在します。

どちらも同じOSとして提供されますが、大きく違うのは、どこから買うのかという点と1 ～ 2次サポートの提供者がどこの会社なのかという点です。3次サポートとしては、どちらでもRed Hatが対応します。

また、細かいところでは、サーバーベンダーが提供するOEM版の場合には、自社のサーバーで最適に利用するためのデバイスドライバーやハードウェア監視プログラムをセットで提供して、そのサポートを提供している場合があります。

さらに最近では、Red Hat認定クラウドプロバイダーから提供される、従量課金のCCP版RHELがあります。クラウドプロバイダーから、インスタンス費用と合算で利用した分だけ請求され、個別のサブスクリプション契約は不要です。クラウドプロバイダーによっては、RHELを1時間単位で提供してくれますので、突発的なトラフィックが予想されるキャンペーンサイトなどで一時的に多数利用する場合にはリーズナブルです。この場合、RHELの1次サポートはクラウドプロバイダーが行います。

サブスクリプションで提供されているコンテンツにカスタマーポータルからアクセスできる

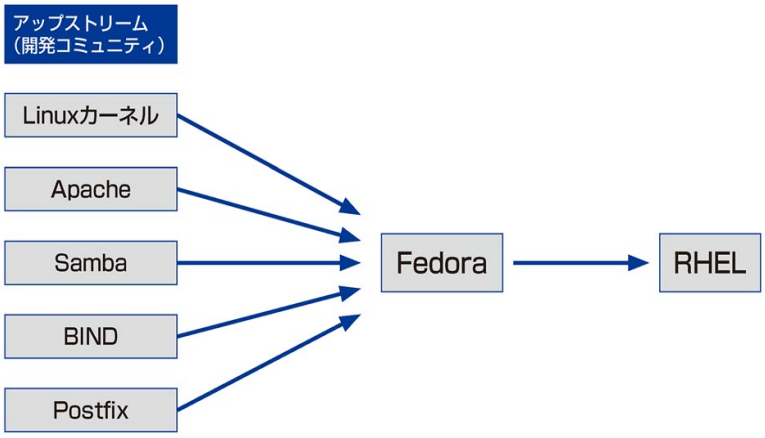
RHELの開発形態

RHELは、Fedoraが開発のベースとなっています。Fedoraから約3年間隔で派生してエンタープライズブランチが作られます。RHELに収録されているソフトウェアは、必ずしもコミュニティで開発されているソフトウェアの最新版ではなく、少し枯れたバージョンがピックアップされて収録されます。

まずはコミュニティで開発されたソフトウェアをパッケージングしてFedoraとしてリリースします。Fedoraで、バグ修正や他のソフトウェアとの組み合わせが確認されます。その結果が月日を経てRHELとなり、長期安定版として多くの人へ届けられるという流れです。

昔からRed Hatは、コミッターと呼ばれる決定権を持つ世界中の開発者をフルタイムで雇っています。以前はLinuxカーネル中心でしたが、現在では主要なコミュニティにはRed Hat社員がいる状況となっています。そして日夜、さまざまなコミュニティのプロジェクトの中で開発しています。このような個々のソフトウェアの集合から生まれる最終形がRHELと言えます。

また、RHELでは「アップストリームファースト」という開発形態を採用しています。これは、開発コミュニティに対してバグ修正や機能改善を行った上で、自社のディストリビューションであるRHELの中のパッケージに修正を取り込むというやり方です。オープンソースソフトウェアにおいて、アップストリームにマージしない独自拡張をメンテナンスするためにフォークするのは手間もコストもかかり合理的な選択肢ではありません。20年以上Linuxでビジネスしている Red Hatが導き出した最適解とも言えるでしょう。



次のページに続く

サポートライフサイクル

Red Hat Enterprise Linuxのライフサイクルは、2012年5月に、7年から10年に延長されました。現在では、標準サポートライフサイクルで10年間、一部のサーバーベンダーから提供するミッションクリティカル向けの拡張サポートを含めると13年間サポートが受けられるOSです。企業で利用する汎用OSの中では、最も長いサポートライフサイクルです（執筆時点）。

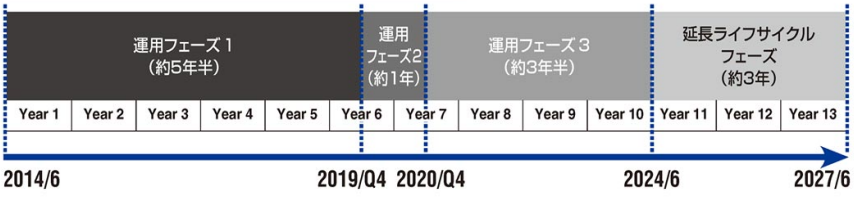
しかも、この10年間のライフサイクルは、サーバー向けのRHELだけではなく、デスクトップやワークステーション向けのRHELに対しても適用されます。

サブスクリプション契約を更新すると、バグ修正やセキュリティ修正を行った更新パッケージが提供されます。そして、無制限回数のテクニカルサポートを受けられるようになります。また、バージョンアップやバージョンダウンに対する追加費用も発生しません。

Red Hat Enterprise Linuxのライフサイクル

https://access.redhat.com/ja/support/policy/updates/errata

Red Hat Enterprise Linux 7のサポートライフサイクル



※延長ライフサイクルフェーズは、限定的なサポートしか提供されないが、Red Hat CDNからソフトウェアを入手できる延命期間

適切な問い合わせ先

Red Hatは、年間サブスクリプションモデルと呼ばれる形式で提供されており、ソフトウェアの販売ではありません。では、Red Hatは何を提供するかというと、Enterprise Agreement（EA）という契約文書のもと、サブスクリプション契約が有効な期間だけ、製品や修正パッケージ、テクニカルサポートを提供します。

テクニカルサポートの拠点は日本を含む世界中にあります。日本で購入したサブスクリプションでは、日本のオフィスに勤務するサポートスタッフが日本語もしくは英語で対応します。カスタマーポータルのグローバルサポートサービスのページから新規サポートケースの受付およびサポートに関するSLAなどの情報を提供しております。

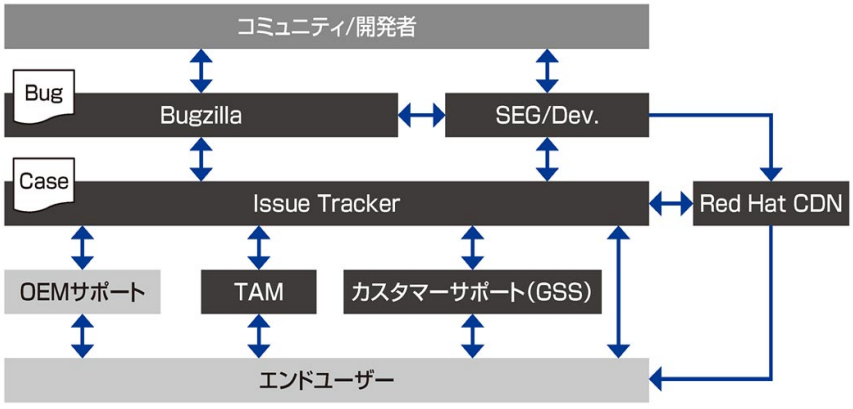
カスタマーポータル - グローバルサポートサービス
https://access.redhat.com/support/

Bugzillaをバグ報告先だと思われて、直接Bugzillaへ報告されている方がときどきいます。過去は、バグ報告先として利用していた時期もあります。現在、カスタマーポータルからサポートケースをオープンして頂き、そこでテクニカルサポートがバグだと判断して、開発チームへエスカレーションする仕組みになっています。では、現在のBugzillaは何かというとテクニカルサポートと開発チームとの間のバグ追跡システムという位置づけです。

ちなみに、サブスクリプションを購入していない時点において、Red Hatに対して質問したい場合の質問先もあります。レッドハットへのお問い合わせフォームがあり、ここから質問すると、Red Hatの営業が購入前の質問に対して答えてくれます。

・購入前の製品に対するご質問
→ レッドハットへのお問い合わせ http://jp.redhat.com/contact/sales.html
・障害の報告、機能要求、バグ報告先
→ Red Hatカスタマーポータル https://access.redhat.com/support/
・サブスクリプションのアクティベート
→ Red Hatカスタマーポータル https://access.redhat.com/subscriptions/activate/
・カスタマーポータルへログインできないなど
→ Red Hatカスタマーサービス https://access.redhat.com/ja/support/contact/customerService
・Red Hat製品のダウンロード
→ Red Hatカスタマーポータル https://access.redhat.com/
・Fedoraのバグ報告先
→ Bugzilla https://bugzilla.redhat.com/enter_bug.cgi?classification=Fedora

Red Hat Enterprise Linuxのサポートフロー



1-3 Red Hat Enterprise Linux 7を知ろう

RHEL 7の特徴

RHEL 7には、RHELの開発用Linuxディストリビューションであ るFedoraで開発された多くの新機能が追加されています。その新機能には、カーネルからユーザーランドの仕組みまで、広範囲なものがあります。最新テクノロジーや最新のハードウェア、新しい業界規格への対応はもちろんのこと、RHEL 7には企業のITインフラの10年後を見据えた仕組みが新機能となって追加されています。

このレッスンでは、RHEL 7の機能的な特徴について、今回RHEL 7で搭載されたキーテクノロジーを中心に紹介します。

64bitアーキテクチャーのみサポート

RHEL 7は、3年半の開発期間を経て2014年6月10日にリリースされました。RHELでは初めてLinuxカーネル3.x系のバージョン3.10を採用しており、Intel EM64T（x86_64）、IBM POWER（ppc64）、IBM System z（s390x）向けアーキテクチャーをサポートしています。

RHEL 7の正式公開（GA：Generally Available）から、ライフサイクルの期間の間、主要ソフトウェアコンポーネントのバージョン番号が変わらず（リベースされず）にバックポート（同じバージョンに不具合修正などを加えること）を繰り返して、更新パッケージがリリースされていきます。つまりRHEL 7に含まれるLinuxカーネルのバージョン3.10は、今後10年間以上メンテナンスされるわけです。

また、RHEL 7から64bit版のRHELのみが提供されます。従来の32bitアプリケーション向けには、32bitの互換ライブラリが提供されます。近年ではデータ量の増加により、32bitのアドレス幅の限界が見えてきたことと、4GBを超えるメモリー搭載量のサーバーが一般的になってきたことから、64bitのみに限定した方が合理的だという結果に落ち着きました。

その他、NUMA（Non-Uniform Memory Access）型のマルチプロセッサシステムにおいて、プロセスとメモリーをNUMAノード間で自動的に移動する仕組みを搭載し、さらなる性能向上が行われました。

systemdとfirewalldを採用

システム全体を管理する仕組みが、従来のUpstartからsystemdへ置き換わりました。システム起動時のサービスの起動がすべてsystemdによって行われ、サービス間の依存性の管理や、起動順序の管理などもsystemdが行います。

これにともない、サービスを起動したり停止したりする際に使っていたinitスクリプトが廃止され、すべてsystemdの「ユニットファイル」にて管理されます。

ファイアウォールの管理の仕組みとしては、firewalldが搭載されています。従来のiptablesも利用できますが、firewalldがRHEL 7のデフォルトとなっています。

そして、仮想化技術のLinux KVMの機能改善はもちろんのこと、コンテナー管理ツールのDockerも搭載されました。

ファイルシステムはXFSがデフォルト

デフォルトのファイルシステムが、従来のLinux ext4から、RHEL 7ではXFSになりました。近年のデータ増加量のトレンドから、1つのファイルシステムの容量上限および性能面でのスケーラビリティが求められることが変更の背景です。また、XFSでサポートされるファイルシステムの容量上限も、100TBから500TBと、5倍に引き上げられました。

従来のRHEL 5やRHEL 6でも別売りの「Scalable Filesystem Add-on」にてXFSのサポートを行っていました。それに対してRHEL 7からは、標準でXFSのサポートが提供されます。

Linux ext4も、従来どおりサポートされます。サポートされるファイルシステムの容量上限は、16TBから50TBに引き上げられました。

「Resilient Storage Add-on」で提供されるクラスタリングファイルシステムの「GFS2 (Global File System 2)」も、容量上限が100TBから250TBに引き上げられました。

Red Hat Enterprise Linux 7で利用可能なファイルシステム

種類	容量上限	ルートファイルシステム	/bootファイルシステム
XFS	500 TB	Yes	Yes
ext4	50 TB	Yes	Yes
Btrfs	50 TB	Yes	Yes
GFS2	250 TB	No	No

※BtrfsはRHEL 7.1ではTechnology Previewとなっておりサポート対象外

次のページに続く

GNOMEクラシックとGNOME Shellが選べる

RHEL 7のデスクトップ環境としては、GNOME 3や KDE 4が搭載されて、デスクトップの見た目も変わっています。

リリースされているGNOME 3では、GNOME Shellというモダンなインターフェイスがデフォルトとなっています。しかし、RHEL 7に搭載されているGNOME 3では、GNOMEクラシックという古典的なデスクトップ環境がデフォルトとなっています。以前から左上にあったアプリケーションメニューのツリーメニューから辿ってアプリケーションを起動するという操作スタイルです。

GNOME Shellが好みの方は、ログイン時にGNOME Shellに切り替えて使うこともできます。GNOME Shellでは「アクティビティ」という概念が新しく加わっているのが、操作上の特徴です。左上の［アクティビティ］をクリックすると、起動中のアプリケーションのウィンドウ一覧と、ダッシュボードが表示されます。アプリケーションを起動する場合には、ダッシュボードから選んで起動します。GNOME Shellは、デスクトップPCやノートPC以外に、タブレットPCでも操作がしやすいように設計されているインターフェイスです。

GNOME Classicの
アプリケーションメニュー

GNOME Shellの
ダッシュボード

RHEL 7に含まれる主なパッケージ

RHEL 7に含まれる主なパッケージは、Webサーバー、DNSサーバー、データベースサーバー、メールサーバー、ファイルサーバーなど、範囲は多岐に及びます。パッケージのバージョンはFedora 18／19／20をベースとしています。

これらのパッケージがRHEL 7のリリース日（2014年6月10日）からライフサイクルの期間（10年間）、バージョンアップをせずにバックポートを繰り返しながらバージョン固定で提供されます。

RHELに含まれるパッケージのバージョンは古いとよく言われますが、きちんと更新パッケージを適用していれば然るべきセキュリティ修正も適用済みですし、また、マイナーリリースが出るタイミングで新しいハードウェアのサポートや機能拡張も行われます。

そのほか、コミュニティ版で言語仕様が変わりやすいスクリプト型言語に対しては、RHELに含まれることにより安定した実行環境を長期間安全に使うことができるという大きなメリットもあります。

主なパッケージのバージョン

カテゴリー	ソフトウェアとバージョン
カーネル	Kernel 3.10.0
主要サーバー	Apache httpd 2.4.6／Tomcat 7.0.42／Squid 3.3.8／Bind 9.9.4／MariaDB 5.5.35／PostgreSQL 9.2.7／SQLite 3.7.17／memcached 1.4.15 Postfix 2.10.1／sendmail 8.14.7／dovecot 2.2.10／cyrus-imapd 2.4.17／spamassassin 3.3.2／vsftpd 3.0.2／Samba 4.1.1／cups 1.6.3／OpenLDAP 2.4.39／FreeRADIUS 3.0.1／Kerberos5 1.11.3
各種言語処理系	OpenJDK 7／Perl 5.16／PHP 5.4.16／Python 2.7.5／Ruby 2.0.0／GCC 4.8.2
ライブラリ	glibc 2.17／libstdc++ 4.8.2

STEP UP

オープンソースとは何か

近頃、聞くことが多くなったオープンソースとは、そもそも何でしょうか？

古くは1980年代に遡ります。当時、ソフトウェア開発者の権利が厳しいことが、ソフトウェアの発展を妨げているとの意見が高まりました。そこでGNUプロジェクトの創始者であるRichard M. Stallman氏が「使用、学習、コピー、改変、再頒布を自由に行えるソフトウェア」が必要だと主張し、それをフリーソフトウェアと命名しました。

しかし、フリーソフトウェアが「自由」という本来の意味ではなく、「無料」のソフトウェアという意味で解釈されることもあり、1998年にフリーソフトウェアの否定的なイメージを払拭するために作られた「オープンソース」という名称を用いる動きから、今に至ると言われています。ちょうどこの頃は、Netscape Communications社がAOL社に買収され、自社で開発していたWebブラウザー「Netscape Navigator」のソースコードを公開した時期です。

後にThe Open Source Initiative（OSI）という組織が、オープンソース・ライセンスの要件として、「The Open Source Definition（OSD）」という以下の定義を掲げています。

1. 自由な再頒布ができること
2. ソースコードを入手できること
3. 派生物が存在でき、派生物に同じライセンスを適用できること
4. 差分情報の配布を認める場合には、同一性の保持を要求してもかまわない
5. 個人やグループを差別しないこと
6. 適用領域に基づいた差別をしないこと
7. 再配布において追加ライセンスを必要としないこと
8. 特定製品に依存しないこと
9. 同じ媒体で配布される他のソフトウェアを制限しないこと
10. 技術的な中立を保っていること

The Open Source Initiative：オープンソースの定義（日本語）
http://www.opensource.jp/osd/osd-japanese.html

第2章 RHEL 7をインストールする

RHEL7を使ってサーバーを構築していくために、まずはインストールが必要です。本書にインストールメディアは付属していませんので、入手方法からインストールまで順に説明していきます。Red Hatカスタマーポータルからインストールイメージをダウンロードすることによって、ご自身の環境でインストールDVDまたはインストールUSBメモリーを作ることができます。

●この章の内容

2-1 インストールについて確認しよう

インストールの準備

RHEL 7をインストールするために、まずはリリースノートを読んでシステム要件を確認し、インストール先のコンピューターを用意しましょう。また、RHEL 7のインストールイメージを入手する必要があります。

この章ではまず、Red Hat認定ハードウェアの調べ方から、RHEL 7のインストールイメージの入手方法、30日間評価版の申請方法、インストールDVDやインストールUSBメモリーの作成方法を解説します。その上で、RHEL 7のインストールする手順をステップバイステップで解説します。

RHEL 7をインストールするまでの流れ

RHEL 7のインストールを始める前には、環境要件やシステム要件などの調査が必要となります。

このレッスンで解説している内容は、実際にシステムを構築する際に調査／検討すべき点です。本題に入る前に、まずはRHEL 7のリリースノートを一読されることを強くおすすめします。難しくて理解できない部分は、本書を読み進めながら少しずつ理解していただければと思います。

Red Hat Enterprise Linux 7.1リリースノート

https://access.redhat.com/documentation/ja-JP/Red_Hat_Enterprise_Linux/7/html/7.1_Release_Notes/index.html

まず、RHEL 7のリリースノートでシステム要件を確認し、インストール先となるコンピューターを用意しましょう。

続いて、RHEL 7のインストールを行う前に、RHEL 7のインストールイメージを入手します。インストールイメージを入手する際には、Red Hatログインのアカウントと、有効なサブスクリプションが必要です。もしお持ち出ない場合には、手順に沿ってアカウントを作成したのちに、30日間評価版の申請を行ってください。

そして、ダウンロードしたインストールイメージからRHEL 7のインストールDVDやインストールUSBメモリーを作成したのちに、RHEL 7をインストールします。

HINT!

**物理サーバー以外に
インストールする場合**

仮想化技術を使い、RHEL 7を仮想マシン上にインストールして動かすという方法もあります。VMware WorkstationやMicrosoft Hyper-V、もしくはLinux KVM上のゲストOSとして利用できます。そのほか、Amazon EC2やGoogle Compute Engineなどのパブリッククラウドサービス上でRHEL 7を利用する方法もあります。

システム要件を確認しよう

RHEL 7をインストールするハードウェアのシステム要件は、Red Hatのインストールガイドに記載されています。以下に、インストールガイドから抜粋します。

- **BIOSもしくはUEFIが搭載されていること**
- **Intel ／ AMDの64bit CPU（x86_64）が搭載されていること**
- **最低1GBのメモリーが搭載されていること（1論理CPUあたり最低1GBを推奨）**
- **最低7.5GB（推奨10GB）のストレージ領域**
- **ストレージについては以下のどれかの条件を満たすこと**
  - SCSI、SATA、SCSIなどで接続された内蔵ディスク
  - Firmware RAIDで構成されたRAIDデバイス
  - Fibre Channelで接続された外部ディスク
  - iSCSIで接続された外部ディスク(iSCSI Boot)
  - Xenのブロックデバイス
  - VirtIOのブロックデバイス

なお、USBメモリーやSDカードへのインストールはサポート外です。

次のページに続く

インストールの種類を確認しよう

RHEL 7では、インストールするベース環境として、下の表にある種類が用意されています。なお、昔のRHELのように全部入りでインストールするという選択肢は、RHEL 7では存在しません。

インストールの種類

ベース環境	意味
最低限のインストール	OSとして最小限の基本的な機能です
インフラストラクチャサーバー	DHCPサーバーやDNSサーバー、認証サーバーなど、ネットワーク基盤となるサーバーです
ファイルとプリントサーバー	企業向けのファイルサーバー、プリントサーバー、ストレージサーバーです
ベーシックWebサーバー	静的および動的なインターネットコンテンツを提供するサーバーです
仮想化ホスト	最小の仮想化ホストです
サーバー（GUI利用）	GUIを使用してネットワークインフラストラクチャのサービスを動作させるサーバーです

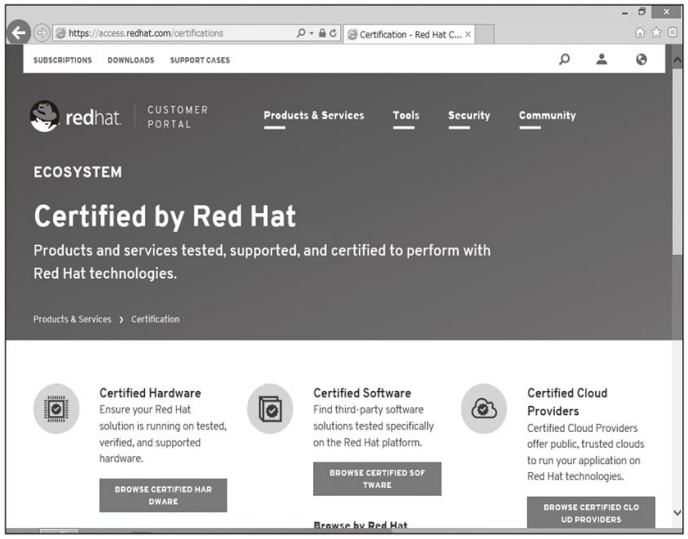
コンピューターはRed Hatの認定したものを

Red Hatが認定しサポートするコンピューターは、Red Hat Certified products catalog（https://access.redhat.com/certifications）のCertified Hardwareにメーカーと型番が掲載されています。ここに載っているコンピューターでないと、Red Hatがサポートを提供することはできません。

そのため、Certified Hardwareに掲載されているかどうかを確認した上で、RHEL 7をインストールしましょう。サーバーメーカー側がRed Hat社で認定を受けていると言っている場合でも、認定を受けているRHELのバージョンをCertified Hardwareできちんと確認して、RHEL 7に対して認定を受けているコンピューターを用意してください。そうでないと、インストールがうまく行えないケースや、GUIで画面が正常に映らなかったりハードディスクやネットワークカードが認識しなかったりするハードウェアトラブルに遭遇するケースがあります。

Certified Hardwareに掲載されていない機種のハードウェア絡みの障害については、Red Hat社はテクニカルサポートで対応ができません。ご了承ください。

Certified products catalog



2-2 RHEL 7を入手するには

インストールイメージのダウンロード

RHEL 7をインストールするためには、インストールイメージをダウンロードして、インストールのためのDVD-RやUSBメモリーを作成する必要があります。インストールイメージを入手するには、Red Hatログインのアカウントでログインします。アカウントをお持ちでない場合には、まずRed Hatログインのアカウントを作成します。評価版の利用のためであれば、個人のアカウントを作成すればよいでしょう。その後に、手持ちのサブスクリプションを利用するか、RHELの30日間評価版を申請して、インストールイメージをダウンロードします。

Red Hatログインのアカウントが必要

Red Hatのサブスクリプションサービスを受けるためには、「Red Hatログイン」と呼ばれるアカウントを作成する必要があります。これはRHEL 7の評価版を利用する場合にも必要です。

もうすでに何かしらのRed Hat製品をご利用であれば、1度は作成しているはずです。もしお持ちでない場合には、作成してください。

HINT!

最新のマイナーリリースを選ぶ

ダウンロードする時点で、RHEL 7のマイナーリリースが7.2や7.3などにバージョンアップしていた場合、一番新しいDVDイメージをダウンロードしてください。古いDVDイメージからインストールすると、インストール後のセキュリティアップデートの数が増えて、時間がかかります。

HINT!

DVDイメージのサイズに注意

ダウンロードするDVDイメージファイルは約3.5～4.0GBのサイズがあります。転送量に制限がある回線では、注意してください。

Red Hatログインのアカウントを作成する

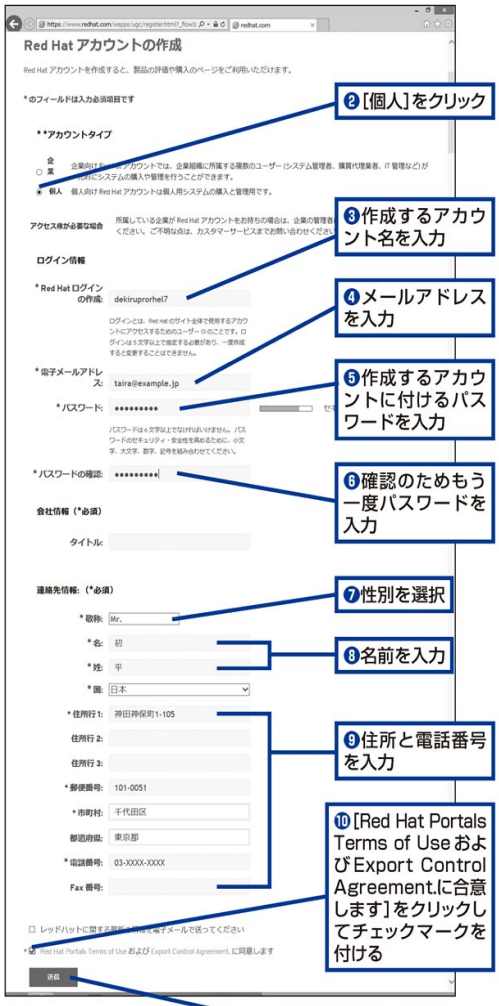
1 アカウント作成フォームに入力する

❶Webブラウザーで以下のURLにアクセス

https://www.redhat.com/wapps/ugc/register.html

アカウント作成フォームが表示される

ここでは評価版の利用のためだけなので個人のアカウントとする



2 アカウントの登録を確認する

登録の確認メール中のURLにアクセス

アカウントが作成された

3 ログインする

登録手続きをしていたウィンドウからログインする

次のページに続く

30日間評価版をダウンロードする

1 カスタマーポータルを表示する

❶Webブラウザーで以下のURLにアクセス

https://access.redhat.com/

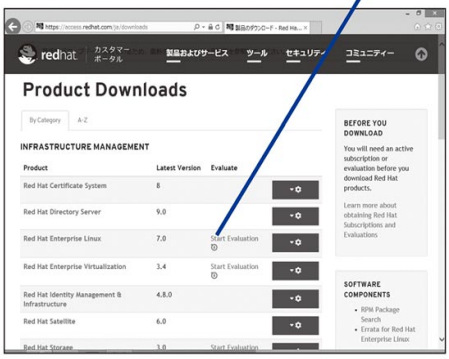
カスタマーポータルが表示された

❷[ダウンロード]をクリック

2 RHEL 7の評価版サブスクリプションを選ぶ

ソフトウェアの一覧が表示された

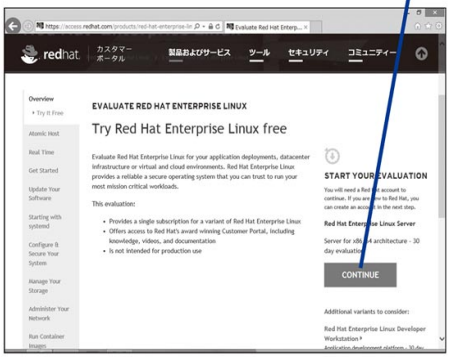
[Red Hat Enterprise Linux]の[Start Evaluation]をクリック



3 評価を開始する

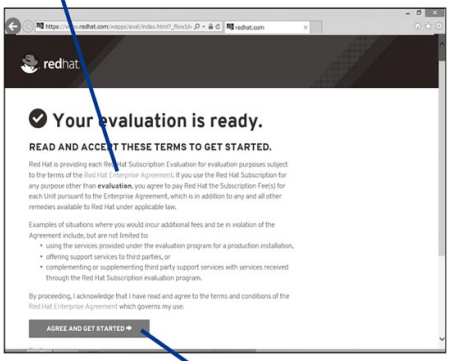
評価の開始を確認される

[Continue]をクリック



4 規約に同意する

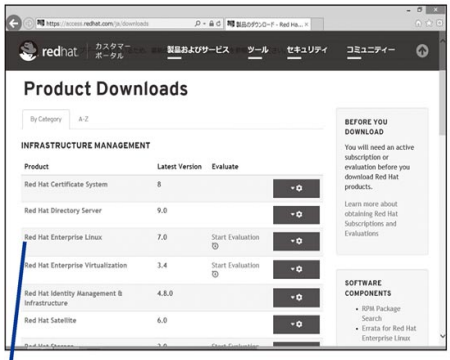
❶評価版規約を確認



❷[AGREE AND GET STARTED]をクリック

5 ダウンロードする

❶ソフトウェアの一覧ページに移動

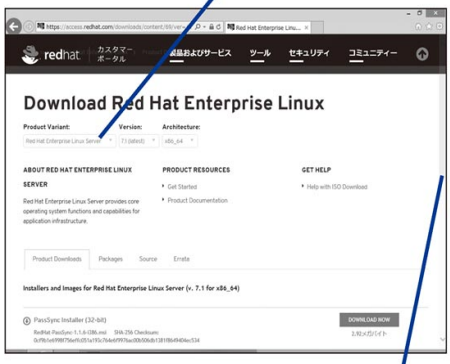


❷[Red Hat Enterprise Linux]をクリック

6 ダウンロードする製品を確認する

ダウンロードページが表示された

❶RHELの最新版であることを確認

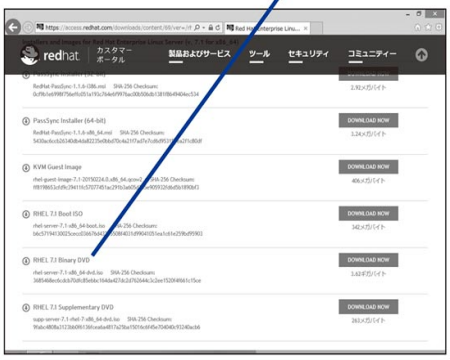


❷下にスクロール

7 インストールイメージをダウンロードする

ダウンロードを開始する

[RHEL 7.1 Binary DVD]をクリック



ダウンロードが実行される

2-3 インストールメディアを作成するには

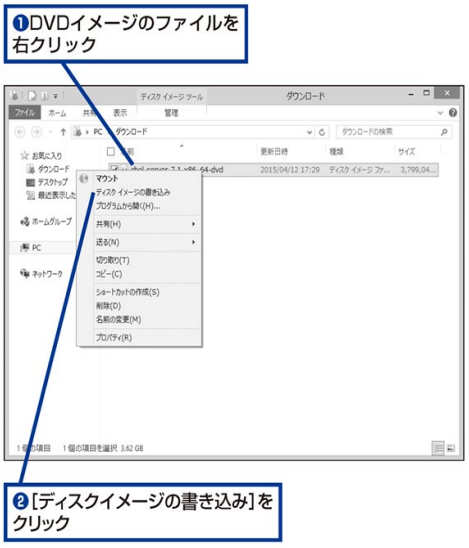
DVDやUSBメモリーの作成

このレッスンでは、ダウンロードしたインストールイメージを使って、Windows、Mac、Fedoraの環境からインストールDVDを作成する一般的な手順を解説します。

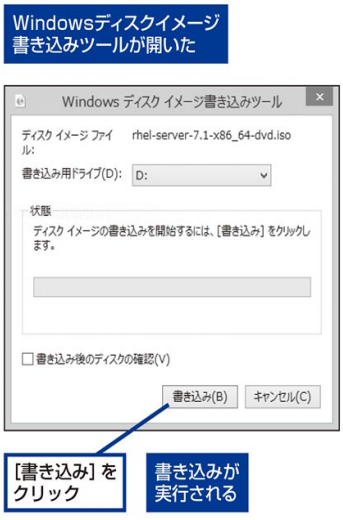
ただし、最近ではDVDドライブを搭載しないサーバーやパソコンが増えてきました。そこでインストールDVDを作ることや、サーバーでDVDからインストールするのが難しいケースもあるかと思います。そのような場合、USBメモリーにRHEL 7のインストールイメージを書き込んでインストールUSBメモリーを作成することもできます。

Windowsでインストール DVDを作成する

1 DVDイメージを選ぶ



2 書き込みを実行する

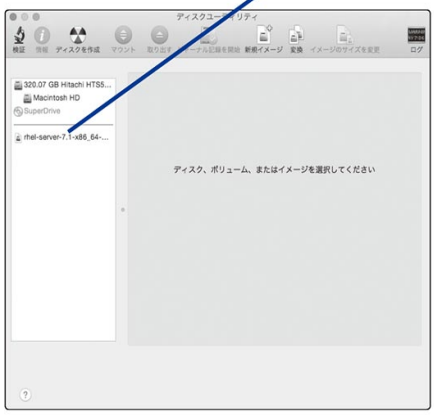


MacでインストールDVDを作成する

1 DVDイメージを指定する

❶[ディスクユーティリティ]を起動

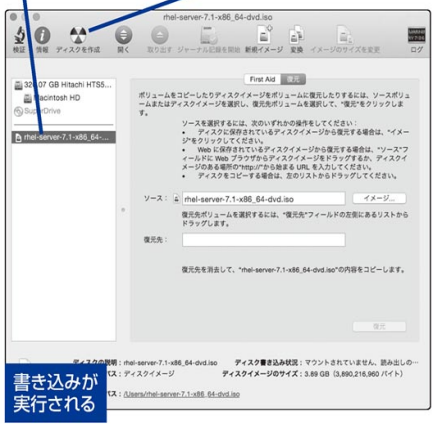
❷ウィンドウ左の一覧にDVDイメージのファイルをドラッグ&ドロップ



2 書き込みを実行する

❶DVDイメージをクリック

❷[ディスクを作成]をクリック



書き込みが実行される

LinuxでインストールDVDを作成する

1 DVDイメージを選ぶ

ここではFedora 21での手順を説明する

❶「brasero」「brasero-nautilus」のパッケージをインストール

❷DVDイメージのファイルのあるフォルダーを開く

❸DVDイメージのファイルを右クリック

❹[書き込む]をクリック

2 書き込みを実行する

ディスクに書き込む画面が開いた

❶書き込み先を確認

❷[書き込む]をクリック

書き込みが実行される

次のページに続く

インストールUSBメモリーを作成する

RHEL 7のDVDイメージをUSBメモリーに書き込むことによって、USBメモリーからインストーラーを起動することもできます。4GB以上のUSBメモリーを用意してください。

Linuxで書き込む場合は、RHEL 7のインストールイメージをddコマンドでUSBメモリーに書き込みます。

Windowsで書き込む場合は、インストールUSB作成ツールとして「Fedora LiveUSB Creator」を使うのがよいでしょう。Fedora LiveUSB Creatorをダウンロードしてインストールしたあと、DVDイメージを指定して書き込みます。

Linuxからインストールusbメモリーを作る

1 DVDイメージを書き込む

USBメモリーのデバイスが「/dev/sdb」だとする

コマンドを入力

DVDイメージのファイル名

書き込み先のデバイス名

```
[root@localhost]# dd if=rhel-server-7.1-x86_64-dvd.iso of=/dev/sdb bs=1M
```

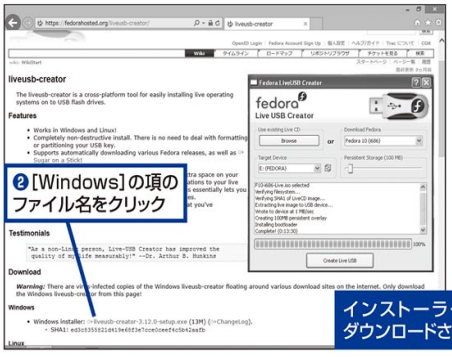
Windowsからインストールusbメモリーを作る

1 LiveUSB Creatorをダウンロードする

❶Webブラウザーで以下のURLにアクセス

LiveUSB Creatorのサイトが表示される

https://fedorahosted.org/liveusb-creator/



❷[Windows]の項のファイル名をクリック

インストーラーがダウンロードされる

2 インストールを開始する

❶ダウンロードしたインストーラーを起動

インストーラーが起動する

❷[次へ]をクリック

3 インストール先を指定する

インストール先を尋ねられる

ここでは特に変更しない

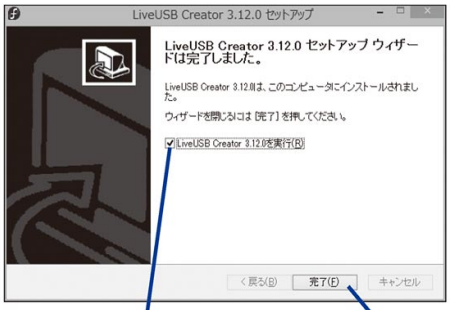
[インストール]をクリック

インストールが実行される

4 インストールが完了した

[次へ]をクリック

5 LiveUSB Creatorを起動する



❶チェックマークが付いていることを確認

❷[完了]をクリック

6 DVDイメージを書き込む

LiveUSB Creatorが起動した

❶[Browse]をクリックしてDVDイメージのファイルを選ぶ

❷USBメモリーが選ばれていることを確認

起動時にUSBメモリーを指していなかった場合は、右の再読み込みをクリックする

❸[Create Live USB]をクリック

書き込みが実行される

2-4 RHEL 7をインストールするには

サーバーへのインストール

このレッスンでは、実際にサーバーへRHEL 7をインストールする作業について解説します。インストール対象のサーバーのハードディスクは初期化されますので、もしも重要なものがあれば忘れずにバックアップをしてください。

RHEL 7では質問してくる項目が以前のバージョンと比べて最小化されています。最低限、管理者用に設定するパスワードをあらかじめ決めておいてください。インストールが済むとRHEL 7が起動します。ネットワークの設定変更やソフトウェアの追加は、インストール後に必要に応じて行っていきます。

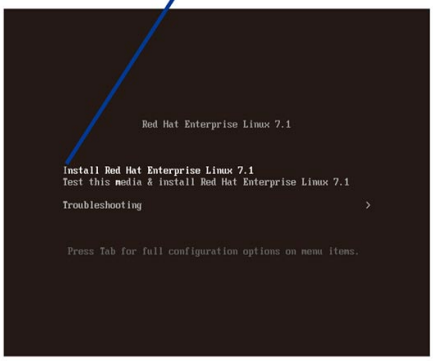
1 イントールを開始する

❶サーバーマシンに電源を入れてすぐインストールDVDをセット

インストールUSBメモリーからインストールする場合は、USBメモリーをセットしてから電源を入れる

起動画面が表示された

❷カーソルキーで[Install Red Hat Enterprise Linux 7.1]を選んでEnterキーを押す

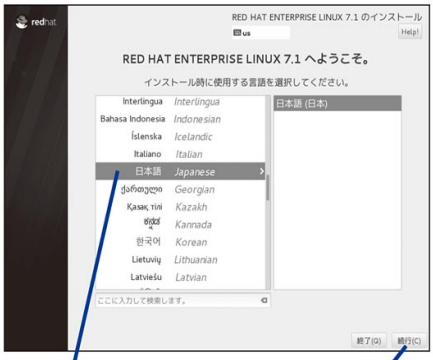


メモ DVDやUSBメモリーから起動する設定や操作は、マシンによって異なります。

注意 RHELをインストールすると、マシンに入っていたOSやアプリケーション、データはすべて失われます。

2 インストール時の言語を選ぶ

言語を選ぶ画面が表示された



❶[日本語]をクリック

❷[続行]をクリック

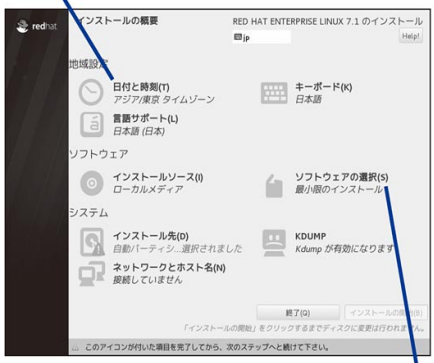
3 ソフトウェアの選択に進む

設定の画面が表示された

インストール時の設定項目はこの画面に集約されている

変更したい項目をクリックして各項目を設定する

ここではまずインストールの種類を選ぶ



[ソフトウェアの選択]をクリック

HINT!

キーボード配列に気をつけよう

読者の多くが日本語106／109キーボード配列をお使いだと思います。RHEL 7のインストール時に言語で英語を選択すると、英語101／102キーボード配列が選択されてしまいますのでインストール時にキーボードの設定を見なおしてください。*（アスタリスク）や&（アンパサンド）を入力しようとすると他の記号が入力されたり、|（パイプ）や_（アンダーバー）が入力できなくなったりといったトラブルが発生します。

4 ベース環境を選ぶ

ここではGUI付きサーバーを選ぶ

❶[サーバー（GUI使用）]をクリック

❷[完了]をクリック

5 ネットワークの設定に進む

設定の画面に戻った

続いて、ネットワークが有効になっていないため設定する

[ネットワークとホスト名]をクリック

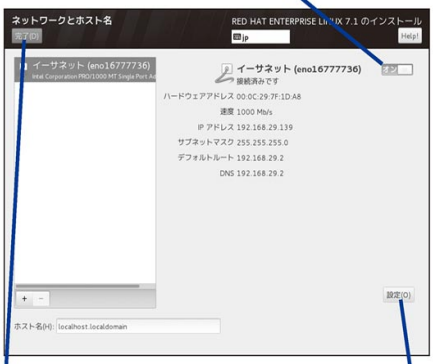
次のページに続く

6 ネットワークを有効にする

ネットワークインターフェイスが表示されている

❶ここをクリック

[オン]に表示が変わる



❷[完了]をクリック

細かい設定を変更する場合はここをクリックする

7 インストール先の設定に進む

設定の画面に戻った

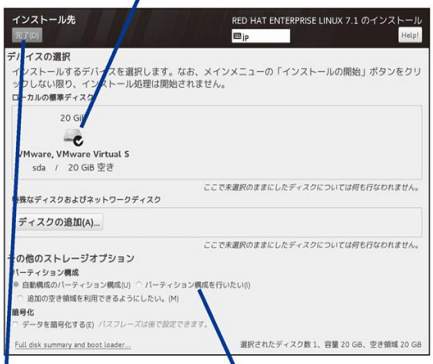
続いて、インストール先を確認する

[インストール先]をクリック

8 インストール先を確認する

❶インストール先を確認

ディスクが複数ある場合は並んで表示される



❷[完了]をクリック

パーティション構成を変更する場合はここで選ぶ

HINT!

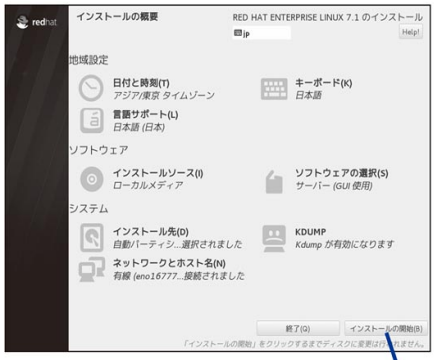
IPアドレスの重複に気をつけよう

手順❻で［設定］をクリックすると、DHCPを使わずに任意のIPアドレスを指定できます。ネットワーク上のコンピューターにはそれぞれIPアドレスが付けられます。このIPアドレスはユニークなものでなければなりません。IPアドレスを指定するときは、そのIPアドレスがネットワーク上で本当に使われていないかどうか、ネットワーク管理者に確認してください。IPアドレスが重なってしまうと、ネットワーク接続時に不思議な挙動のトラブルが発生してしまい、解決までに時間がかかります。

9 インストールを開始する

設定の画面に戻った

インストールを実行する



[インストールの開始]をクリック

HINT!

強度の強いパスワードとは？

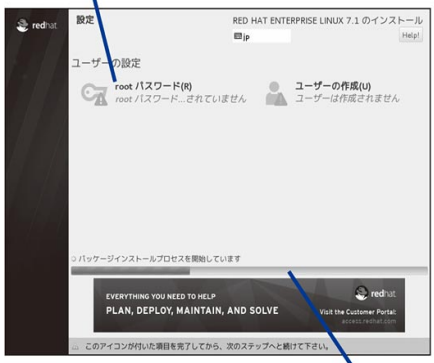
パスワードはシステムのアクセス権を証明する1つの方法です。手順⓫や手順⓭でパスワードを設定するときに、パスワードの強度が低いと警告が表示されますので、強度の強いパスワードを設定しましょう。強度の強いパスワードには、大文字／小文字／数字／記号を混ぜること、辞書に含まれている単語は使わないこと、十分に長くすることが重要になります。では何文字だとよいかというと、現在のところ、10文字以上にすれば総当りの解析に1000年以上かかると言われています。しかし、1000台使って解析すれば1年で見つかるかもしれません。悪意のあるユーザーに侵入されないためにも、定期的にパスワードを変更することも重要です。

10 rootパスワードの設定に進む

ユーザーの設定の画面が表示された

rootのパスワードを設定する

[rootパスワード]をクリック

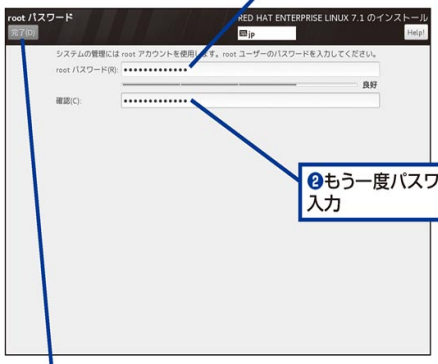


並行してインストールが進んでいる

11 rootパスワードを設定する

管理者ユーザー「root」のパスワードを設定する

❶パスワードを入力



❷もう一度パスワードを入力

❸[完了]をクリック

メモ　パスワードの強度が低い場合は警告が表示されます。その場合、もう一度[完了]をクリックしてそのまま強行することもできます。

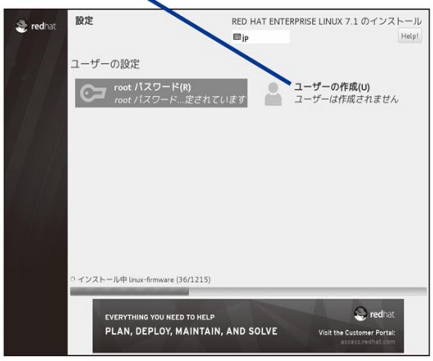
次のページに続く

12 一般ユーザーの作成に進む

ユーザーの設定の画面に戻った

続いて一般ユーザーを作成する

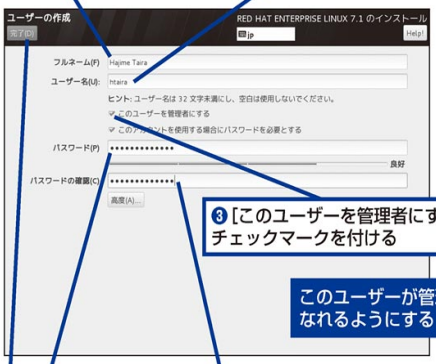
[ユーザーの作成]をクリック



13 一般ユーザーを作成する

❶フルネームを入力

❷ログインなどで使うユーザー名を入力



❸[このユーザーを管理者にする]にチェックマークを付ける

このユーザーが管理者になれるようにする

❹パスワードを入力

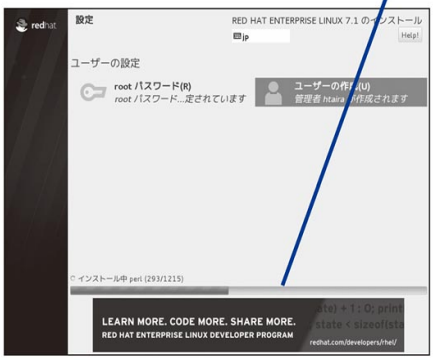
❺もう一度パスワードを入力

❻[完了]をクリック

14 インストールが完了するまで待つ

ユーザーの設定の画面に戻った

インストールが進んでいる



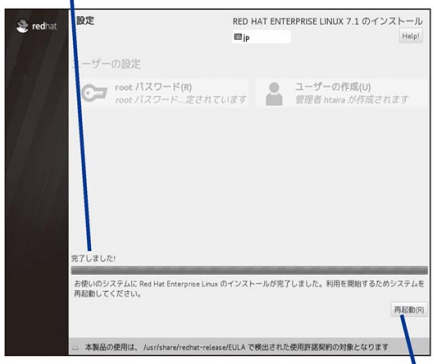
HINT!

一般ユーザーの必要性

Linuxには、管理者権限を持つrootユーザーと、一般ユーザーが存在します。rootユーザーだけでも日々の運用管理は行えますが、デスクトップ用途でGUIを利用したり、システム上で解析や計算プログラムを動かしたりするのであれば、一般ユーザーを作成して利用した方がよいでしょう。万一、rootユーザーで実行中のプログラムが暴走した場合、システムが応答しなくなるぐらいリソースを食い潰すことや、OS上の重要なシステムファイルを破壊することもあります。

15 再起動する

インストールが完了した



[再起動]をクリック

マシンが再起動する

メモ インストール用のDVDやUSBメモリーは再起動中に取り外してください。

16 起動画面が表示される

この画面では特に何もしなくてよい

HINT!

ブートローダー GRUB2

インストール完了後、再起動したあとの手順⓰で出てくる黒い画面は、RHEL 7で採用されているブートローダーの「GRUB2」です。システム上にカーネルが複数インストールされている場合は、GRUB2の画面で起動するカーネルを選択できます。また、この画面でキー操作を行うことで起動時のカーネルに渡すオプションを一時的に変更したりすることができます。

17 ライセンスの確認に進む

続きの設定の画面が表示された

ライセンスを確認する

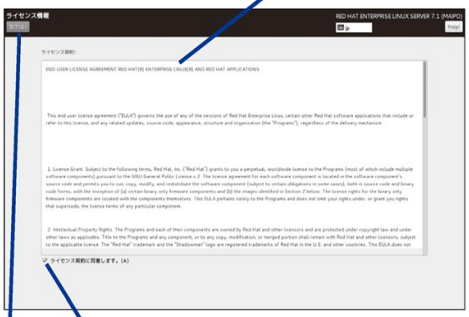
[ライセンス情報]をクリック

次のページに続く

18 エンドユーザーライセンス契約を確認する

エンドユーザーライセンス契約の条項が表示される

❶契約条項を読む



❷[ライセンス契約に同意します]にチェックマークを付ける

❸[完了]をクリック

19 設定を完了する

設定の画面に戻った

[設定の完了]をクリック

HINT!

Kdumpとは

RHEL 7にはシステムがクラッシュした場合にメモリーの内容をハードディスク上にダンプするKdumpという仕組みがあります。Kdumpが吐き出したダンプは、クラッシュダンプもしくはカーネルダンプと呼ばれています。

手順❾などでこのKdumpをきちんと設定していると、エラーメッセージが出力されないような状況で突然システムが再起動してしまうなどの難解な事象において、問題解決に役立つ場合があります。

HINT!

システムの登録とは

手順⓴では「システムの登録」について聞かれます。これは、インストールしたシステムを、Red Hatの管理サーバーにおいて購入済みサブスクリプションと紐付けする作業です。RHELのインストール後に行うこともできます。
システムを登録することにより、Red Hat CDNから更新パッケージを入手できるようになります。本書では、**第4章**において、システムの登録方法のレッスンで解説します。

20 システム登録が表示される

システムの登録の画面が表示される

ここでは登録せず、後から登録する

21 RHEL 7が起動した

ログイン画面が表示された

STEP UP

パブリッククラウドでも使えるRHEL 7

近年、コンピューターリソースの利用形態の1つとして、クラウドコンピューティングと呼ばれる形が普及しています。ハードウェアを購入してそこにシステムを構築するのではなく、クラウドプロバイダーが提供するリソースを必要なときに必要な分だけ、クラウドの利用料を支払って使います。クラウドの環境の中でも、インターネット経由でアクセスが可能なものをパブリッククラウドと呼びます。

Amazon Web ServicesやGoogleをはじめとするRed Hat認定クラウドプロバイダーでは、RHEL 7のシステムイメージがすでにインストールされた状態でサービスを利用できます。1時間あたり約8円(2015年執筆時点のAmazon EC2時間単価)で、コンピューティングリソースとRHEL 7を、インターネット経由で利用できます。

本書で説明している内容のほとんどは、パブリッククラウドの環境でも試すことができます。パブリッククラウドを活用することで、技術を習得する場合にネックとなるイニシャルコストを低減することができます。クレジットカードと少しの英語力さえあれば、手元のパソコンから、RHEL 7をすぐに試せます。

Red Hat 認定クラウドプロバイダーのリスト

https://access.redhat.com/ecosystem/search/#/category/Cloud% 20Provider

第3章 RHEL 7を使い始める

RHEL 7をサーバーで利用する場合の操作や設定はコマンド入力が基本となります。
この章では、まず、RHEL 7へGUIでログインする方法およびコマンドラインでログインする方法を説明し、そのあとに基本知識としてファイル操作を中心とした基本的なコマンドラインの使い方を説明します。

●この章の内容

3-1 Linuxの操作を始めるには

ログインとログアウト

RHEL 7へのログイン方法にはGUIからのログインとコマンドラインからのログインの2種類のログイン方法があります。ここでは、**第2章**で「サーバー（GUI利用）」でインストールしている場合を前提に、GUIでのログイン方法からご紹介します。インストーラーの途中で一般ユーザーを作成しているので、GUIログインの画面にユーザー名の一覧が表示されます。管理者ユーザー（root）でログインするには、アカウントを変更してログインします。また、コマンドラインからは、コンソールのログインプロンプトから入力してログインします。

GUIで一般ユーザーとしてログインする

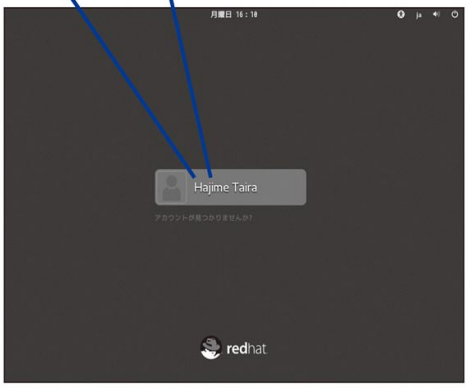
1 ユーザーを指定する

ログイン画面が表示されている

レッスン2-4手順⓭で作成した一般ユーザーとしてログインする

名前が表示されている

名前をクリック



2 パスワードを入力する

パスワードの入力画面が表示される

❶一般ユーザーのパスワードを入力

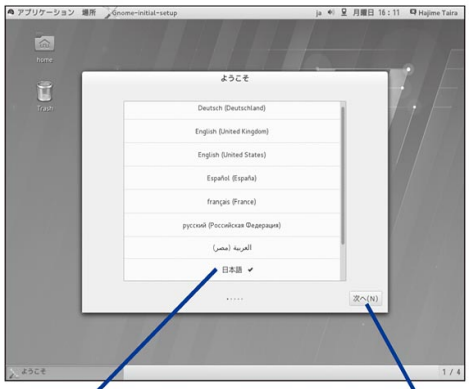
❷[サインイン]をクリック

アカウントの初期設定をする

1 言語を選ぶ

初めてログインしたときにはアカウントの初期設定画面が表示される

最初に言語の設定が表示される



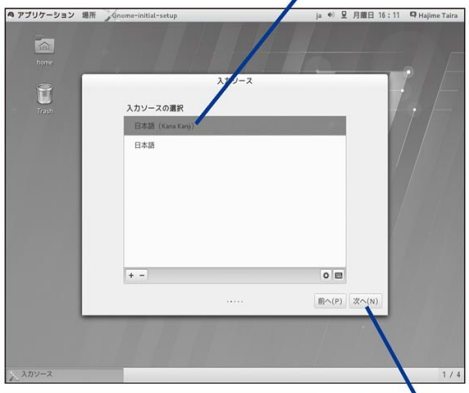
❶[日本語]をクリック

❷[次へ]をクリック

2 入力ソースを選ぶ

文字入力の設定が表示される

❶[日本語(Kana Kanji)]をクリック



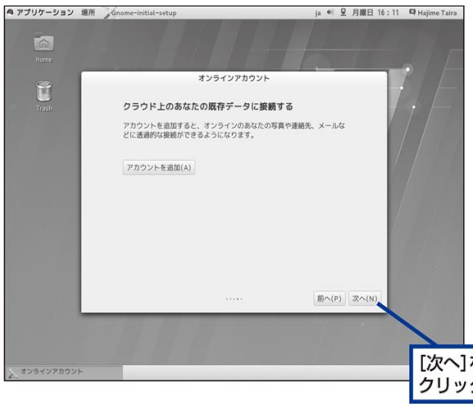
[日本語]を選ぶと、日本語変換の含まれない日本語入力が選ばれるので注意

❷[次へ]をクリック

3 オンラインアカウントをスキップする

ここでGoogleやMicrosoft Exchangeのアカウントを設定して連携できる

ここでは特に設定しない

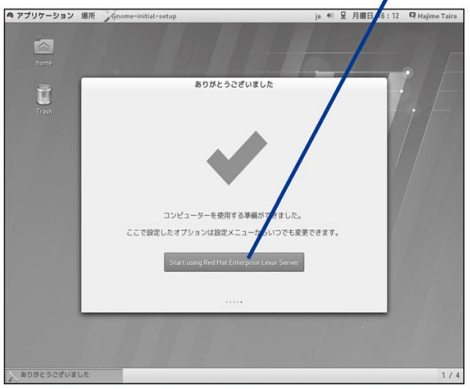


[次へ]をクリック

4 アカウントの初期設定が完了した

設定が完了した

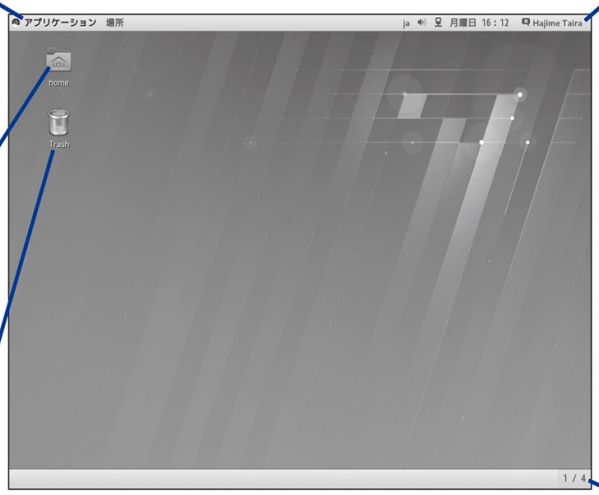
[Start using Red Hat Enterprise Linux Server]をクリック



メモ ヘルプ画面が表示されます。必要な項目を読み終えたらウィンドウ右上の[×]をクリックして閉じてください。

次のページに続く

GNOMEクラシックのGUI画面



HINT!

GNOMEクラシックとそれ以外のデスクトップ環境

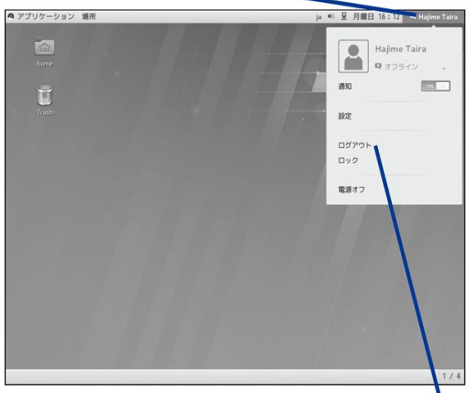
RHEL 7ではデフォルトで、GNOMEクラシックのデスクトップ環境が用意されます。GNOMEクラシックは、RHEL 6のGNOMEデスクトップ環境に似せているので、操作性が少し違うだけで馴染みやすいと思います。また、ログインのパスワード入力の画面で、[サインイン]の左にある歯車のアイコンをクリックするとメニューが出現します。ここから他のデスクトップ環境を選択することもできます。

GUIでログアウトする

1 ログアウトする

❶画面右上の名前をクリック

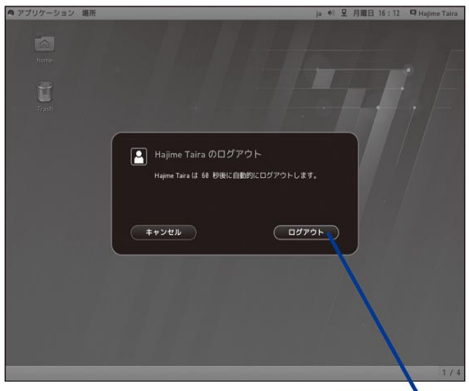
ステータスメニューが表示される



❷[ログアウト]をクリック

2 確認が表示される

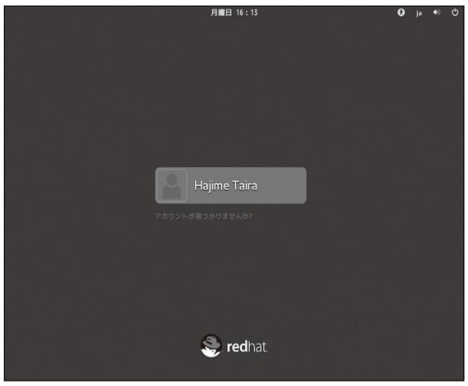
確認のメッセージが表示される



[ログアウト]をクリック

3 ログアウトした

ログイン画面に戻った



HINT!

ステータスメニューの便利機能

GNOMEの右上にあるステータスメニューは、ログアウトや電源オフの操作のときによく使います。そのほか、メッセンジャーへの状態通知や、設定画面へのショートカットなどが集まっています。[設定]という項目を選ぶと、Windowsのコントロールパネルに相当する画面が表示されます。コマンドラインを使いこなさなくてもシステムのさまざまな設定ができるので、覚えておきましょう。

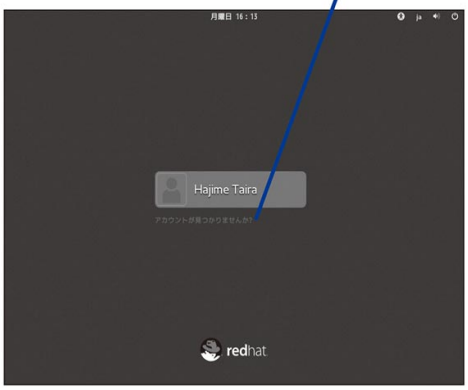
次のページに続く

GUIでrootとしてログインする

1 アカウントを切り替える

管理者ユーザー(root)としてログインする

[アカウントが見つかりませんか?]をクリック



2 ユーザー名を入力する

ユーザー名の入力画面が表示された

❶「root」と入力

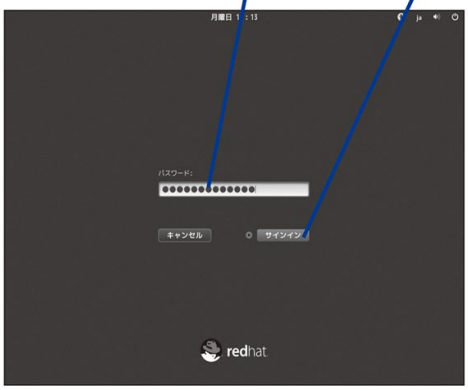
❷[次へ]をクリック

3 パスワードを入力する

パスワードの入力画面が表示される

❶rootのパスワードを入力

❷[サインイン]をクリック



一般ユーザーと同様に、初回は初期設定が表示される

デスクトップが表示される

HINT!

rootで作業する場合の注意点

rootユーザーはシステムの最大権限ユーザーです。rootユーザーはシステムを管理することもできますが、同時にシステムを壊してしまうことも簡単にできてしまいます。

特にGUIにてrootでログインする際には注意してください。rootユーザーで起動したアプリケーションが暴走した場合、システムリソースを使い果たし、OSごと巻き込んでシステムダウンする場合も考えられます。

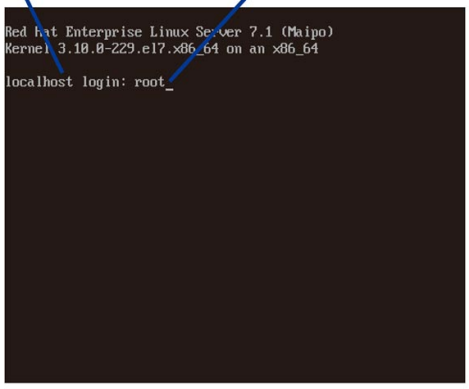
コマンドラインでログインとログアウトをする

1 ユーザー名を入力する

❶GUIのログイン画面が表示されている場合は Ctrl ＋ Alt ＋ F2 キーを押してコンソール画面に切り替える

ログインプロンプトが表示されている

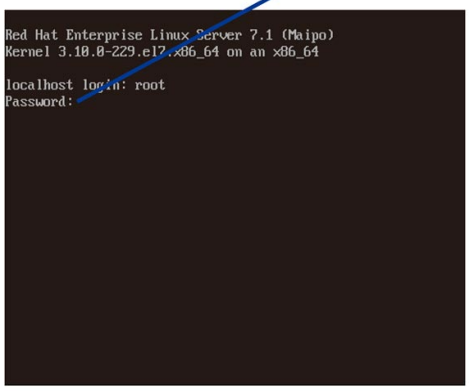
❷ユーザー名を入力して Enter キーを押す



2 パスワードを入力する

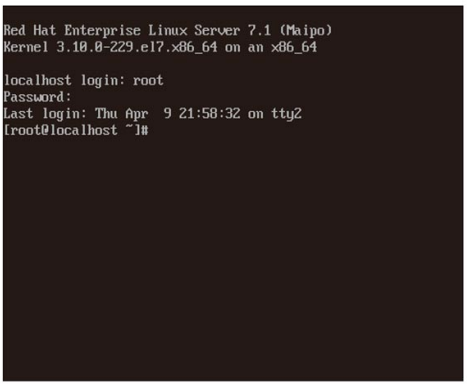
パスワードのプロンプトが表示された

パスワードを入力して Enter キーを押す



3 ログインした

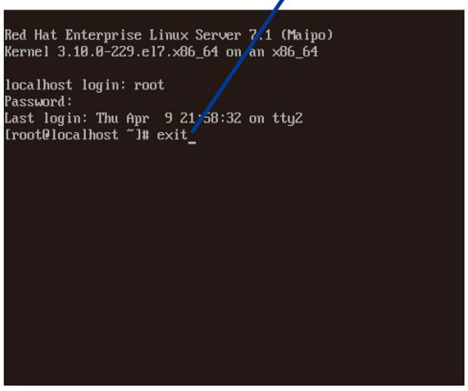
コマンドプロンプトが表示された



4 ログアウトする

ログアウトしてログインプロンプトに戻る

「exit」と入力して Enter キーを押す



ログインプロンプトが表示された状態に戻る

3-2 Linuxを終了するには

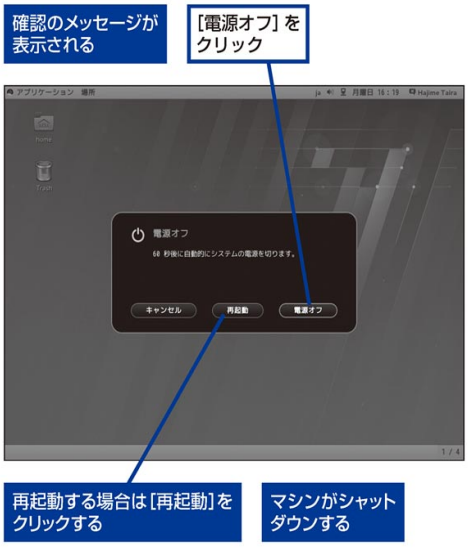
シャットダウン、再起動

サーバーは電源をこまめに切ることはありません。ただし、システムを運用しているとRHELのカーネルアップデートや計画停電などのメンテナンス時に、どうしてもシステムをシャットダウンもしくは再起動しなければならないシチュエーションがあります。計画停電で電力供給がシステム稼働中に途絶えたり、電源ボタンを押して強制的にシャットダウンしたりすると、ファイルシステムが破損する場合もあります。そのようなことが起きないように、このレッスンではシステムの正しいシャットダウン方法を解説します。

GUIからシャットダウンする

1 シャットダウンする

2 確認が表示される



HINT!

ログイン画面からもシャットダウンできる

シャットダウン作業の際に、システムをログインするまでもない場合や、ハードウェアメンテナンス時に間違えて電源が入ってしまった場合には、ログイン画面の右上をクリックしましょう。ログイン画面でもステータスメニューが表示されるので、［電源オフ］をクリックすることで、システムを安全にシャットダウンできます。

コマンドラインからシャットダウンする

1 シャットダウンする

コマンドラインでログインしておく

「systemctl poweroff」と入力してEnterキーを押す

再起動する場合は「systemctl reboot」と入力する

```
[root@localhost ~]# systemctl poweroff
```

マシンがシャットダウンする

HINT!

Ctrl＋Alt＋F1 ～ F6キーで表示される画面はそれぞれ別画面

GUIでCtrl＋Alt＋F2キーを押すと、黒いバックに白い文字でコマンドラインの画面が表示されます。F2のほかに、F3 ～ F6キーでもそれぞれ同じような独立した画面になっています。F1が元のGUI画面です。

3-3 端末を起動するには

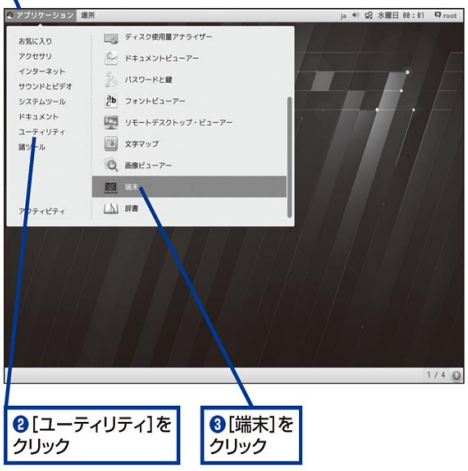
アプリケーションの起動

GUIからRHEL 7をコマンドラインで操作するには、「端末」というアプリケーションを使います。端末のウィンドウの中は仮想端末となっており、コマンドラインインターフェイスの入り口となります。

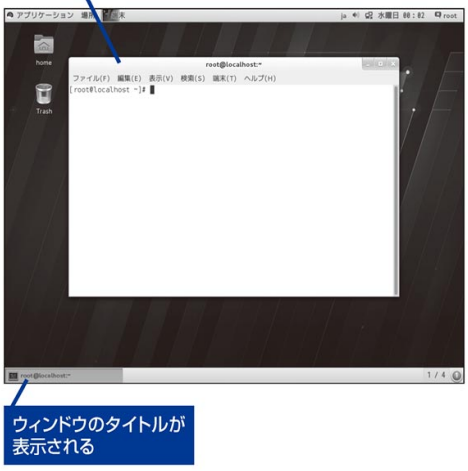
端末をはじめ、インストールされているアプリケーションは、アプリケーションメニューに一覧として表示されます。ここから目的のアプリケーションを選択することで、起動してウィンドウが開きます。GNOMEクラシックでのウィンドウの操作方法は、Windowsなどとだいたい同じです。

アプリケーションを起動する

1 メニューを開く



2 端末が起動した



HINT!

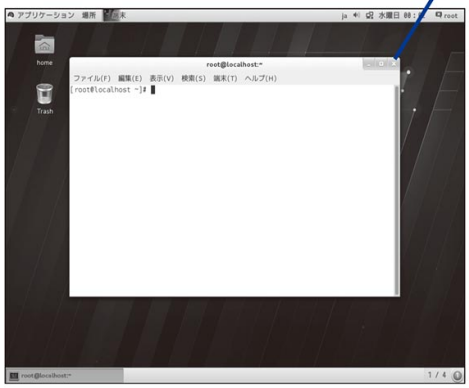
画面のロック

GUIでログインした状態で操作せずに放置しときや、ステータスメニューから［ロック］をクリックしたときには、GNOMEのロック画面のカーテンがかかります。マウスを下から上に向かってドラッグアンドドロップするか、Enterキーを押すことでカーテンを解除できます。

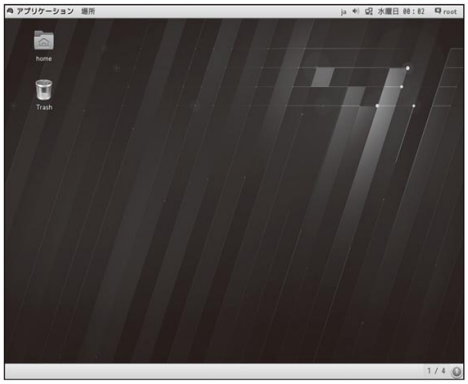
豆知識として、［ロック］のショートカットキーであるCtrl+Alt+Lキーを覚えておくとよいでしょう。サーバーから少し離れている間に不正利用されることを防ぎ、また、液晶のバックライトも消してくれるので節電効果もあります。

アプリケーションを終了する

1 端末を終了する



2 端末が終了した



3-4 コマンドラインの使い方をマスターしよう

コマンド入力の基本

RHEL 7には、GUIもありますが、サーバーの構築や管理には、主にコマンドラインインターフェイス（CUI）で作業を行います。よって、コマンドプロンプトにコマンド入力して実行し、その応答結果を見るという繰り返しです。そのため、たくさんコマンドを覚えて、的確な場面でコマンドを間違えなく打ち込めるようになりましょう。

このレッスンでは、コマンド入力の基本から解説します。最初は少し難しいかもしれませんが、慣れると段々コマンド入力するのが快くなってきます。

コマンドと引数について理解しよう

Linux上での操作はコマンドラインもしくはGUIにて行います。サーバーでは多くの場合、コマンドラインからコマンドを入力することによっての操作します。

コマンドはプログラムを呼び出すための文字列です。コマンドプロンプトが表示されているときに入力し、最後にEnterキーを押すことにより受け付けられます。

受け付けられたコマンドはプログラムを呼び出して、指示どおりの処理を行い、処理が終わった後に、次のコマンドを受け付けるためのコマンドプロンプトを再び表示します。

コマンドが呼び出したプログラムの中の処理で画面出力があった場合には、何かしら文字が表示されます。ただし、何も画面に返さずにコマンドプロンプトを返すコマンドもあります。

コマンドによっては、コマンドの後ろに「引数」という文字列を与えることができます。引数はプログラムを呼び出すときにプログラムに渡される文字列です。ファイル操作の場合には対象のファイル名が引数として渡されます。スペース区切りで複数の引数を渡すこともできます。

引数にスペースや予約文字が含まれている場合には「'」（クォート）や、「"」（ダブルクォート）で挟むことで回避できます。

引数には対象となるファイルのパスやオプションなどを入力する

引数の中にスペースが入る場合は、' や " でくくる

```
[root@localhost ~]# command 引数 引数 ...
```

コマンドと引数の間にはスペースを入れる

引数が複数あるときは、前の引数のあとにスペースを入力して続ける

HINT!

コマンドプロンプトは誰が処理するの？

ログインしたユーザーに対して対話型インターフェイスを提供し、コマンドプロンプトへ入力したプログラムの呼び出しを担当する「シェル」というプログラムがあります。

シェルにもいろいろとありますが、RHEL 7のデフォルトのシェルは「/usr/bin/bash」にあるBash（Bourne again shell）です。そのほかに、tsch、zsh、kshがRHEL 7では提供されます。

calコマンドと引数

コンソールにカレンダーを表示するcalコマンドを例に引数の例を見てみましょう。まずはcalコマンドをそのまま実行してみましょう。今月のカレンダーが表示されます。

続いて、表示する月を引数で指定してみましょう。calコマンドの後にスペースを置いて月を、さらにスペースを置いて年を指定して実行すると、その月のカレンダーが表示されます。

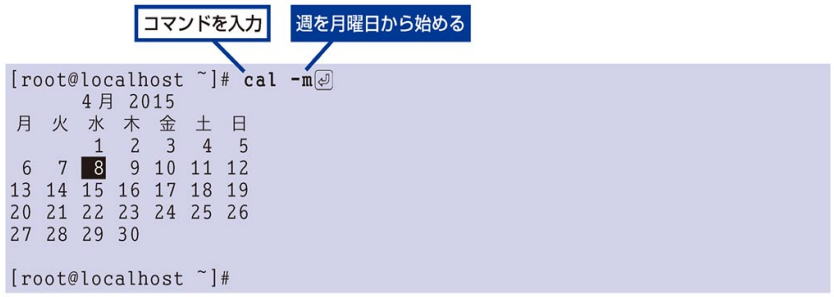
このように、コマンドに対して操作対象のような情報を指定する場合などに引数が使われます。

次のページに続く

calコマンドのオプション

コマンドの引数の中には、プログラムの動作を変えるために指定する特別な引数である「オプション」を受け付けるものもあります。オプションは呼び出すプログラムによって異なりますが、いずれも「-」（ハイフン）もしくは「--」（ハイフン2つ）で始まる文字列を引数に指定します。

calコマンドでは、日曜日から始まるカレンダーが表示されましたが、月曜日から始まるカレンダーが好みな人もいると思います。そこでcalコマンドには -mというオプションが用意されています。



HINT!

シェルの組み込みコマンド

RHEL 7 にはコマンドがいろいろとあります。そうしたコマンドの中でも、プログラムの実体が/usr/sbinや/usr/binの中に存在しない、シェルの組み込みコマンドと呼ばれるコマンドがあります。分かりやすい例としては、変数を処理するsetとexport、条件式やループを処理するif、for、whileなどがあります。コマンドプロンプトでhelpと実行すると、シェルで提供される組み込みコマンドの一覧が表示されます。

HINT!

プロンプトってなに？

プロンプト（prompt：促す）は、その名のとおり入力を促すものです。プロンプトは画面に表示される文字ですが、カーソルのある位置に何の入力を求められているかを意味しています。

HINT!

入力の編集

コマンドラインに入力した文字は、左右のカーソルキーでカーソルを動かし、文字を追加したり、Back spaceキーやDeleteキーで文字を削除したりできます

簡易ヘルプの表示

これ以外のオプションにも興味はありませんか？ ほとんどのコマンドには「--help」というオプションが用意されています。これはオプションに対する簡易ヘルプを表示してくれるものです。


```
[root@localhost ~]# cal --help↵

Usage:
 cal [options] [[[day] month] year]

オプション：
 -1, --one         show only current month (default)
 -3, --three       show previous, current and next month
 -s, --sunday      Sunday as first day of week
 -m, --monday      Monday as first day of week
 -j, --julian      output Julian dates
 -y, --year        show whole current year
 -V, --version     display version information and exit
 -h, --help        display this help text and exit
[root@localhost ~]#
```

ユリウス歴でカレンダーを表示するオプション（--julian）までありますね。

本書で紹介しているコマンドに、オプションで--helpを指定してオプションを探してみると、面白いオプションが見つかることでしょう。

オプションがたくさんあって全部覚えるのが大変そうという印象を受けるかもしれませんがご安心ください。筆者も、よく使う主要なコマンドしかオプションは覚えていません。

HINT!

入力の取り消し

Ctrl＋Uキーで、それまでに入力した文字を取り消すこともできます。コマンドラインでは使うことは少ないかもしれませんが、パスワードの入力のように入力内容が見えない場面で、間違えたかもしれないと思ってやり直すときなどに使えます。

HINT!

プログラムを中断するには

多くのプログラムは、実行中にユーザーがCtrl＋Cキーを押すと、実行を中断できます。ただし、キーで中断しないプログラムや、Ctrl＋Cキーに他の役割を割り当ててあるプログラムもあります。

3-5 ディレクトリを理解しよう

Linuxのディレクトリ

これからいくつかコマンドを紹介する前に、Linuxが持つディレクトリの概念を理解しておきましょう。

RHEL 7のディレクトリ階層は、LSBのFHSという標準規格に従っています。この取り決めでは、ディレクトリごとに、何を格納すべき場所なのか定められています。好き勝手なディレクトリにファイルを格納してしまうとシステム運用時に支障が出ます。

このレッスンでは、ディレクトリ階層とカレントディレクトリという概念について、コマンド操作を交えながら解説します。

ルートディレクトリ以下にすべて格納

多くの読者におなじみのWindowsでは、ハードディスクに「C:」や「D:」というドライブレターが振られて、そこにフォルダーやファイルが格納されています。

Linuxではフォルダーを「ディレクトリ」と呼びます。Linuxでは、ハードディスクごとのドライブレターは存在しません。「/」と表記されるルートディレクトリを起点として、すべてのディレクトリとファイルが、ルートディレクトリ以下のどこかのディレクトリに格納されています。

ディレクトリAの中にディレクトリBがある場合、AはBの「親ディレクトリ」といいます。また、BをAの「サブディレクトリ」といいます。ルートディレクトリは親ディレクトリを持たない、根っこのディレクトリです。

HINT! Linuxでは大文字と小文字が区別される

Linuxのディレクトリ名やファイル名は、大文字と小文字が区別され、別ファイルとして管理されます。たとえば /homeと/Homeは別のディレクトリとして扱われます。コマンドラインで指定するときには大文字と小文字の入力ミスに注意しましょう。

HINT! 入力補完でファイル名を楽に入力

シェルの機能として、ファイル名やディレクトリ名の途中まで入力したところで、Tabキーを押すだけで残りの文字を自動的に入力してくれる、「入力補完」という機能があります。また同じ文字で始まるものがあったときは、Tabキーを2回押すと、その文字で始まる名前の一覧を表示してくれます。

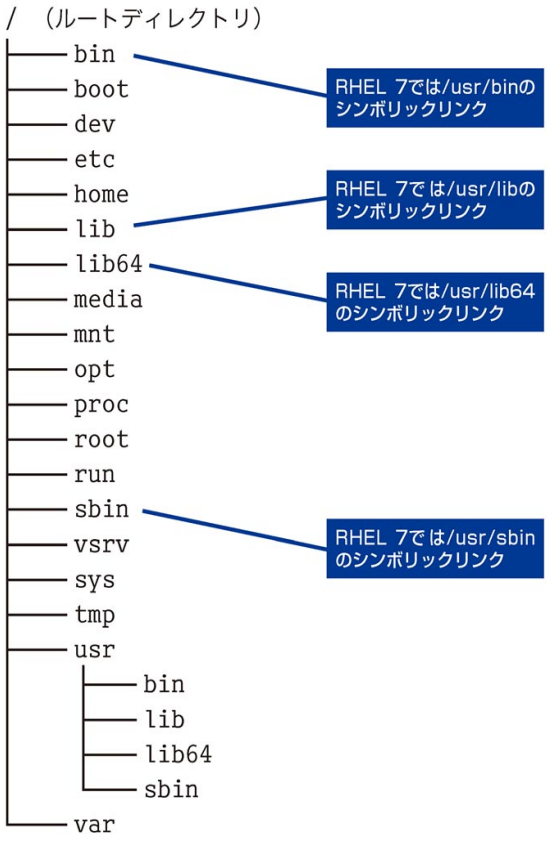
RHEL 7のディレクトリ階層

ルートディレクトリから見ると、サブディレクトリが木のように階層化されています。これをディレクトリツリーと呼びます。

一般的に使われるディレクトリ名は、どのLinuxディストリビューションでも一緒になっています。これはLSB（Linux Standard Base）のFHS（Filesystem Hierarchy Standard）という仕様で規定されています。

RHEL 7では一部の合理化されており、古いLinuxで個別に存在していたディレクトリ（/binや/sbin、/lib、/lib64）が /usr以下の同一ディレクトリに集約されています。なお、昔のファイルの位置で指定しても問題が出ないようにシンボリックリンクによって飛ばされます。

RHEL 7のディレクトリ階層



次のページに続く

一般的なディレクトリ

ディレクトリ	役割
/usr/bin	一般的なコマンドが格納されている
/usr/sbin	管理用のコマンドが格納されている
/boot	起動に必要なブートローダーやカーネルが格納されている
/etc	システムの設定ファイルが格納されている
/usr/lib、/usr/lib64	ライブラリが格納されている
/tmp	一時的なファイル格納場所

ファイルの位置を指定する「パス」

ディレクトリ名とファイル名を組み合わせたファイルの位置を示す表記を「パス」と言います。

たとえば、「/」ディレクトリから「home」→「htaira」→「Documents」と辿った場所にある「hello.txt」というファイルを、「/home/htaira/Documents/hello.txt」と表記します。このように/（ルートディレクトリ）から見たときのパスを、特別に「絶対パス」もしくは「フルパス」と呼びます。

また、/home/htairaを現在位置として、相対的に見た場合、hello.txtはDocuments→hello.txtという位置にあります。これを「./Documents/hello.txt」と表記します。このような表記を「相対パス」と呼びます。

現在位置を示すカレントディレクトリ

コマンドラインでは、現在位置を示す「カレントディレクトリ」という考え方があります。コマンドラインなどでファイル名を指定するときに、絶対パスでなかった場合には、カレントディレクトリの中にあるファイルが処理対象のファイルとなります。また、相対パスも、カレントディレクトリから見た相対位置で表現するパスの指定方法です。

カレントディレクトリを知るには

現在のカレントディレクトリを知るにはpwdコマンドを使います。pwdコマンドを実行するとカレントディレクトリがフルパスで表示されます。


```
[root@localhost ~]# pwd
/root
[root@localhost ~]#
```

「/root」と表示されましたね。これはLinuxのファイルシステム上の/rootがカレントディレクトリだということを意味します。

なお、/rootはrootユーザーのホームディレクトリでもあります。「ホームディレクトリ」はユーザーがログインしたときに最初にカレントディレクトリとなる、特別なディレクトリです。一般ユーザーの場合は「/home/ユーザー名」がホームディレクトリとなります。コマンドラインではホームディレクトリを「~」と表すこともあります。

カレントディレクトリを移動するには

カレントディレクトリを移動するにはcdコマンドを使います。cdコマンドを引数なしで実行するとホームディレクトリに移動します。


```
[root@localhost ~]# cd
[root@localhost ~]#
```

cdコマンドの引数に/tmpを指定して実行してみましょう。

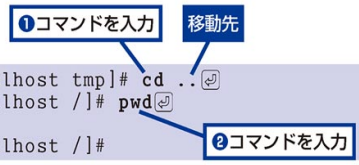

```
[root@localhost ~]# cd /tmp
[root@localhost tmp]#
```

そして、もう一度、pwdコマンドを実行します。


```
[root@localhost tmp]# pwd
/tmp
[root@localhost tmp]#
```

カレントディレクトリが/tmpに移動しました。

続いて、相対パスで「..」と指定してみましょう。




```
[root@localhost tmp]# cd ..
[root@localhost /]# pwd
/
[root@localhost /]#
```

今度は/（ルートディレクトリ）に移動しました。

3-6 ファイル名の一覧を取得するには

lsとドットファイル

ディレクトリについて学んだら、次にカレントディレクトリのファイル名の一覧を取得してみましょう。ファイル名の一覧の取得にはlsコマンドを使います。

lsコマンドはファイル名の一覧だけではなく、目的ごとに用意されたオプションを指定することで、ファイルの所有者やグループ、パーミッション、作成日時や更新日時、ファイルサイズなど、ファイルに紐付いたさまざまな情報を取得することができます。

このレッスンでは、lsコマンドの使い方とドットファイルについて解説します。

lsコマンドの使い方

ルートディレクトリにいる状態で何も指定せずに実行してみましょう。

コマンドを入力

```
[root@localhost /]# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
[root@localhost /]#
```

いくつかファイルの一覧が表示されました。

特定のディレクトリを指定する場合は、引数にディレクトリのパスを指定します。

コマンドを入力　一覧表示するディレクトリ

```
[root@localhost /]# ls /usr
bin  etc  games  include  lib  lib64  libexec  local  sbin  share  src  tmp
[root@localhost /]#
```

今度はオプション「-l」を指定してみましょう。

HINT!

ワイルドカードで複数のファイルをまとめて指定

ファイルを指定するときには、文字のかわりにワイルドカードを利用すると、複数のファイルをまとめて指定できます。「*」は、そこにどんな文字列があっても当てはまります。「?」は、そこにどんな1文字があっても当てはまります。「[」と「]」の間に複数の文字を指定すると、その中のどれか1文字と、ファイル名中の同じ位置にある1文字が同じ場合に当てはまります。


```
[root@localhost /]# ls -l
合計 32
lrwxrwxrwx.   1 root root    7  3月 15 13:21 bin -> usr/bin
dr-xr-xr-x.   3 root root 4096  3月 15 13:31 boot
drwxr-xr-x.  20 root root 3260  4月  8 20:43 dev
drwxr-xr-x. 135 root root 8192  4月  8 20:42 etc
drwxr-xr-x.   3 root root   19  3月 15 13:29 home
lrwxrwxrwx.   1 root root    7  3月 15 13:21 lib -> usr/lib
lrwxrwxrwx.   1 root root    9  3月 15 13:21 lib64 -> usr/lib64
drwxr-xr-x.   2 root root    6  3月 13  2014 media
drwxr-xr-x.   2 root root    6  3月 13  2014 mnt
drwxr-xr-x.   3 root root   15  3月 15 13:26 opt
dr-xr-xr-x. 355 root root    0  4月  8 20:42 proc
dr-xr-x---.  14 root root 4096  4月  8 20:52 root
drwxr-xr-x.  35 root root 1040  4月  8 20:44 run
lrwxrwxrwx.   1 root root    8  3月 15 13:21 sbin -> usr/sbin
drwxr-xr-x.   2 root root    6  3月 13  2014 srv
dr-xr-xr-x.  13 root root    0  4月  8 20:42 sys
drwxrwxrwt.  14 root root 4096  4月  8 20:52 tmp
drwxr-xr-x.  13 root root 4096  3月 15 13:21 usr
drwxr-xr-x.  22 root root 4096  4月  8 20:42 var
[root@localhost /]#
```

ファイルの所有者名と所有グループ名、ファイルサイズ、作成日時が表示されました。

先頭には、「drwxr-xr-x」などのような文字列も表示されています。これは**レッスン3-8**で説明するパーミッション情報（ファイルのアクセス権）です。

また、「-> usr/bin」といった矢印で表現されているファイル名もあります。こうしたファイルはシンボリックリンクといいます。シンボリックリンクについては、**レッスン3-7**で説明します。

続いて、ホームディレクトリに移動し、lsコマンドにオプション「-a」を付けて実行してみましょう。


```
[root@localhost /]# cd
[root@localhost ~]# ls -a
.              .bash_profile  .dbus            initial-setup-ks.cfg  ビデオ
..             .bashrc        .esd_auth        ダウンロード          音楽
.ICEauthority  .cache         .local           テンプレート          画像
.bash_history  .config        .tcshrc          デスクトップ          公開
.bash_logout   .cshrc         anaconda-ks.cfg  ドキュメント
[root@localhost ~]#
```

「.」（ドット）から始まるファイルがたくさんあることが分かります。これらは「ドットファイル」と呼ばれ、Linux上では隠しファイルとして扱われます。GNOMEのファイルマネージャーでも、通常ではファイル一覧にドットファイルは表示されません。

ドットファイルには、主にアプリケーションの設定がテキスト形式で格納されています。Windowsのレジストリに相当する仕組みです。また、ドットファイルの一部はディレクトリになっており、その中に設定ファイルを格納します。

3-7 ファイルの基本操作をマスターしよう

mkdir、cp、mv、rm、ln

レッスン3-6では、lsコマンドでファイルの一覧を表示しました。Linuxには、ls以外にも、ファイル操作のコマンドがたくさんあります。

このレッスンでは、より実用的なファイル操作を学びましょう。ここでは、ファイルの基本操作ができるように、Linuxで日常的によく利用する、最低限覚えて頂きたいコマンドを解説します。

具体的には、ディレクトリの作成や、ファイルのコピー／移動、シンボリックリンクなどについて解説します。それぞれ、mkdir、cp、mv、lnを使います。

mkdir：ディレクトリの作成

ディレクトリの作成にはmkdirコマンドを使います。

コマンドを入力　作成するディレクトリ

```
[root@localhost ~]# mkdir dir1
[root@localhost ~]#
```

また、階層化されたディレクトリを一度に作成したい場合には、-pオプションを付けて実行します。たとえば、カレントディレクトリに「MyProjects」というディレクトリが存在しない場合に、オプションを付けずにmkdirで「MyProjects/project1」を作成しようとすると失敗します。そこで、-pオプションを付けると、ディレクトリ「~/MyProjects」を先に作った上で、その中にディレクトリproject1を作成してくれます。

コマンドを入力　作成するディレクトリ

```
[root@localhost ~]# mkdir -p MyProjects/project1
[root@localhost ~]#
```

HINT!

履歴でコマンドを楽に再入力

一度入力したコマンドは、履歴機能を使って、再度入力できます。コマンドラインで↑キーを押すと、前に入力したコマンドを1つずつ遡れます。また、↓キーで最近入力したコマンドに1つずつ戻ります。表示されたコマンドは、そのまま実行するだけでなく、編集してから実行することもできます。今の画面だけでなく、過去にログインしたときに入力したコマンドも、ある程度まで遡れます。

cp：ファイルのコピー

ファイルのコピーにはcpコマンドを使います。

cpコマンドの1つ目の引数で対象のファイルの名前を指定し、2つ目の引数で新しいファイル名を指定します。


```
[root@localhost ~]# cp ~/.bashrc file1
[root@localhost ~]#
```

また、1つ目の引数で対象のファイルの名前を与え、2つ目の引数にディレクトリ名を与えると、指定したディレクトリの中に対してコピー元と同じ名前でファイルをコピーします。

たとえば、カレントディレクトリにあるファイルfile1をディレクトリ /tmpの中にコピーするには、次のように実行します。


```
[root@localhost ~]# cp file1 /tmp
[root@localhost ~]#
```

ディレクトリと、その中に含まれるファイルやディレクトリをまとめてコピーするには、cpコマンドに-rオプションを付けて実行します。たとえば、「/usr/share/doc/words-3.0」ディレクトリの中身をホームディレクトリの「sample」ディレクトリ以下にコピーするには、次のように実行します。


```
[root@localhost ~]# cp -r /usr/share/doc/words-3.0 ~/sample
[root@localhost ~]#
```

cpコマンドでファイルをコピーすると、デフォルトで、新しく作られたファイルの作成日時はコピーしたときのものになります。また、ほかの所有者がほかのユーザーになっているファイルをコピーすると、所有者情報が変わってしまいます。設定ファイルを編集する前にコピーして編集前のファイルをバックアップしておくときなどは、次のように「-p」オプションを付けて実行します。

```
[root@localhost ~]# cp -p ~/.bashrc file2
[root@localhost ~]#
```

次のページに続く

mv：ファイルのリネームと移動

ファイルのリネームや移動にはmvコマンドを使います。

mvコマンドの1つ目の引数で対象のファイルの名前を与え、2つ目の引数で新しいファイル名を指定することで、ファイル名をリネームします。


```
[root@localhost ~]# mv file1 new1
[root@localhost ~]#
```

また、mvコマンドを使うことで、ほかのディレクトリの中にファイルを移動することができます。たとえば、カレントディレクトリにあるファイルnew1をディレクトリdir1へ移動するには、次のように実行します。


```
[root@localhost ~]# mv new1 dir1
[root@localhost ~]#
```

rm：ファイルの削除

ファイルを削除するにはrmコマンドを使います。


```
[root@localhost ~]# rm dir1/new1
[root@localhost ~]#
```

rmdir：ディレクトリの削除

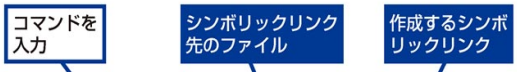
ディレクトリを削除するには、rmdirコマンドを使います。このとき、対象のディレクトリの中はファイルが1つもない状態になっている必要があります。


```
[root@localhost ~]# rmdir dir1
[root@localhost ~]#
```

ln：シンボリックリンクの作成

Windowsのショートカットファイルやシンボリックリンクのような仕組みとして、Linuxには「シンボリックリンク」という仕組みがあります。

このシンボリックリンクを作成するには、lnコマンドを使います。たとえば、ホームディレクトリの中に、/usr/share/doc/HTMLディレクトリに対するシンボリックリンク「document」を作成する場合には、次のようにlnコマンドに-sオプションを付けて実行します。




```
[root@localhost ~]# ln -s /usr/share/doc/HTML ~/document
[root@localhost ~]#
```

シンボリックリンクを作ることで、長いファイルパスの場所にあるファイルを簡単に指定できるようになります。また、特定のファイルシステム上に格納されているデータが大きくて容量を圧迫している場合に、他のファイルシステム上にデータを移動して、元の場所からそのデータに対してシンボリックリンクを張ることもできます。これによって、アプリケーションを書き換えることなく、あたかもデータの実体が元の場所にあるように扱えます。

シンボリックリンクと類似の仕組みとして、同じファイルシステム内で対象のファイルの実体に対して別のファイル名を定義する「ハードリンク」という仕組みもあります。シンボリックリンクはファイルシステムをまたいでリンクできるという点にハードリンク大きな違いがあります。ハードリンクもlnコマンドで作れますが、最近ではあまり使う機会はありません。

HINT!

リダイレクトでファイルと入出力

Linuxでは、通常使う入出力が「標準入力」と「標準出力」と名付けられています。多くのコマンドは、標準入力と標準出力を利用してデータの受け渡しをしています。通常は、標準入力はキーボード、標準出力は画面に設定されています。これらをファイルに切り替えることを「リダイレクト」といいます。次のような指定をコマンドの後ろに付けることで、リダイレクトを指定してコマンドを実行できます。

< ファイル名	ファイルから入力
> ファイル名	ファイルに出力
>> ファイル名	ファイルの最後に追加

HINT!

パイプでコマンドをつなげる

コマンドの標準出力を別のコマンドの標準入力に流し込む「パイプ」という仕組みもあります。パイプは、コマンドとコマンドを「|」という記号でつないで指定します。パイプを使えば、長い出力から必要な場所だけを抜き出したりといったことができます。パイプを使うと、コマンドをいくつでもつなげられます。

コマンド1 | コマンド2

3-8 ファイルのアクセス制御を理解しよう

オーナーとパーミッション

Linux上では、データが格納された一般的なファイルはもちろんのこと、ディレクトリや、ハードディスクなどのブロックデバイス、キーボードやマウスなどのキャラクターデバイスもすべてファイルとして扱います。

また、ファイルにはオーナー（所有権）とパーミッション（アクセス権限）という属性情報があります。これによって、ファイルの読み込みや書き込みができるかどうかをユーザーごとに指定できます。

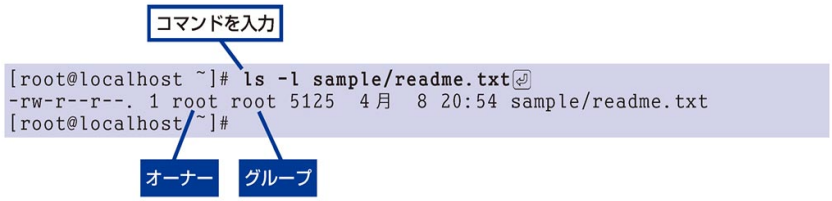
ここでは、ファイルのオーナーやパーミッションについて学びます。

ファイルには所有権がある

Linuxは元々、複数のユーザーでシステムを利用することを意識して作られているOSです。各ユーザーにはUID（ユーザー ID）が割り当てられています。

また、各ユーザーは1つ以上、最低限ユーザー名と同じ名前のグループに所属します。各グループにはGID（グループID）が割り当てられています。

Linux上のそれぞれのファイルには、オーナーと呼ばれる所有権が付与されています。具体的には、ファイルを作成すると、ファイルの作成者のUIDとGIDが付与されます。オーナーとグループは、ls -lコマンドで表示されます。



RHEL 7のインストール時にファイルシステム上に展開されたアプリケーションや設定ファイルの多くは、管理者ユーザーのroot（UID：0）がオーナーとなっています。これによって、一般ユーザーの誤操作などによってシステムが壊れることを防いでいます。

また、サーバーで動くプログラム（サービス）がユーザーとしてUIDとGIDを持っていることがあります。特定のサービスが利用する設定ファイルやデータファイルでは、そのサービスがオーナーになっている場合もあります。

一方、/homeディレクトリの中にある一般ユーザーのデータファイルについては、各ユーザーがオーナーとなっています。

chown：ファイルのオーナーを変更する

ファイルのオーナーやグループを変更するコマンドとして、chownコマンドがあります。主にrootユーザーが管理のために実行します。

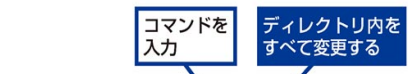
たとえば、「~/sample/readme.txt」というファイルのオーナーを、「htaira」ユーザーと「htaira」グループへ変更する場合は、次のように実行します。


```
[root@localhost ~]# chown htaira:htaira ~/sample/readme.txt
[root@localhost ~]#
```

オーナーが変更されたことをls -lコマンドで確認してみます。

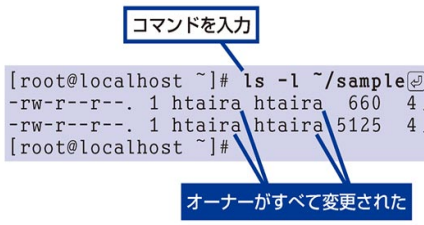

```
[root@localhost ~]# ls -l ~/sample/readme.txt
-rw-r--r--. 1 htaira htaira 5125  4月  8 20:54 sample/readme.txt
[root@localhost ~]#
```

特定のディレクトリの中のすべてのファイルのオーナーを一括で変更する場合には、「-R」オプションを指定します。次のコマンドを実行すると、~/sample自身を含む、ディレクトリ内のすべてのファイルやディレクトリのオーナーがhtairaユーザーとhtairaグループに変更されます。




```
[root@localhost ~]# chown -R htaira:htaira ~/sample
[root@localhost ~]#
```

再びls -lコマンドで確認してみます。




```
[root@localhost ~]# ls -l ~/sample
-rw-r--r--. 1 htaira htaira  660  4月  8 20:54 license.txt
-rw-r--r--. 1 htaira htaira 5125  4月  8 20:54 readme.txt
[root@localhost ~]#
```

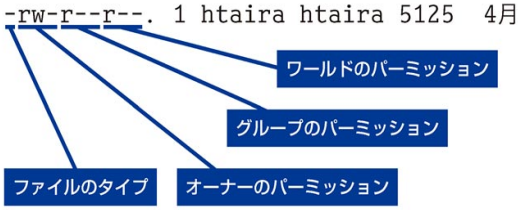
次のページに続く

パーミッションは対象ごとに設定する

ファイルのアクセス権は、オーナー（ユーザー）、グループ、ワールド（その他のユーザー）の3種類に対して設定できます。それぞれに対して、読み込み（Read）、書き込み（Write）、実行（eXecute）のアクセス制御ができます。これをLinuxでは「パーミッション」と呼びます。

ls -lコマンドを実行したときに行の先頭に表示される「-rw-r--r--」や「drwxrwxr-x」といった記号のような表記が、ファイルのパーミッションを表しています。

```
-rw-r--r--. 1 htaira htaira 5125  4月  8 20:54 readme.txt
```



先頭の1文字は、ファイルのタイプを意味しています。よく使われるファイルのタイプには、次のものがあります。

- …… 一般のファイル
d …… ディレクトリ
b …… ブロックデバイスファイル
c …… キャラクターデバイスファイル
l …… シンボリックリンク

続く文字は3文字ごとに、それぞれオーナー、グループ、ワールドのパーミッションを意味します。3文字は順に、「r」「w」「x」の文字、またはそれぞれ許可されていないことを意味する「-」が入ります。

各文字の意味は次のとおりです。

r …… 読み込みが許可されている
w …… 書き込みが許可されている
x …… 実行が許可されている

つまり「-rw-r--r--」は、一般のファイルであり、オーナーのみ読み書きが許可されており、その他のユーザーは読むことのみ許可されているというパーミッションです。

同じように「drwxrwxr-x」は、ディレクトリであり、オーナーもしくはグループが一緒のユーザーは読み書きが許可されており、その他のユーザーは読むことのみ許可されているというパーミッションです。

chmod：ファイルのパーミッションを変更する

パーミッションの変更はchmodコマンドで行います。

chmodコマンドでは、パーミッションを表現する表記方法がいくつかあります。まず、ファイルに付与する権限やファイルから抜く権限を指定する方法です。たとえば、「file1」に対して、オーナーおよびグループの書き込み権限を付与する場合には、chmodコマンドで次のように実行します。


```
[root@localhost ~]# chmod +w file1
[root@localhost ~]#
```

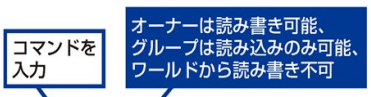
さらに、対象としてオーナーやグループ、ワールドを指定することもできます。たとえば、「file2」に対して、オーナー（u）およびグループ（g）から書き込み権限を抜いて、ワールド（o）から読み込み権限と書き込み権限の両方を抜く場合には、chmodコマンドで次のように実行します。


```
[root@localhost ~]# chmod u-w,g-w,o-rw file2
[root@localhost ~]#
```

また、ファイルのパーミッションは、「775」や「640」といった8進数の3桁の数字で指定することもあります。各桁の数字は以下のように、r（4）、w（2）、x（1）の組み合わせをビット演算した結果で表現されます。これを上の桁から、オーナー、グループ、ワールドの順で3桁に並べます。

```
 0    1    2    3    4    5    6    7
---  --x  -w-  -wx  r--  r-x  rw-  rwx
```

たとえば、640であれば「rw-r-----」を、775であれば「rwxrwxr-x」を意味します。つまり、「file3」に対して、オーナーは読み書き可能、グループは読み込みのみ可能、ワールドから読み書き不可としたい場合には、次のようにパーミッション「640」を設定します。




```
[root@localhost ~]# chmod 640 file3
[root@localhost ~]#
```

3-9 ファイルの圧縮や展開をするには

tar

ファイル操作の続きとして、このレッスンでは、バックアップにも使えるtarコマンドについて学んでいきます。

Linux環境では、複数のファイルをまとめるアーカイブ形式として、tar形式が多く使われます。tar形式はアプリケーションのソースコードやログファイルなどをまとめるときなどに、よく使われています。

tarコマンドは、元々、テープメディアにファイルを格納するコマンドとして使われていました。現在では、tarファイルへのバックアップやアーカイブに使われています。

tarでアーカイブを作成する

Linux環境においてファイルやディレクトリをまとめて格納するアーカイブ形式として、tar（tape archiver）形式がよく使われます。tar形式のファイルの拡張子は、圧縮しない場合は.tar、圧縮する場合は圧縮形式に応じて.tar.gz、.tar.bz2、.tar.xzなどが使われます。

tar形式のアーカイブは、tarコマンドによって作成します。最初の引数であるオプションにより動作を指定します。tar.gz形式のファイルを作成する場合は、オプションとして「zcvf」を指定します。「z」は「.gz」形式の圧縮を、「c」はアーカイブの作成（Create）を、「v」は経過の表示を、「f」は続く引数で指定したファイルを対象にすることを意味します。引数の最後に、アーカイブするファイルやディレクトリを指定します。

たとえば、システムの設定ファイルが置かれるディレクトリ「/etc」の内容をアーカイブ「etc.tar.gz」に格納する場合は、次のようにtarコマンドを実行します。

コマンドを入力　作成するアーカイブ　アーカイブの対象

```
[root@localhost ~]# tar zcvf etc.tar.gz /etc
tar: メンバ名から先頭の '/' を取り除きます
/etc/
/etc/fstab
/etc/resolv.conf
/etc/aliases.db
[root@localhost ~]#
```

tarでアーカイブを展開する

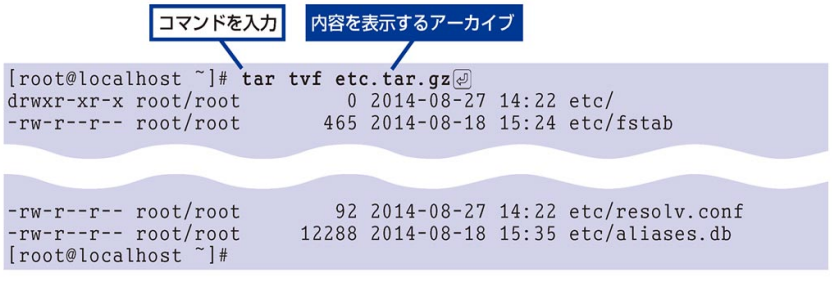
tarによるアーカイブを展開する場合は、オプションとして「xvf」を指定します。「x」はアーカイブの展開（eXtract）を意味します。以前のtarコマンドは、展開するときにも圧縮方式を指定しなければなりませんでしたが、RHEL 7に含まれるtarコマンドは圧縮の有無や圧縮方式を自動検出して展開してくれます。

たとえば、さきほど作成したアーカイブファイル「etc.tar.gz」をカレントディレクトリに展開する場合は、次のようにtarコマンドを実行します。

これにより、カレントディレクトリにetcというディレクトリが作られ、その中にファイルが展開されます。アーカイブを作成したときに「メンバ名から先頭の '/' を取り除きます」と警告が表示されたように、作成時に絶対パスで指定したファイルは相対パスとして格納されています。

アーカイブ内のファイルを一覧表示する

アーカイブに格納されたファイルの一覧を表示する場合は、オプションとして「tvf」を指定します。「t」はアーカイブ内容の一覧表示（lisT）を意味します。たとえば、「etc.tar.gz」のファイル一覧を表示場合は、次のようにtarコマンドを実行します。



tarコマンドのオプションは非常に多いため、困ったときには「man tar」コマンドを実行してマニュアルを参照してください。

3-10 外部メディアを使うには

mount、umount、eject

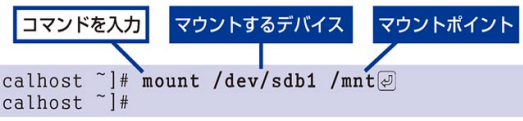
WindowsではUSBメモリーやDVD-ROMなどの外部メディアを、内蔵ディスクとは別のドライブとして扱い、「E:」などのドライブレターで識別します。それに対してLinuxでは、すべてのディレクトリとファイルが、ルートディレクトリ以下のどこかに置かれます。

そこでLinuxでは、外部メディアを使うには、ルートディレクトリ以下のディレクトリ構造の中につなげる必要があります。これをマウントといいます。マウントの方法は、GUI環境と、サーバーなどのGUIを使わないコマンドラインの環境とで異なります。

mount：外部メディアをマウントする

LinuxでUSBメモリーやCD/DVDなどの外部メディアを使うには、システム上で認識されている外部メディア（ブロックデバイス）を、システム上のどこかのディレクトリにマウントする必要があります。

コマンドラインから外部メディアをマウントするには、mountコマンドを使います。mountコマンドの引数で、マウントするブロックデバイス名と、マウント先のディレクトリ名（マウントポイント）を指定します。なお、mountコマンドはrootユーザーで実行します。




```
[root@localhost ~]# mount /dev/sdb1 /mnt
[root@localhost ~]#
```

SATAやSASのドライブや、USBメモリーなどのデバイスは、システムで認識された順に、/dev/sdaや/dev/sdbというブロックデバイス名が割り当てられます。さらに、/dev/sdbの中のパーティションには、順に/dev/sdb1や/dev/sdb2という別のブロックデバイス名が割り当てられます。

また、CD/DVDドライブなどの光学ドライブには、/dev/sr0や/dev/sr1というブロックデバイス名が割り当てられます。光学ドライブを使う場合には、戸惑わないようにしてください。なお、/dev/sr0はシンボリックリンクにより/dev/cdromという名前でもアクセスできるようになっています。

umount：アンマウントする

使いおわった外部メディアをシステム上から外す（アンマウントする）場合には、umountコマンドを実行します。引数として、マウントポイントまたはブロックデバイス名を指定します。コマンド名が、「unmount」ではなく「umount」なので間違えないようにしましょう。なお、umountコマンドもrootユーザーで実行します。


```
[root@localhost ~]# umount /mnt
[root@localhost ~]#
```

eject：外部メディアを取り出す

ejectコマンドを使うと、CD/DVDドライブなどの中に入っているメディアを取り出せます。ドライブがマウントされている場合は、アンマウントされてから取り出されます。フロッピーディスクドライブなどに対しても実行できます。

システムに搭載されている光学ドライブが1つの場合、ejectコマンドを実行すると、光学ドライブにイジェクト命令が送られてメディアが出てきます。


```
[root@localhost ~]# eject
[root@localhost ~]#
```

システムに複数の光学ドライブが搭載されている場合には、対象のドライブのブロックデバイス名を引数で指定します。

また、光学ドライブが対応していれば、-tオプションでトレイを閉めることもできます。


```
[root@localhost ~]# eject -t
[root@localhost ~]#
```

GUIでは自動的にマウントされる

GUIからログインしている状態では、USBメモリーやDVD-ROMなどをマシンに入れると、自動的にマウントされます。取り出すためのボタンも表示されます。

STEP UP

マニュアルを読もう

RHEL 7では、システムにインストールされているマニュアルや、Red Hat Customer Portalから参照できる公式ドキュメントが充実しています。最大限活用しましょう。

特にマニュアルを参照するmanコマンドは覚えておくとよいでしょう。使い方は以下のとおりです。

man［オプション］［セクション］ページ

たとえば、lsコマンドの使い方のページを参照するには、次のように実行します。

```
[root@localhost ~]# man ls
```

manコマンドでページが分からなくても、-kオプションを指定してキーワードを指定すれば、キーワードが含まれるページの一覧が表示されます。たとえば、プロセス管理のコマンドを調べたくて、「process」がキーワードに含まれるページを参照するには、次のように実行します。

```
[root@localhost ~]# man -k process
```

その他、manにはセクションという考え方があり、コマンドのページを調べるのか、設定ファイルのページを調べるのか、明示的に指定できます。よく指定するものとしては、セクション1は一般利用者コマンドの説明、セクション5はファイルの説明、セクション8は管理者向けコマンドの説明です。

```
[root@localhost ~]# man 1 hostname
[root@localhost ~]# man 5 hostname
```

そのほか、Red Hatが提供する製品ドキュメントは、RHELを管理する上で非常に役立ちます。日々のシステム管理で分からないことがあった場合に、ぜひ目を通してください。

Customer Portal製品ドキュメント

https://docs.redhat.com/

第4章 ネットワークを準備する

RHEL 7をサーバーで利用するには、ネットワークの設定が必要です。この章では、RHEL 7のネットワークインターフェイスの設定を行う方法を説明し、そのあとにネットワークの各種状態を確認するためのコマンドをご紹介します。そして、最後に最新版のソフトウェアアップデートを適用する方法を説明します。

●この章の内容

4-1 ネットワークインターフェイスの命名ルールを知ろう

ネットワークインターフェイス

RHEL 6までや多くのLinuxディストリビューションでは、ネットワークインターフェイスに「eth0」や「eth1」といった名前が付けられます。それに対してRHEL 7からは、ネットワークインターフェイスのデフォルトの命名ルールとして「Predictable Network Interface Names」が採用されました。また、Dell社のサーバーに限り、「biosdevname」という命名ルールが適用されます。

従来のeth0やeth1といった命名ルールは、Linux KVMの仮想化環境では残りますが、目にすることは減るでしょう。

Predictable Network Interface Namesのルール

Predictable Network Interface Namesは、systemdが提供する命名ルールの仕組みです。イーサーネットデバイスは「en」から始まり、無線LANカードは「wl」から始まります。

Predictable Network Interface Namesのルール

名前の例	条件
eno1	ファームウェアやBIOSが提供してくれたハードウェア構成情報があり、オンボードNICだと判別できている場合の命名ルール
ens1	ファームウェアやBIOSが提供してくれたハードウェア構成情報があり、PCI Expressスロット番号が判別できている場合の命名ルール
enp2s0	ファームウェアやBIOSが提供してくれたハードウェア構成情報がなく、物理的なロケーションしか分からない場合の命名ルール
enx78e7d1ea46da	ネットワークインターフェイスのMACアドレスに基づく命名ルール
eth0	ファームウェアやBIOSが提供してくれたハードウェア構成情報がなく、ロケーションが特定できない場合の命名ルール

HINT!

命名ルールを変更するには

Predictable Network Interface Namesを無効にしたい場合は、カーネルオプションで「net.ifnames=0」を指定して再起動します。また、biosdevnameを無効にしたい場合は、カーネルオプションで「biosdevname=0」を指定して再起動します。
Predictable Network Interface Namesとbiosdevnameの両方を無効化すると、RHEL 6までのネットワークインターフェイスと同じ、eth0やeth1などの命名ルールに戻せます。
カーネルオプションは起動時にも指定できますが、ネットワークインターフェイスの名前を変更する場合は、毎回同じカーネルオプションで起動する必要があります。そのためには、GRUB2の設定を変更します。GRUB2の設定については、**第17章**で解説します。

biosdevnameのルール

biosdevnameは、RHEL 6の頃からDell社のサーバーに限って適用されていました。Predictable Network Interface Namesと似たような仕組みですが、命名ルールが少しシンプルです。

biosdevnameのルール

名前の例	条件
em1	ファームウェアやBIOSが提供してくれたハードウェア構成情報があり、オンボードNICだと判別できている場合の命名ルール
p1p1	ファームウェアやBIOSが提供してくれたハードウェア構成情報があり、PCI Expressスロット番号が判別できている場合の命名ルール
eth0	ファームウェアやBIOSが提供してくれたハードウェア構成情報がない場合

4-2 ネットワークを設定するには

nmtui、nm-connection-editor

RHEL 7のネットワークは、NetworkManagerというソフトウェアで管理されています。以前から存在するネットワークインターフェイスの設定ファイルも引き続き使えますが、ネットワークの設定は、NetworkManagerを通して行うのがよいでしょう。

このレッスンでは、NetworkManagerによりネットワークを簡単に設定するツールを解説し、固定IPアドレスを設定します。RHEL 7では、端末などのキャラクター画面向けにnmtui、GUI向けにnm-connection-editorというツールが用意されています。

本書のネットワーク構成

本書では、下の表のようなネットワーク構成を前提に、各種サーバーを構築していきます。RHEL 7のインストール時に、IPアドレスをDHCPにて自動取得する設定になっているものとして、固定IPアドレスの設定に変更する方法を解説します。実際の設定では、ご自身のネットワーク構成に置き換えながら読み進めてください。

なお、外部向けサーバーでは、別途ルーターやファイアウォールなどネットワーク機器の設定も必要になる場合がありますが、本書では解説しません。ルーターやファイアウォールのマニュアルを参考に適切な設定を行ってください。

本書のネットワーク構成

設定項目	設定内容
ホスト名	host1.dekiru.gr.jp
ネットワークインターフェイス名	eno1
IPアドレス	192.168.0.1
ネットマスク	255.255.255.0
ゲートウェイ	192.168.0.254
DNSサーバー	192.168.0.254

キャラクター画面で設定する場合

1 nmtuiを起動する

システムのネットワーク設定はroot権限で実行する

ここでは端末から実行してみる

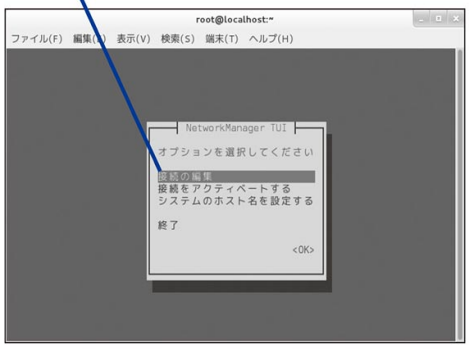
「nmtui」と入力してEnterキーを押す

2 設定の変更を選ぶ

キャラクターでダイアログが表示された

項目間はカーソルキーで移動する

[接続の編集]を選んでEnterキーを押す



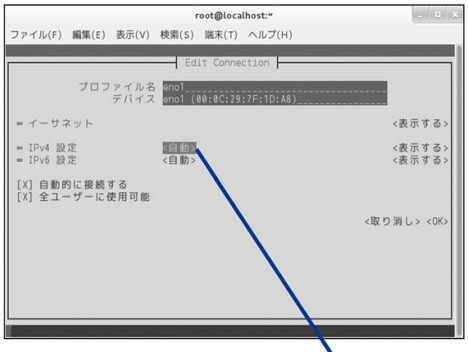
3 ネットワークインターフェイスを選ぶ

ネットワークインターフェイスの一覧が表示された

設定するネットワークインターフェイスを選んでEnterキーを押す

4 IPv4の設定を開始する

接続の設定画面が表示された



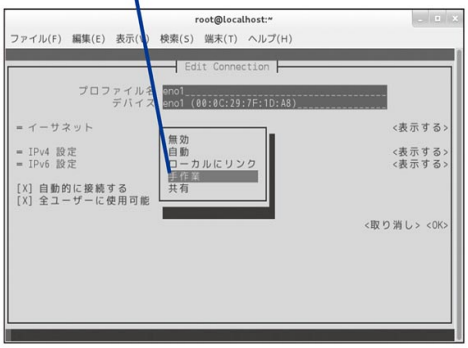
[IPv4 設定]の[自動]を選んでEnterキーを押す

次のページに続く

5 手動設定を選ぶ

メニューが表示された

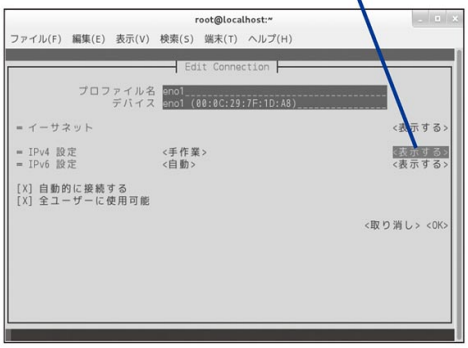
[手作業]を選んでEnterキーを押す



6 設定の変更を選ぶ

[手作業]に変わった

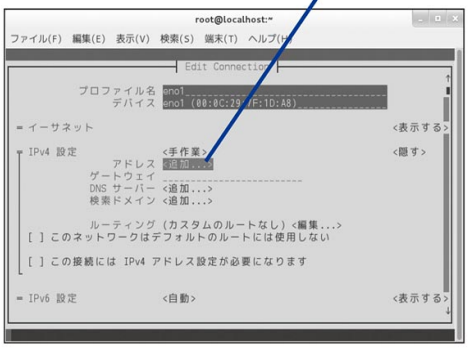
[IPv4 設定]の[表示する]を選んでEnterキーを押す



7 IPアドレスを追加する

設定項目が表示された

[アドレス]の[追加]を選んでEnterキーを押す



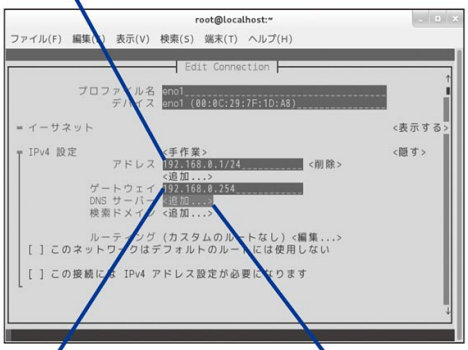
8 IPアドレスを入力する

ここでは「本書のネットワーク構成」の例から入力する

自分の環境に合わせて入力する

❶設定するIPアドレスとネットマスクを入力

IPアドレスのあとに「/」で区切ってネットマスクを指定する

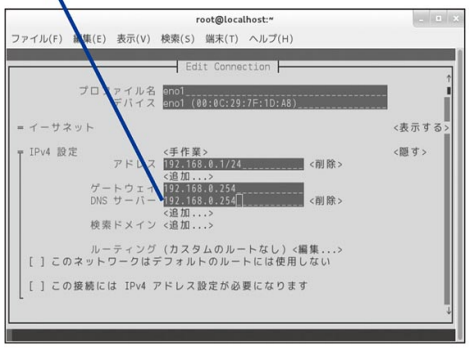


❷ゲートウェイのIPアドレスを入力

❸[DNS サーバー]の[追加]を選んでEnterキーを押す

9 DNSサーバーを指定する

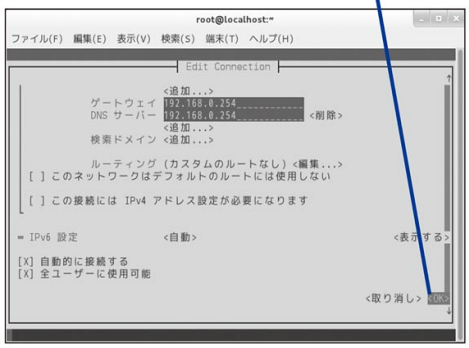
DNSサーバーのIPアドレスを入力



10 設定を完了する

カーソルキーで下の項目を選んでいくと画面がスクロールする

❶[OK]を選んでEnterキーを押す



ネットワークインターフェイスを選ぶ画面に戻る

❷[終了]を選んでEnterキーを押す

nmtuiが終了する

❸システムを再起動する

GUIで設定する場合

1 nm-connection-editorを起動する

rootユーザーでログインしておく

「nm-connection-editor」と入力してEnterキーを押す

2 ネットワークインターフェイスを選ぶ

ネットワークインターフェイスの一覧が表示された

❶設定するネットワークインターフェイスをクリック

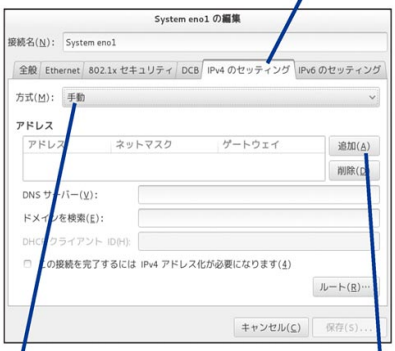
❷[編集]をクリック

次のページに続く

3 IPv4の設定を開始する

接続の設定画面が表示された

❶[IPv4のセッティング]をクリック



❷[手動]を選択

❸[追加]をクリック

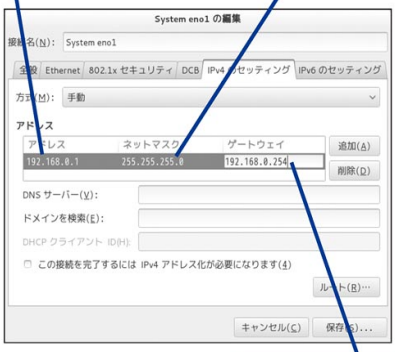
4 IPアドレスを入力する

ここでは「本書のネットワーク構成」の例から入力する

自分の環境に合わせて入力する

❶設定するIPアドレスとネットマスクを入力

❷設定するネットマスクを入力

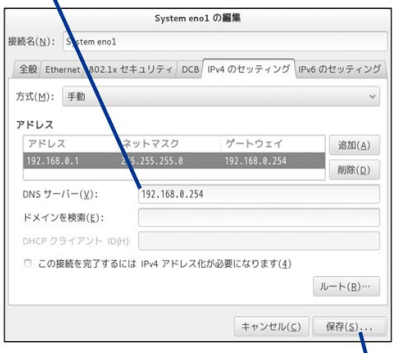


❸ゲートウェイのIPアドレスを入力

5 設定を完了する

❶DNSサーバーのIPアドレスを入力

複数指定する場合は、「,」(カンマ)区切りで入力する



❷[保存]をクリック

ネットワークインターフェイスを選ぶ画面に戻る

❸[閉じる]をクリック

nm-connection-editorが終了する

❹システムを再起動する

HINT!

ネットワークだけ更新するには

このレッスンでは、ネットワークの設定変更を反映するために、システムを再起動しています。そのかわりに、コマンドラインから一度ネットワークを切断し、再接続することでも反映できます。

```
# nmcli device disconnect eno1
# nmcli device connect eno1
```

ネットワークインターフェイスの設定ファイル

ネットワーク設定ツールのnmtuiおよびnm-connection-editorでネットワークを設定しました。さて、設定ファイルはどこに保存されているのでしょうか？

「/etc/sysconfig/network-scripts」のディレクトリの中には、「ifcfg-」から始まるファイルがあります。ネットワークインターフェイス名が「enol」ならば、「ifcfg-enol」が対応する設定ファイルです。この中に、ネットワークの設定ツールで設定した情報が格納されています。

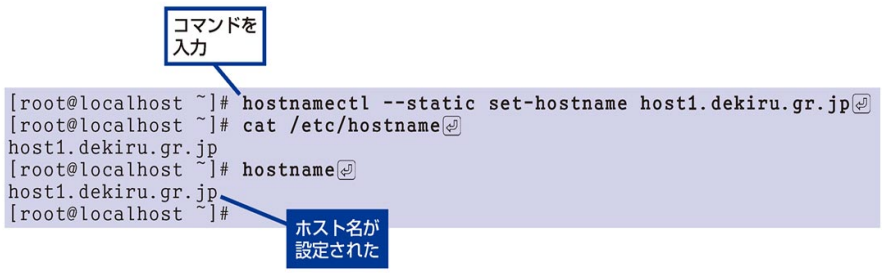
ifcfg-eth1の例

```
HWADDR=68:05:ca:12:34:56
TYPE="Ethernet"
BOOTPROTO="none"
IPADDR0="192.168.0.1"
PREFIX0="24"
GATEWAY0="192.168.0.254"
DEFROUTE="yes"
DNS1="192.168.0.254"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="no"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
NAME="System eno1"
UUID="80e09b4c-a1d0-45fe-936c-b5165b371dee"
DEVICE="eno1"
ONBOOT="yes"
```

ホスト名の変更

RHEL 7では、システムのホスト名の情報は「/etc/hostname」ファイルに記述されています。また、hostnameコマンドでホスト名を表示できます。

ホスト名の変更のためには、新しくhostnamectlコマンドが用意されています。次のようにコマンドを実行し、ホスト名を設定します。



4-3 ネットワークを確認するには

ip、ss、ping

ipコマンドやssコマンドを使うと、ネットワークインターフェイスの状態の確認や設定、ルーティングテーブルの表示や追加、削除、ARPテーブルの確認と削除など、ネットワークに関するさまざまな操作ができます。

RHEL 6までは、これらの操作に、net-toolsパッケージに含まれるifconfigやroute、arp、netstatといったコマンドが使われていました。RHEL 7ではnet-toolsは推奨されず、最小構成の場合はインストールされません。

このレッスンでは、ipコマンドやssコマンドのよく使う機能をいくつか解説します。

IPアドレスを確認する

IPアドレスやMACアドレスを確認するには、ip addr showコマンドを実行します。これは、従来のifconfigコマンドに相当します。

コマンドを入力

```
[root@host1 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN
group default qlen 1000
    link/ether f0:1f:af:34:56:78 brd ff:ff:ff:ff:ff:ff
[root@host1 ~]#
```

ネットワークインターフェイスの情報が表示された

ルーティングテーブルを確認する

ルーティングテーブルを確認するには、ip route showコマンドを実行します。これは、従来のrouteコマンドに相当します。

コマンドを入力

```
[root@host1 ~]# ip route show
default via 192.168.0.254 dev eno1 proto static  metric 1024
192.168.0.0/24 dev eno1  proto kernel  scope link  src 192.168.0.1
[root@host1 ~]#
```

ルーティングテーブルが表示された

ARPテーブルを確認する

ARPテーブルを確認するには、ip neighbor showコマンドを実行します。これは、従来のarpコマンドに相当します。

コマンドを入力

```
[root@host1 ~]# ip neighbor show
192.168.0.101 dev eno1 lladdr 00:a0:de:44:66:88 REACHABLE
192.168.0.102 dev eno1 lladdr f0:1f:af:34:34:34 REACHABLE
192.168.0.250 dev eno1 lladdr 1c:b1:7f:12:34:56 STALE
192.168.0.108 dev eno1  FAILED
[root@host1 ~]#
```

ARPテーブルが表示された

次のページに続く

セッションを確認する

ssコマンドを使うと、ネットワークで通信するのに使われるソケットの各種統計情報を確認できます。従来のnetstatコマンドに代わって利用されます。

TCPポートで通信を行っているすべての通信状態を表示するには、ss -natコマンドを実行します。

コマンドを入力

```
[root@host1 ~]# ss -nat
State      Recv-Q Send-Q        Local Address:Port          Peer Address:Port
LISTEN     0      100               127.0.0.1:25                 *:*
LISTEN     0      128                     *:22                   *:*
LISTEN     0      100                   ::1:25                  :::*
LISTEN     0      128                    :::22                  :::*
[root@host1 ~]#
```

TCPポートでリッスンして待ち受けているすべてのポートを表示するには、ss -nltコマンドを実行します。

コマンドを入力

```
[root@host1 ~]# ss -nlt
State      Recv-Q Send-Q        Local Address:Port          Peer Address:Port
LISTEN     0      100               127.0.0.1:25                 *:*
LISTEN     0      128                     *:22                   *:*
LISTEN     0      100                   ::1:25                  :::*
LISTEN     0      128                    :::22                  :::*
[root@host1 ~]#
```

UDPポートで通信しているすべての通信状態を表示するには、ss -nauコマンドを実行します。

コマンドを入力

```
[root@host1 ~]# ss -nau
State      Recv-Q Send-Q        Local Address:Port          Peer Address:Port
UNCONN     0      0                    *:34275                 *:*
UNCONN     0      0                    *:5353                  *:*
[root@host1 ~]#
```

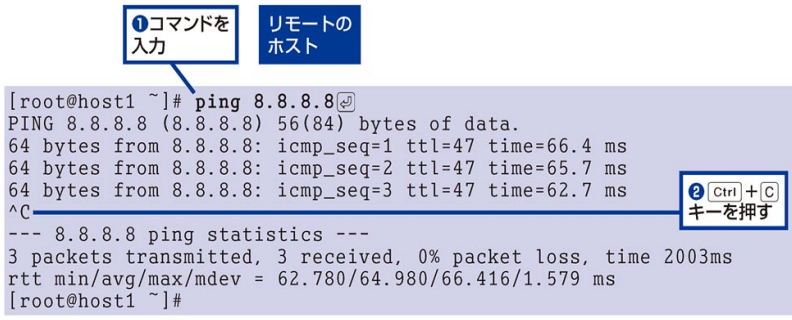
UDPポートでリッスンして待ち受けているすべてのポートを表示するには、ss -nluコマンドを実行します。

コマンドを入力

```
[root@host1 ~]# ss -nlu
State      Recv-Q Send-Q        Local Address:Port          Peer Address:Port
UNCONN     0      0                    *:34275                 *:*
UNCONN     0      0                    *:5353                  *:*
[root@host1 ~]#
```

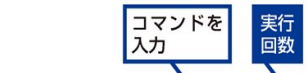
ネットワークの疎通を確認する

ネットワーク経由でリモートのホストにパケットが到達できるか確認するには、pingコマンドを実行します。




```
[root@host1 ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=47 time=66.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=47 time=65.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=47 time=62.7 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 62.780/64.980/66.416/1.579 ms
[root@host1 ~]#
```

Ctrl+Cを実行するまで、無制限に実行されますのでご注意ください。-cオプションを使うと実行回数を指定することもできます。




```
[root@host1 ~]# ping -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=47 time=67.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=47 time=66.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=47 time=64.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=47 time=62.8 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=47 time=69.7 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 62.868/66.276/69.707/2.349 ms
[root@host1 ~]#
```

「time=」で表示されるレスポンス結果（RTT）の目安としては、同じネットワークセグメント上のホストの場合で1 ms（ミリ秒）以内、国内のインターネット上にあるホストの場合で10ms ～ 30ms以内、地球の裏側で500ms程度です。

なお、pingコマンドはICMPプロトコルを使います。ICMPプロトコルがブロックされているネットワーク環境では、実行してもうまく結果が帰ってこない場合があります。

4-4 システムをRed Hat カスタマーポータルに登録するには

subscription-manager

最新版のソフトウェアアップデートを適用する場合や、ソフトウェアを追加インストールする場合などには、Red Hat CDNもしくはRed Hat Satelliteからパッケージを取得する必要があります。

そのためには、事前にシステムがサブスクリプション管理サーバー（カスタマーポータル、Subscription Asset、Red Hat Satellite）に登録されている必要があります。

このレッスンでは、システムを登録する手続きについて、そのためのコマンドと、その流れについて説明します。

登録にはsubscription-managerを使う

RHELのソフトウェアをアップデートするには、システムをRed Hat カスタマーポータルに登録する必要があります。

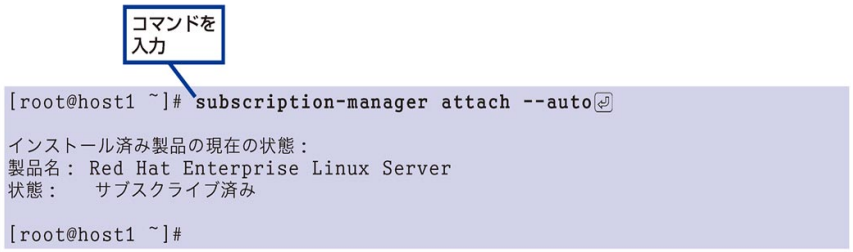
RHEL 6までRed Hat Networkにシステムを登録する際に利用していたrhn_registerコマンドは、RHEL 7ではローカル環境にパッケージ配信システムを提供するRed Hat Satelliteへ登録するために限定されたコマンドとなり、Red Hat Satelliteを利用していないユーザーには必要ないコマンドになりました。

RHEL 7では、subscription-managerコマンドを利用します。subscription-managerには、コマンドラインとGUIツールの両方が提供されていますが、利用可能なサブスクリプションとの紐付けを行うだけなので、初めて行う場合でもコマンドラインからで大丈夫です。

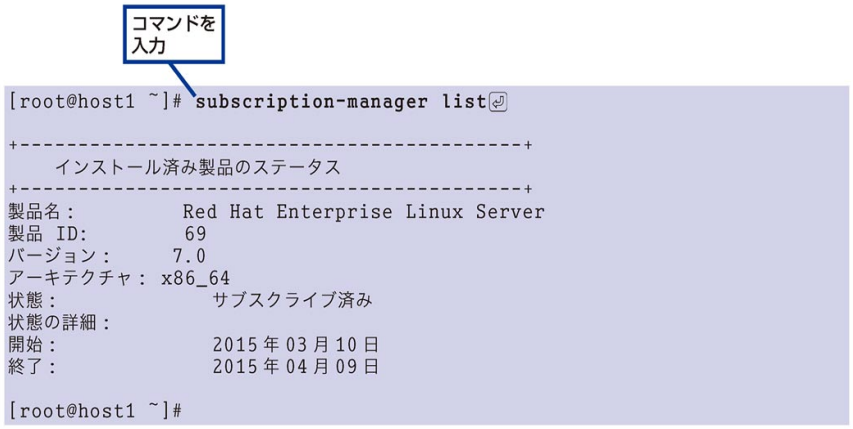
システムを登録する

まずは､次のようにサブスクリプションを登録します。「ユーザー」に入れるユーザー名は、RHEL 7のISOイメージを入手する際にRed Hatのサイトに入力した「Red Hat アカウント」を指定します。

次に、システムにサブスクリプションをアタッチします。次のようにコマンドを実行します。



サブスクリプションが適切にアタッチできたか確認するには、次のようにコマンドを実行します。



「状態:」(英語メッセージの場合は「Status:」) が「サブスクライブしています」(英語メッセージの場合は「Subscribed」) になっており、サブスクリプションの「開始:」と「終了:」の日付が表示されていれば、アタッチできています。

これでシステムとサブスクリプションが紐付けられました。yumコマンドや、GUIツールのgpk-applicationやgpk-update-viewerを通じて、更新パッケージを取得できます。

4-5 ソフトウェアをインストールするには

yum

Red Hatから提供されるソフトウェアは「RPMパッケージ」という単位で管理されています。RHEL 7は、RPMパッケージが集まってできています。ソフトウェアのインストールやアンインストール、アップデートといった作業も、RPMパッケージ単位で行います。

yumコマンドを実行すると、Red Hat CDNから最新のRPMパッケージを入手してインストールできます。RPMパッケージ間の依存性を確認し、関連するRPMパッケージも同時にインストールされます。ファイルとして入手したRPMパッケージもyumコマンドでインストールできます。

パッケージをインストールする

追加ソフトウェアをRed Hat CDNから入手してインストールするには、yum installコマンドを使います。最初にyum installコマンドを実行したときには、Red Hat CDNの公開鍵がインストールされます。

1 インストールを実行する

ここでは「zsh」パッケージをインストールする

❶コマンドを入力

インストールするパッケージ

```
[root@host1 ~]# yum install zsh
読み込んだプラグイン：langpacks, product-id, subscription-manager
rhel-7-server-rpms                                       | 3.7 kB     00:00
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> パッケージ zsh.x86_64 0:5.0.2-7.el7_1.1 を インストール
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package    アーキテクチャー
                          バージョン                  リポジトリー                容量
================================================================================
インストール中：
 zsh        x86_64        5.0.2-7.el7_1.1           rhel-7-server-rpms        2.4 M

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

総ダウンロード容量： 2.4 M
インストール容量： 5.6 M
Is this ok [y/d/N]: y
Downloading packages:
```

必要なパッケージが検索される

zshパッケージが表示される

❷「y」と入力してEnterキーを押す

インストールが実行される

メモ Red Hat CDNにアクセスするには、インターネットへのアクセスと、**レッスン4-4**のようなシステムの登録が必要です。

HINT!

海外のDNSサーバーは使わない

Red Hat CDNは、アクセスしてきたマシンに対して、そこからネットワーク的に最も近いサーバーを参照させます。したがって、国内からGoogle Public DNSなど海外のDNSサーバーを参照していると、Red Hat CDNの海外のキャッシュサーバーを参照してしまい、ダウンロード測度が低下する場合があります。契約しているインターネットサービスプロバイダーが提供するDNSサーバーを参照することをおすすめします。

2 公開鍵をインストールする

最初にyum installを実行したときのみ公開鍵の警告が表示される

公開鍵を2つインストールする

```
警　告 :  /var/cache/yum/x86_64/7Server/rhel-7-server-rpms/packages/zsh-5.0.2-7.
el7_1.1.x86_64.rpm: ヘッダー V3 RSA/SHA256 Signature、鍵 ID fd431d51: NOKEY
zsh-5.0.2-7.el7_1.1.x86_64.rpm の公開鍵がインストールされていません
zsh-5.0.2-7.el7_1.1.x86_64.rpm                             | 2.4 MB    00:08
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release から鍵を取得中です。
Importing GPG key 0xFD431D51:
 Userid     : "Red Hat, Inc. (release key 2) <security@redhat.com>"
 Fingerprint: 567e 347a d004 4ade 55ba 8a5f 199e 2f91 fd43 1d51
 Package    : redhat-release-server-7.1-1.el7.x86_64 (@anaconda/7.1)
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
上記の処理を行います。よろしいでしょうか？ [y/N]y
Importing GPG key 0x2FA658E0:
 Userid     : "Red Hat, Inc. (auxiliary key) <security@redhat.com>"
 Fingerprint: 43a6 e49c 4a38 f4be 9abf 2a53 4568 9c88 2fa6 58e0
 Package    : redhat-release-server-7.1-1.el7.x86_64 (@anaconda/7.1)
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
上記の処理を行います。よろしいでしょうか？ [y/N]y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  インストール中            : zsh-5.0.2-7.el7_1.1.x86_64                      1/1
rhel-7-server-rpms/7Server/x86_64/productid             | 1.7 kB     00:00
  検証中                    : zsh-5.0.2-7.el7_1.1.x86_64                      1/1

インストール :
  zsh.x86_64 0:5.0.2-7.el7_1.1
```

❶「y」と入力してEnterキーを押す

❷「y」と入力してEnterキーを押す

パッケージのインストールが実行される

インストールが完了した

次のページに続く

ローカルのRPMパッケージをインストールする

yumコマンドでは、Red Hat CDNから以外にも、インストールDVDに入っているRPMパッケージや、ダウンロードしてきたRPMパッケージなど、ローカルに存在するRPMパッケージもインストールできます。ローカルのRPMパッケージをインストールするには、yum localinstallコマンドを使います。

なお、インストールしたいRPMパッケージに依存性がある場合は、依存するRPMパッケージも同時に引数で指定する必要があります。

1 RPMパッケージをダウンロードする

Red Hat CDNからRPMパッケージをyumdownloaderコマンドでダウンロードする

コマンドを入力　ダウンロードするパッケージ　ダウンロードされるRPMパッケージのファイル

```
[root@host1 ~]# yumdownloader expect
読み込んだプラグイン：langpacks, product-id
expect-5.45-12.el7.x86_64.rpm                                | 260 kB   00:02

[root@host1 ~]#
```

2 RPMパッケージをインストールする

ダウンロードしたパッケージをインストールする

❶コマンドを入力　インストールするパッケージ

```
[root@host1 ~]# yum localinstall expect-5.45-12.el7.x86_64.rpm
読み込んだプラグイン：langpacks, product-id, subscription-manager

================================================================================
 Package     アーキテクチャー
                        バージョン              リポジトリー                  容量
================================================================================
インストール中：
 expect      x86_64      5.45-12.el7          /expect-5.45-12.el7.x86_64     558 k

合計容量： 2.4 M
総ダウンロード容量： 1.9 M
インストール容量： 4.9 M
Is this ok [y/d/N]: y
Downloading packages:

完了しました！
[root@host1 ~]#
```

❷「y」と入力してEnterキーを押す

インストールが完了した

パッケージをアンインストールする

パッケージをアンインストールするには、yum removeコマンドを使います。

1 アンインストールを実行する

ここでは「zsh」パッケージをアンインストールする

❶コマンドを入力　　アンインストールするパッケージ

```
[root@host1 ~]# yum remove zsh
読み込んだプラグイン：langpacks, product-id, subscription-manager
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> パッケージ zsh.x86_64 0:5.0.2-7.el7_1.1 を 削除
--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package    アーキテクチャー
                          バージョン                  リポジトリー                容量
================================================================================
削除中：
 zsh        x86_64         5.0.2-7.el7_1.1           @rhel-7-server-rpms        5.6 M

トランザクションの要約
================================================================================
削除  1 パッケージ

インストール容量： 5.6 M
上記の処理を行います。よろしいでしょうか？ [y/N]y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  削除中                  : zsh-5.0.2-7.el7_1.1.x86_64                        1/1
  検証中                  : zsh-5.0.2-7.el7_1.1.x86_64                        1/1

削除しました：
  zsh.x86_64 0:5.0.2-7.el7_1.1

完了しました！
[root@host1 ~]#
```

削除するパッケージを確認される

❷「y」と入力してEnterキーを押す

アンインストールが実行される

アンインストールが完了した

4-6 RHEL 7を最新の状態にする

yum update

RHEL 7は、RPMパッケージが集まってできています。パッケージを最新のものに更新することで、システムを最新の状態に保てます。

Red Hatでは、サブスクリプション契約が有効な期間において、製品のバグ修正、セキュリティ修正、機能拡張のエラータが提供されます。各エラータには相当するRPMパッケージが用意され、yumコマンドによってこのRPMパッケージを適用することにより、システムを最新の状態に保てます。yumコマンドを1回実行するだけで、最新のソフトウェアをチェックして更新パッケージを適用します。

1 更新を開始する

コマンドを入力

更新するパッケージがチェックされる

```
[root@host1 ~]# yum update
読み込んだプラグイン：langpacks, product-id, subscription-manager
依存性の解決をしています
---> パッケージ bind-libs.x86_64 32:9.9.4-18.el7 を 更新
---> パッケージ bind-libs.x86_64 32:9.9.4-18.el7_1.1 を アップデート
---> パッケージ bind-libs-lite.x86_64 32:9.9.4-18.el7 を 更新

--> 依存性解決を終了しました。

依存性を解決しました

================================================================================
 Package                アーキテクチャー
                               バージョン                  リポジトリー          容量
================================================================================
インストール中：
 kernel                 x86_64 3.10.0-229.1.2.el7          rhel-7-server-rpms   31 M
更新します：
 bind-libs              x86_64 32:9.9.4-18.el7_1.1         rhel-7-server-rpms 1.0 M
 bind-libs-lite         x86_64 32:9.9.4-18.el7_1.1         rhel-7-server-rpms 712 k
 bind-license           noarch 32:9.9.4-18.el7_1.1         rhel-7-server-rpms  80 k
 bind-utils             x86_64 32:9.9.4-18.el7_1.1         rhel-7-server-rpms 199 k
 binutils               x86_64 2.23.52.0.1-30.el7_1.1      rhel-7-server-rpms 5.0 M
 dnsmasq                x86_64 2.66-13.el7_1               rhel-7-server-rpms 228 k
 dracut                 x86_64 033-241.el7_1.1             rhel-7-server-rpms 300 k
 dracut-config-rescue   x86_64 033-241.el7_1.1             rhel-7-server-rpms  44 k
 dracut-network         x86_64 033-241.el7_1.1             rhel-7-server-rpms  82 k
 firefox                x86_64 31.6.0-2.el7_1              rhel-7-server-rpms  61 M
 flac-libs              x86_64 1.3.0-5.el7_1               rhel-7-server-rpms 169 k
 freerdp-libs           x86_64 1.0.2-5.el7_1.1             rhel-7-server-rpms 221 k
```

更新するパッケージが表示された

2 更新を実行する

```
トランザクションの要約
================================================================================
インストール    1 パッケージ
更新           52 パッケージ

総ダウンロード容量： 147 M
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for rhel-7-server-rpms
(1/53): bind-libs-lite-9.9.4-18.el7_1.1.x86_64.rpm          | 712 kB   00:02
(2/53): bind-libs-9.9.4-18.el7_1.1.x86_64.rpm               | 1.0 MB   00:02

--------------------------------------------------------------------------------
合計                                                2.6 MB/s | 147 MB  00:56
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  更新します              : libxml2-2.9.1-5.el7_1.2.x86_64                1/105

  検証中                  : 10:qemu-img-1.5.3-86.el7.x86_64             105/105

インストール：
  kernel.x86_64 0:3.10.0-229.1.2.el7

更新：
  bind-libs.x86_64 32:9.9.4-18.el7_1.1
  bind-libs-lite.x86_64 32:9.9.4-18.el7_1.1
  bind-license.noarch 32:9.9.4-18.el7_1.1
  bind-utils.x86_64 32:9.9.4-18.el7_1.1
  binutils.x86_64 0:2.23.52.0.1-30.el7_1.1
  dnsmasq.x86_64 0:2.66-13.el7_1
  dracut.x86_64 0:033-241.el7_1.1
  dracut-config-rescue.x86_64 0:033-241.el7_1.1
  dracut-network.x86_64 0:033-241.el7_1.1
  firefox.x86_64 0:31.6.0-2.el7_1
  flac-libs.x86_64 0:1.3.0-5.el7_1
  freerdp-libs.x86_64 0:1.0.2-5.el7_1.1
  freetype.x86_64 0:2.4.11-10.el7_1.1
  java-1.7.0-openjdk.x86_64 1:1.7.0.75-2.5.4.7.el7_1
  java-1.7.0-openjdk-headless.x86_64 1:1.7.0.75-2.5.4.7.el7_1
  kernel-tools.x86_64 0:3.10.0-229.1.2.el7
  kernel-tools-libs.x86_64 0:3.10.0-229.1.2.el7
  libcacard.x86_64 10:1.5.3-86.el7_1.1
  libgudev1.x86_64 0:208-20.el7_1.2
  libsss_idmap.x86_64 0:1.12.2-58.el7_1.6
  libsss_nss_idmap.x86_64 0:1.12.2-58.el7_1.6

完了しました！
[root@host1 ~]#
```

「y」と入力してEnterキーを押す

ダウンロードが実行される

更新が実行される

更新が完了した

STEP UP

なぜシステムの登録が必要なのか

RHELでは、インターネット経由でシステムを登録します。登録が必要なのは、システムを最新の状態に保つための更新パッケージのアップデートを行えるようにするためです。

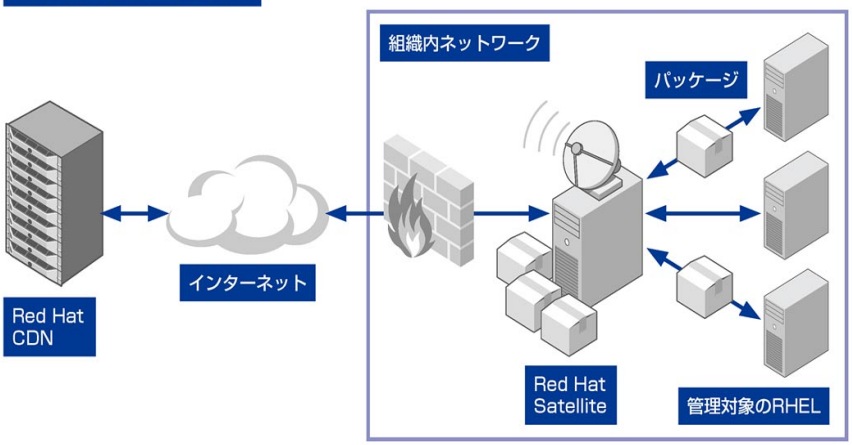
Red HatはRHEL以外にも、仮想化管理ツールのRed Hat Enterprise Virtualizationや、ミドルウェア製品のRed Hat JBossなど、すべての製品をソフトウェアチャンネルから提供しています。

subscription-managerコマンドを使って、システムを登録してサブスクリプションと紐付けることで、そのシステムに付与されているサブスクリプションに従い、RHELのベースチャンネルや、RHELのアドオンのチャンネル、その他のRed Hat製品のチャンネルが使えるようになります。

つまり、Red Hatとサブスクリプション契約をしているすべての製品をyumコマンドでインストールできます。そして、すべての製品を最新の状態に保つことができます。

インターネットに接続できない環境のサーバーに対しては、ローカル環境に同じような機能を用意する、Red Hat Satelliteという製品があります。Red Hat Satelliteが一括して、Red Hat CDNからチャンネルのメタデータとRPMパッケージをあらかじめ取得しておくことで、ローカル環境のRHELサーバーに対して、Red Hat Satelliteから更新パッケージを提供できます。

◆Red Hat Satelliteの仕組み



第5章
サーバーを準備する

RHEL 7でサーバーを構築する基本的な作業として、本章ではユーザーアカウントの仕組みと管理方法、systemdによる各種サービスの管理方法、ファイアウォールの仕組みと設定方法、SSHを使ってサーバーのコマンドラインにリモートから接続する方法について説明していきます。なお、RHEL 6との違いも随所に交えながらご紹介していきますので、従来からRHELをお使いの方も目を通しておくとよいでしょう。

●この章の内容

5-1 ユーザーを管理するには

ユーザーとグループ

Linuxは複数のユーザーで利用できるOSです。システムの利用者は事前に登録されているユーザーアカウントでログインしてから利用を開始します。システムを1人で専有して利用している場合であっても、システムのバックグラウンドでサービスとして稼働している複数のプロセスは、それぞれ別々のユーザーとして動いています。また、ユーザーの集合であるグループという単位を作り、ユーザーを参加させることもできます。このレッスンでは、ユーザーの追加やグループの追加、グループへのユーザーの加入、ユーザーとグループの削除を学びます。

ユーザーを追加する

各ユーザーには、ユーザー名とユーザーID（UID）が与えられます。システムを制御する特権を持つ管理者ユーザーのrootは、UIDが0で固定されています。一方、一般ユーザーは、1000以降のUIDが通し番号で振られます。ユーザー名は好きに決められます。

ユーザーを追加するには、root権限でuseraddコマンドを実行します。作成するユーザー名を引数に指定します。

ユーザー名以外に何も指定せずに実行した場合には、一般ユーザーとしてユーザーが作成され、/homeの下にホームディレクトリが用意されます。また、ユーザーを識別するUIDが自動的に振られます。そのユーザーのグループも作られ、GIDが自動的に振られます。UIDとGIDは、明示的に番号を指定することもできます。

コマンドを入力　作成するユーザー名

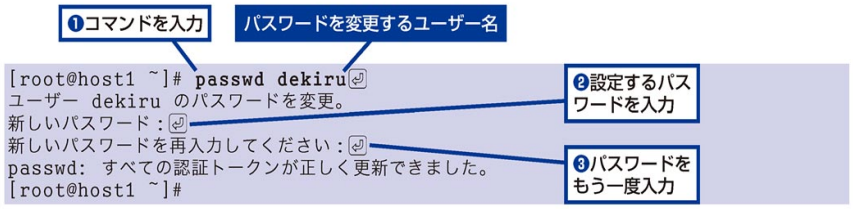
```
[root@host1 ~]# useradd dekiru
[root@host1 ~]#
```

なお、RHEL 6ではUIDとGIDが、500から連番で振られていました。そのため、RHEL 6からRHEL 7へデータを移行するときには注意してください。RHEL 6で振られたUIDとGIDをRHEL 7でも活かして運用を続けたい場合には、設定ファイル/etc/login.defsの中の「UID_MIN」と「GID_MIN」の値を「500」に、「SYS_UID_MAX」と「SYS_GID_MAX」の値を499に変更する必要があります。設定した上で、useraddコマンドを実行してください。

パスワードを変更する

各ユーザーには、ユーザーごとにパスワードを設定します。管理者がユーザーを作った直後はパスワードが設定されておらず、パスワードを設定するまでログインできません。実際の運用では、rootがパスワードを設定したあと、一般ユーザーが最初にログインしたときに改めて自分でパスワードを変更すべきです。

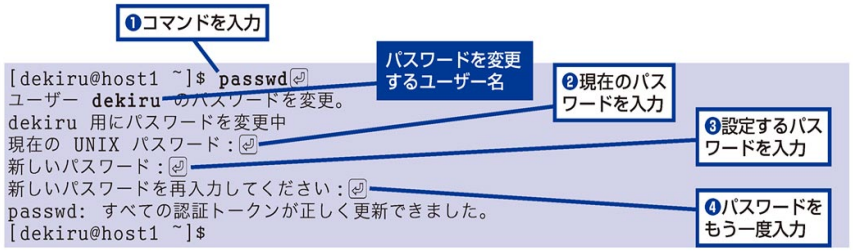
パスワードを変更するには、passwdコマンドを使います。rootアカウントから変更する場合には、コマンドの引数にユーザー名を指定します。



メモ 環境によっては英語でメッセージが表示されます。

一般ユーザーが自分のパスワードを変更するには、ログインしている状態で引数なしでpasswdコマンドを実行します。

この場合、すでに設定されている現在のパスワードを質問された後に、新しいパスワードを入力します。



グループを追加する

useraddコマンドで作られるグループのほかに、作業や所属する組織に対して追加グループを作成する必要が生じてくることでしょう。

グループを作成するにはgroupaddコマンドを使います。たとえば、GIDが300の「management」グループを作成するには、root権限で次のように実行します。

次のページに続く


```
[root@host1 ~]# groupadd -g 300 management
[root@host1 ~]#
```

ユーザーをグループへ加入させる

特定のユーザーについて、所属するグループを追加できます。1人のユーザーは、複数のグループに加入できます。

ユーザーの所属するグループを追加するには、usermodコマンドに-aオプションと-Gオプションを付けて実行します。たとえば、dekiruユーザーを、さきほど作成したmanagementグループに加入させるには、次のように実行します。


```
[root@host1 ~]# usermod -a -G management dekiru
[root@host1 ~]#
```

ユーザーを削除する

ユーザーを削除するにはuserdelコマンドを使います。たとえば、「dekiru」ユーザーを削除するには、次のように実行します。


```
[root@host1 ~]# userdel dekiru
[root@host1 ~]#
```

ホームディレクトリも削除したい場合には、-rオプションを付けて実行します。たとえば、「dekiru」ユーザーをホームディレクトリごと削除するには、次のように実行します。


```
[root@host1 ~]# userdel -r dekiru
[root@host1 ~]#
```

グループを削除する

グループを削除するにはgroupdelコマンドを使います。たとえば、「management」グループを削除したい場合には次のように実行します。


```
[root@host1 ~]# groupdel management
[root@host1 ~]#
```

ユーザーとパスワードの仕組み

Linuxのユーザー情報は、/etc/passwdと/etc/shadowの2つのファイルに格納されています。両者のファイルは、「:」（コロン）で区切られた情報が列挙されているだけのように見えますが、1行分の情報が1ユーザーに対応しています。

/etc/passwdというファイル名から推測できますが、昔はこのファイルに暗号化されたパスワードが記述されていました。1つ目のフィールドはユーザー名で、2つ目のフィールドはパスワードを意味します。最近のLinuxディストリビューションでは、パスワードの代わりに「x」というダミー文字が入っています。

/etc/passwdの1ユーザーの情報

```
htaira:x:1000:1000:Hajime Taira:/home/htaira:/bin/bash
```

ユーザー名　パスワード　UID　GID　コメント　ホームディレクトリ　ログインシェル

たとえば、ls -lコマンドなどで、UIDからユーザー名を参照する場合など、/etc/passwdファイルを間接的に参照することがあります。そのため、システムにログインできる人であれば誰でも /etc/passwdの読み取り権限を持っています。ここにパスワードが記載されていると、セキュリティ的に問題です。

そこで現在、Linuxを利用するユーザーのパスワードは、/etc/shadowファイルに記載されています。このファイルには「----------.」という特殊なパーミッションが付与されており、管理者しか見ることができません。

/etc/passwdと/etc/shadowのパーミッション

```
-rw-r--r--. 1 root root 2020  4月 13 17:21 /etc/passwd
----------. 1 root root 1156  4月 13 17:21 /etc/shadow
```

/etc/shadowの1ユーザーの情報

```
htaira:$6$C8VFSKQI$GoFYjquR816Ww.Xq4sr6EsLzRbMJzaFpG6ZlfPF18ukRD5FH/Rhl
KRCIuAFLkXN5H9UYVmPJHqd8x5GAMqQun1:16296:0:99999:7:::
```

また、/etc/shadowに記述されているパスワードは、ハッシュ化（符号化）されており、その値を盗み見られたとしても、元のパスワードを当てることは大変難しくなっています。そのためには時間をかけて総当りでパスワードを解析しなければなりません。10文字以上であれば1000年以上かかると言われています。

5-2 サービスを管理するには

systemd

Linuxではサーバー機能を「サービス」というプログラムによって提供しています。RHEL 7では、サービスの管理を「systemd」という仕組みから行うように変更されました。従来のLinuxと操作方法や管理の体系が大きく変わっていて、RHEL 7の最大の変更点といえます。

このレッスンでは、RHEL 7でのサービスの起動や停止、再起動、自動起動の設定の方法を解説します。また、動作モードをGUIやテキストコンソールなど切り替える仕組みもsystemdから行うように変更されたので、動作モードの操作方法も解説します。

systemdでサービスを管理

Linuxでは、クライアントからの要求に応えるサーバー機能を、それぞれプログラムによって提供しています。こうしたサーバー機能を「サービス」と呼びます。Webサーバーやメールサーバー、DNSサーバーなどは、すべてサービスです。

サービスを起動し管理する仕組みとして、Linuxではinitデーモンというプログラムが動いています。Linuxでは古くから、SysVinitがinitデーモンとして使われてきました。また、RHEL 6ではUpstartが採用されました。

このinitデーモンが、RHEL 7からは「systemd」に変更されました。これはRHEL 7の中で一番大きい変更だといえます。

systemdでは、サービスなどの管理対象を「Unit（ユニット）」として管理しています。systemdは、ユーザーやシステムが起こしたイベントに基づいてUnitを実行する、イベント駆動型となっています。また、各Unitには、そのUnitを実行するのに必要なUnitが依存関係として定義されています。

サービスの起動と停止をするUnitは、サービスユニット（*.service）と呼ばれます。サービスユニットの定義ファイルは/usr/lib/systemd/systemディレクトリに格納されています。

これにより、必要に応じて必要なサービスを起動する、賢いシステムを実現しています。また、依存関係のないサービスの起動などを並列化し、システムの初期化と起動を格段に高速化しました。

サービス一覧の表示

systemdでは、サービスの起動や停止などの操作に、systemctlコマンドを使います。RHEL 6までで使われていたserviceコマンドは実質的に廃止され、systemctlコマンドを呼び出す代用コマンドになっています。

まず、どのようなサービスがあるか一覧で表示してみましょう。Unitの一覧を表示するには、systemctl list-unit-filesコマンドを実行します。Unitの中からサービスだけを表示するには、「-type=service」オプションを指定します。

❶コマンドを入力

```
[root@host1 ~]# systemctl list-unit-files --type=service↵
UNIT FILE                                   STATE
abrt-ccpp.service                           enabled
abrt-oops.service                           enabled
abrt-pstoreoops.service                     disabled
abrt-vmcore.service                         enabled
abrt-xorg.service                           enabled
abrtd.service                               enabled
accounts-daemon.service                     enabled
alsa-restore.service                        static
alsa-state.service                          static
alsa-store.service                          static
anaconda-direct.service                     static
anaconda-noshell.service                    static
anaconda-shell@.service                     static
anaconda-sshd.service                       static
anaconda-tmux@.service                      static
anaconda.service                            static
arp-ethers.service                          disabled
atd.service                                 enabled
auditd.service                              enabled
autovt@.service                             disabled
avahi-daemon.service                        enabled
blk-availability.service                    disabled
lines 1-23
```

一覧が表示された

spaceキーで次の画面を表示

Bキーで前の画面を表示

❷Qキーを押して終了

RHEL 6の「chkconfig --list」コマンドに相当します。

次のページに続く

サービスの停止

サービスを停止するには、systemctl stopコマンドにUnit名を指定してroot権限で実行します。たとえば、「bluetooth.service」を停止するには次のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl stop bluetooth.service
[root@host1 ~]#
```

サービスが停止した

RHEL 6までの「service サービス名 stop」に相当します。なお、「bluetooth.service」のようなUnit名のうち、「.service」の部分は省略してもかまいません。

systemctl stopコマンドを実行しても、そのUnitに依存しているUnitが動いていると、停止できません。以下は、印刷用サービス「cups.service」を停止しようとした例です。

コマンドを入力

```
[root@host1 ~]# systemctl stop cups.service
Warning: Stopping cups.service, but it can still be activated by:
  cups.path
  cups.socket
[root@host1 ~]#
```

サービスの停止に失敗した

サービスの起動

サービスを起動するには、systemctl startコマンドにUnit名を指定して実行します。たとえば、さきほど起動した「bluetooth.service」を起動するには次のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl start bluetooth.service
[root@host1 ~]#
```

サービスが起動した

RHEL 6までの「service サービス名 start」に相当します。

サービスの再起動

サービスを再起動するには、systemctl restartコマンドにUnit名を指定して実行します。たとえば、「bluetooth.service」を再起動するには次のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl restart bluetooth.service
[root@host1 ~]#
```

サービスが再起動した

RHEL 6までの「service サービス名 restart」に相当します。

サービスの自動起動

OSが起動するときにサービスが自動的に起動するように、あるいはしないように設定するのにも、systemctlコマンドを使います。RHEL 6までのchkconfigコマンドに相当します。

たとえば、bluetooth.serviceを自動的に起動しないようにするには、以下のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl disable bluetooth.service
rm '/etc/systemd/system/dbus-org.bluez.service'
rm '/etc/systemd/system/bluetooth.target.wants/bluetooth.service'
[root@host1 ~]#
```

サービスが自動起動しないようになった

また、bluetooth.serviceを自動的に起動させるには、以下のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl enable bluetooth.service
ln -s '/usr/lib/systemd/system/bluetooth.service' '/etc/systemd/system/
dbus-org.bluez.service'
ln -s '/usr/lib/systemd/system/bluetooth.service' '/etc/systemd/system/
bluetooth.target.wants/bluetooth.service'
[root@host1 ~]#
```

サービスが自動起動するようになった

次のページに続く

動作モードを表すtarget

RHEL 6などの従来のLinuxシステムには、動作モードを複数切り替えられる「ランレベル」という概念がありました。

RHEL 7ではランレベルの概念がなくなり、代わりにsystemdの「target」という概念が用意されています。targetもUnitの一種で、サービスなどのUnitを束ねるためのUnitとなっています。

RHEL 6の各ランレベルに対応するRHEL 7のtargetは、以下の表のとおりです。

ランレベルとtargetの対応

ランレベル	target
0	poweroff.target
1	rescue.target
3	multi-user.target
5	graphical.target
6	reboot.target
emergency	emergency.target

起動時の動作モードを変更する

RHEL 6などの従来のLinuxシステムでは、起動時のランレベルを設定ファイル/etc/inittabで指定していました。

RHEL 7では、起動時のtargetを「default.target」として設定します。具体的には、指定するターゲットを、/etc/systemd/system/default.targetとしてシンボリックリンクします。

たとえば、本書の**レッスン2-4**のとおりにRHEL 7をインストールした場合は、起動時にGUI画面の動作モードになっています。このとき、default.targetとして「graphical.target」が設定されています。

起動時のtargetの切り替えは、systemctl set-defaultコマンドで行います。テキストコンソール画面の動作モード「multi-user.target」に変更するには、以下のように実行します。

コマンドを入力　　以降先の動作モード

```
[root@host1 ~]# systemctl set-default multi-user.target
rm '/etc/systemd/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/
default.target'
[root@host1 ~]#
```

HINT!

起動時にtargetを指定するには

起動時に指定することで、default.targetで指定されたのと異なるtargetで起動することもできます。それには、ブートローダー GRUB2から、カーネルオプションとして「systemd.unit=<ターゲット>」という項目を追加します。たとえば、レスキューモードで起動するには「systemd.unit=emergency.target」を追加します。

反対に、multi-user.targetからgraphical.targetに変更するには、以下のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl set-default graphical.target
rm '/etc/systemd/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/graphical.target' '/etc/systemd/system/
default.target'
[root@host1 ~]#
```

起動後に動作モードを変更する

すでに起動している状態で、その場で動作モードを変更するには、systemctl isolateコマンドを使います。たとえば、GUIで動作しているときに、以下のように実行すると、テキストコンソールの動作モードに変更されます。

コマンドを入力

```
[root@host1 ~]# systemctl isolate multi-user.target
```

テキストコンソール画面に変わり、ログインプロンプトが表示される

RHEL 6までのtelinitコマンドに相当します。

テキストコンソールからGUIに動作モードを変更するには、以下のように実行します。

コマンドを入力

```
[root@host1 ~]# systemctl isolate multi-user.target
```

GUI画面に変わり、ログイン画面が表示されるインプロンプトが表示される

5-3 旧方式でファイアウォールを設定するには

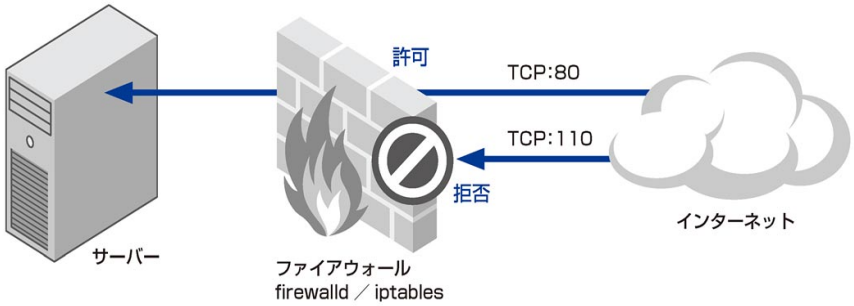
iptables

Linuxにもカーネルの中にファイアウォール機能が搭載されています。RHEL 6まではiptablesという機構が使われていましたが、RHEL 7からはfirewalldが追加されました。iptablesも引き続き提供されますが、デフォルトはfirewalldです。これからはfirewalldが標準になっていくと思いますので、本書では各種サーバーの設定でfirewalldを使います。しかし、iptablesの知識が基礎として必要になるため、このレッスンでは先にiptablesについて紹介します。ファイアウォール機能についてすでに知っている方は、**レッスン5-4**に進んでください。

ファイアウォールってなに？

ネットワーク経由でのアクセスをフィルタリングする仕組みに、ファイアウォールというものがあります。あらかじめ許可されたパケットを通過させて、それ以外のパケットの通過を拒否するという、ネットワークにおけるアクセスコントロールの仕組みです。

ファイアウォールの仕組み



iptables ってなに？

iptablesは、Linux 2.4の時代から使われてきたファイアウォールの仕組みです。実際にはLinuxのカーネルの中にある「Netfilter」がパケットフィルタリング処理を行っており、iptablesはその処理を設定します。なお、**レッスン5-4**で説明するfirewalldも、Netfilterを使ってパケットフィルタリング処理を行っています。

iptablesでは、「チェーン」に対して、パケットを通過させたり拒否したりするルールを定義してフィルタリングします。チェーンには、パケットの受信に対する「INPUT」チェーン、パケットの送信に対する「OUTPUT」チェーン、ネットワークインターフェイスの間でのパケットの転送に対する「FORWARD」チェーンの3つが用意されています。

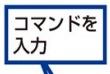
これらのチェーンに対してフィルタリングルールを定義していくことで、ファイアウォールとして機能させます。いわゆる古典的なACL（Access Control List）型のファイアウォールと言えます。

具体的には、最初に基本ポリシーとしてDROP（基本無視）やREJECT（基本拒否）、ACCEPT（基本通過）を定義します。その上で、対象ごとに細かいフィルタリングルールを定義していきます。

それぞれのフィルタリングルールは通常、TCPやUDPのポート番号か、単一ホストのIPアドレスまたはネットワークアドレス、あるいはその両者を対象として定義します。

iptablesを有効にする

もし実際にRHEL 7上でiptablesを試してみたい場合は、事前に設定が必要です。iptablesはfirewalldと同時に起動することはできません。そのため、まずデフォルトで動いているfirewalldサービスを停止し、iptablesサービスを開始します。




```
[root@host1 ~]# systemctl stop firewalld.service
[root@host1 ~]# systemctl start iptables.service
[root@host1 ~]#
```

次のページに続く

iptablesの初期設定

RHEL 7の初期設定で定義されているiptablesのルールを見てみましょう。iptablesのルール定義は、/etc/sysconfig/iptablesというファイルに保存されています。下にその内容を挙げます。このルールは上から順に適用されます。

まず、INPUTとFORWARD、OUTPUTの3つのチェーンについて、デフォルトはACCEPTという基本ポリシーとなっています。その上で、INPUTチェーンとFORWARDチェーンでは、途中でACCEPTと明記したもの以外は、最後でREJECTしています。INPUTチェーンでは、リプライパケット、ICMPパケット、loインターフェイスのパケット、SSHのポート22に対するパケットを許可しています。

初期設定のルール

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

基本ポリシー

許可ルール

遮断ルール

iptablesコマンドの使い方

ルールはiptablesコマンドで1つずつ定義します。ルールを定義するときには、INPUTとFORWARD、OUTPUTのそれぞれのチェーンを指定します。

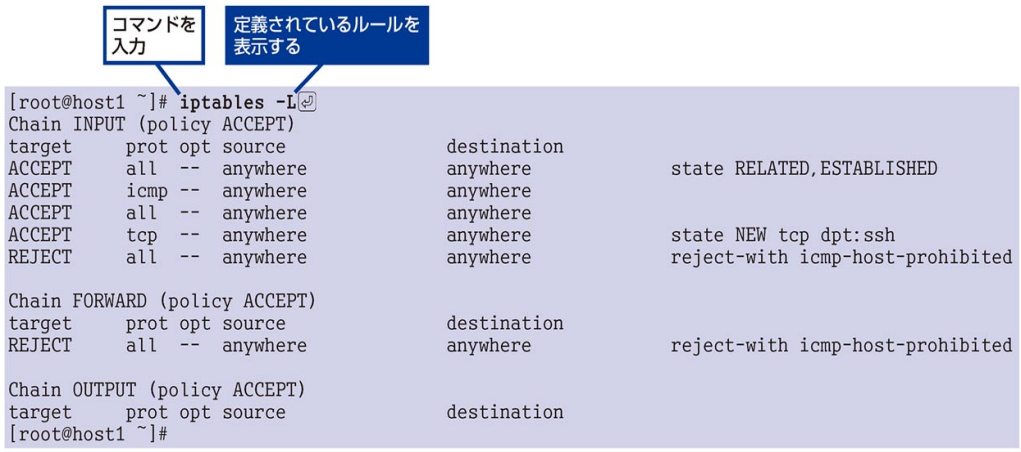
ではこれから、RHEL 7にデフォルトで定義されているiptablesのルールを確認し、一度ルールを初期化したあと、手動で1つずつ再定義してみます。

HINT!

ルールの種類

このレッスンでは、iptableを使って、許可ルールの「ACCEPT」と、遮断ルールの「REJECT」を設定しています。こうした処理の種類（ターゲットと呼びます）にはそのほか、「DROP」「QUEUE」「RETURN」「LOG」「REDIRECT」「DNAT」「SNAT」などさまざまなものがあります。

1 ルールを表示する



2 チェーンのルールを初期化する

3 基本ポリシーを設定する

次のページに続く

4 リプライパケットの許可ルールを定義する

返信のパケットは許可する

コマンドを入力　ルールを追加する

```
[root@host1 ~]# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
[root@host1 ~]#
```

リプライパケット　許可する

5 許可ルールを定義する

最小限の通信を許可する

コマンドを入力　ICPのパケットを許可　自分自身への通信を許可　sshの接続を許可

```
[root@host1 ~]# iptables -A INPUT -p icmp -j ACCEPT
[root@host1 ~]# iptables -A INPUT -i lo -j ACCEPT
[root@host1 ~]# iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
[root@host1 ~]#
```

6 遮断ルールを定義する

許可に該当しないパケットを遮断する　INPUTとFORWARDのチェーンに設定する

コマンドを入力　遮断する

```
[root@host1 ~]# iptables -A INPUT -j REJECT --reject-with icmp-host-prohibited
[root@host1 ~]# iptables -A FORWARD -j REJECT --reject-with icmp-host-prohibited
[root@host1 ~]#
```

7 ルールを保存する

ここまでの設定はメモリー上にしか格納されていない

ルールを/etc/sysconfig/iptablesに保存する

コマンドを入力

```
[root@host1 ~]# /usr/libexec/iptables/iptables.init save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
[root@host1 ~]#
```

ルールが保存された

8 ルールを再読み込みする

ルールを/etc/sysconfig/iptablesから読み込む

コマンドを入力

再読み込み

```
[root@host1 ~]# systemctl reload iptables
[root@host1 ~]#
```

iptablesを無効にする

ここまでiptablesを試しましたが、本書では実際の設定にfirewalldを使います。iptablesを有効にした場合は、iptablesを無効にしてfirewalldを有効にしてください。

コマンドを入力

```
[root@host1 ~]# systemctl stop iptables.service
[root@host1 ~]# systemctl start firewalld.service
[root@host1 ~]#
```

5-4 新方式でファイアウォールを設定するには1

firewalld

RHEL 7からはfirewalldが追加されており、本書でも積極的に利用します。ネットワーク管理者とサーバー管理者とで、ネットワークの見方に少しギャップがあります。ネットワークインターフェイスがどのゾーンのネットワークに属するかというポリシー管理をファイアウォールに実装し、ネットワーク管理者の思考で設定できるようにした仕組みが、firewalldという実装だと筆者は考えています。このレッスンではfirewalldの仕組みやゾーンの考え方、サービスの定義内容について解説していきます。まずは、きちんと概念を理解してから次のレッスンに進んでください。

firewalld ってなに？

ファイアウォールには静的ファイアウォールと動的ファイアウォールがあります。RHEL 7からは動的ファイアウォールの仕組みとして、firewalldが新たに追加されました。

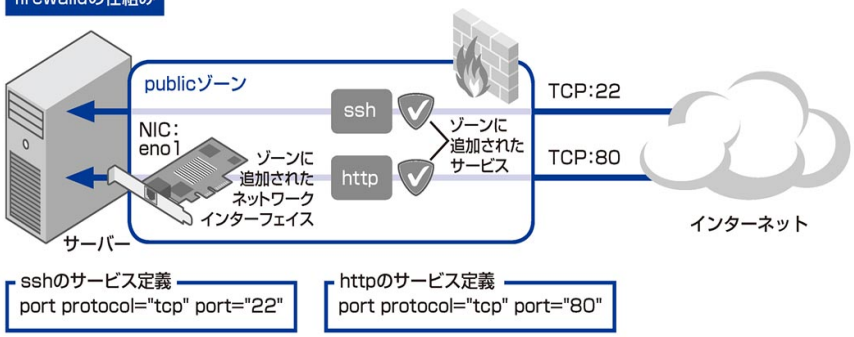
firewalldではネットワークを抽象化して「ゾーン」に分けて管理します。これは、iptablesには存在しなかった仕組みです。今までネットワーク設計者が頭の中で考えていたゾーンの概念を、ファイアウォールにそのまま実装できます。

firewalldではネットワークインターフェイスは、どこかのゾーンに属します。何も設定していないネットワークインターフェイスはデフォルトゾーンに所属します。そして、ゾーンに対して各種サービスの許可ルールを追加していきます。

サービスは事前にある程度定義されており、そのサービスの定義の中にTCPやUDPのポート番号などの情報が定義されています。

この、ゾーンにサービスを割り当てるという概念は、従来型の静的ファイアウォールのルールを書いていた人には斬新な仕組みと受け取れることでしょう。

firewalldの仕組み



firewall-cmdで設定する

firewalldを設定するコマンドラインツールとして、firewall-cmdが用意されています。firewall-cmdのほかに、GUIから設定するfirewall-configや、D-BusのAPIを使ったアプリケーションからfirewalldを設定できます。

firewall-cmdコマンドにより、ゾーンにサービスを追加したり、ネットワークインターフェイスが属するゾーンを変更したりできます。

ゾーンという考え方

ネットワーク管理者が、管理対象ネットワークのセキュリティを確保するためにTrusted ／ Untrustedという分け方でネットワークを分離する設計手法があります。分離されたくくりをゾーン（セキュリティゾーン）と呼びます。特にネットワーク間のファイアウォールを設計する際に、このような分け方をするでしょう。

ただし、現実にはTrusted ／ Untrusted以外にも、おおむね信用できるが攻撃されるかもしれないというゾーンが存在するときもあります。

firewalldには、事前定義済みゾーンが9つあります。

publicゾーンもしくはdmzゾーンは、インターネットに公開するサービスがある場合に利用され、おおむね信用できるが攻撃されるかもしれないゾーンと言えます。

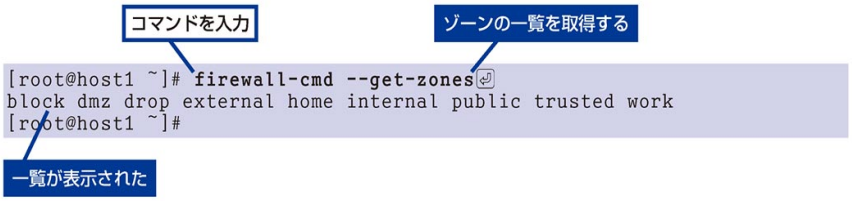
trustedゾーンは、すべてのパケットを通過させます。家庭内LANなど絶対的に信用できるゾーンと言えます。

一見すると重複するゾーンがありますが、実際どれを使うかはユーザーの自由です。どのゾーンにしようか迷ったらデフォルトのpublicゾーンをベースに許可するサービス追加していくとよいでしょう。

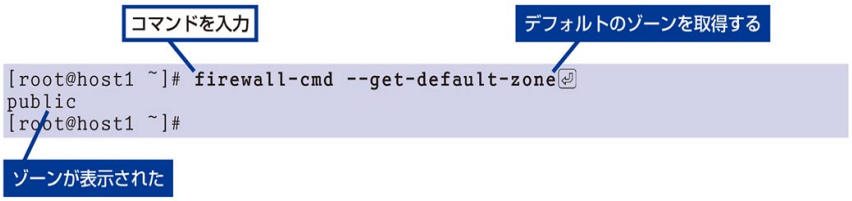
ゾーンの一覧を取得する

ゾーンという考え方を理解するために、まずはfirewalldに事前に定義されているゾーンの一覧を取得してみましょう。次のように、firewall-cmdに--get-zonesオプションを付けて実行します。

次のページに続く



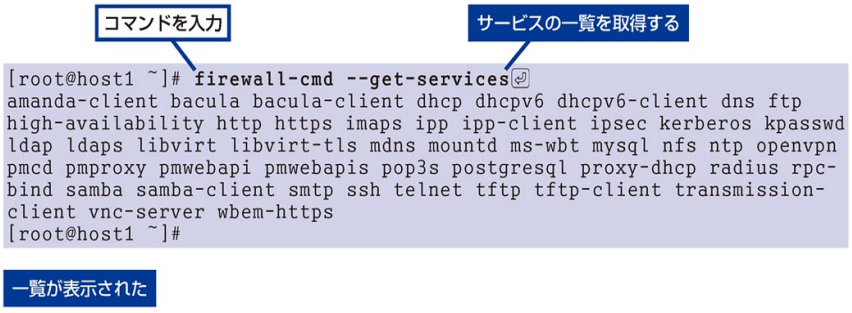
デフォルトのゾーンを確認するには –get-default-zoneオプションを付けて実行します。



サービスの一覧を取得する

事前に定義されているサービスの一覧を取得するには –get-servicesオプションを付けて実行します。

これらのサービスの定義は、/usr/lib/firewalld/services/ディレクトリ以下にXML形式で格納されています。ここは直接書き換えてはいけません。ユーザー定義のサービスをfirewalldに追加したい場合は、/etc/firewalld/services/の中へ追加する必要があります。なお、同じサービス名の場合、/usr/lib/firewalld/services/よりも/etc/firewalld/services/の中にあるファイルが優先されます。



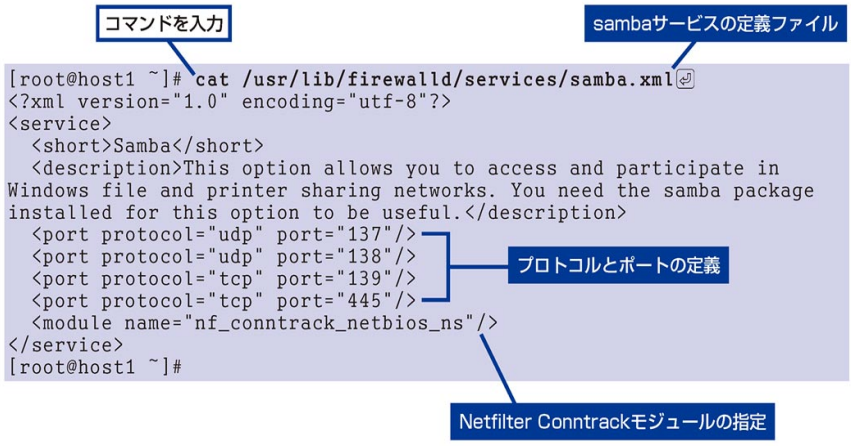
httpサービスの定義を見る

たとえば、firewalldのhttpサービスが定義されているXMLファイルを確認してみましょう。httpサービスの説明と共に、プロトコルがTCPで、ポートが80であると定義されているのが分かります。


```
[root@host1 ~]# cat /usr/lib/firewalld/services/http.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>WWW (HTTP)</short>
  <description>HTTP is the protocol used to serve Web pages. If you
plan to make your Web server publicly available, enable this option.
This option is not required for viewing pages locally or developing Web
pages.</description>
  <port protocol="tcp" port="80"/>
</service>
[root@host1 ~]#
```

sambaサービスの定義を見る

また、今度はfirewalldのsambaサービスの定義を見てみましょう。sambaサービスではhttpサービスと異なり、複数のポート指定とNetfilter Conntrackモジュールの指定が行われています。




```
[root@host1 ~]# cat /usr/lib/firewalld/services/samba.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>Samba</short>
  <description>This option allows you to access and participate in
Windows file and printer sharing networks. You need the samba package
installed for this option to be useful.</description>
  <port protocol="udp" port="137"/>
  <port protocol="udp" port="138"/>
  <port protocol="tcp" port="139"/>
  <port protocol="tcp" port="445"/>
  <module name="nf_conntrack_netbios_ns"/>
</service>
[root@host1 ~]#
```

このように、firewalldで事前に定義されたサービスを利用すると、1つ1つポート番号を指定しなくとも、一括で複数の関連するポートに対して通過を許可できます。

5-5 新方式でファイアウォールを設定するには2

firewall-cmdによる設定

firewalldは旧方式のiptablesと同じく、Linuxが持つパケットフィルタリングの仕組みの「Netfilter」を利用してフィルタリング処理を行っています。しかし両者は、ファイアウォールを設定するときの考え方が違います。firewalldでは、ACLを定義するスタイルではなく、目的のサービスに関連するポートを開放するというスタイルで設定していきます。

このレッスンでは、firewalldの基礎知識をつけてるために、firewalldの仕組みと重要な概念であるゾーンとサービス、それらの中身について解説します。

サービスの許可と禁止

ゾーンに対して通信を許可するサービスを追加するには、firewall-cmdに--add-serviceを付けて実行します。対象のゾーンを指定しなかった場合には、デフォルトゾーンが対象になります。この場合、firewalldに即時に反映されます。

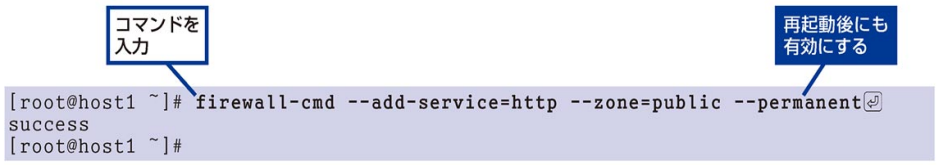
追加しただけでは再起動後には忘れられてしまいます。再起動後にも有効にするには、--parmanentオプションを付けて再度実行します。ただし、--permanentオプションの場合は、即座には反映されません。そのため、即座に反映して、再起動後も有効にするには、両方を実行する必要があります。

また、サービスの通信を禁止するには、ゾーンからサービスを外します。firewall-cmdに--remove-serviceを付けて実行します。同じく再起動後も有効になるように--permanentオプションを付けて再度実行しましょう。

1 ゾーンにサービスを追加する


```
[root@host1 ~]# firewall-cmd --add-service=http --zone=public
success
[root@host1 ~]#
```

2 再起動後にも追加を有効にする

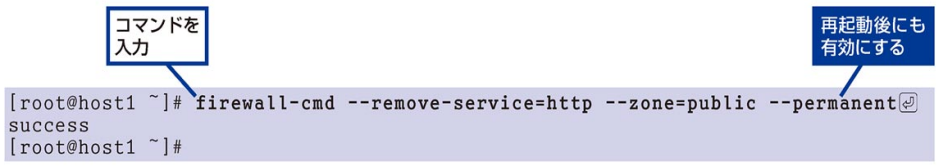



```
[root@host1 ~]# firewall-cmd --add-service=http --zone=public --permanent
success
[root@host1 ~]#
```

3 ゾーンからサービスを外す


```
[root@host1 ~]# firewall-cmd --remove-service=http --zone=public
success
[root@host1 ~]#
```

4 再起動後も外したのを有効にする

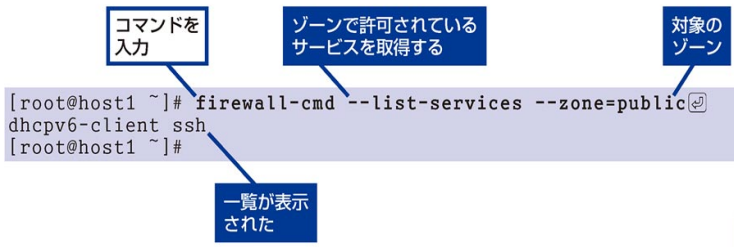



```
[root@host1 ~]# firewall-cmd --remove-service=http --zone=public --permanent
success
[root@host1 ~]#
```

許可されたサービスの一覧

指定のゾーンに追加されて許可されているサービスの一覧を取得するには、firewall-cmdに--list-servicesオプションを付けて実行します。

1 許可されたサービスを表示する




```
[root@host1 ~]# firewall-cmd --list-services --zone=public
dhcpv6-client ssh
[root@host1 ~]#
```

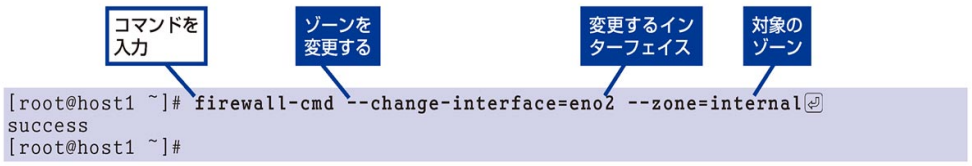
次のページに続く

インターフェイスのゾーンの変更

インターフェイスを異なるゾーンへ変更したい場合には、firewall-cmdに--change-interfaceを付けて実行します。下の例ではeno2をinternalゾーンへ変更します。変更する場合には対象のゾーンを指定する必要があります。

この場合も、再起動後も有効になるように --permanentオプションを付けて再実行しましょう。

1 ゾーンを変更する




```
[root@host1 ~]# firewall-cmd --change-interface=eno2 --zone=internal
success
[root@host1 ~]#
```

2 再起動後にも変更を有効にする


```
[root@host1 ~]# firewall-cmd --change-interface=eno2 --zone=internal --permanent
success
[root@host1 ~]#
```

設定の再読み込み

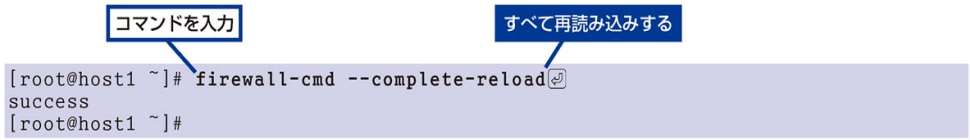
/etc/firewalld/services/ディレクトリにサービス定義を追加した場合などには、firewalldの設定の再読み込みを行います。設定の再読み込みには、firewall-cmdに--reloadオプションを付けて実行します。

また、firewall-cmdに--complete-reloadオプションを付けるとNetfilter Conntrackのステート情報も初期化されます。

1 設定を再読み込みする


```
[root@host1 ~]# firewall-cmd --reload
success
[root@host1 ~]#
```

2 ステート情報も初期化して再読み込みする



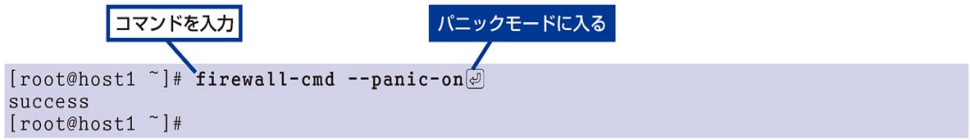

```
[root@host1 ~]# firewall-cmd --complete-reload
success
[root@host1 ~]#
```

パニックモード

firewalldでは、緊急時にすべてのネットワーク通信を遮断するパニックモードが用意されています。ただし、通常運用時にはサービス障害の原因となりますので、むやみに実行しないでください。

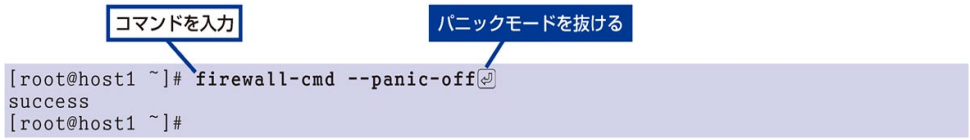
パニックモードに入るには、firewall-cmdに--panic-onを付けて実行します。また、パニックモードを抜けるには、firewall-cmdに--panic-offを付けて実行します。

1 パニックモードに入る




```
[root@host1 ~]# firewall-cmd --panic-on
success
[root@host1 ~]#
```

2 パニックモードを抜ける




```
[root@host1 ~]# firewall-cmd --panic-off
success
[root@host1 ~]#
```

5-6 コマンドラインにリモート接続するには

OpenSSH

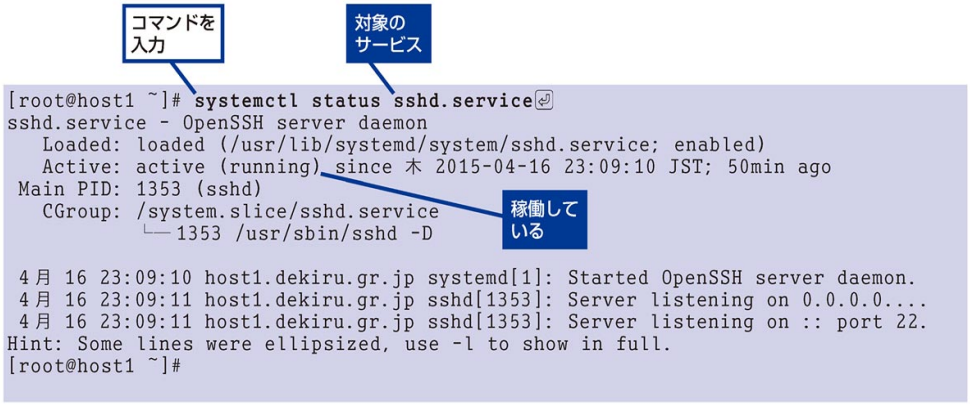
SSHは従来のTelnetに変わる安全なつなぎやすいリモートシェル環境を提供します。RHEL 7ではSSHサービスは、デフォルトで稼働しているサービスの1つです。また、ファイアウォールの設定でもデフォルトでSSHサービス（TCPポート：22）が許可されています。

SSHプロトコルは、デフォルトの状態でユーザー認証もデータも暗号化されているため、インターネット越しのリモートログインの場合にも広く使われています。また、パスワード認証のみでなく、さまざまな公開鍵認証、ワンタイムパスワードを併用した2要素認証にも対応しています。

OpenSSHの起動

OpenSSHの設定ファイルの/etc/ssh/sshd_configでは、何も変更しなくとも、デフォルトでOpenSSHサーバーとして稼働するようになっています。DVDからインストールした場合のデフォルトでは、管理者ユーザー rootでもパスワード認証を利用してログインできます。

OpenSSHの動作状況を確認するにはsystemctl statusコマンドにサービスのUnitである「sshd.service」を指定して実行します。もし起動していない場合は、「systemctl start sshd.service」で起動してください。

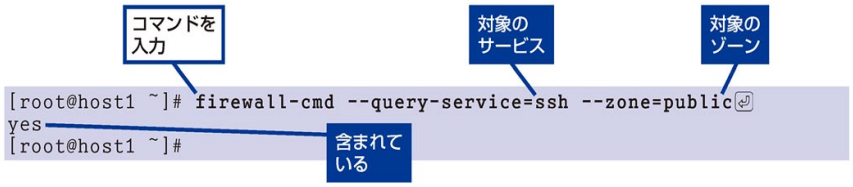



```
[root@host1 ~]# systemctl status sshd.service
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since 木 2015-04-16 23:09:10 JST; 50min ago
 Main PID: 1353 (sshd)
   CGroup: /system.slice/sshd.service
           └─1353 /usr/sbin/sshd -D

 4月 16 23:09:10 host1.dekiru.gr.jp systemd[1]: Started OpenSSH server daemon.
 4月 16 23:09:11 host1.dekiru.gr.jp sshd[1353]: Server listening on 0.0.0.0....
 4月 16 23:09:11 host1.dekiru.gr.jp sshd[1353]: Server listening on :: port 22.
Hint: Some lines were ellipsized, use -l to show in full.
[root@host1 ~]#
```

ファイアウォールの設定

ファイアウォールの設定においても、sshサービスはデフォルトで許可されています。設定を変更する必要はありません。

sshサービスがゾーンに含まれ、通信が許可されているかどうか確認するには、firewall-cmdコマンドに--query-serviceオプションを付けて実行します。結果が「yes」となっていれば通信が許可されています。もしも対象とするゾーンにsshサービスが含まれていなかった場合は、firewall-cmdコマンドに--add-serviceを付けて実行して、ゾーンに追加してください。

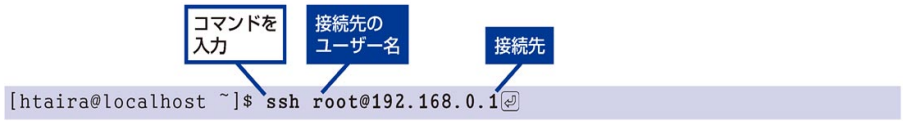


リモートからログインする（Linux、Mac OS X）

OpenSSHの動いているRHEL 7に、ほかのマシンからリモートでログインしてみましょう。

SSHのクライアントは、LinuxやMac OS Xでは、端末内で動作するsshコマンドが最初からインストールされています。

1 sshコマンドを実行する



次のページに続く

2 接続先を登録する

接続先の確認メッセージが表示される

このメッセージは初めて接続するサーバーのときにだけ表示される

「yes」と入力

```
The authenticity of host '192.168.0.1 (192.168.0.1)' can't be established.
ECDSA key fingerprint is 12:ad:9d:d6:72:7b:4e:be:ed:10:95:cb:54:e0:f4:52.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.1' (ECDSA) to the list of known hosts.
```

接続先が接続元のリストに登録された

メモ 2回目以降でこのメッセージが表示された場合は、偽物のサーバーの可能性があります。

3 パスワードを入力する

接続先サーバーのパスワードを尋ねられる

パスワードを入力

```
root@192.168.0.1's password:
Last login: Sat Oct 18 16:40:48 2014
[root@host1 ~]#
```

ログインした

4 ログアウトする

作業を終えてログアウトする

コマンドを入力

```
[root@host1 ~]# exit
ログアウト
Connection to 192.168.0.1 closed.
[htaira@localhost ~]$
```

HINT! パスワード認証方式と公開鍵認証方式

パスワード認証方式は、ブルートフォースアタックと呼ばれる総当り攻撃にてパスワードを連番で試された場合には、とても脆弱な認証方式です。インターネットに公開するサーバーには、より安全な公開鍵認証方式のみ利用するようにしましょう。秘密鍵が漏洩しない限り、不正侵入をするのが極めて難しくなります。具体的な方法は、本章のSTEP UPをご覧ください。

HINT! 接続できないときは

もしもSSHで接続できない場合は、以下の点を確認してください。

- IPアドレスやパスワードに不備がないか確認
- systemctlコマンドでsshサービスが稼働しているか確認
- OpenSSHが動いているホストで自分自身（localhost）に接続できるか確認

```
# ssh root@localhost
```

sshコマンドでlocalhostに接続できるのであれば、ファイアウォールで遮断されているか、通信キャリアによってポート規制を受けている場合があります。

リモートからログインする（Windows）

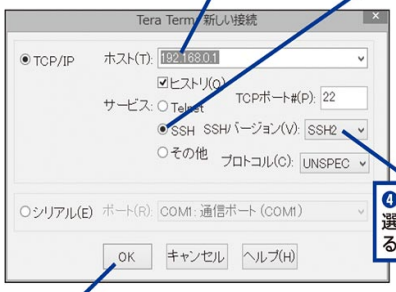
WindowsからSSHで接続するには、Tera TermやPuTTYといったSSHクライアントを使います。ここでは、すでにTera Termをインストールしてある場合を説明します。

1 サーバーを指定する

❶Tera Termを起動

❷接続先を入力

❸[SSH]が選択されていることを確認



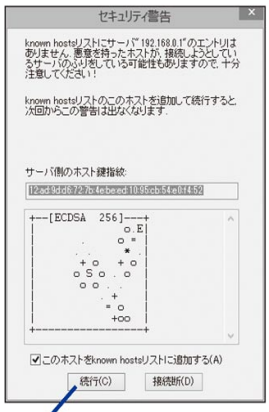
❹[SSH2]が選択されていることを確認

❺[OK]をクリック

2 サーバーを指定する

接続先の確認メッセージが表示される

このメッセージははじめて接続するサーバーのときにだけ表示される



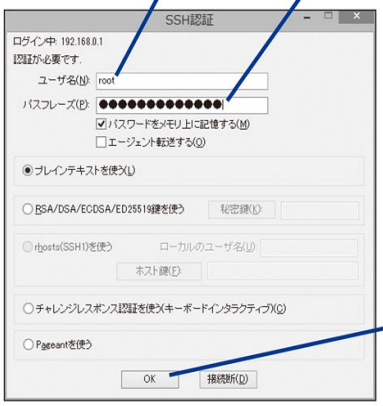
[続行]をクリック

接続先が登録される

3 サーバーを指定する

❶接続先のユーザー名を入力

❷パスワードを入力



❸[OK]をクリック

4 サーバーを指定する

接続先のプロンプトが表示された

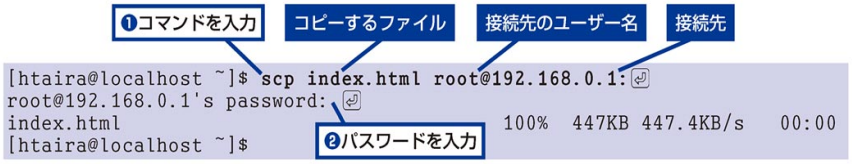
次のページに続く

scpでファイルをコピーする

ネットワーク経由でローカルからリモートホストへファイルをコピーする仕組みとして、SSHプロトコルを利用したscp（secure copy）があります。

ローカルからリモートへのコピー

ローカルからリモートホストへファイル（index.html）をコピーする場合には、次のコマンドを実行します。



リモートからローカルへのコピー

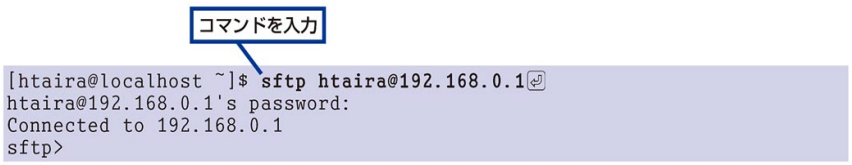
リモートホストからローカルへファイル（index.html）をコピーする場合には、次のコマンドを実行します。

sftpでファイルをコピーする

scpの他にSSHプロトコルを使いファイル操作を行うためのsftp（secure ftp）という仕組みがあります。scpコマンドはファイルのコピーしか行えませんが、sftpコマンドではファイルの削除やファイル名の変更、パーミッションの変更など、ftpコマンドを代替するファイル操作が行えます。

sftpの起動

sftpコマンドを実行し、「sftp>」というプロンプトになれば、リモートホスト上のファイル操作が行えます。



ファイル一覧

ファイルの一覧を表示するには、Linuxのコマンドと同じくlsコマンドを使います。

コマンドを入力

```
sftp> ls -l
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 ダウンロード
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 テンプレート
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 デスクトップ
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 ドキュメント
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 ビデオ
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 公開
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 画像
drwxr-xr-x    2 htaira   htaira          6 Apr 17 00:08 音楽
sftp>
```

ファイルのアップロード

ファイルをアップロードするには、putコマンドを使います。

コマンドを入力

```
sftp> put index.html
index.html                                    100%  447KB 447.4KB/s   00:00
sftp>
```

ファイルのダウンロード

ファイルをダウンロードするには、getコマンドを使います。

コマンドを入力

```
sftp> get index.html
/home/htaira/index.html                       100%  447KB 447.4KB/s   00:00
sftp>
```

ファイルの削除

ファイルを削除するには、Linuxのコマンドと同じくrmコマンドを使います。

コマンドを入力

```
sftp> rm index.html
Removing /home/htaira/index.html
sftp>
```

接続の終了

リモートホストから切断するには、quitコマンドを実行します。

コマンドを入力

```
sftp> quit
[htaira@localhost ~]$
```

STEP UP

OpenSSHのセキュリティを高めるには

OpenSSHには、パスワード認証や公開鍵認証方式など、さまざまな認証方式があります。外部に公開するサーバーのSSHサービスは、基本的には公開鍵認証方式しか受け付けないようにしてください。

パスワード認証方式は、ブルートフォースアタックと呼ばれる総当り攻撃にてパスワードを連番で試された場合には、とても脆弱な認証方式です。特に、ユーザーのパスワードに一般的な英単語や生年月日などを設定していた場合には、すぐに不正侵入されてしまうことでしょう。

パスワード認証を無効にする場合には、OpenSSHの設定ファイル/etc/ssh/sshd_configのPasswordAuthenticationの項目を「no」にします。

```
PasswordAuthentication no
```

「no」に変更

設定の反映にはsshd.serviceの再起動が必要です。

```
[root@host1 ~]# systemctl restart sshd.service
[root@host1 ~]#
```

コマンドを入力

しかし、パスワード認証を無効化したとしても、アタックを試みた痕跡が/var/log/secureに残ると思います。そこでrootユーザーのSSH経由での直接ログインを禁止し、SSHを許可するユーザーを列挙してしまい、特定のユーザーでなければ認証フェーズまで到達しないようにします。

OpenSSHの設定ファイル/etc/ssh/sshd_configの「# Authentication:」の行のすぐ下に、AllowUsersの記述を追記します。

```
# Authentication:
AllowUsers htaira
```

1行を追加

メモ ユーザー名は記述例です。実際のユーザー名を指定してください。

メモ 複数ユーザーを指定するには、半角スペースで区切って記述します。

こちらも設定の反映にはsshd.serviceの再起動が必要です。

```
[root@host1 ~]# systemctl restart sshd.service
[root@host1 ~]#
```

コマンドを入力

第6章 簡易DNSサーバーを作る

本章では、LAN内部で簡易DNSサーバーを簡単に作って運用できるdnsmasqを紹介します。また、コマンドライン環境で使われる一般的なテキストエディターであるviコマンドの使い方も紹介します。

●この章の内容

6-1 LANの中に簡易DNSサーバーを作ろう

dnsmasqの概要

RHELに含まれるDNSサーバーといえば「BIND」が有名です。しかし、検証環境で動かすアプリケーションがDNSを要求することも多々あります。そのために、LANの内部だけで参照できるようなDNSサーバーを作るには、BINDは大袈裟すぎます。また、アクセスする側でホスト名とIPアドレスを管理する/etc/hostsを、全マシンにコピーして運用する方法もありますが、管理が煩雑になります。

そこでこの章では、LANの内部で簡易DNSサーバーを作って簡単に設定できるdnsmasqを解説します。

/etc/hostsで設定

dnsmasqがどのように簡単なのか、その仕組みを解説します。

DNSサーバーとしては「BIND」が有名です。BINDはDNSの機能をフルに実装したサーバーです。そのため、設定には複雑なゾーンファイルを記述する必要があります。

それに対してdnsmasqは、/etc/hostsに記述されている情報からDNSの情報を構成します。/etc/hostsはLinuxにおいて、ほかのホストの名前を独自に設定してそのマシンで参照するファイルです。/etc/hostsでは、IPアドレスとホスト名の組を並べたホストテーブルだけを記述します。

そのためdnsmasqでは、/etc/hostsを書き換えられる程度のスキルさえあれば、DNSサーバーを用意できます。

/etc/hostsの例

```
127.0.0.1   localhost localhost.localdomain
192.168.0.1 host1.dekiru.gr.jp
```

IPアドレス　ホスト名

/etc/resolv.confで上位を参照

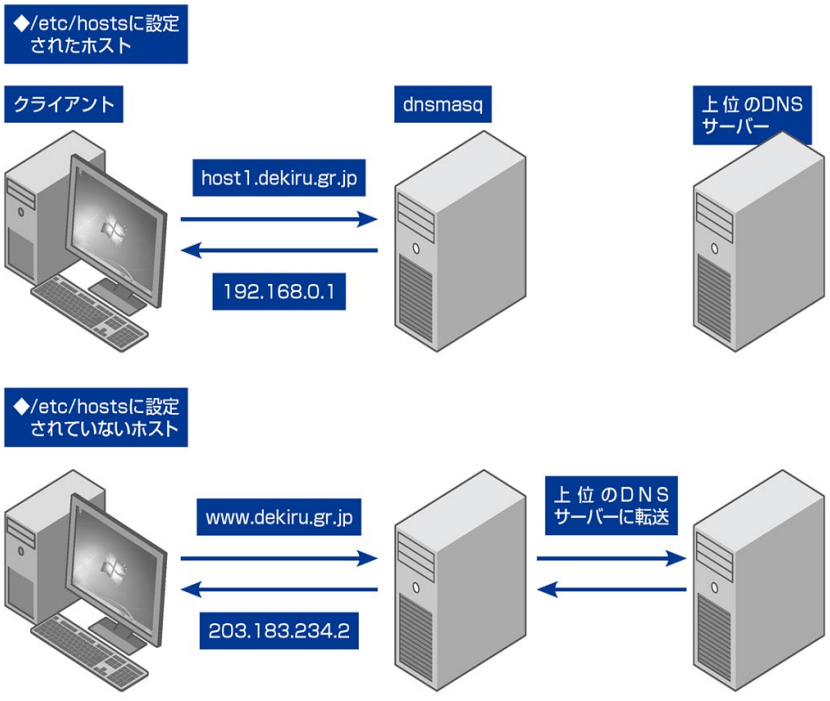
dnsmasqでは、/etc/hostsに記述されていないホスト名が問い合わせられた場合には、上位のDNSサーバーへ転送します。これによってdnsmasqは、/etc/hostsの最小限の設定だけでDNSサーバーの役割を果たします。

dnsmasqでは上位のDNSサーバーを、/etc/resolv.confで設定します。/etc/resolv.confは、dnsmasqに限らず、そのホストからネットワーク接続するときに参照するDNSサーバーを記述したファイルです。

たとえば、/etc/resolv.confに次のように記述していた場合は、4行目に記述されている8.8.8.8が上位のDNSサーバーとなり、dnsmasqがDNSの問い合わせを転送します。

/etc/resolv.confの例

```
domain localdomain
search localdomain dekiru.gr.jp
nameserver 127.0.0.1
nameserver 8.8.8.8
```



6-2 簡易DNSサーバーを作るには

dnsmasqのインストール

dnsmasqはインストール後に、特に設定しなくてもすぐに動かすことができます。この点がdnsmasqの一番の魅力です。dnsmasqの管理外のホスト名に対しては、上流のDNSサーバーにクエリを転送して中継参照する仕組みもあります。

dnsmasqはRPMパッケージとして提供されているため、dnsmasqのインストールは簡単です。多くの場合は最初からインストールされていますし、後からインストールする場合でも、それほど難しい作業はありません。

このレッスンでは簡易DNSサーバーdnsmasqのインストールと有効化方法を解説します。

dnsmasqのインストール

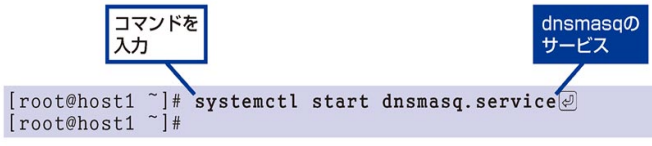
dnsmasqは、最小限環境でインストールした場合でも最初からインストールされています。もし何らかの理由で再インストールする場合には、dnsmasqパッケージをインストールしてください。

```
[root@host1 ~]# yum install dnsmasq
```

dnsmasqの起動

dnsmasqは初期状態では起動していません。そこで、systemctlコマンドでdnsmasq.serviceを起動します。また、ほかのマシンから問い合わせができるように、firewall-cmdを使って外部からのDNS問い合わせをファイアウォールで許可します。

1 dnsmasqを起動する




```
[root@host1 ~]# systemctl start dnsmasq.service
[root@host1 ~]#
```

HINT!

/etc/hostsファイルではダメなのか

Linuxに備わっている名前解決の仕組みとして、/etc/hostsファイルを使うという方法もあります。名前解決をしようとするマシンの/etc/hostsにホストを登録しておけば、DNSサーバーに問い合わせることなく、マシン内で名前を解決できます。
しかし、/etc/hostsは、名前解決をするマシンにそれぞれホストの情報を登録する必要があります。1つのホストが増えるたびに、1行を追記して、すべてのマシンに配布しなくてはなりません。頻繁にホストの追加を行う環境であれば、間違いなく運用が破綻します。BINDのゾーンファイルを記述することが面倒なのが理由であれば、dnsmasqをうまく活用しましょう。

2 システム起動時にも起動させる

コマンドを入力

```
[root@host1 ~]# systemctl enable dnsmasq.service
ln -s '/usr/lib/systemd/system/dnsmasq.service' '/etc/systemd/system/multi-user.
target.wants/dnsmasq.service'
[root@host1 ~]#
```

3 DNSでのアクセスを許可する

コマンドを入力　　DNSでのアクセス

```
[root@host1 ~]# firewall-cmd --add-service=dns --zone=public
success
[root@host1 ~]#
```

4 システム起動時にも許可させる

コマンドを入力

```
[root@host1 ~]# firewall-cmd --add-service=dns --zone=public --permanent
success
[root@host1 ~]#
```

6-3 テキストファイルを編集するには

vi

vi（visual editor）は、LinuxなどUNIX系OSの環境で歴史のある、一般的なテキストエディターです。約40年の間、使い続けられています。

RHEL7にはviの改良版のVim（vi improved）が搭載されています。最小限のサーバー構成でシステムをインストールした場合でも、Vimの最小版であるviコマンドは最初から入っています。viコマンドの使い方を覚えておけば、さまざまな設定ファイルの編集ができますので、ここでマスターしましょう。なお、すでにviなどのエディターを使いこなせる方は、このレッスンを読み飛ばして次に進んでください。

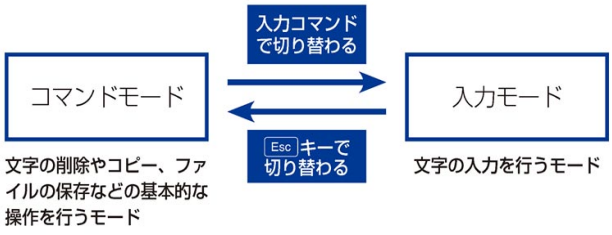
viの2つのモード

viには、主に「入力モード」と「コマンドモード」の2つのモードがあります。この2つのモードを切り替えながらファイルの編集作業を行います。

viの起動時には、コマンドモードになっています。コマンドモードでは、多くのキーにそれぞれ操作が割り当てられています。たとえば、「l」のキーを押すとカーソルが右に、「h」のキーを押すとカーソルが左に動きます。このように、すべての操作がキーボードのみで実行でき、マウス操作は不要です。

コマンドモードで「i」、「a」、「o」などのキーを押すと、入力モードに切り替わります。入力モードでは、入力した文字がファイルに入力されます。入力モードでEscキーを押すと、またコマンドモードに戻ります。

なお、コマンドモードで「:」のキーを押すと、画面の最下行でLinuxのコマンドラインのように文字列でviのコマンドを入力する状態になります。これも一種のモードで、「コマンドラインモード」と呼ばれることもあります。



viを操作する

1 viを起動する

コマンドを入力

編集するファイル名

```
[root@host1 ~]# vi hello.txt
```

viが起動して空のファイルが開かれる

```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"hello.txt" [New File]
```

2 入力モードに移る

「i」キーを押す

```
-- INSERT --
```

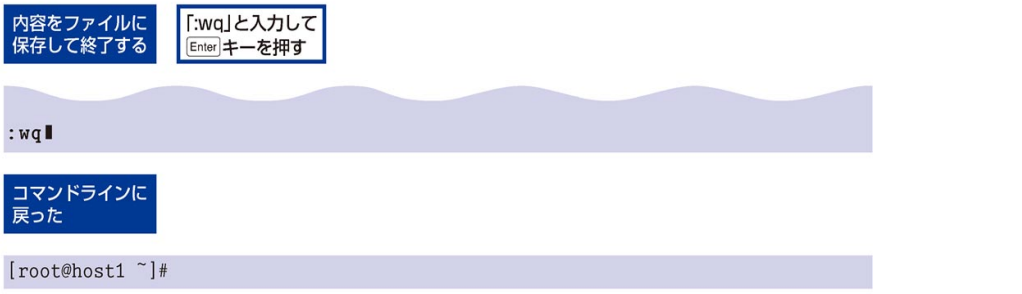
画面下端に「INSERT」と表示された

入力モードに切り替わった

次のページに続く

3 文字を入力する

4 保存して終了する



5 もう一度viで開く

6 文字を追加する

7 保存しないで終了する

変更内容をファイルに保存しないで終了する

「:q」と入力して Enter キーを押す

```
:q!
```

コマンドラインに戻った

```
[root@host1 ~]#
```

HINT!

viで操作が分からなくなったときは

設定ファイルの編集中に操作が分からなくなった場合には、Esc キーを押して、:q!で保存せずに抜けましょう。打ち込んだ文字は、もう一度打ち直せばどうにかなりますし、設定ファイルを破壊するよりはよいでしょう。
なお、viのコマンドの中に:helpというコマンドが用意されており、オンラインヘルプを見ることもできます。

viの主なコマンド

テキスト入力（コマンドモード→入力モード）

i	挿入（カーソル位置から入力）
I	行頭に挿入
a	追記（カーソルの次から入力）
A	行末に挿入
R	現在のカーソル位置から行末までを上書き変更
o	カーソル位置の行の後に新しい行を作成
O	カーソル位置の行の前に新しい行を作成

カーソル移動

← or h	左へ移動
→ or l	右へ移動
↑ or k	上へ移動
↓ or j	下へ移動
0	行頭へ移動
^	空白を除く行頭へ移動
$	行末へ移動
%	対応する括弧へ移動
w	次の単語の語頭へ移動
e	単語の末尾へ移動

行移動

gg	最初の行へ移動
G	最終の行へ移動
11G	11行目へ移動

削除・コピー・貼り付け

x	カーソル位置の1文字を削除
X	カーソルの左の1文字を削除
dw	現在のカーソル位置の単語を削除
dd	現在のカーソル位置の行を削除
D	現在のカーソル位置から行末まで削除
yy or Y	カーソル位置の行をコピー
3yy	カーソル位置から3行をコピー
p	貼り付け
p	カーソル位置の前に貼り付け

ファイル操作

:e *ファイル名*↵	ファイルを開く
:w↵	ファイルを上書き保存する
:w *ファイル名*↵	ファイルを名前を付けて保存する
:q↵	終了する
:q!↵	強制的に終了する
:wq↵ or ZZ	ファイルを上書き保存して終了する

6-4 dnsmasqを設定するには

/etc/hostsの編集

/etc/hostsファイルは通常ローカルホスト内での名前解決のために利用しています。

このレッスンでは、前のレッスンで紹介したテキストエディターのviを使って、実際にファイルを編集していきます。

/etc/hostsファイルは、システムが稼働する上で必要なファイルの1つですが、多少記述を間違えても、システムがクラッシュするようなことはないでしょう。viの編集操作の練習にも、ちょうどよい題材です。

操作に分からなくなったらcpコマンドでバックアップしていたファイルから戻してください。

/etc/hostsを編集する

dnsmasqでは、登録するホスト名とIPアドレスの対の情報を、/etc/hostsに記述します。新しいホストの情報を追記してみましょう。

1 設定ファイルをコピーする

ファイルをコピーしてバックアップする　コマンドを入力

```
[root@host1 ~]# cp -p /etc/hosts /etc/hosts.orig
[root@host1 ~]#
```

2 設定を追加する

設定ファイルをviで開く　❶コマンドを入力

```
[root@host1 ~]# vi /etc/hosts
```

ファイルが開かれた　❷「G」キーを押して最後の行へ移動　❸「o」キーを押して入力モードに切り替え　新しい行の作成になる

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.1 host1.dekiru.gr.jp
```

❹IPアドレスとホスト名を入力　❺Escキーを押す　コマンドモードへ切り替わる

3 保存して終了する

ファイルに保存して終了する

「:wq」と入力してEnterキーを押す

```
:wq
```

コマンドラインに戻る

4 設定を反映する

dnsmasqサービスを再起動する

コマンドを入力

```
[root@host1 ~]# systemctl restart dnsmasq.service
[root@host1 ~]#
```

HINT!

設定ファイルのバックアップをとる

設定ファイルを間違えて記述してしまい、元の記述も分からなくなった! ということがないように、設定ファイルを編集する前にcpコマンドでバックアップをとる習慣を身につけましょう。本書では、既存の設定ファイルを編集する必要がある場面では、手順に随所にcpコマンドによるバックアップを入れてあります。

dnsmasqを参照する

LAN内のPCからDNSサーバーとしてdnsmasqに問い合わせるようにするには、参照するDNSサーバーの設定を変更する必要があります。

1台のPCから参照できるようにするのであれば、PCで設定を変更します。LAN内のPCがみなdnsmasqに問い合わせるようにするには、ブロードバンドルーターなどDHCPサーバーになっている機器で、LAN側のDNSサーバー設定を変更します。

設定が変更されたら、PCからhost1.dekiru.gr.jpにpingなどでアクセスして確認します。

コマンドを入力

```
C:¥Users¥htaira>ping -n 3 host1.dekiru.gr.jp

host1.dekiru.gr.jp [192.168.0.1] に ping を送信しています 32 バイトのデータ:
192.168.0.1 に ping を送信しています 32 バイトのデータ:
192.168.0.1 からの応答: バイト数 =32 時間 =1ms TTL=64
192.168.0.1 からの応答: バイト数 =32 時間 <1ms TTL=64
192.168.0.1 からの応答: バイト数 =32 時間 <1ms TTL=64

192.168.0.1 の ping 統計:
    パケット数: 送信 = 3、受信 = 3、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 = 1ms、平均 = 0ms

C:¥Users¥htaira>
```

STEP UP

特定のサーバーにアクセスできないようにする

dnsmasqは簡易DNSサーバーということもあって、/etc/hostsに他人のドメインのホストも記述できるようになっています。これを使って、外部のサーバーのホスト名に対応するIPアドレスを、LAN内では別のものに差し替えることもできます。

たとえば、LAN内のユーザーが特定のサイトにアクセスできないようにしたい場合に使えます。dnsmasqが稼働するホストの/etc/hostsに、差し替えるサイトのホスト名と、IPアドレスの「127.0.0.1」を組にして設定します。すると、ユーザーがWebブラウザーなどでそのサイトにアクセスしようとすると、dnsmasqに問い合わせた結果、IPアドレス「127.0.0.1」、つまりWebブラウザーの動いているマシンにアクセスすることになります。これにより、対象のホストにアクセスできないようになります。

ただし、厳密なファイアウォールではないので、IPアドレスを直接指定されたり、違うDNSサーバーを参照されたりした場合にはアクセスできてしまいます。少しでも抑制効果はあることでしょう。

この仕組みを応用すると、サーバーのホスト名を一部のテスト用端末からのみ差し替えて、テスト用サーバーに向けるといったこともできます。

/etc/hostsの設定例

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.1 host1.dekiru.gr.jp


127.0.0.1   akiba-pc.watch.impress.co.jp
```

「akiba-pc.watch.impress.co.jp」にアクセスできないようにする

第7章 Webサーバーを作る

Webブラウザーに対してWebコンテンツを提供するには、HTTPプロトコルで通信するWebサーバーを用意する必要があります。この章では、WebサーバーであるApache HTTP Serverの構築方法について説明します。

●この章の内容

7-1 Webサーバーを作ろう

Apacheの概要

Apache HTTP Server（Apache）は世界中のWebサーバーで古くから広く使われており、RHEL 7にも含まれています。Apacheは比較的簡単に動作し、モジュールでさまざまな機能が追加できます。

配信場所となるディレクトリにHTMLや画像などのコンテンツを格納すれば、ApacheがHTTPプロトコルにてクライアントとやりとりします。もし、Webサーバーにグローバルipが付与されているか、ルーターでポート転送が設定されていれば、Apacheを起動した瞬間から世界中にコンテンツが公開されます。

HTTPとURL

HTTP（Hypertext Transfer Protocol）は、WebブラウザーとWebサーバーとの間でHTMLファイルやCSSファイルなどのWebコンテンツを送受信する際に用いられる通信プロトコルです。HTTPで通信を行うHTTPサーバーのことをWebサーバーとも呼びます。HTTPではテキスト形式のHTMLファイル以外にも、GIFやJPEGといったバイナリ形式のファイルも送受信できます。

日頃よく目にする「http://www.impress.co.jp/」といったアドレスは、URL（Uniform Resource Locator）と呼ばれ、インターネット上の住所です。その中でも「http://」から始まる場合は、HTTPプロトコルで通信することを意味しています。

手元にWebサーバーを作ることによってローカルネットワークでのみ利用できるWebサーバーを作ることもできます。また、グローバルIPアドレスを付与することで、インターネット経由でもアクセスできるようになります。

URLの構造

http://www.dekiru.jp:80/about/index.html

プロトコル　ホスト名　ポート　パス名

HINT!

RPMパッケージを使う

世の中の参考書では、The Apache Software Foundationからソースを取ってきて、コンパイルしてから利用するという古典的なインストール方法が記載されていることがあります。しかし、この方法ではApacheがRed Hatのサポート外となり、自己責任となります。Red Hatの提供するパッケージ以外の利用は控えてください。日々報告される脆弱性情報を毎日ウォッチしてコンパイルし直すのは現実的な運用とは言えません。Red Hatが配布するRPMパッケージをお使いください。

HINT!

RHELのApacheのバージョン

RHELに含まれるApacheのバージョンは、RHELのメジャーリリースが出荷されたタイミングで採用されたApacheのバージョンがベースとなります。メジャーリリース以降のセキュリティ修正やバグ修正は、ベースのバージョンにバックポートという形で移植されます。RHEL 7で採用されているApacheは2.4.6で、一見古いバージョンに思えますが、きちんと修正アップデートを適用すると最新のコミュニティ版で見つかった脆弱性やバグに対する修正がバックポートされています。

Apache ってなに？

Apacheとは、Apache Software FoundationのApache HTTP Server Projectで開発が行われているオープンソース・ソフトウェアのWebサーバーです。正式名称はApache HTTP Serverですが、一般的にApacheと略して呼ばれます。

Apache Software Foundationには、これ以外にもアプリケーションサーバーのApache Tomcatや、分散データベースソフトウェアのApache Cassandra、ビックデータ処理のApache Hadoopなどもあります。

Apacheは1995年に開発が始まり、UNIX系OSやWindows、IBM OS/2、Novell NetWareなどさまざまなOSに移植されて多くのユーザーに愛されてきました。もちろん、Linuxでも利用可能であり、RHELの過去すべてのメジャーリリースに収録されています。

RHEL 7に搭載されているApache 2.4は、前バージョンの2.2と比べて、次のような特徴があります。

- 同時アクセスを処理する機構のマルチプロセッシングモジュール（MPM）をローダブルモジュールにできるようになった
- MPMの1つとしてEvent MPMが正式サポートされた
- 一部のI/O処理が非同期処理となった
- リバースプロキシーの設定を動的に変更できるようになった
- メモリー消費量が低減した
- モジュールごと、ディレクトリごとの細かなログレベルの制御が可能になった

なお、RHEL 7に含まれるApache 2.4（httpd-2.4.6）は、The Apache Software Foundationが2.4系の開発のメインストリームのサポートを終了したとしても、RHEL 7のサポートライフサイクルが終了するまで（現在2024年6月30日を予定）利用できます。

7-2 Webサーバーを作るには

Apacheのインストール

Apacheはインストール後にすぐにWebサーバーとして動かすことができます。RHEL 7ではRPMパッケージで提供されているため、yumコマンドでインストールが完了します。また、モジュールを追加することでApache上でPHPやPythonなどのスクリプトを動かすこともできます。そのほかにもさまざまな認証や、コンテンツを配信時に圧縮する機能、プロキシー機能を提供するモジュールなどもあります。このレッスンではApacheのインストール方法と、有効化方法、DocumentRootの概念について解説します。

インストールする

1 Apacheをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install httpd
読み込んだプラグイン：langpacks, product-id, subscription-manager
```

httpdのパッケージと必要なパッケージが表示される

```
================================================================================
 Package          アーキテクチャー
                                バージョン              リポジトリー                容量
================================================================================
インストール中：
 httpd            x86_64       2.4.6-31.el7         rhel-7-server-rpms       1.2 M
依存性関連でのインストールをします：
 apr              x86_64       1.4.8-3.el7          rhel-7-server-rpms       103 k
 apr-util         x86_64       1.5.2-6.el7          rhel-7-server-rpms        92 k
 httpd-tools      x86_64       2.4.6-31.el7         rhel-7-server-rpms        79 k
 mailcap          noarch       2.1.41-2.el7         rhel-7-server-rpms        31 k

インストール容量： 4.3 M
Is this ok [y/d/N]: y
Downloading packages:
```

❷「y」と入力してEnterキーを押す

```
完了しました！
[root@host1 ~]#
```

インストールが完了した

設定する

RHEL 7のApacheの設定ファイルは/etc/httpd/conf/httpd.confです。何も変更しなくても動くようになっていますが、サーバー管理者のメールアドレスとホスト名は変更しておきましょう。

1 元の設定ファイルをバックアップする

変更する前の設定ファイルをコピーしておく

コマンドを入力

```
[root@host1 ~]# cp -p /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.orig
[root@host1 ~]#
```

2 設定を書き換える

❶コマンドを入力

```
[root@host1 ~]# vi /etc/httpd/conf/httpd.conf
```

viでファイルが開かれた

```
#
# This is the main Apache HTTP server configuration file.  It contains the

#
ServerAdmin webmaster@dekiru.gr.jp

#
ServerName host1.dekiru.gr.jp
```

❷この行に管理者のメールアドレスを入力

❸この行の先頭の「#」を消してWebサーバーのホスト名を入力

❹上書き保存して終了

コンテンツを置く

/etc/httpd/conf/httpd.confの中に「DocumentRoot」という設定項目があります。

```
DocumentRoot "/var/www/html"
```

次のページに続く

Apacheで公開コンテンツを保存するディレクトリの最上位をDocumentRootと呼びます。DocumentRootで指定されているディレクトリにindex.htmlを代表とするコンテンツを配置してください。

このディレクトリに格納するコンテンツのファイルのパーミッションは、「755」や「644」など、ユーザー「apache」が読める値に設定してください。

また、コンテンツのファイルのSELinuxのセキュリティコンテキストは「httpd_sys_content_t」である必要があります。cpコマンドでファイルをコピーした場合は正しく設定されますが、mvコマンドでファイルを置いた場合には設定されないことがあり、HTTPのエラー「403 Forbidden」が発生します。DocumentRootに置いたファイルのセキュリティコンテキストを設定するには、セキュリティコンテキストを再設定する「restorecon」コマンドを実行するのが簡単です。

1 ファイルをDocumentRootに置く

すでに「index.html」を用意しているものとする

コマンドを入力

```
[root@host1 ~]# mv index.html /var/www/html
[root@host1 ~]#
```

2 セキュリティコンテキストを設定する

コマンドを入力

変更を表示する

ディレクトリの中のファイルを変更する

対象のディレクトリ

セキュリティコンテキストが設定された

```
[root@host1 ~]# restorecon -v -R /var/www/html
restorecon reset /var/www/html/index.html context unconfined_u:object_r:admin_home_
t:s0->unconfined_u:object_r:httpd_sys_content_t:s0
[root@host1 ~]#
```

起動する

1 Apacheを起動する

httpd.serviceを起動する

❶コマンドを入力

```
[root@host1 ~]# systemctl start httpd.service
[root@host1 ~]# systemctl enable httpd.service
ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.
wants/httpd.service'
[root@host1 ~]#
```

システム起動時に起動するようにする

❷コマンドを入力

2 ファイアウォールで許可する

HTTPでのアクセスを許可

❶コマンドを入力

```
[root@host1 ~]# firewall-cmd --add-service=http --zone=public
success
[root@host1 ~]# firewall-cmd --add-service=http --zone=public --permanent
success
[root@host1 ~]#
```

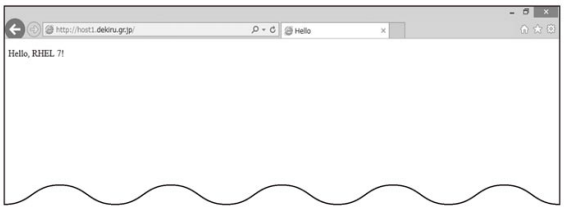
システム起動時に許可させる

❷コマンドを入力

3 Webブラウザーでアクセスする

ほかのマシンからWebブラウザーでアクセスしてみる

第6章でdnsmasqを設定してあればホスト名でアクセスできる



ホスト名を設定していない場合はIPアドレスを指定する

設置したコンテンツが表示された

アクセスできない場合は

もしもアクセスできない場合は、以下の点を確認してください。

- Apacheの設定ファイルに不備がないか確認
- systemctl statusコマンドでhttpdサービスが稼働しているか確認
- Apacheが動いているホストで下のようにcurlコマンドを使ってアクセス確認

curlコマンドでindex.htmlの内容が表示されるのであれば、Apacheは正常に稼働しています。ファイアウォールにてブロックされている場合や、Webブラウザーのプロキシー設定に問題がある場合が考えられます。

コマンドを入力

```
[root@host1 ~]# curl http://localhost/ | head -5
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   104  100   104    0     0    673      0 --:--:-- --:--:-- --:--:--   675
<!DOCTYPE html>
<html>
<head>
<title>Hello</title>
</head>
[root@host1 ~]#
```

7-3 Webの通信を暗号化するには1

SSL/TLSの概要

インターネットの普及とサービスの充実により、HTTPプロトコルを通じて、ショッピングやチケット予約、インターネットバンキングによる振込み、税金の確定申告などといろいろなことができるようになりました。

しかし、生活が便利になった反面で、多くの個人情報が日々通じてやりとりされています。そのため、通信が盗聴されないように身を守る仕組みが必要となります。

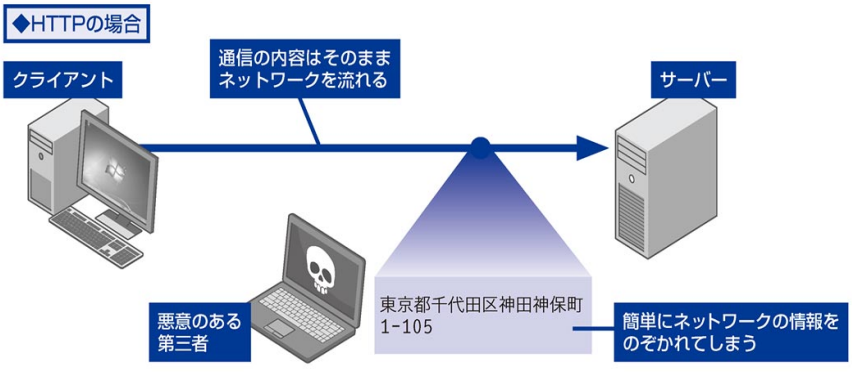
そこでHTTPプロトコルを暗号化する、SSL/TLSという仕組みをこのレッスンでは説明していきます。

SSLってなに？

HTTPプロトコルによる通信は、暗号化されていません。そのため、悪意のあるユーザーがネットワークをキャプチャリングして解析した場合、通信の内容のすべてを覗き見ることができてしまいます。ユーザーが見ているページや、入力フォームに入力した文字などが分かるため、入力フォームに入れた個人情報やクレジットカードまで取得できてしまいます。

そこで、HTTPプロトコルの通信を暗号化する仕組みとして、SSL（Secure Socket Layer）やTLS（Transport Layer Security）が用意されています。SSL/TLSで暗号化されたHTTPプロトコルの通信を「HTTPS」と表現する場合もあります。SSL/TLSは、個人情報を入力する必要のあるサイトを運営するのに必須と言えます。

なお、TLSはSSLプロトコルをベースとしたプロトコルです。SSL 3.0の次にあたるのがTLS 1.0です。SSLという名称が広く普及したため、SSLとTLSは特別な場合を除き総称してSSLと呼ばれます。本書でもSSLと記載します。



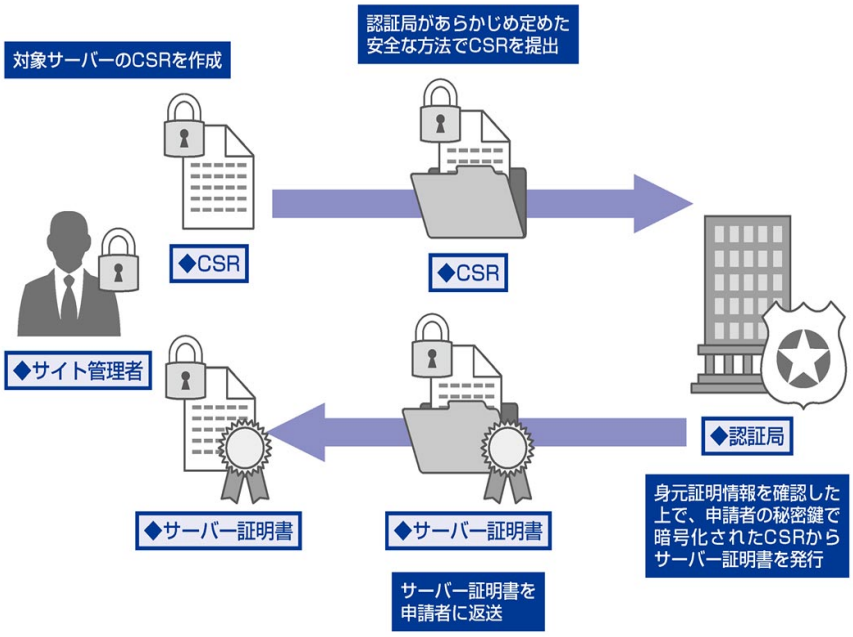
ApacheでSSLを使う

Apacheは、もちろんSSLに対応しています。暗号化通信を行うためには以下の事務手続きが必要です。

・OpenSSLのツールで秘密鍵ファイルを生成する
・OpenSSLのツールでサーバー証明書を発行するための署名要求書（CSR：Certificate Signing Request）を生成する
・SSL証明局（CA：Certificate Authority）に身元証明情報と一緒に署名要求書（CSR）を送付する
・SSL証明局によって署名されたサーバー証明書を受け取る
・SSL証明局によって署名されたサーバー証明書をApacheに設定する

気をつけなければならないのは、SSLのサーバー証明書の有効期限です。SSLで使用するサーバー証明書には1年間や数年などの有効期限があり、定期的に更新する必要があります。更新手続きは、期限前に行う必要があります。

また、更新手続きの際には、秘密鍵のパスフレーズが必要となります。忘れずに覚えておいてください。



7-4 Webの通信を暗号化するには2

サーバー証明書の取得手続き

WebサーバーでSSL/TLSによる暗号化通信に対応するためにはサーバー証明書というものが必要となります。サーバー証明書を入手するには、第三者機関である証明局に対象のWebサーバーの身元を証明してもらう必要があります。証明依頼をCSRと呼びます。証明局から署名付きのサーバー証明書を発行してもらって初めてSSL/TLSの暗号化通信ができます。

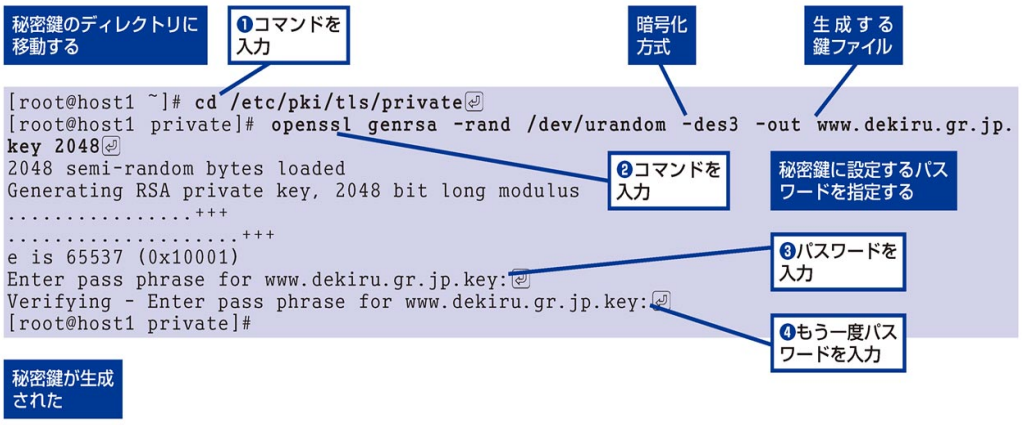
初めて証明局に申請する場合には、その手続きの複雑さに若干戸惑うこともあるでしょう。このレッスンではCSRを生成する上での具体的な作業と証明局に提出するまでの流れを説明します。

秘密鍵ファイルを生成する

RHEL 7の場合、秘密鍵ファイルはOpenSSLに含まれるopensslコマンドで生成します。

なお、秘密鍵ファイルの生成作業は/etc/pki/tls/privateディレクトリで行いましょう。ほかのディレクトリでは秘密鍵ファイルが他のユーザーから参照できてしまう可能性があります。

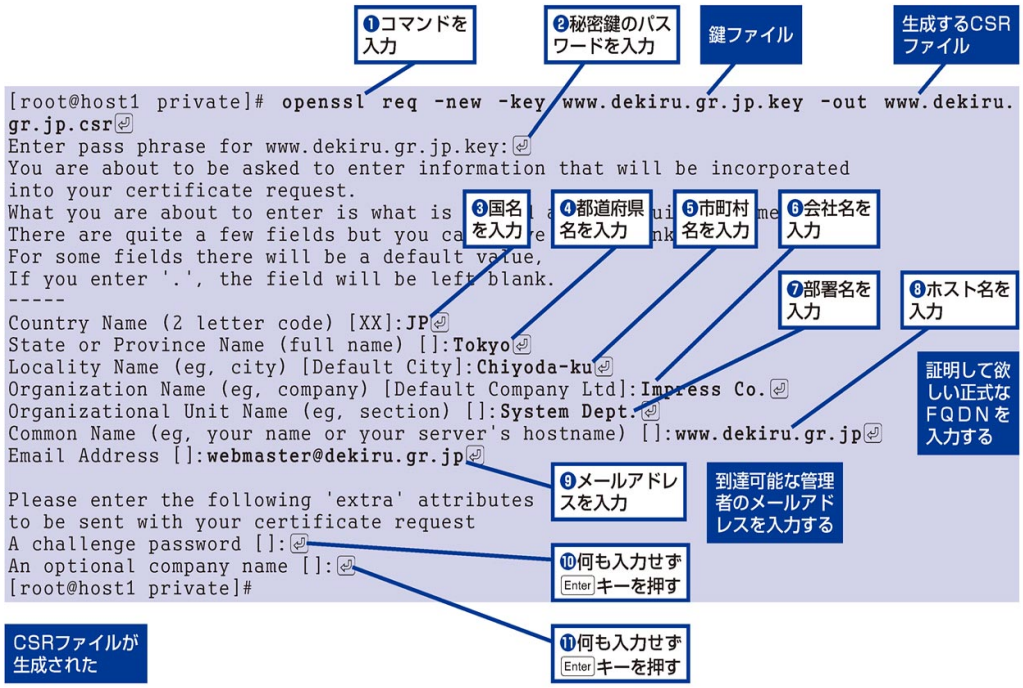
1 秘密鍵を生成する



CSRファイルを生成する

証明局に申請するために、CSR（Certificate Signing Request）という証明書の署名要求を行います。署名要求によって生成されるファイルをCSRファイルと呼びます。このCSRファイルを証明局へあらかじめ定められた方法で提出します。

2 CSRファイルを生成する



CSRファイルを提出する

準備ができたら、生成したCSRファイルを証明局の運営組織へ申請しましょう。運営組織としては、Symantec（旧VeriSign）、GeoTrust、Comodo、Digicertなどが有名で、日本国内ではGMO GlobalSign、SECOM TRUST SYSTEMSなどといった国内企業も運営しています。具体的な証明書発行のための手数料や提出方法は運営組織ごとに異なります。詳しくは証明局の運営組織にお問い合わせください。

7-5 Webの通信を暗号化するには3

サーバー証明書の設定

証明局（CA）から署名済みのサーバー証明書が返ってきたら、CSRを生成する際に利用した秘密鍵と一緒に、Apacheの設定ファイルで指定します。

その際にパスワード付きの秘密鍵を設定するとApache起動時に毎回パスワードを聞かれてしまいます。そのため、元の秘密鍵から事前にパスワードを解除した秘密鍵を作成し、このパスワード解除済みの秘密鍵をApacheに設定しましょう。なお、パスワード付き秘密鍵とパスワード解除済みの秘密鍵は、厳重に管理してください。

1 mod_sslをインストールする

ApacheのSSLモジュールをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install mod_ssl
読み込んだプラグイン：langpacks, product-id, subscription-manager
```

mod_sslのパッケージが表示される

```
================================================================================
 Package         アーキテクチャー
                                 バージョン              リポジトリー                    容量
================================================================================
インストール中：
 mod_ssl         x86_64          1:2.4.6-31.el7          rhel-7-server-rpms          99 k
```

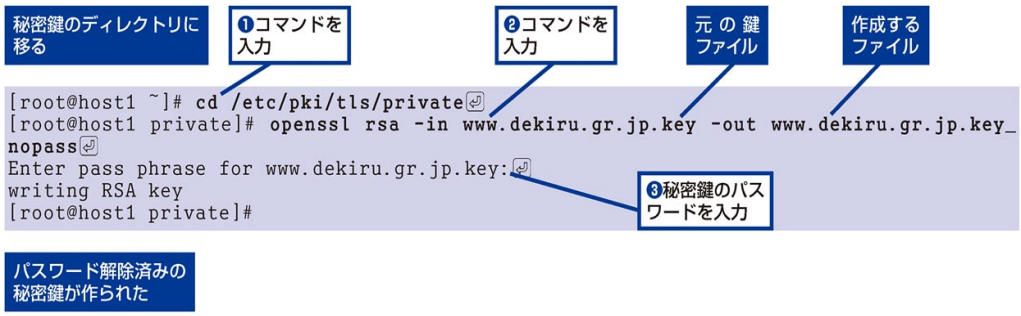
❷「y」と入力してEnterキーを押す

```
インストール容量： 219 k
Is this ok [y/d/N]: y
Downloading packages:
```

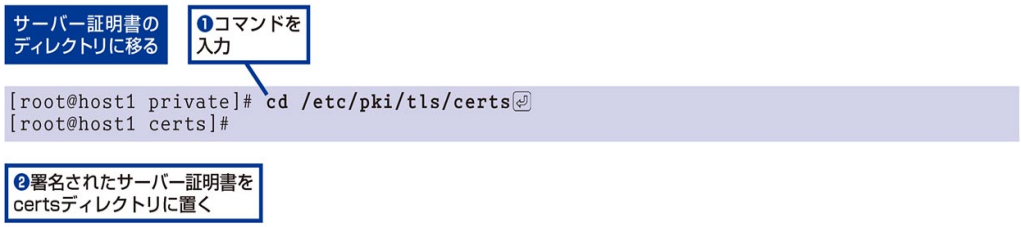
```
完了しました！
[root@host1 ~]#
```

インストールが完了した

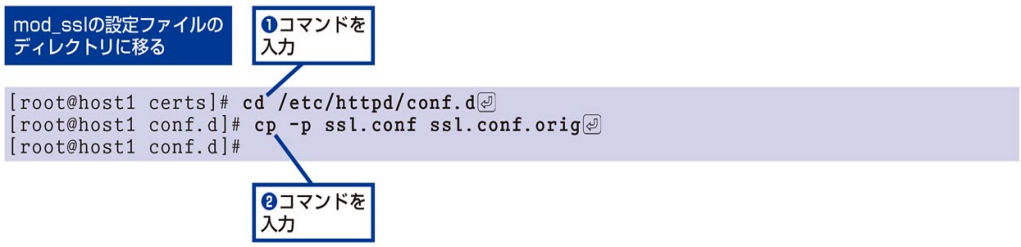
2 パスワードを解除した秘密鍵を作る



3 サーバー証明書を置く



4 設定ファイルのバックアップをとる



5 ApacheのSSLの設定ファイルを開く

次のページに続く

6 ホスト名を設定する

```
#DocumentRoot "/var/www/html"
#ServerName www.example.com:443
ServerName www.dekiru.gr.jp:443
```

1行を追加

CSR生成時に指定したFQDNを指定する

HINT!

OpenSSLの脆弱性「POODLE」の対策

RHEL 7で採用されているhttpd-2.4.6-18.el7では、2014年4月に発表されたOpenSSLの脆弱性「Heartbleed」については、リリース時に修正パッチがバックポートされており、対策済みです。
しかし、もう1つ大きな脆弱性「POODLE」への対策には、ApacheのSSL設定ファイルの変更が必要です。手順❼では、その対応をしています。

7 SSLv3.0を無効にする

```
#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv2 -SSLv3
```

「-SSLv3」を追加

8 サーバー証明書を指定する

```
# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate.  If
# the certificate is encrypted, then you will be prompted for a
# pass phrase.  Note that a kill -HUP will prompt again.  A new
# certificate can be generated using the genkey(1) command.
SSLCertificateFile /etc/pki/tls/certs/www.dekiru.gr.jp.crt
```

設置した署名済みサーバー証明書に変更

9 秘密鍵を指定する

```
# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file.  Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/pki/tls/private/www.dekiru.gr.jp.key_nopass
```

パスワード解除済みの秘密鍵に変更

HINT!

サーバー証明書の更新方法は？

SSLのサーバー証明書には失効期限があり、定期的に更新手続きが必要です。サーバー証明書の更新時にもCSRを再発行し、期限切れになる前に証明局へ提出します。更新時は約90日前からCSRを受け付けてもらえます。ギリギリになってCSRを申請すると更新手続きに何営業日か必要な場合があります。詳しくは、証明局の運営組織に問い合わせてください。
新しいサーバー証明書を受け取ったら、速やかにサーバー上のものを差し替えてApacheを再起動してください。更新手続きだけでは失効期限は延びません。

10 中間証明書を指定する

この手順は必要であれば実行する

```
#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt
SSLCertificateChainFile /etc/pki/tls/certs/intermediate.crt
```

❶1行を追加

証明局の中間証明書を指定する

❷上書き保存して終了

11 ファイアウォールで許可する

HTTPSでのアクセスを許可

❶コマンドを入力

```
[root@host1 private]# firewall-cmd --add-service=https --zone=public
success
[root@host1 private]# firewall-cmd --add-service=https --zone=public --permanent
success
[root@host1 private]#
```

システム起動時に許可させる

❷コマンドを入力

12 Apacheを再起動する

設定を反映させる

コマンドを入力

```
[root@host1 private]# systemctl restart httpd.service
[root@host1 private]#
```

STEP UP

SSLはどこまで安全なのか

暗号化通信のSSL/TLSが保護する対象は、Webブラウザーが起動しているクライアントから、HTTPSプロトコルを使ってアクセスするWebサーバーまでの間の通信です。もし、クライアントのキーボード入力が不正に記録されていた場合には、SSL/TLSでは保護できません。また、Webサーバーのローカルで保存されたファイルや、バックエンドに接続されているデータベースの中身も、SSL/TLSでは保護できません。

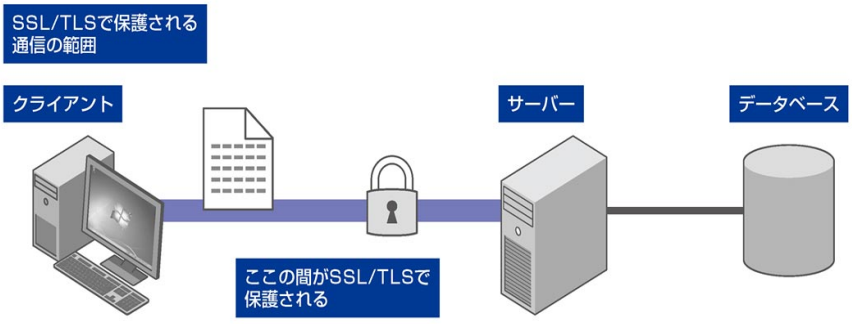
そのほかWebサーバーがSSLのプログラム自体に脆弱性が残ったまま運用されていると、表面的には暗号化通信が行われているように思えても、暗号が解読されてしまう場合もあります。したがって、稼働した後も定期的なセキュリティ修正の適用が重要です。

OpenSSLでは、HeartbleedやPOODLEと呼ばれる大きな脆弱性が報告されています。これらの脆弱性に共通するのは、Webサーバーとしては正常に動いていても、横からデータを盗まれる可能性があるという点です。

この2つの脆弱性について、Red Hatからは下に挙げたように情報や対応を提供しています。このような対応を行ってきちんと運用するのでなければ、暗号化通信していたとしても、悪意のあるユーザーからは通信内容が筒抜けです。

・Announcements#791943: OpenSSLのセキュリティーに関する重要な最新情報: CVE-2014-0160
https://access.redhat.com/announcements/791943

・Articles#1232403: POODLE: SSLv3.0脆弱性（CVE-2014-3566）
https://access.redhat.com/ja/articles/1232403



第8章 FTPサーバーを作る

ファイル転送プロトコルのFTPプロトコルを使うと、双方向にファイルを転送することができます。本章では、FTPのサーバーをvsftpdにより作ります。

●この章の内容

8-1 FTPサーバーを作るには

vsftpd

RHEL 7にはFTPサーバーとしてvsftpdが用意されています。インストールすると、デフォルトで匿名FTPサーバーとして動作するようになっています。ソフトウェアを配布する匿名FTPサーバーなどでは、まだFTPも多く使われています。

なおFTPプロトコルは、そのままでは認証もデータも暗号化されていないため、インターネット越しで利用する場合にはセキュリティ面で問題があります。重要なファイルを転送するときには、**レッスン5-6**で解説したscpやsftpを使いましょう。

1 vsftpdをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install vsftpd
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package        アーキテクチャー
                                バージョン              リポジトリー                    容量
================================================================================
インストール中：
 vsftpd         x86_64          3.0.2-9.el7          rhel-7-server-rpms          166 k
```

vsftpdのパッケージが表示される

```
インストール容量： 343 k
Is this ok [y/d/N]: y
Downloading packages:


完了しました！
[root@host1 ~]#
```

❷「y」と入力してEnterキーを押す

インストールが完了した

2 vsftpdを起動する

vsftpd.serviceを起動する

❶コマンドを入力

```
[root@host1 ~]# systemctl start vsftpd.service
[root@host1 ~]# systemctl enable vsftpd.service
ln -s '/usr/lib/systemd/system/vsftpd.service' '/etc/systemd/system/multi-user.target.
wants/vsftpd.service'
[root@host1 ~]#
```

システム起動時に起動するようにする

❷コマンドを入力

3 ファイアウォールで許可する

FTPでのアクセスを許可

❶コマンドを入力

```
[root@host1 ~]# firewall-cmd --add-service=ftp --zone=public
success
[root@host1 ~]# firewall-cmd --add-service=ftp --zone=public --permanent
success
[root@host1 ~]#
```

システム起動時に許可させる

❷コマンドを入力

4 FTPでアクセスする

ほかのマシンからWebブラウザーでアクセスしてみる

以下のURLにアクセス

ftp://（ホスト名）/

第6章でdnsmasqを設定してあればホスト名でアクセスできる

ホスト名を設定していない場合はIPアドレスを指定する

匿名FTPのディレクトリが表示された

メモ FTPサーバーで配布したいファイルがある場合には、/var/ftp/pub/ディレクトリ以下にファイルを配置します。

HINT!

SELinuxのコンテキストに注意

匿名FTPでコンテンツを公開するために、/var/ftp/pubにmvコマンドでファイルを移動した場合には、移動元のSELinuxコンテキストが引き継がれます。これにより、vsftpサーバーにアクセス権限がなく、FTPからアクセスしたときに存在が確認できなかったり、アクセスできなかったりする場合があります。そのため、cpコマンドでコピーしてファイルを置くのがよいでしょう。

8-2 ホームディレクトリにFTP経由でアクセスさせるには

vsftpdの設定

vsftpdでは一般ユーザーのホームディレクトリをFTP経由でアクセスさせるようにできます。これを許可するには、vsftpdの設定ファイルである/etc/vsftpd/vsftpd.confで「local_enable」を設定します。デフォルトで有効になってるはずですが、念のため確認しましょう。

また、SELinuxのブール値という設定で、vsftpがホームディレクトリにアクセスできるように設定します。

ただし、繰り返しますが、FTPでは認証もデータも暗号化されていないため、LAN内部で利用する程度に留めましょう。

1 元の設定ファイルをバックアップする

変更する前の設定ファイルをコピーしておく

コマンドを入力

```
[root@host1 ~]# cp -p /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.orig
[root@host1 ~]#
```

2 設定ファイルを開く

コマンドを入力

```
[root@host1 ~]# vi /etc/vsftpd/vsftpd.conf
```

HINT!

FTPで使うポート

ファイル転送プロトコルのFTPでは、コマンド用とデータ用で別々のTCPポートを利用します。コマンド用は21番ポートで固定されていますが、データ用として利用するポート番号は変動します。そのために、Linuxカーネルが FTPの通信を追跡して適切なポートを通過させる仕組みとして、nf_conntrack_ftpモジュールがあります。firewalldでftpサービスを有効にすると、21番ポートが許可されるだけではなく、nf_conntrack_ftpモジュールも読み込まれます。

HINT!

SELinuxのブール値とは

Linuxのセキュリティを最高レベルに高めるためのSELinuxという仕組みは、アプリケーションごとにポリシーを定義するため、昔は取り扱いにくいものでした。SELinuxのブール値という仕組みを使うと、ポリシーを記述する知識がなくても、システム利用時にポリシーの再コンパイルなしに、setseboolコマンドにてポリシーの一部の挙動を変更することができます。

3 匿名FTPサーバーを無効にする

viでファイルが開かれた

「NO」に変更

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
```

4 ホームディレクトリにアクセスする設定を確認する

❶「local_enable」が「YES」になっていることを確認

❷上書き保存して終了

```
# Uncomment this to allow local users to log in.
# When SELinux is enforcing check for SE bool ftp_home_dir
local_enable=YES
```

5 ホームディレクトリにアクセスする権限を有効にする

コマンドを入力

再起動後も有効にする

ftpでホームディレクトリにアクセスできるブール値

```
[root@host1 ~]# setsebool -P ftp_home_dir on
[root@host1 ~]#
```

6 設定変更を反映する

vsftpd.serviceを再起動する

コマンドを入力

```
[root@host1 ~]# systemctl restart vsftpd.service
[root@host1 ~]#
```

8-3 FTPクライアントから接続するには

ftpコマンド

ホームディレクトリにアクセスするには、FTP接続するときにユーザー名を指定してログインします。ここでは、FTPサーバーにFTPクライアントのftpコマンドをインストールし、localhostに接続して確認します。

localhostで接続を確認してみて問題なくつながれば、ほかのマシンからFTPクライアントで接続してみましょう。

なお、rootユーザーでのFTPのログインは禁止されています。rootユーザーでログインを試みると「530 Permission denied.」とエラーが返ってきます。

1 ftpをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install ftp
読み込んだプラグイン：langpacks, product-id, subscription-manager

================================================================================
 Package     アーキテクチャー
                           バージョン                リポジトリー                容量
================================================================================
インストール中：
 ftp         x86_64        0.17-66.el7             rhel-7-server-rpms          61 k

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

インストール容量： 96 k
Is this ok [y/d/N]: y
Downloading packages:

完了しました！
[root@host1 ~]#
```

ftpのパッケージが表示される

❷「y」と入力してEnterキーを押す

インストールが完了した

2 FTPでログインする

❶コマンドを入力

```
[root@host1 ~]# ftp localhost
Trying ::1...
Connected to localhost (::1).
220 (vsFTPd 3.0.2)
Name (localhost:root): htaira
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

❷ユーザー名を入力

❸パスワードを入力

FTPクライアントのプロンプトが表示された

3 ファイルを送信する

ftpを実行したマシンからFTPサーバーにファイルを送る

コマンドを入力

送信するファイル

```
ftp> put hello.txt
local: hello.txt remote: hello.txt
229 Entering Passive Mode (|||18076|).
150 Ok to send data.
226 Transfer complete.
6 bytes sent in 5.7e-05 secs (105.26 Kbytes/sec)
ftp>
```

ファイルが送信された

4 FTPクライアントを終了する

コマンドを入力

```
ftp> quit
221 Goodbye.
[root@host1 ~]#
```

Linuxのプロンプトに戻った

STEP UP

FTPのアクティブモードとパッシブモード

FTPプロトコルでは通常、TCPの21番ポートでセッションを確立し、TCPの20番ポートでデータ転送を行います。これをアクティブモードと呼んでいます。しかしアクティブモードでは、ファイアウォールやNATを超えてFTPでファイル転送する際に不都合が発生します。

そこでFTPには、クライアント側がデータ転送用のポートを明示的に指定する、パッシブモードというモードが用意されています。一般的に使われているFTPクライアントのほとんどは、パッシブモードでも接続できます。

クラウド上で利用するときの注意点

OpenStackや、Amazon EC2、Google Compute Engineなどのクラウドサービスにおいて、外部のIPアドレスと内部のIPアドレスとで異なるIPアドレスを持つ、スタティックNATされた環境が多くなってきました。その場合、アクティブモードではつながりますが、パッシブモードでは接続が失敗するケースがあります。

vsftpdにはパッシブモード時に外部向けのIPアドレスを明示的に指定するためのpasv_addressという変数があります。もしグローバルIPが1.2.3.4の場合、/etc/vsftpd/vsftpd.confで次のように指定してください。

```
pasv_address=1.2.3.4
```

また、クラウド環境内のセキュリティグループに許可ポートとして指定するTCPポートの範囲をvsftpdの設定で定義する必要があります。同じTCPポートの範囲をセキュリティグループとFTPクライアント側にも設定する必要があります。

```
pasv_min_port=50000
pasv_max_port=51000
```

第9章 ファイルサーバーを作る

コンピューターどうしがネットワークでつながるメリットのひとつに、ファイル共有があります。ファイル共有を行うためには、ファイルサーバーが必要です。この章ではLinux向けのファイルサーバー（NFS）と、Windows向けのファイルサーバー（Samba）の作り方を、それぞれ説明します。

●この章の内容

9-1 Linux用のファイルサーバーを作ろう

NFSの概要

Linuxでファイルサーバーと言えば、古くからある仕組みでNFS（Network File System）があります。RHEL 7にはNFSv4（Version 4）に準拠したNFSサーバーとNFSクライアントの実装が含まれています。

NFSを使うことで、他のLinuxからアクセスができる共有ディレクトリを提供（エクスポート）することができます。

NFSサーバーが提供する共有ディレクトリをローカルにマウントすることで、他のLinuxマシンの共有ディレクトリをローカルファイルシステムの一部であるかのように利用できます。

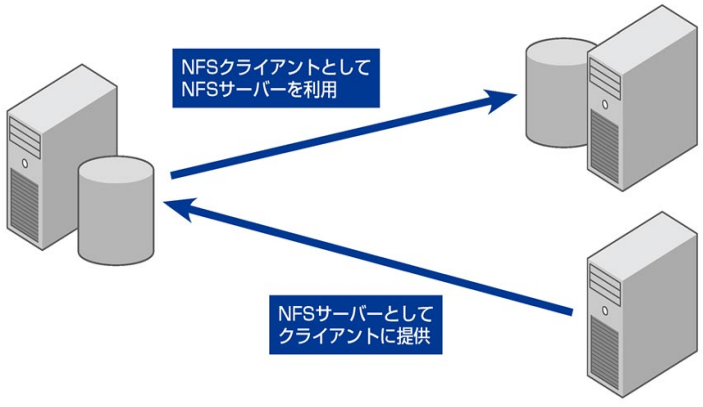
NFS ってなに？

UNIX系システムでファイルシステムを共有する仕組みとして、NFS（Network File System）があります。

RHEL 7は、NFSサーバーにもNFSクライアントにもなれます。つまり、リモートにあるNFSサーバー上のディスクリソースをネットワーク経由で利用できますし、ディスクリソースをほかのクライアントへ提供することもできます。

NFSでは、共有されたディスクリソースを、NFSクライアントからファイルシステムとして透過的に利用できます。そのため、ディスクリソースが足りなくなったときのディスク容量追加のようなシチュエーションでも活用できます。ディスクリソースを有効活用することにもつながります。

また、ネットワーク経由で起動するPXEブートという仕組みとNFSを組み合わせることで、ディスクレスで起動するクライアントの構成を作ることも可能です。



NFSの仕組み

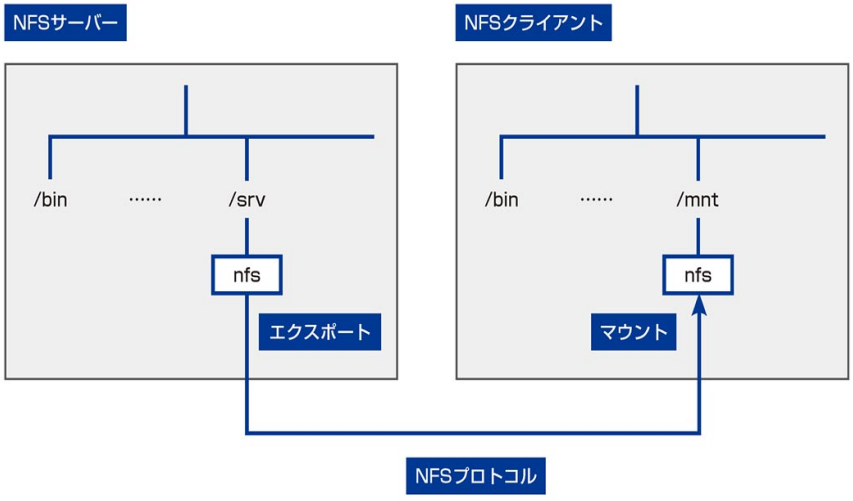
NFSは元々Solaris向けに作られた仕組みで、UNIXシステムにおけるファイルシステムの共有の仕組みとして作られました。NFSサーバー上の特定のディレクトリをネットワーク経由で共有し、複数のNFSクライアントから同時にマウントして使えます。

たとえば、NFSサーバー上の100GBの共有ディレクトリ「/srv/nfs」を、NFSプロトコルにてNFSクライアントで/mnt/nfsとしてマウントした場合、あたかも100GBの/mnt/nfsというディスク領域が増えたと認識されます。

NFSサーバー上のディレクトリをNFSクライアントから見えるようにすることを、NFSでは「エクスポート」と呼びます。

NFSクライアント側でマウントしたNFSサーバー上の共有ディレクトリは、直接接続されたハードディスクや、DVDなどの光学ドライブをマウントしたときと同様に、ローカルディレクトリの一部として扱うことができます。

この章ではNFSサーバーの構築方法とNFSクライアントからのマウント方法を解説していきます。



9-2 Linux用のファイルサーバーを作るには

NFSサーバー

RHEL 7が稼働しているサーバーであれば、特別な要件なしに、NFSサーバーを構築することができます。NFSサーバーを開始し、外部に提供したいディレクトリをNFSエクスポートとして宣言するだけで 、他のLinuxマシンにディスクリソースを分け与えることができます。NFSサーバーの開始やファイアウォールの設定は、既存の設定から行えます。

このレッスンでは、RHEL 7でNFSファイルサーバーを動かすための設定作業と、任意のディレクトリをNFSエクスポートするまでの流れを解説します。

NFSサーバーを構築する

NFSサーバーの本体はカーネルモジュールとして実装されているため、RHEL 7がインストールされているシステムにもれなく組み込まれています。

そのほか、NFSサーバー機能を制御するためのツール類としてnfs-utilsパッケージが必要になります。RHEL 7をGUIサーバーとしてインストールした場合はnfs-utilsパッケージが最初からインストールされますが、インストールされていない場合はyumコマンドでインストールしてください。

```
[root@host1 ~]# yum install nfs-utils
```

1 NFSサーバーを開始する

❶コマンドを入力

```
[root@host1 ~]# systemctl start nfs-server.service
[root@host1 ~]# systemctl enable nfs-server.service
ln -s '/usr/lib/systemd/system/nfs-server.service' '/etc/systemd/system/nfs.target.
wants/nfs-server.service'
[root@host1 ~]#
```

システム起動時に開始するようにする

❷コマンドを入力

2 ファイアウォールで許可する

ここではNFSクライアントと接続するNICがinternalゾーンに属しているものとする

❶コマンドを入力　　ゾーン

```
[root@host1 ~]# firewall-cmd --add-service=nfs --zone=internal
success
[root@host1 ~]# firewall-cmd --add-service=nfs --zone=internal --permanent
success
[root@host1 ~]#
```

システム起動時に許可させる　　❷コマンドを入力

3 公開用ディレクトリを作る

コマンドを入力　　公開用ディレクトリ

```
[root@host1 ~]# mkdir /srv/nfs /srv/nfs2
[root@host1 ~]#
```

メモ これらのディレクトリをマウントポイントとしてディスクをマウントして公開することもできます。

4 公開するディレクトリを設定する

NFSサーバーの設定ファイルを編集する　　❶コマンドを入力

```
[root@host1 ~]# vi /etc/exports
```

ファイルが開かれた　　❷以下の内容を追加　　1行で1つの共有ディレクトリを設定する

```
/srv/nfs        192.168.0.0/255.255.255.0(rw,no_root_squash)
/srv/nfs2       192.168.0.0/255.255.255.0(ro)
```

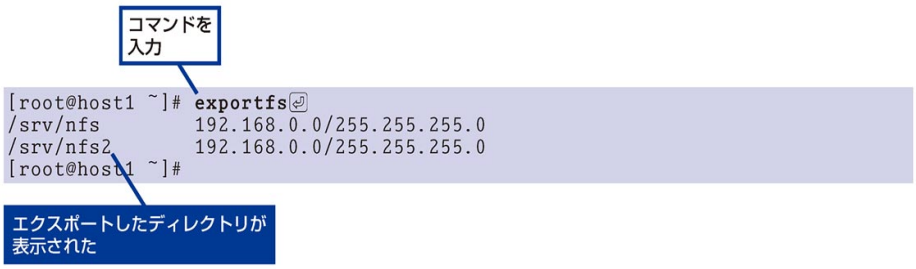
共有ディレクトリ　　許可するクライアントまたはネットワーク　　NFSオプション

❸保存して終了

次のページに続く

5 ディレクトリをエクスポートする

6 エクスポートしたディレクトリを確認する



HINT!

ホームディレクトリをエクスポートするときは

サーバー上の/homeをNFSエクスポートして、複数台のマシンで共通のホームディレクトリを利用する場合があります。その場合には、SELinuxのブール値「use_nfs_home_dirs」をonにして、ホームディレクトリのNFSでの共有を許可してください。

```
# setsebool -P use_nfs_home_dirs on
```

NFS共有をマウントする

NFSサーバーに対して、別のRHEL 7のマシンをクライアントとしてマウントしてみましょう。RHEL 7であれば、追加インストールなしでNFSクライアントとして使えます。なお、RHEL 7以外のLinuxのマシンからもNFSマウントできます。

1 マウントポイントを作る


```
[root@host2 ~]# mkdir /mnt/nfs
[root@host2 ~]#
```

2 共有ディレクトリをマウントする


```
[root@host2 ~]# mount 192.168.0.1:/srv/nfs /mnt/nfs
[root@host2 ~]#
```

問題がなければ何も出力されない

HINT!

NFSv4とNFSv3

NFSでは、バージョン4のNFSv4とバージョン3のNFSv3が使われています。mountコマンドに-oでオプション指定しなければ、NFSv4で読み書き可能な領域としてマウントされます。明示的にNFSv3を指定したいときは、「-o nfsvers=3」とオプションを指定してください。

9-3 Windows用のファイルサーバーを作ろう

Sambaの概要

Windowsをネットワークに接続すると、簡単な設定だけで他のWindowsパソコンとファイル共有ができます。それと同じプロトコルをLinux上に実装することで、Windowsのネットワークに Linuxを参加させるソフトウェアがSambaです。

Sambaを使うことで、Windowsからアクセスができる共有フォルダーや共有プリンターをRHEL 7から提供できます。これにより、それまでWindows Serverで提供していたファイルサーバーの仕組みを、RHEL 7に置き換えることができます。また、Sambaを使うことで、LinuxとWindows間のファイルのやりとりが容易になります。

Samba ってなに？

Sambaを使うことで、Windowsに対してファイル共有サービスを提供できます。

Windowsでは共有フォルダーという仕組みを使い、同一ネットワーク内のWindowsのコンピューター間でファイルを共有できます。このときに利用されているファイル転送のプロトコルは、「SMB」もしくは「CIFS」と呼ばれます。

SMBは、かつてNetBIOSという仕組みを利用していたときに使っていたプロトコルです。SMBを元にした、Windows 2000以降のWindows OSでTCP/IPを介してファイル共有サービスのプロトコルは、CIFSプロトコルと呼びます。

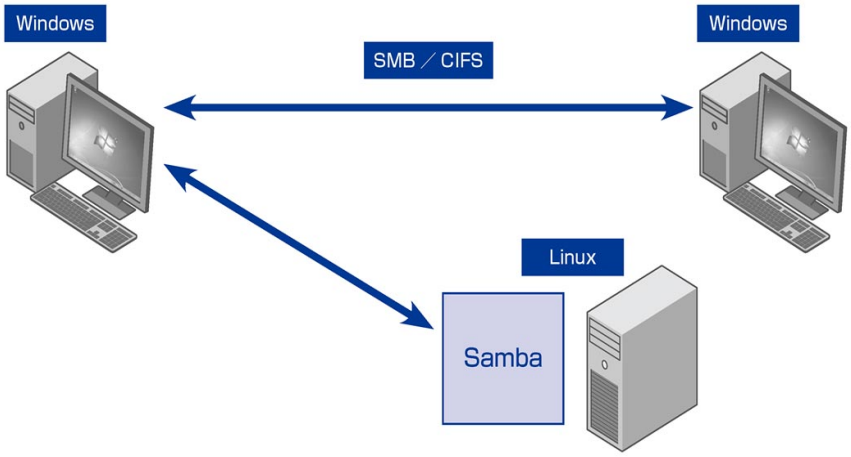
このCIFSプロトコルを実装し、ファイル共有サービスをLinuxで提供する仕組みがこの章でご紹介するSambaです。

RHEL 7に収録されているSamba 4では、Windows Server 2012およびWindows 8から対応になったCIFSプロトコル「SMB 3.0」の一部にも対応しており、より高速なデータ転送や、データ暗号化などの機能を使えます。

HINT!

Sambaの名前の由来

Sambaは元々「smbserver」と呼ばれていました。しかし、すでに「SMBserver」を商標登録していたSyntax社から指摘され、プロトコル名の「SMB」という3文字を含む適当な単語をシステム上の辞書ファイルからgrepコマンドで探して、Sambaになったと言われています。



Sambaのさまざまな機能

Sambaはファイル共有サービス以外にも機能を持っています。その一つとして、プリンター共有サービスを提供することもできます。ただし、最近ではネットワークプリンター機能を持つ家庭向けの複合機も一般的になってきたので、使われることも少なくなりました。

また、SambaはWindowsと同じようにActive Directoryに参加できます。ユーザー認証はもちろんのこと、ユーザーやグループの情報をActive Directoryから参照してLinuxのファイルシステム上のファイルに付与するといったこともできます。Samba 4からはActive Directoryのドメインコントローラーとなることもできるようになりました。

なお、以前はSambaでファイル名の文字コードの問題が多く起こっていました。しかし、それは昔の、Windows側の文字コードがShift_JIS（CP932）でLinux側の文字コードがEUC-JPを使っていた頃の話です。最近ではUnicode（UTF-8）で統一されてきたため、極めて発生しにくくなりました。

9-4 Windows用のファイルサーバーを作るには

Sambaサーバー

RHEL 7上にSambaを導入して簡単な設定をするだけで、Windowsに対して共有フォルダーを提供できます。最初は慣れないかもしれませんが、設定はそれほど難しいものではありません。

Sambaの設定には、smb.confという設定ファイルを変更する必要があります。しかし、Sambaは歴史が古いソフトウェアということもあり、多くの設定項目があります。

このレッスンではSambaをファイルサーバーとして動かすための必要最低限の設定作業と、Sambaのユーザーの追加まで、共有フォルダーを作る流れを解説します。

インストールする

1 Sambaをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install samba
読み込んだプラグイン：langpacks, product-id, subscription-manager

================================================================================
 Package      アーキテクチャー
                          バージョン                 リポジトリー                 容量
================================================================================
インストール中：
 samba        x86_64       4.1.12-21.el7_1        rhel-7-server-rpms       555 k
```

Sambaのパッケージが表示される

```
総ダウンロード容量： 555 k
インストール容量： 1.6 M
Is this ok [y/d/N]: y
Downloading packages:
```

❷「y」と入力してEnterキーを押す

```
完了しました！
[root@host1 ~]#
```

インストールが完了した

2 元の設定ファイルをバックアップする

変更する前の設定ファイルをコピーしておく

コマンドを入力

```
[root@host1 ~]# cp -p /etc/samba/smb.conf /etc/samba/smb.conf.orig
[root@host1 ~]#
```

3 Sambaを設定する

❶コマンドを入力

```
[root@host1 ~]# vi /etc/samba/smb.conf
```

ファイルが開かれた

❷ワークグループ名を変更

```
#
        workgroup = DEKIRU
        server string = Samba Server Version %v

        max protocol = SMB2

        load printers = no
        cups options = raw
```

❸行頭の「;」を削除

❹「no」に変更

❺保存して終了

次のページに続く

4 設定を確認する

testparmコマンドで設定を確認する

❶コマンドを入力

```
[root@host1 ~]# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
        workgroup = DEKIRU

        browseable = No
[root@host1 ~]#
```

確認結果が表示される

❷Enterキーを押す

設定内容が表示された

Sambaのユーザーを作成する

Sambaのユーザーとパスワードは、Linuxのユーザーとは別に管理されます。

Sambaのユーザー管理方法には数種類ありますが、RHEL 7に搭載されているSambaのデフォルトでは、TDB形式のユーザーデータベースにより管理されます。TDB形式のユーザーデータベースの場合、ユーザーはpdbeditコマンドで作成します。

なお、pdbeditコマンドでSambaのユーザーを追加する前に、RHEL上のローカルユーザーとしても同じ名前のユーザーを作成しておく必要があります。

1 ユーザーを追加する

❶コマンドを入力

作成するユーザー名

```
[root@host1 ~]# pdbedit -a -u htaira
new password:
retype new password:
Unix username:        htaira
NT username:

Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
[root@host1 ~]#
```

❷設定するパスワードを入力

❸パスワードをもう一度入力

ユーザーが追加された

SELinuxを設定する

ここまでの設定で、Sambaでは各ユーザーが自分のホームディレクトリを共有できるようになりしました。

ただし、実際に共有するには、Sambaの設定以外に、SELinuxのブール値「samba_enable_home_dirs」をonにして、ホームディレクトリのSambaでの共有を許可する必要があります。もしもonにしなかった場合、クライアント側のウィンドウにホームディレクトリの共有フォルダーは出現しますが、アクセスできないという現象になります。

1 ホームディレクトリの共有を許可する

コマンドを入力

```
[root@host1 ~]# setsebool -P samba_enable_home_dirs on
[root@host1 ~]#
```

起動する

1 Sambaを開始する

❶コマンドを入力　システム起動時に開始するようにする　❷コマンドを入力

```
[root@host1 ~]# systemctl start smb.service
[root@host1 ~]# systemctl enable smb.service
ln -s '/usr/lib/systemd/system/smb.service' '/etc/systemd/system/multi-user.
target.wants/smb.service'
[root@host1 ~]#
```

2 ファイアウォールで許可する

Sambaへのアクセスを許可　❶コマンドを入力

```
[root@host1 ~]# firewall-cmd --add-service=samba --zone=public
success
[root@host1 ~]# firewall-cmd --add-service=samba --zone=public --permanent
success
[root@host1 ~]#
```

システム起動時に許可させる　❷コマンドを入力

次のページに続く

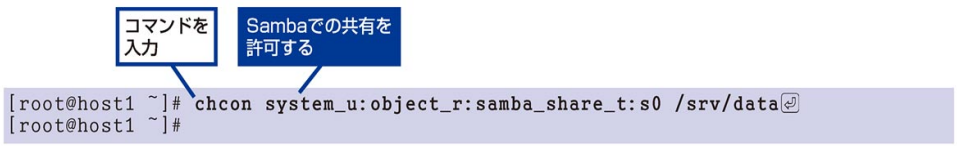
共有フォルダーを作る

新しく、ユーザーがみな使える共有フォルダーを作ってみましょう。新しく共有するためのディレクトリを作成し、ディレクトリのパーミッションとSELinuxコンテキストを変更して、Sambaのユーザーが誰でも読み書きできるようにします。その上で、Sambaに設定を追加します。

1 ディレクトリを作る

2 パーミッションを設定する

3 SELinuxコンテキストを設定する



4 Sambaを設定する

```
[data]
        comment = Data Share
        path = /srv/data
        public = yes
        writable = yes
        printable = no
```

❸保存して終了

5 Sambaを再起動する

コマンドを入力

Sambaでの共有を許可する

```
[root@host1 ~]# systemctl restart smb.service
[root@host1 ~]#
```

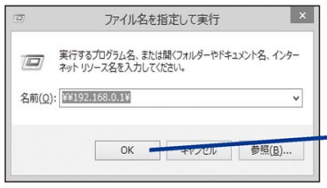
Windowsから接続する

1 接続する

❶⊞-Rキーを押す

❷以下の内容を入力

¥¥（Sambaサーバー）¥

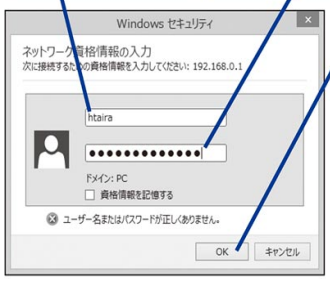


❸[OK]をクリック

2 ログインする

❶Sambaのユーザー名を入力

❷Sambaのパスワードを入力

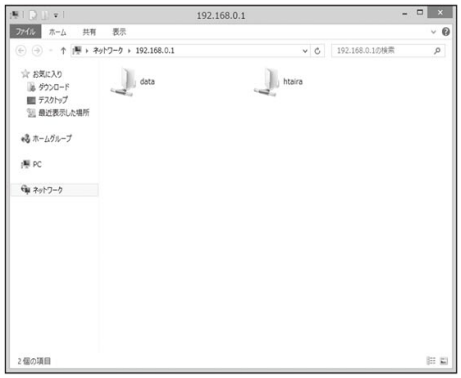


❸[OK]をクリック

メモ Windowsのユーザー名とパスワードがSambaと同じであれば、この画面は表示されません。

3 Sambaサーバーに接続した

設定した「data」とホームディレクトリの2つが表示された



STEP UP

NFSの設定ディレクトリ/etc/exports.d

RHEL 6までは、NFSの設定ファイルはデフォルトで/etc/exportsの1つだけでした。そのため、NFSの設定をプログラムから追加や削除しようとしたときや、RPMパッケージで管理しようとしたときに、いろいろと不都合が発生していました。

そこでRHEL 7からは、/etc/exports.dというディレクトリを使えるようになりました。このディレクトリを作成して、その中に「*.exports」という形式のファイル名で複数の設定ファイルを置けます。これにより、NFSエクスポートのエントリーを用途ごとに分割して記述できるようになりました。

複数のシステムに対してNFSで共有ディレクトリをエクスポートする場合に、それぞれを分割することで、設定ファイルの見通しがよくなります。それだけではなく、運用時のオペレーションミスの低減にもつながります。

/etc/exports.dによる設定の例

❶コマンドを入力

```
[root@host1 ~]# mkdir /etc/exports.d
[root@host1 ~]#
```

❷コマンドを入力

```
[root@host1 ~]# vi /etc/exports/nfs1.exports
```

❸以下の内容を追加

```
/srv/nfs        *(rw,no_root_squash)
```

❹保存して終了

第10章 DHCPサーバーやプロキシサーバーを作る

クライアントPC向けのネットワークで、IPアドレスの自動割り当てを行う仕組みがDHCPです。また、LANとインターネットを接続するにはIPマスカレードによりLANとインターネットの2種類のIPアドレスを変換します。そのほか、インターネットに対してアクセスするマシンを限定したり、許可されたサイトのみを閲覧可能にしたりする方法として、プロキシサーバーを用意して中継させることがあります。この章では、IPマスカレードとDHCPサーバー、プロキシサーバーの構築の方法について説明します。

●この章の内容

10-1 インターネットアクセスを共有するには

IPマスカレード

インターネットに接続するためには、グローバルIPアドレスが必要です。初期のインターネットでは、すべてのコンピューターにグローバルIPアドレスが振られていました。しかし、IPマスカレードを使用して通信を中継するルーターにグローバルIPを振ることで、そのルーター配下にある複数のコンピューターに対して、インターネットアクセスを提供することができます。多くのユーザーにインターネットアクセスを共有して利用することができます。

このレッスンでは、IPマスカレードの仕組みを説明していきます。

IPv4アドレスの枯渇とIPマスカレードの必要性

IPv4アドレスは32ビットで表現される情報で、単純計算で42億（2^{32}）のIPアドレスが定義できます（割り当て可能なIPアドレスはもっと少ない）。現在では世界人口の約4割の29億人がインターネットを利用しています（ITU World Communication / ICT Indicators database 2014年調査結果）。

1人が2つ以上の情報端末を利用するとグローバルIPアドレスがすぐに枯渇します。そこでIPマスカレードは、NAPT（Network Address and Port Translation）という仕組みを使います。クライアントのプライベートIPアドレスからインターネット上のホスト宛てのリクエストがルーターなどに来ると、パケット内の送信元IPアドレスとポート番号を動的に書き換えて、ルーターのグローバルIPアドレスから送信されたパケットとしてインターネット上に送出します。また、レスポンスとして受け取ったパケットは、宛先を任意のプライベートIPアドレスとポートに書き換えて、LAN内の対象のクライアントに返します。

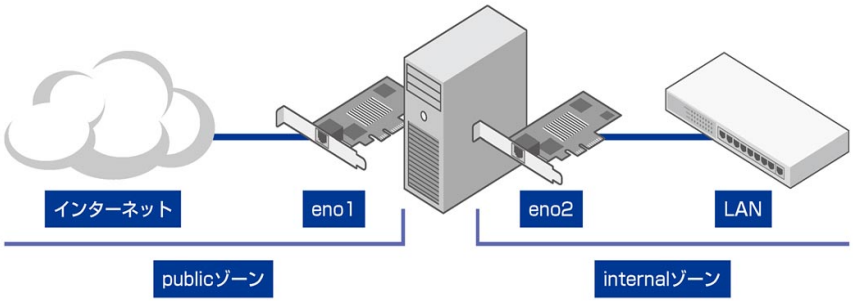
IPマスカレードは、グローバルIPアドレスの枯渇を防ぐ目的と、LAN内部のセキュリティを確保するために一般的に使われています。また、最近ではスマートフォンにもプライベートIPアドレスが振られることがほとんどで、通信キャリア側でIPマスカレードが行われています。

デメリットとしては、IPマスカレードされるクライアントではサーバーアプリケーションとしてポートを待ち受けることはできません。また、ルーター側の処理能力の限界を超えると、すべての通信がうまく行えなくなる場合があります。

IPマスカレードの設定

以前はiptablesで指定していたIPマスカレードの設定も、firewalldに含まれています。IPマスカレードもゾーンに割り当てます。

ここで解説する構成は次の図のとおりです。ここでは、インターネットにつながっているpublicゾーンを設定してIPマスカレードを有効化します。すると、LANからインターネットに向けたパケットが、eno1のインターフェイスのIPアドレスで外部に出ます。



1 IPフォワーディングの設定を追加する

カーネルの設定ファイルを作成する　コマンドを入力　設定内容　新しい設定ファイル

```
[root@host1 ~]# echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/00-ipv4_forward.conf
[root@host1 ~]#
```

2 SELinuxのセキュリティコンテキストを設定する

作成した設定ファイルのセキュリティコンテキストを変更する　コマンドを入力　セキュリティコンテキスト　対象のファイル

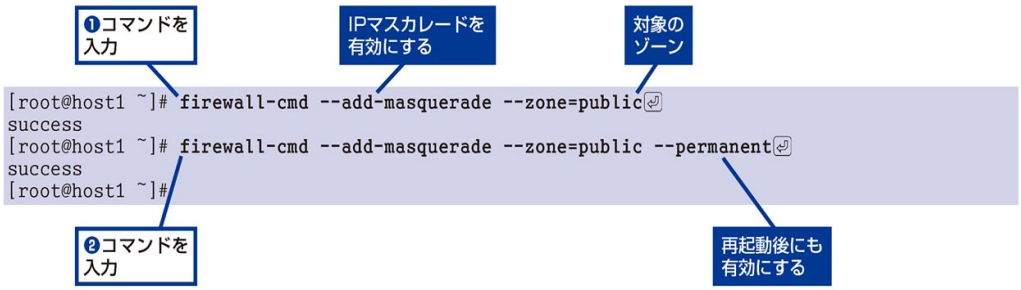
```
[root@host1 ~]# chcon system_u:object_r:system_conf_t:s0 /etc/sysctl.d/00-ipv4_forward.conf
[root@host1 ~]#
```

次のページに続く

3 設定を反映する


```
[root@host1 ~]# sysctl --system
* Applying /etc/sysctl.d/00-ipv4_forward.conf ...
net.ipv4.ip_forward = 1
* Applying /usr/lib/sysctl.d/00-system.conf ...
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
kernel.shmmax = 4294967295
kernel.shmall = 268435456
* Applying /usr/lib/sysctl.d/50-default.conf ...
kernel.sysrq = 16
kernel.core_uses_pid = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
* Applying /etc/sysctl.d/99-sysctl.conf ...
[root@host1 ~]#
```

4 IPマスカレードを有効にする



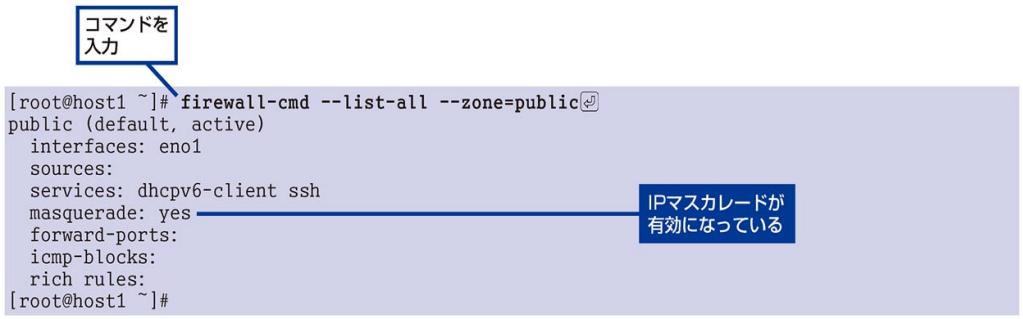

```
[root@host1 ~]# firewall-cmd --add-masquerade --zone=public
success
[root@host1 ~]# firewall-cmd --add-masquerade --zone=public --permanent
success
[root@host1 ~]#
```

HINT!

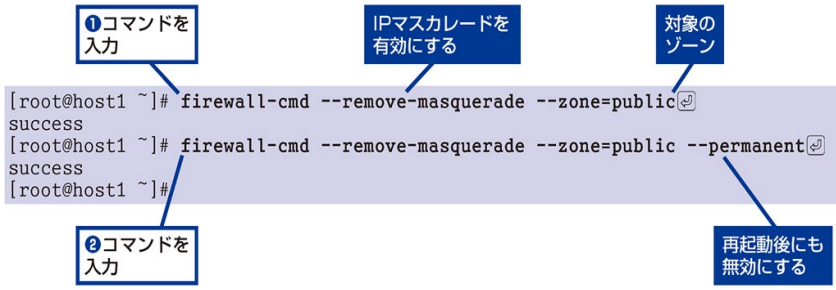
NATとNAPT

グローバルIPアドレスとプライベートIPアドレスを変換する仕組みには、IPアドレスのみを書き換えるNAT（Network Address Translation）と、IPアドレスとポートを書き換えるNAPT（Network Address and Port Translation）の2種類があります。なお、一般にNATと言われたときには、NAPTの意味を含めてNATと呼んでいる場合もあります。

5 設定を確認する



6 IPマスカレードを無効にする



10-2 DHCPを知ろう

DHCP

DHCPは、インターネットやイントラネットでIPアドレスを割り当てるために広く使われています。IPアドレスの割り当てを行う仕組みは、いくつか実装がありました。DHCPが開発される前は、RARPやBOOTPというプロトコルが使われ、最終的にDHCPプロトコルが主流になりました。

DHCPがない場合、コンピューターに1台ずつIPアドレスを手動で割り振る必要があります。現在、IPアドレスを手動で設定する機会は、ルーターの設定か、もしくはサーバー構築のときぐらいだと思います。このレッスンでは、DHCPの必要性と、その仕組みについて説明します。

なぜDHCPが必要なの？

本書でも何度か出てきたように、ネットワークには「IPアドレス」という、ネットワーク上のコンピューターに割り振られるアドレスがあります。

このIPアドレスには、大きく分けて、静的IPアドレス（Static IP address）と動的IPアドレス（Dynamic IP address）の2種類が存在します。静的IPアドレスは一般的に、サーバーやプリンターなどに割り当てます。動的IPアドレスは一般的に、ノートパソコンやスマートフォンなどに割り当てます。一般家庭や事業者向けにインターネットプロバイダーから割り当てられるIPアドレスも大半は動的IPアドレスです。

サーバーのように特定のネットワークに固定されて利用されるコンピューターの場合には、静的IPアドレスを割り当てるのが便利でしょう。サーバーはクライアントに対してサービスを提供する必要があるので、クライアントから特定されてアクセスされる必要があります。

一方、ノートパソコンのように複数のネットワークに参加する可能性があるコンピューターの場合は、コンピューターに静的IPアドレスを割り振ろうとすると、ネットワークを利用するまでにネットワークごとに大変面倒な設定変更が必要になってしまい、不便になってしまいます。そこで、ネットワーク接続時にネットワーク上から動的IPアドレスを割り当ててもらいます。動的IPアドレスを割り当てるプロトコルをDHCP（Dynamic Host Configuration Protocol）と呼び、動的IPアドレスを払い出してくれるサーバーをDHCPサーバーと呼びます。

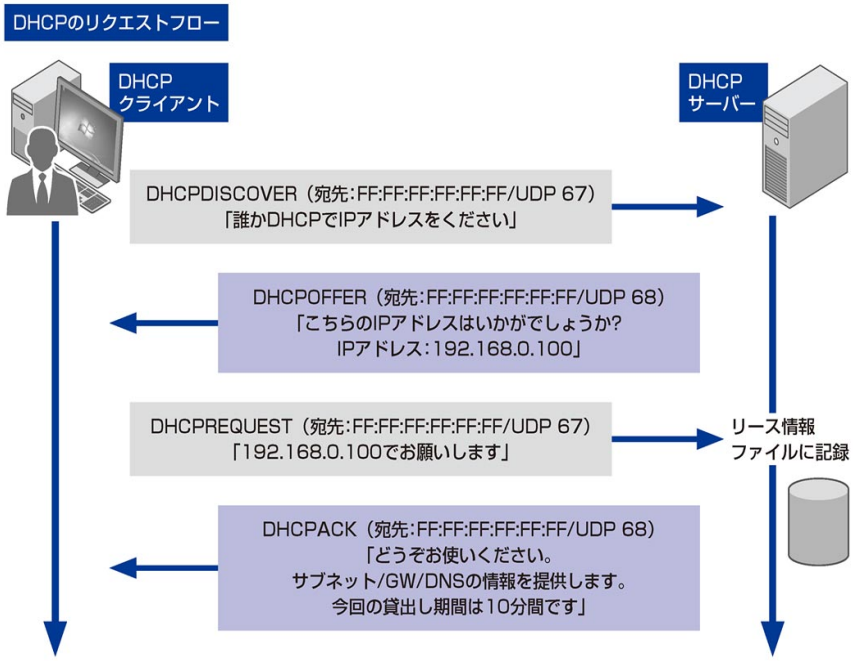
DHCPサーバーからはDHCPを通じて、IPアドレス以外にも、ホスト名やデフォルトゲートウェイの経路情報、DNSサーバーのIPアドレス、NTPサーバーのIPアドレスなどの情報がネットワークパラメーターとして渡されます。これらをネットワーク接続時にDHCPクライアントが受け取ることにより、ネットワークの設定を自動化できます。

DHCPの仕組み

ここでDHCPの仕組みについて解説しましょう。DHCPサーバーを構築する上で、この流れを理解しておくことが重要です。

まず、DHCPクライアントがネットワークに接続されると、宛先MACアドレスFF:FF:FF:FF:FF:FFに対してブロードキャストのUDPパケット（UDPポート67）を送信します（DHCPDISCOVER）。このパケットを受信したDHCPサーバーは、DHCPプールからIPアドレスの候補を選択し、クライアントのMACアドレスに対して返答します（DHCPOFFER）。受け取ったIPアドレスの候補が、接続されているネットワークのブロードキャストドメイン内で使用されていないかを検証した上で、DHCPサーバーに対して候補のIPアドレスの承認要求をします（DHCPREQUEST）。承認要求を受け取ったDHCPサーバーは承認（DHCPACK）を送信すると同時に、リース済みのIPアドレスとリース先のMACアドレスを割り当て管理テーブルに追加します。

DHCPはブロードキャストを利用してIPアドレスを取得する仕組み上、同じネットワークセグメント内（ブロードキャストドメイン内）に複数DHCPサーバーが存在していると誤動作の原因となります。組織内にネットワーク管理者がいる場合には許可を得てからDHCPサーバーを構築しないとネットワーク障害を引き起こします。また、ネットワーク上にルーターが存在する場合には、ルーターのDHCPサーバーが不要かどうか事前に確認した上で設定を無効化してからDHCPサーバーを構築してください。



10-3 DHCPサーバーをインストールするには

dhcpd

DHCPサーバーの機能は、現在ではルーターの中に実装されていることも多く、ネットワーク起動の環境などを作らなければ、ルーターの機能でことが足ります。もちろん、RHELでもDHCPサーバーを作ることができます。

RHELでDHCPサーバーを作ることにより、アクセスログを詳細に記録したり、特定のMACアドレスに対して、特定のIPアドレスを割り当てたり、ネットワークブートのための特殊なオプションを指定したりできます。

このレッスンでは、RHEL 7でのDHCPサーバーのインストールと設定について説明します。

1 dhcpdをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install dhcp
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package     アーキテクチャー
                          バージョン                    リポジトリー                  容量
================================================================================
インストール中：
 dhcp        x86_64       12:4.2.5-36.el7          rhel-7-server-rpms       510 k
```

dhcpdのパッケージが表示される

```
インストール容量： 1.4 M
Is this ok [y/d/N]: y
Downloading packages:
```

❷「y」と入力してEnterキーを押す

```
完了しました！
[root@host1 ~]#
```

インストールが完了した

2 DHCPサーバーを設定する

変更する前の設定ファイルをコピーしておく

❶コマンドを入力

```
[root@host1 ~]# cp -p /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.orig
[root@host1 ~]# vi /etc/dhcp/dhcpd.conf
```

設定ファイルを編集する

❷コマンドを入力

viでファイルが開かれた

❸以下の内容を追加

メモ 設定内容は各自のネットワークに合わせてください。

```
subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.100 192.168.0.199;
  option routers 192.168.0.1;
  option domain-name-servers 192.168.0.1;
  option domain-name "dekiru.internal";
  default-lease-time 600;
  max-lease-time 7200;
}
```

LAN側のネットワーク

配布するIPアドレス

ルーターのIPドレス

DNSサーバーのIPアドレス

ドメイン名

デフォルトのIPアドレスの貸出し期間。ここでは10分間

最大のIPアドレスの貸出し期間。ここでは2時間

❹上書き保存して終了

3 ファイアウォールで許可する

LAN側のNICがinternalゾーンに属しているものとする

dhcpでのアクセスを許可

❶コマンドを入力

ゾーン

```
[root@host1 ~]# firewall-cmd --add-service=dhcp --zone=internal
success
[root@host1 ~]# firewall-cmd --add-service=dhcp --zone=internal --permanent
success
[root@host1 ~]#
```

システム起動時に許可させる

❷コマンドを入力

4 DHCPサーバーを開始する

❶コマンドを入力

```
[root@host1 ~]# systemctl start dhcpd.service
[root@host1 ~]# systemctl enable dhcpd.service
ln -s '/usr/lib/systemd/system/dhcpd.service' '/etc/systemd/system/multi-user.target.
wants/dhcpd.service'
[root@host1 ~]#
```

システム起動時に開始するようにする

❷コマンドを入力

10-4 プロキシサーバーをインストールするには

Squid

Squidはイントラネットのプロキシサーバーやキャッシュサーバーとして広く使われています。インターネット黎明期にはダイヤルアップ接続で帯域幅が非常に限られていたこともあり、Squidはキャッシュサーバーとして使われていました。

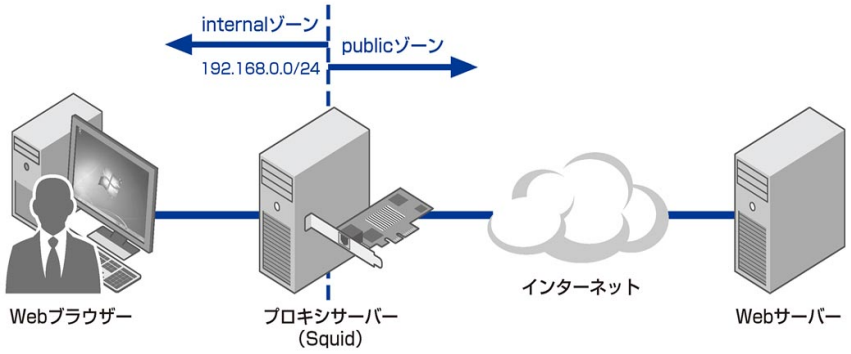
最近ではアクセスできるサイトを限定したりするプロキシサーバーとして主に使われます。また、サーバー側のリバースキャッシュプロキシとして使われることが多くなってきています。

このレッスンでは、プロキシサーバーの必要性と仕組みを説明し、プロキシサーバーSquidのインストールおよび設定について説明します。

プロキシサーバーってなに？

異なるネットワークと通信を行う際に、その通信を代わりに行い対象のサーバーにアクセスする中継（Proxy）するサーバーを、プロキシサーバーといいます。主に、インターネットに対してアクセスさせるマシンを限定させ、許可されたアクセスのみ通過させるといった用途で利用されます。プロキシサーバーはさまざまなプロトコルを中継できますが、HTTPとFTPを中継させることがほとんどです。

HTTPとFTPに対応しているプロキシサーバーとして、RHELにはSquidがあります。Squidはキャッシングプロキシサーバーです。つまり、ウェブから取得したコンテンツをキャッシュして、同じリクエストに対しては実際のサーバーではなくキャッシュから応答する機能があります。これによって、ウェブのレスポンス速度の向上や、WAN回線の使用帯域幅の削減にもつながります。



1 squidをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install squid
読み込んだプラグイン：langpacks, product-id, subscription-manager

================================================================================
 Package                    アーキテクチャー
                                    バージョン              リポジトリー             容量
================================================================================
インストール中：
 squid                      x86_64    7:3.3.8-12.el7_0     rhel-7-server-rpms  2.6 M
```

squidと依存パッケージが表示される

```
インストール容量： 12 M
Is this ok [y/d/N]: y
Downloading packages:

完了しました！
[root@host1 ~]#
```

❷「y」と入力してEnterキーを押す

インストールが完了した

2 設定ファイルをバックアップする

変更する前の設定ファイルをコピーしておく

❶コマンドを入力

```
[root@host1 ~]# cp -p /etc/squid/squid.conf /etc/squid/squid.conf.orig
[root@host1 ~]#
```

次のページに続く

3 Squidを設定する

設定ファイルを編集する　コマンドを入力

```
[root@host1 ~]# vi /etc/squid/squid.conf
```

4 LANからのアクセスを許可する

viでファイルが開かれた　LANからSquidへのアクセスを許可する

サンプルの設定を無効にする　❶行頭に「#」を挿入

```
#acl localnet src 10.0.0.0/8    # RFC1918 possible internal network
#acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
#acl localnet src fc00::/7       # RFC 4193 local private network range
#acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machines
acl localnet src 192.168.0.0/24
```

LANを許可する　❷1行を追加

5 匿名FTPのアカウントを追加する

匿名FTPを中継するときのユーザー名を設定する　❶ファイル末尾に1行を追加

```
ftp_user squid@dekiru.gr.jp
```

❷上書き保存して終了

6 Squidを開始する

❶コマンドを入力　システム起動時に開始するようにする　❷コマンドを入力

```
[root@host1 ~]# systemctl start squid.service
[root@host1 ~]# systemctl enable squid.service
ln -s '/usr/lib/systemd/system/squid.service' '/etc/systemd/system/multi-user.target.
wants/squid.service'
[root@host1 ~]#
```

7 ファイアウォールの設定を追加する

ファイアウォールでのSquidの設定を新しく作る

❶コマンドを入力

新しく作る設定ファイル

```
[root@host1 ~]# vi /etc/firewalld/services/squid.xml
```

viで新しいファイルが開かれた

❷以下の内容を追加

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>Squid(Proxy)</short>
  <description>Squid is a proxy caching server for Web clients.</description>
  <port protocol="tcp" port="3128"/>
</service>
```

❸保存して終了

8 ファイアウォールの設定を読み込み直す

追加した設定を使えるようにする

コマンドを入力

```
[root@host1 ~]# firewall-cmd --reload
success
[root@host1 ~]#
```

9 ファイアウォールで許可する

LAN側のNICがinternalゾーンに属しているものとする

❶コマンドを入力

ゾーン

```
[root@host1 ~]# firewall-cmd --add-service=squid --zone=internal
success
[root@host1 ~]# firewall-cmd --add-service=squid --zone=internal --permanent
success
[root@host1 ~]#
```

システム起動時に許可させる

❷コマンドを入力

メモ ゾーンには、Webに接続させたいクライアントが属するLAN側のネットワークが含まれたゾーンを指定してください。

次のページに続く

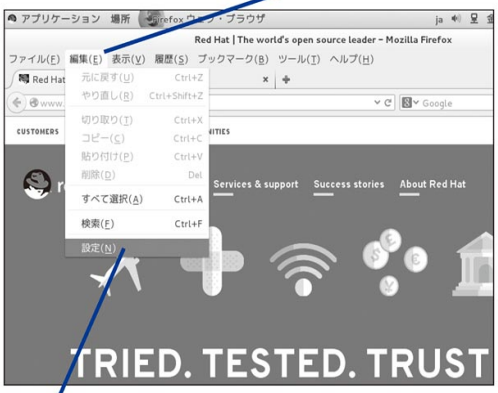
ブラウザーの設定（RHEL 7の場合）

1 Firefoxの設定を表示する

❶Firefoxを起動

❷Alt キーを押す

❸[編集]をクリック

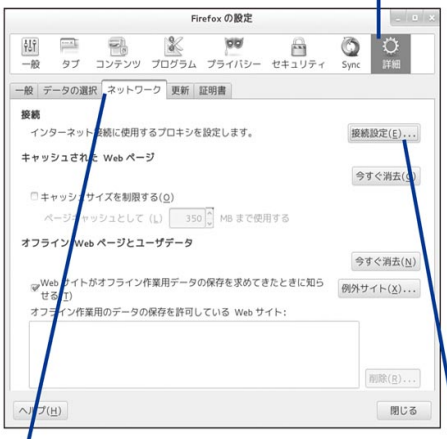


❹[設定]をクリック

2 接続設定を表示する

Firefoxの設定が表示された

❶[詳細]をクリック



❷[ネットワーク]をクリック

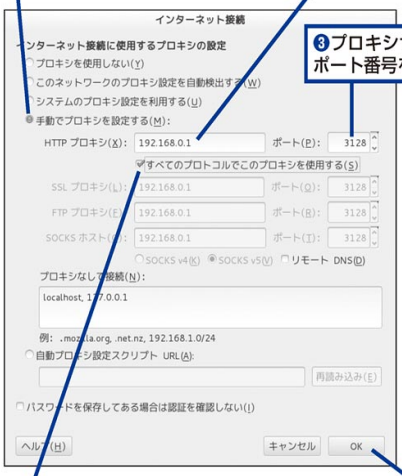
❸[接続設定]をクリック

3 プロキシサーバーの情報を入力する

❶[手動でプロキシを設定する]をクリック

❷プロキシサーバーのIPアドレスを入力

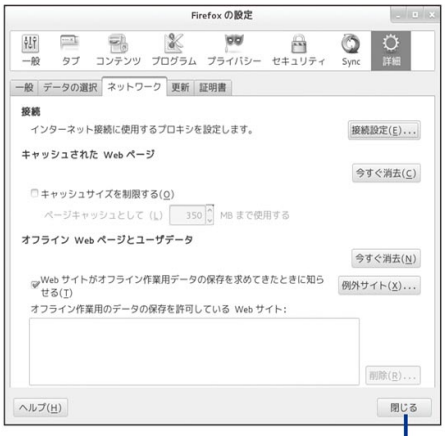
❸プロキシサーバーのポート番号を入力



❹[すべてのプロトコルでこのプロキシを使用する]をクリックしてチェックマークを付ける

❺[OK]をクリック

4 Firefoxの設定を閉じる



Firefoxの設定に戻った

[閉じる]をクリック

ブラウザーの設定（Windowsの場合）

1 Internet Explorerの設定を表示する

❶Internet Explorerを起動

❷歯車アイコンをクリック

❸[ネットワークオプション]をクリック

2 LANの設定を表示する

インターネットオプションが表示された

❶[接続]をクリック

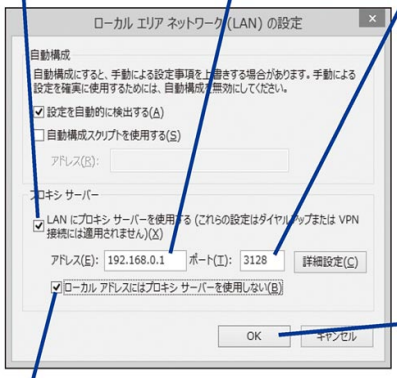
❷[LANの設定]をクリック

3 プロキシサーバーの情報を入力する

❶[LANにプロキシサーバーを使用する]をクリックしてチェックマークを付ける

❷プロキシサーバーのIPアドレスを入力

❸プロキシサーバーのポート番号を入力



❹[ローカルアドレスにはプロキシサーバーを使用しない]をクリックしてチェックマークを付ける

❺[OK]をクリック

4 インターネットオプションを閉じる

インターネットオプションに戻った

[OK] をクリック

10-5 リバースプロキシサーバーを作るには

リバースプロキシ

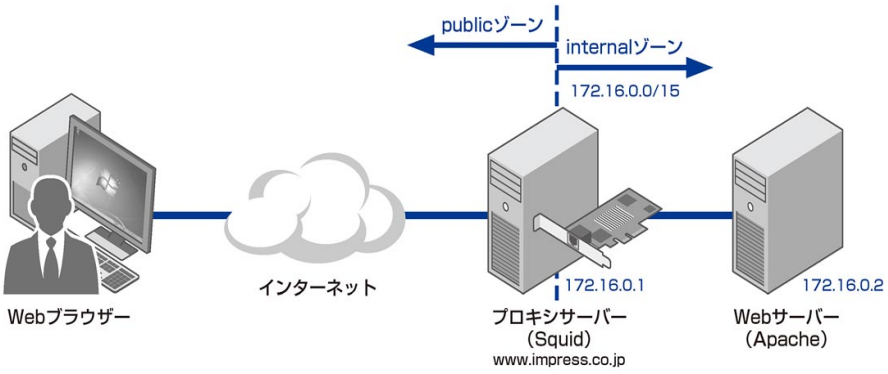
通常のプロキシは、インターネットに接続するクライアントを保護するためにアクセス制限を行ったり、アクセスログを残したりする目的でLAN内部に設置されます。反対にリバースプロキシは、サーバーに対してアクセス制限を行ったりするために、サーバーの前段に設置されます。

リバースプロキシは主にHTTPプロトコルに対し、URLをフィルタリングして特定のURLへのアクセスを禁止したり、特定のIPアドレスのみアクセスを許可したりできます。また、このレッスンで説明するSquidには、特定のコンテンツをキャッシュする仕組みもあります。

リバースプロキシってなに？

SquidにはWebサーバーの前段で中継するリバースプロキシとしての機能もあります。

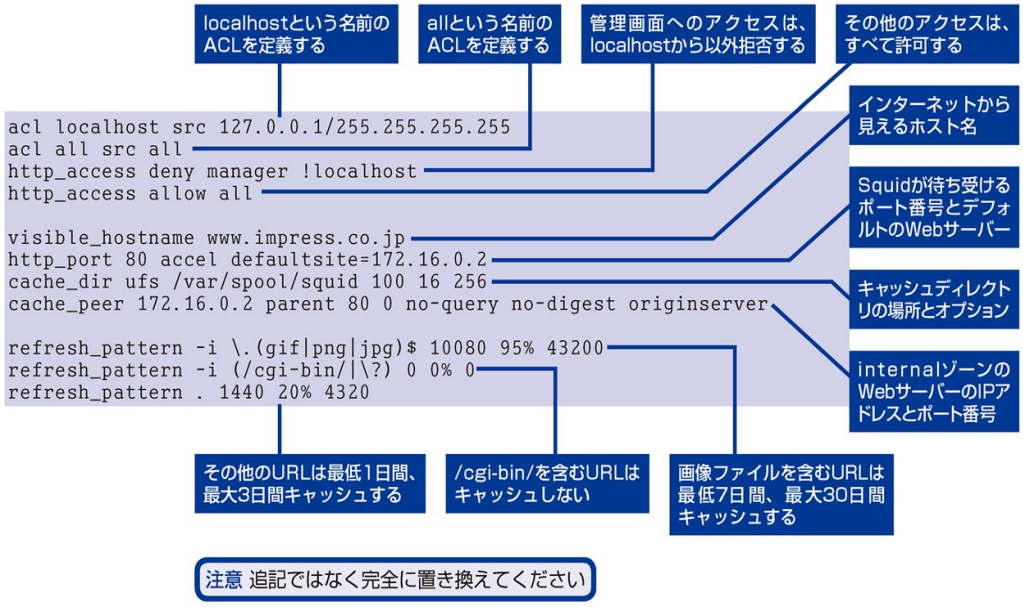
リバースプロキシを設置することで、静的コンテンツをSquidがキャッシングすることができます。これにより、Webサーバーの負荷を低減することができます。WebサーバーはCGIやPHPなどの動的コンテンツの処理にリソースを割くことができます。



リバースプロキシの設定

このシステム構成でリバースプロキシを構成したい場合には、Squidの設定ファイルの/etc/squid/squid.confを下のように記述します。

ユーザーから見ると、http://www.impress.co.jp/にアクセスするとあたかも1台のWebサーバーにアクセスしているように見えます。その向こうでは、背後のWebサーバーにあるコンテンツをプロキシし、必要に応じて静的コンテンツをキャッシュして配信してくれます。



注意 追記ではなく完全に置き換えてください

STEP UP

特定のマシンに特定のIPアドレスを割り当てる

DHCPプロトコルにおいて、ネットワーク内で特定の役割を行うコンピューターに対して、あらかじめ指定されたMACアドレスを元に特定のIPアドレスを割り当てることができます。

DHCPによるネットワーク設定しか行えないデバイスに対して、特定のIPアドレスを割り当てて、監視サーバーから追跡しやすくするという場合にも有効な手法です。

DHCPの設定ファイル/etc/dhcp/dhcpd.confの最後に以下のように追記します。MACアドレスに応じてfixed-addressで指定したIPアドレスを割り当てることができます。なお、対象のMACアドレスは、事前にipコマンドなどで調べてから記述してください。

```
host printer1 {
        hardware ethernet 12:34:56:78:90:A1;
        fixed-address 192.168.0.201;
}

host printer2 {
        hardware ethernet 12:34:56:78:90:A2;
        fixed-address 192.168.0.202;
}
```

メモ subnetの括弧の中に記述しても、括弧の外に記述してもかまいません。

第11章 DNSサーバーを作る

本章では、インターネットに公開するDNSサーバーを動かすためのソフトウェアであるBINDを紹介します。
ここでは、BINDの設定方法から、ゾーンファイルの定義、正引きゾーンファイルと逆引きゾーンファイルの仕組みまで説明します。

●この章の内容

11-1 BINDを動くようにするには

BINDのインストールと基本設定

BINDは、広く使われているDNSサーバーの1つです。外部から参照される本格的なDNSサーバーを構築できます。

BINDは元々はカリフォルニア大学バークレイ校で開発されましたが、その後、DEC社の開発者へ移り、現在ではISC（Internet Software Consortium）によって開発されています。

このレッスンでは、まず、BINDのインストール方法とキャッシュサーバーとして動作させる設定について説明します。これは、DNSサーバーの仕組みを理解するために、一度は試しておいて頂きたい内容です。

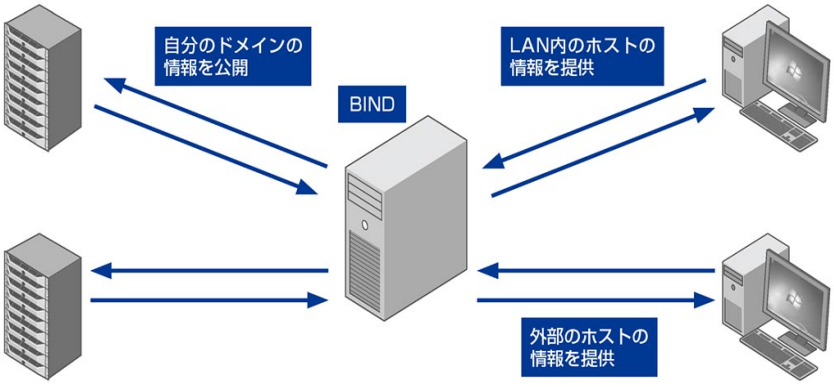
BINDってなに？

第6章では、LAN内部向けの簡易DNSサーバーを提供するソフトウェアとしてdnsmasqを紹介しました。RHEL 7には本格的なDNSサーバーのBINDも用意されています。BINDはインターネット上で最もよく使われているDNSサーバーです。

BINDを使うことで、LAN内部だけではなく、自分が所有するドメインの情報を管理することができます。BINDの設定ファイルにホスト名とIPアドレス、レコードタイプなどを記述したDNSのゾーン情報を格納することにより、外部のDNSサーバーからの問い合わせに対応します。

外部に公開するウェブサーバーがある場合や、インターネット向けの外部メールサーバーを構築するのであれば、必ずBINDは押さえておきたいソフトウェアの1つです。

初期定義までは難しいですが、一度導入できてしまえば、新規ホストの登録作業などは比較的ルーチンワークに近い形でできます。



ファイアウォールでDNSを許可する

ファイアウォールでほかのマシンからのDNSのアクセスを許可しましょう。なお、**第6章**でdnsmasqの設定をした場合は、すでに設定してあります。

1 ファイアウォールで許可する

❶コマンドを入力

```
[root@host1 ~]# firewall-cmd --add-service=dns --zone=public
success
[root@host1 ~]# firewall-cmd --add-service=dns --zone=public --permanent
success
[root@host1 ~]#
```

システム起動時に許可させる

❷コマンドを入力

dnsmasqが動いている場合

BINDとdnsmasqを同じホストで同時に動かすと、TCPとUDPのポートがそれぞれバッティングします。もし、dnsmasqを起動しているのであれば、BINDをセットアップする前に停止して無効化しましょう。

1 dnsmasqを停止する

❶コマンドを入力

```
[root@host1 ~]# systemctl stop dnsmasq.service
[root@host1 ~]# systemctl disable dnsmasq.service
rm '/etc/systemd/system/multi-user.target.wants/dnsmasq.service'
[root@host1 ~]#
```

システム起動時に起動しないようにする

❷コマンドを入力

次のページに続く

2 dnsmasqをマスクする

手動での起動も抑制する

コマンドを入力

```
[root@host1 ~]# systemctl mask dnsmasq.service
ln -s '/dev/null' '/etc/systemd/system/dnsmasq.service'
[root@host1 ~]#
```

BINDのインストール

1 BINDをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install bind
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package     アーキテクチャー
                        バージョン                        リポジトリー                容量
================================================================================
インストール中：
 bind        x86_64      32:9.9.4-18.el7_1.1         rhel-7-server-rpms      1.8 M

インストール容量：4.3 M
Is this ok [y/d/N]: y
Downloading packages:

完了しました！
[root@host1 ~]#
```

BINDのパッケージが表示される

❷「y」と入力してEnterキーを押す

インストールが完了した

HINT!

サービスのmaskとunmaskとは

systemdでは、システム起動時にサービスを自動起動させないdisableのほかにも、手動であってもサービスを起動させないmaskと、その状態を解除するunmaskが用意されています。いずれもsystemctlコマンドによって指定します。

BINDの基本設定

BINDは設定ファイル/etc/named.confで設定します。BINDのデフォルトの設定では、loインターフェイスのループバックアドレス127.0.0.1（IPv4）と::1（IPv6）しかリッスンしません。そのため、少し設定を編集しておきましょう。

またDNSでは、ドメインなどの管理する単位をゾーンと呼びます。BINDでは、ゾーン1つにつき1つのゾーンファイルを作成して、ゾーン情報を設定します。そして、これらのゾーンファイルの場所について/etc/named.confに記述します。

BINDを設定する

1 元の設定ファイルをバックアップする

変更する前の設定ファイルをコピーしておく

コマンドを入力

```
[root@host1 ~]# cp -p /etc/named.conf /etc/named.conf.orig
[root@host1 ~]#
```

2 設定ファイルを開く

コマンドを入力

```
[root@host1 ~]# vi /etc/named.conf
```

3 ネットワークのグループに名前を付ける

viでファイルが開かれた

LANに「localnet」という名前を付ける

4行を追加

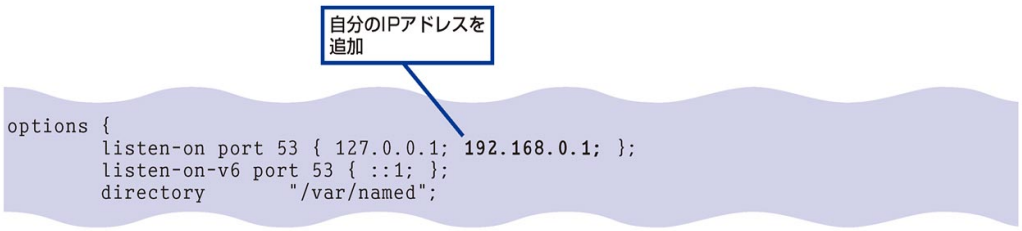
グループの名前

localnetに含まれるIPアドレス

```
acl localnet {
        127.0.0.1;
        192.168.0.0/24;
};

options {
```

次のページに続く

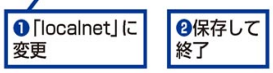
4 リッスンするアドレスを追加する



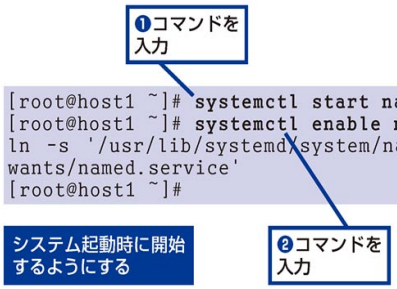

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.0.1; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
```

5 LANからの問い合わせを許可する

```
        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";
        allow-query     { localnet; };
```



6 BINDを開始する




```
[root@host1 ~]# systemctl start named.service
[root@host1 ~]# systemctl enable named.service
ln -s '/usr/lib/systemd/system/named.service' '/etc/systemd/system/mult-user.target.
wants/named.service'
[root@host1 ~]#
```

HINT!

aclグループによるアクセス制御

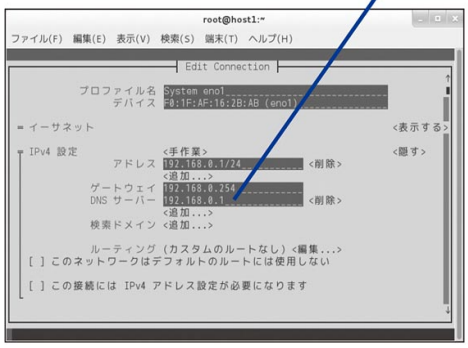
BINDでは、設定ファイル中のaclグループの中に接続元としてDNS問い合わせを許可するホストもしくはネットワークセグメントを列挙する必要があります。ここに記述されていない接続元からのDNS問い合わせは拒否します。

7 サーバーに自分のBINDを参照させる

ネットワーク設定を変更してBINDをサーバー自身から参照させる

❶レッスン4-2を参考にnmtuiを起動

❷自分のIPアドレスに変更



❸保存して終了

8 ネットワーク設定の変更を反映する

ネットワークを一度切断する

❶コマンドを入力

対象のネットワークインターフェイス

```
[root@host1 ~]# nmcli device disconnect eno1
[root@host1 ~]# nmcli device connect eno1
Device 'eno1' successfully activated with '92a15d9e-679f-427b-af2e-2fe29a993364'.
[root@host1 ~]#
```

ネットワークに再び接続する

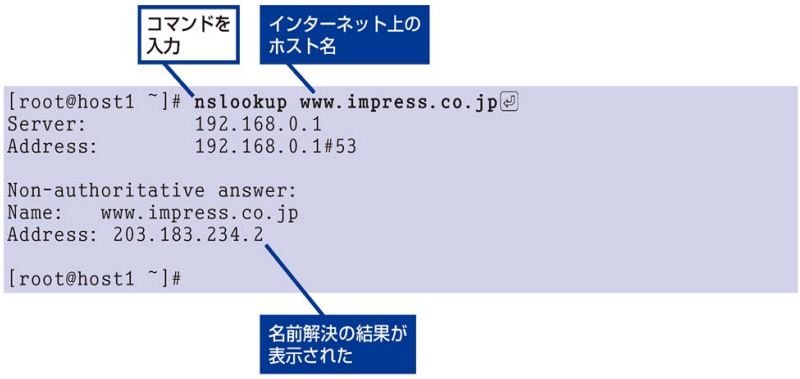
❷コマンドを入力

次のページに続く

BINDでの名前解決を確認する

ここまで設定した上でBINDに名前解決の依頼をすると、ループバックアドレスのゾーン情報しか持たない単なるDNSキャッシュサーバーとして機能することが確認できます。ただし、問い合わせ先はルートネームサーバーとなるので注意が必要です。

1 nslookupから名前解決する



上位のDNSを参照するよう設定する

ルートネームサーバーには、世界中からたくさんのDNSサーバーがアクセスします。そのため、レスポンスはよくありません。

DNSキャッシュサーバーとして実用的に動かすには、ISPのDNSサーバーなど、上位のDNSサーバーをBINDから参照するようにします。BINDの設定ファイルのforwardersで、明示的に上位のDNSサーバーを指定してください。

1 設定ファイルを開く

HINT!

DNSサーバーの役割

DNSサーバーには大きく分けてキャッシュサーバーとコンテンツサーバー（権威サーバー）の2種類の役割を持つものがあります。
キャッシュサーバーはクライアントからの問い合わせに対応して、DNSの応答速度を高める目的で利用されるものです。
コンテンツサーバーは自分の管理しているゾーンに対する問い合わせに対してのみ応答し、それ以外の問い合わせが来たとしても上位のDNSに問い合わせることはありません。

2 上位のDNSサーバーを指定する

viでファイルが開かれた

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.0.1; };
        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";
        allow-query     { localnet; };
        forwarders      { 192.168.0.254; };
```

❶1行を追加

上位のDNSサーバー

❷保存して終了

3 BINDを再起動する

コマンドを入力

```
[root@host1 ~]# systemctl restart named.service
[root@host1 ~]#
```

11-2 ゾーンを定義するには

ゾーンファイル

BINDでは、ドメインなどの管理する単位ごとに、ゾーンファイルを作って設定します。それぞれのゾーンファイルには、ゾーンに属するホスト名やアドレス、それらの属性などの付帯情報を記述します。

ゾーンファイルには、ホスト名からIPアドレスを引く正引きゾーンファイルと、IPアドレスからホスト名を参照する逆引きゾーンファイルが存在します。

ここでは、イントラネット内に「dekiru.internal」というドメインを作る例をもとに、ゾーンの記述方法を紹介します。

コンテンツサーバーとしてのBIND

レッスン11-1では、キャッシュサーバーとしてのBINDの設定について説明してきました。このレッスンでは“dekiru.internal”というゾーン情報を持つコンテンツサーバーとしてのBINDについて説明していきます。

ゾーン情報が記述されたファイルをゾーンファイルと呼びます。ゾーンファイルは、対象のドメイン名に属するホスト名とIPアドレス、レコードタイプ、有効期限などからなる情報を列挙したテキストファイルです。RHEL 7のデフォルトでは、ゾーンファイルは/var/namedディレクトリの中に格納されています。

主なレコードタイプ

ゾーンファイルの中でホスト名とIPアドレス以外に重要な意味を持つのがレコードタイプです。

ホスト名からIPアドレスを参照するレコードをAレコードと呼び、これを正引きといいます。

IPアドレスからホスト名を参照するレコードをPTRレコードと呼び、これを逆引きといいます。

そのほかに、ホスト名からホスト名の別名を参照するCNAMEレコードや、DNSサーバーを意味するNSレコード、メールサーバーを意味するMXレコードなどがよく利用されます。

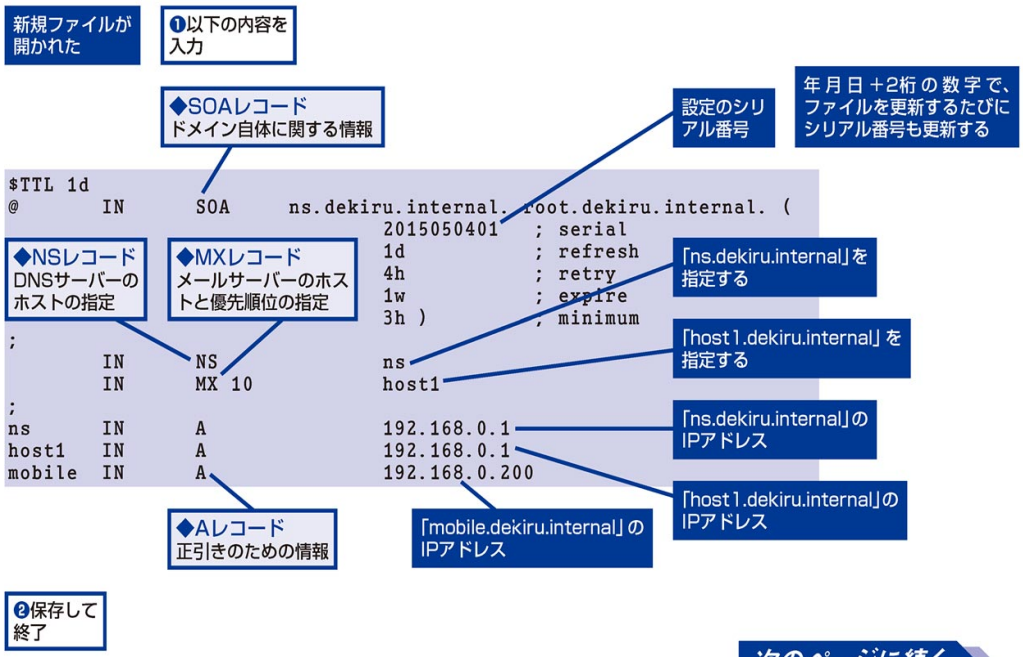
HINT!

ゾーンファイル中のホスト名に付く「.（ドット）」

ゾーンファイル中のSOAレコードの定義では、ホスト名が「ns.dekiru.internal.」のように「.」で終わっています。ホスト名の末尾が「.」で終わっていない場合には、named.confで指定されたゾーン名がホスト名の後ろに自動的に補完されるルールになっています。たとえば、「host1」とだけ記述すると「host1.dekiru.internal」として処理されます。

1 正引きゾーンファイルを新規作成する

2 正引きゾーンファイルを設定する




```
$TTL 1d
@       IN      SOA     ns.dekiru.internal. root.dekiru.internal. (
                                2015050401  ; serial
                                1d          ; refresh
                                4h          ; retry
                                1w          ; expire
                                3h )        ; minimum
;
        IN      NS              ns
        IN      MX 10           host1
;
ns      IN      A               192.168.0.1
host1   IN      A               192.168.0.1
mobile  IN      A               192.168.0.200
```

次のページに続く

3 逆引きゾーンファイルを新規作成する

コマンドを入力

逆引きゾーンファイル

```
[root@host1 ~]# vi /var/named/internal.rev
```

4 逆引きゾーンファイルを設定する

新規ファイルが開かれた

❶以下の内容を入力

```
$TTL 1d
@       IN      SOA     ns.dekiru.internal. root.dekiru.internal. (
                                2015050401   ; serial
                                1d           ; refresh
                                4h           ; retry
                                1w           ; expire
                                3h )         ; minimum
;
        IN      NS              ns.dekiru.local.
;
1       IN      PTR             host1.dekiru.internal.
200     IN      PTR             mobile.dekiru.internal.
```

ファイルを更新するたびにシリアル番号も更新する

ネームサーバーとして「ns.dekiru.internal」を指定する

ここでは「192.168.0.1」のホスト

ここでは「192.168.0.200」のホスト

◆PTRレコード
逆引きのための情報

❷保存して終了

5 BINDの設定を編集する

/etc/named.confにゾーンファイルを指定する

コマンドを入力

```
[root@host1 ~]# vi /etc/named.conf
```

6 ゾーンファイルを指定する

viでファイルが開かれた

❶ファイルの末尾に以下の内容を追加

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

zone "dekiru.internal" {
        type master;
        file "internal.zone";
};

zone "0.168.192.in-addr.arpa" {
        type master;
        file "internal.rev";
};
```

dekiru.intenalの正引きゾーン

ゾーンファイル

192.168.0.*からの逆引きゾーン

ゾーンファイル

❷保存して終了

7 BINDを再起動する

コマンドを入力

```
[root@host1 ~]# systemctl restart named.service
[root@host1 ~]#
```

8 nslookupから名前解決する

コマンドを入力

ゾーンで設定したホスト

```
[root@host1 ~]# nslookup host1.dekiru.internal
Server:         192.168.0.1
Address:        192.168.0.1#53

Name:   host1.dekiru.internal
Address: 192.168.0.1

[root@host1 ~]#
```

名前解決の結果が表示された

11-3 DNSサーバーを公開するには

viewステートメント

このレッスンでは、BIND9の新機能であるviewステートメントについて説明します。BIND 9から実装されたviewステートメントを使うことで、複数のネットワークに属するDNSサーバーが、アクセス元のネットワークアドレスに応じて、ゾーンファイルをコントロールできます。

レッスン11-2ですでに設定してあるBINDのゾーンファイルを内部向けゾーンファイルとして、localnetからアクセスしてきたときに利用します。外部向けゾーンファイルを追加し、localnet以外からアクセスしてきたときにその情報を返すように設定します。

内部のホスト名を内部向け専用にする

DNSサーバーをLANの内側で稼働させるのであれば、**レッスン11-2**までの設定で使えます。しかし、このサーバーをインターネットに公開すると、インターネット経由でLAN内部のホスト名が参照できてしまうという問題が発生します。また、オープンリゾルバとして他のユーザーから広く名前解決用に使われてしまうこともあります。

そこで、内部向けと外部向けに設定を分ける必要があります。BIND 8までは、外部向けと内部向けのDNSサーバーを別々のサーバーで動かすといった工夫が必要でした。

BIND 9からは、viewステートメントという仕組みが実装されています。viewステートメントを指定することで、アクセス元のネットワークアドレスに応じて、ゾーンファイルごとの参照の可否をコントロールすることができます。

1 BINDの設定を編集する

コマンドを入力

```
[root@host1 ~]# vi /etc/named.conf
```

2 設定をviewで内部専用にする

viでファイルが開かれた

ゾーンの設定をviewステートメントの中に入れる

「zone "."」行の前に以下の内容を追加

```
};

view "internal" {
        match-clients { localnet; };
        recursion yes;

zone "." IN {
```

設定を適用する対象

内部から外部ホストの問い合わせを許可する

3 設定を内部用のviewに入れる

❶「zone "."」行以降をインデントする

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


        zone "0.168.192.in-addr.arpa" {
                type master;
                file "internal.rev";
        };
```

❷末尾に「};」を追加

```
                file "internal.rev";
        };
};
```

❸保存して終了

次のページに続く

4 BINDを再起動する

コマンドを入力

```
[root@host1 ~]# systemctl restart named.service
[root@host1 ~]#
```

ドメインがインターネットから参照できるようになるまでの事務手続き

DNSサーバーを公開するためには、ここまでのレッスンで説明してきたサーバーの設定以外にも、固定IPアドレスの取得や、ドメインの取得、上流DNSサーバーへの登録作業など事務手続きがほかにも必要となります。

ドメインの取得は、ICANN公式レジストラにドメインを申請し、ドメイン使用料を支払うことでドメインの使用権が与えられます。

また、マスターサーバーとスレーブサーバーの2台のコンテンツサーバーを用意する必要があります。この2つのコンテンツサーバーは上流DNSサーバーに登録されるため、固定IPアドレスを持つ必要があります。

これらの具体的な手続きは、レジストラごとに少し違うため、詳しくはレジストラの運営企業に問い合わせて頂くのがよいでしょう。まずは取得したいドメイン名を考えることを忘れずに。

HINT!

キャッシュサーバーとしての問い合わせを制御する

内部向け設定の手順❷では「recursion」に「yes」を設定しました。これにより、viewで設定する範囲からキャッシュサーバーとしての問い合わせを許可します。一方、外部向け設定の手順❷では、「recursion」に「no」を設定して、キャッシュサーバーとしての問い合わせを拒否しています。外部からキャッシュサーバーとしてアクセスできるDNSサーバーはオープンリゾルバと呼ばれ、そのDNSサーバーやほかのサーバーへの攻撃を許してしまう可能性があります。キャッシュサーバーとしての問い合わせは、内部からは許可し、外部からは許可しないようにしましょう。

外部向けゾーンファイルを追加する

内部向けに続いて、外部に公開する設定のviewステートメントを設定します。ここでは、dekiru.gr.jpドメインのゾーンファイルを例にしますが、実際には自分の取得したドメイン名とIPアドレスに合わせてください。

1 BINDの設定を編集する

コマンドを入力

```
[root@host1 ~]# vi /etc/named.conf
```

2 外部向けの設定を追加する

viでファイルが開かれた

❶ファイルの末尾に以下の内容を追加

```
view "external" {
        match-clients { any; };
        recursion no;
        allow-query { any; };

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

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

        zone "dekiru.gr.jp" {
                type master;
                file "dekiru.zone";
        };

        zone "113.0.203.in-addr.arpa" {
                type master;
                file "dekiru.rev";
        };
};
```

設定を適用する対象

外部ホストの問い合わせを禁止する

問い合わせを許可する

❷保存して終了

次のページに続く

3 正引きゾーンファイルを新規作成する


```
[root@host1 ~]# vi /var/named/dekiru.zone
```

4 正引きゾーンファイルを設定する


```
$TTL 1d
@       IN      SOA     ns.dekiru.gr.jp. root.dekiru.gr.jp. (
                                2015050401   ; serial
                                1d           ; refresh
                                4h           ; retry
                                1w           ; expire
                                3h )         ; minimum
;
        IN      NS              ns
        IN      MX 10           host1
;
ns      IN      A               203.0.113.1
host1   IN      A               203.0.113.1
```

❷保存して終了

HINT!

「;」はコメントアウト

BINDの設定ファイルのnamed.confやゾーンファイルでは、行の中で「;」という文字から以降はコメントになっています。設定項目の意味を書いておいたり、意味のあった記述をコメントアウトして無効にしたりできます。

HINT!

設定が反映されるまで時間がかかることがある

ゾーンの設定を変更したときに、すぐにはデータが反映されない場合があります。外部のDNSサーバーはコンテンツサーバーに問い合わせると、そのデータを一定期間キャッシュします。1～2日様子を見てみましょう。

5 逆引きゾーンファイルを新規作成する

コマンドを入力　　逆引きゾーンファイル

```
[root@host1 ~]# vi /var/named/dekiru.rev
```

6 逆引きゾーンファイルを設定する

新規ファイルが開かれた　　❶以下の内容を入力

```
$TTL 1d
@       IN      SOA     ns.dekiru.gr.jp. root.dekiru.gr.jp. (
                                2015050401   ; serial
                                1d           ; refresh
                                4h           ; retry
                                1w           ; expire
                                3h )         ; minimum

        IN      NS              ns.dekiru.gr.jp.
;
1       IN      PTR             host1.dekiru.gr.jp.
```

❷保存して終了

7 BINDを再起動する

コマンドを入力

```
[root@host1 ~]# systemctl restart named.service
[root@host1 ~]#
```

11-4 ドメインの情報を調べるには

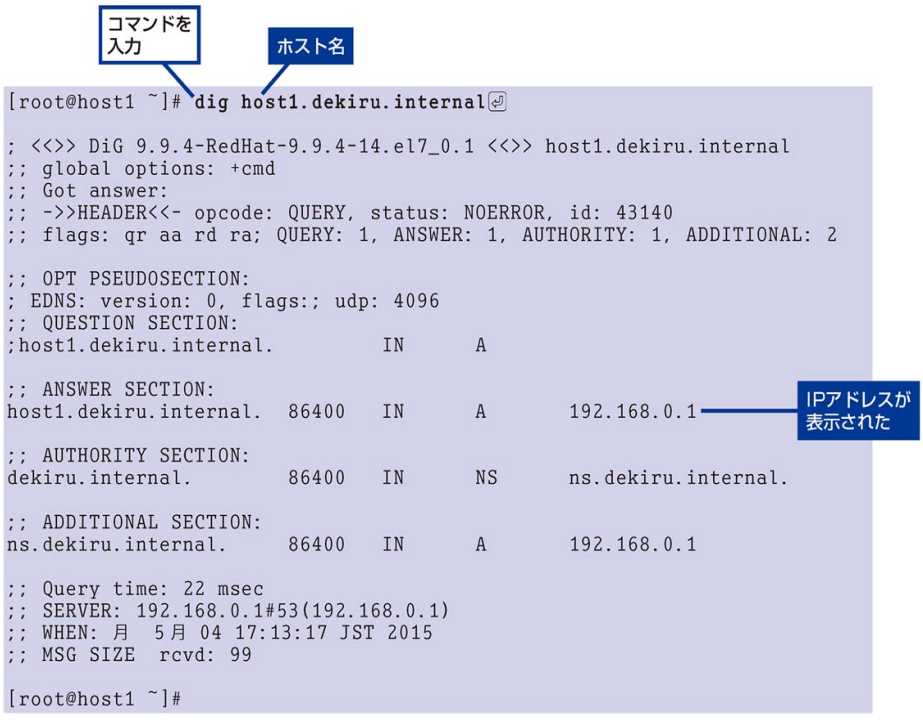
dig

ゾーン情報を調べるにはdigコマンドを使うのが便利です。digコマンドは、nslookupコマンドとほぼ同じ使い方ができますが、表示形式が異なり、ゾーンファイルに記述されているような形式で表示します。任意のレコードをピンポイントで調べることもできます。

また、オプションで指定することでデフォルトのDNSサーバー以外の任意のDNSサーバーに対して問い合わせを行うこともできます。

このレッスンでは、digコマンドを使い実際にAレコードとMXレコードを調べる方法について説明します。

1 ホスト名からIPアドレスを調べる

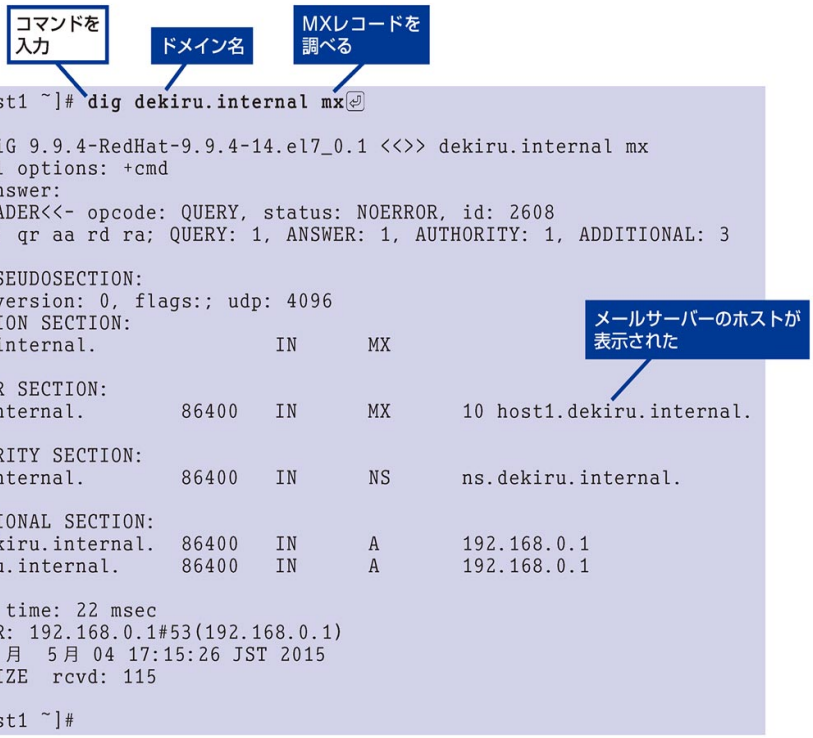



```
[root@host1 ~]# dig host1.dekiru.internal

; <<>> DiG 9.9.4-RedHat-9.9.4-14.el7_0.1 <<>> host1.dekiru.internal
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43140
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;host1.dekiru.internal.          IN      A

;; ANSWER SECTION:
host1.dekiru.internal.  86400   IN      A       192.168.0.1

;; AUTHORITY SECTION:
dekiru.internal.        86400   IN      NS      ns.dekiru.internal.

;; ADDITIONAL SECTION:
ns.dekiru.internal.     86400   IN      A       192.168.0.1

;; Query time: 22 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: 月  5月 04 17:13:17 JST 2015
;; MSG SIZE  rcvd: 99

[root@host1 ~]#
```

ドメインの情報を調べる

digコマンドを使うと、ホスト名からIPアドレスを調べるだけではなく、任意のドメインについて検索し、さまざまなDNS情報を調べられます。たとえば、MXレコードとして設定されているメールサーバーのホストを表示できます。

1 MXレコードを調べる




```
[root@host1 ~]# dig dekiru.internal mx

; <<>> DiG 9.9.4-RedHat-9.9.4-14.el7_0.1 <<>> dekiru.internal mx
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2608
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;dekiru.internal.               IN      MX

;; ANSWER SECTION:
dekiru.internal.        86400   IN      MX      10 host1.dekiru.internal.

;; AUTHORITY SECTION:
dekiru.internal.        86400   IN      NS      ns.dekiru.internal.

;; ADDITIONAL SECTION:
host1.dekiru.internal.  86400   IN      A       192.168.0.1
ns.dekiru.internal.     86400   IN      A       192.168.0.1

;; Query time: 22 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: 月  5月 04 17:15:26 JST 2015
;; MSG SIZE  rcvd: 115

[root@host1 ~]#
```

STEP UP

スレーブDNSサーバーを追加するには

スレーブDNSサーバーに対して、ゾーン転送を許可したい場合には、BINDの設定ファイルにallow-transferの項目を追記する必要があります。

/etc/named.conf

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.0.1; };
        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";
        allow-query     { localnet; };
        forwarders      { 8.8.8.8; 8.8.4.4; };
        allow-transfer  { 123.123.123.234; };
```

スレーブDNSのIPアドレスを指定

/var/named/dekiru.zone

```
$TTL 1d
@       IN      SOA     ns.dekiru.gr.jp. root.dekiru.gr.jp. (
                                2015050402   ; serial
                                1d           ; refresh
                                4h           ; retry
                                1w           ; expire
                                3h )         ; minimum
;
        IN      NS              ns.dekiru.gr.jp.
        IN      NS              ns.dekiru.jp.
        IN      MX 10           host1
;
ns      IN      A               203.0.113.1
host1   IN      A               203.0.113.1
```

スレーブDNSサーバーのホスト名を指定

第12章 メールサーバーを作る

インターネットにおいて日々メールが送受信できるのは、ネットワーク上のメールサーバーが送りたい相手のメールアドレスを手がかりにメールを配送してくれているからです。メールを送るにも受け取るにもメールサーバーが必要です。そこで、実際にRHEL7上でメールサーバーを構築しましょう。この章では、メールの仕組みとSMTPサーバー（Postfix）およびPOP3サーバー（Dovecot）の導入、各種設定方法について説明します。

●この章の内容

12-1 SMTPサーバーを作るには

Postfix

Postfixは、広く使われているメール転送エージェント（MTA）の1つです。Postfixは管理や設定が容易で高速かつ安全であることを指向して開発されています。RHEL 7にもコアパッケージとして標準で含まれています。

初期設定はローカルホスト内のメール配信しか行いませんが、このレッスンで説明する設定を行うことでインターネット経由でのメールの送受信も可能になります。メーラーからPOP3でメールの受信を行うためには、**レッスン12-2**で説明するDovecotと組み合わせる必要があります。

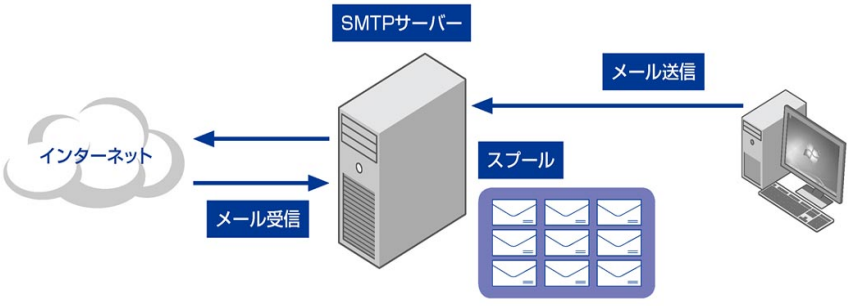
SMTPサーバー Postfix

インターネットでのコミュニケーションとして、メールによる文書のやりとりは日常的なものです。当たり前のようにメールを送ったら届きますが、その裏側では、送信側と受信側で「SMTPサーバー」が動いてメールを配送しています。

送信側のSMTPサーバーは、メールアドレスから適切な組織を判断し、相手側の組織のSMTPサーバーに対してメールを転送します。自分の組織宛てのメールを受け取ったSMTPサーバーはメールを格納しておく「スプール」という保存領域に蓄積します。

まずはSMTPサーバーを作らないことにはメールを送ることも、外部の組織からメールを受け取ることもできません。SMTPサーバーのことを別名MTA（Mail Transfer Agent、メール転送エージェント）とも呼びます。

この章では、MTAの中でも管理や設定が容易で、高速かつ安全であることを指向して開発されている、Postfixを使った構築方法について説明します。



HINT!

MTAとMRA、MUA

メールサーバーは、いくつかの役割をするサーバーソフトからなります。Postfixのような、SMTPによってメールを配信するサーバーを、MTA（Mail Transfer Agent）とも呼びます。一方、MTAが受信してスプールに保存した自分宛てのメールをユーザーが取得するには、**レッスン12-2**のようにPOP3やIMAP4のサーバーを使います。こうしたサーバーをMRA（Mail Retrieval Agent）とも呼びます。なお、メールサーバーに接続してメールを送信するメールクライアントのことを、MUA（Mail User Agent）とも呼びます。

Postfixの設定

PostfixはRHEL 7の「Core」パッケージグループに含まれるため、ほとんどの環境で最初からインストールされています。もしインストールされていない場合は、yumコマンドで追加インストールしてください。

Postfixの主な設定ファイルは/etc/postfix/main.cfです。初期設定では、サーバーの情報やほかのホストからの接続などが設定されていないので、設定を変更します。

1 元の設定ファイルをバックアップする

変更する前の設定ファイルをコピーしておく

コマンドを入力

```
[root@host1 ~]# cp -p /etc/postfix/main.cf /etc/postfix/main.cf.orig
[root@host1 ~]#
```

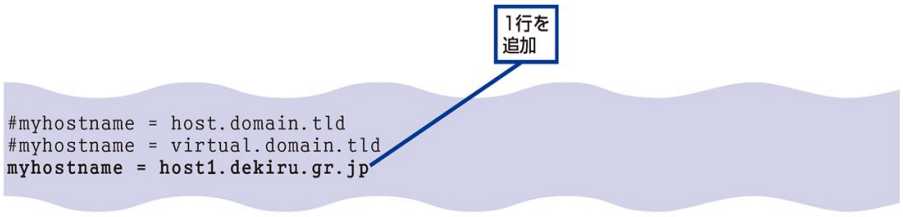
2 設定ファイルを開く

コマンドを入力

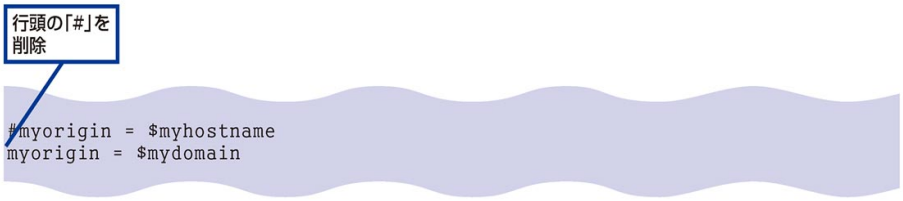
```
[root@host1 ~]# vi /etc/postfix/main.cf
```

次のページに続く

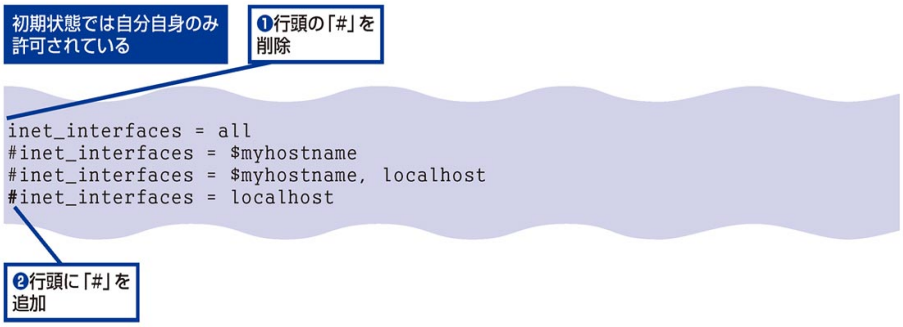
3 ホスト名を指定する



4 メールアドレスのドメイン名を設定する



5 接続を許可する相手を設定する



HINT!

ネットワーク関連の設定項目

手順❸～❽では、ネットワーク関連の項目を設定しています。myhostnameはホスト名の設定で、多くの設定のデフォルト値として使われます。myoriginは、ローカルで送信されたメールに付くドメイン名です。inet_interfacesはメールを受け付ける相手です。mydestinationは、ローカル配送とするドメインのリストです。mynetworksは、信頼できるメールクライアントのリストです。

6 自分自身に配送するドメインを設定する

メールアドレスが指定に合致すればほかのホストに配送しない

❶行頭に「#」を追加

```
#mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain
mydestination = $myhostname, $mydomain, localhost
```

❷1行を追加

7 ローカルネットワークを設定する

1行を追加

```
#mynetworks = 168.100.189.0/28, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table
mynetworks = 192.168.0.0/24, 127.0.0.0/8
```

8 メールボックスを設定する

ここでは受信したメールをユーザーのホームディレクトリの下に保存するようにする

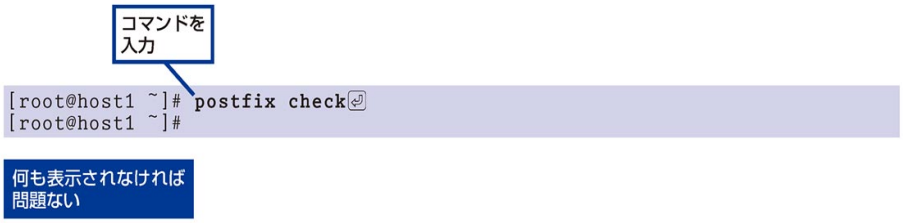
```
#home_mailbox = Mailbox
home_mailbox = Maildir/
```

❶行頭の「#」を削除

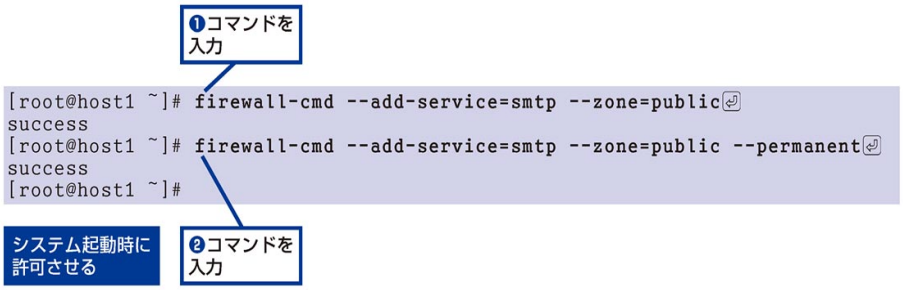
❷保存して終了

次のページに続く

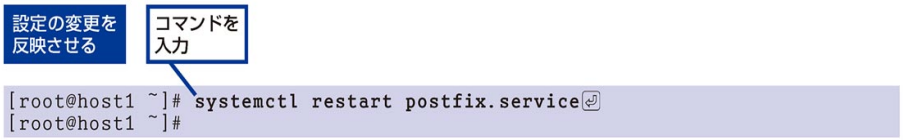
9 設定の書式をチェックする



10 ファイアウォールで許可する



11 Postfixを再起動する



HINT!

メールボックスの設定内容

手順❽では、home_mailboxを有効にして設定しています。home_mailboxを設定すると、Postfixの受信したメールが、システム共通のスプールのディレクトリではなく、宛先となる各ユーザーのホームディレクトリ以下に保存されます。また、「Maildir/」のように最後に「/」を付けると、受信メールをディレクトリで管理する方式となります。このように設定すると、**レッスン12-2**以降のようにDovecotで扱いやすくなります。

コマンドで設定する場合

Postfixの設定項目は非常に多いため、設定ファイルを変更する専用のpostconfコマンドが用意されています。postconfコマンドを使いこなすと、コマンド履歴に残せるので便利です。

ここまでの設定にpostconfコマンドを使う例は、次のようになります。-eオプションは編集を意味し、オプションの後に記述した変数の値を書き換えます。

```
[root@host1 ~]# postconf -e 'myhostname = host1.dekiru.gr.jp'
[root@host1 ~]# postconf -e 'myorigin = $mydomain'
[root@host1 ~]# postconf -e 'inet_interfaces = all'
[root@host1 ~]# postconf -e 'mydestination = $myhostname, $mydomain, localhost'
[root@host1 ~]# postconf -e 'mynetworks = 192.168.0.0/24, 127.0.0.0/8'
[root@host1 ~]# postconf -e 'home_mailbox = Maildir/'
[root@host1 ~]#
```

12-2 POP3サーバーを作るには

Dovecot

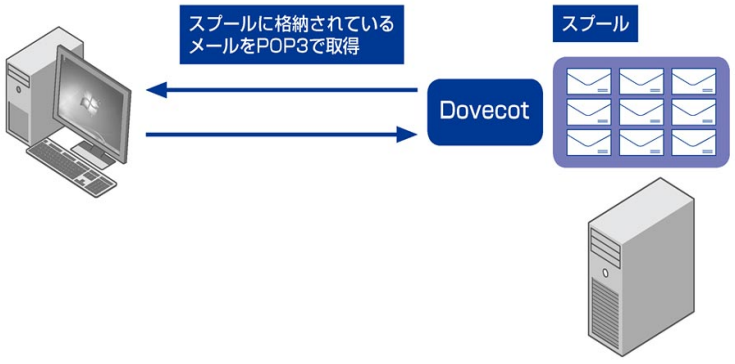
Dovecotは、Postfixと組み合わせて使われることが多いPOP3／IMAP4サーバーです。RHEL 7にも標準で含まれています。POP3やIMAP4といったプロトコルを通じて、ネットワーク経由でメールサーバー上のスプールからメールを取り出すことができます。Dovecotは主に受信サーバーとして機能します。

このレッスンでは、POP3でメールの受信を行うためのDovecotの基本的な設定方法を説明します。なお、インターネット越しにメールの受信を行う場合には、SSL通信などと組み合わせて通信を暗号化するなどの考慮が必要です。

POP3サーバー Dovecot

SMTPサーバーが組織内の他のユーザーや外部の組織から受け取ったメールは、スプールに格納されます。スプールに格納されたメールを閲覧するには、SSH経由でリモートログインしてサーバー上で読むか、もしくはWindowsやMacのメールクライアントで、POP3やIMAP4などのプロトコルを用いてスプールに格納されているメールを受信する必要があります。

ここでは、仕組みが分かりやすいPOP3プロトコルを使ってメールを受信する仕組みを説明します。POP3サーバーとしてはDovecotを使います。DovecotはPOP3とIMAP4に対応しており、管理や設定が容易で、Postfixとの親和性も高いのが特徴です。



基本的な設定

まずは、POP3サーバーとDovecotの理解のために、どのようなメールクライアントでも利用できる設定を説明します。

ここで説明する設定では、パスワードがネットワーク上を、暗号化されていないプレーンテキストで流れます。そのため、インターネット越しにアクセスする場合はセキュリティ的に問題があります。この設定は、LAN内でメールサーバーへアクセスする場合にご利用ください。

実際にPOP3サーバーを立てる場合は、パスワードをチャレンジレスポンス認証にしたり、メールクライアントとメールサーバーの間でSSLによりデータを暗号化したりすることを推奨します。

1 Dovecotをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install dovecot
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package         アーキテクチャー
                                バージョン                  リポジトリー              容量
================================================================================
インストール中：
 dovecot         x86_64         1:2.2.10-4.el7_0.1      rhel-7-server-rpms    3.2 M

インストール容量： 12 M
Is this ok [y/d/N]: y
Downloading packages:


完了しました！
[root@host1 ~]#
```

Dovecotのパッケージが表示される

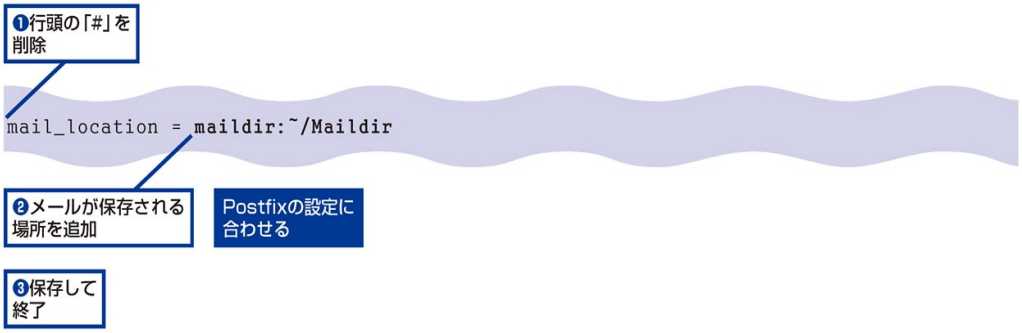
❷「y」と入力してEnterキーを押す

インストールが完了した

次のページに続く

2 メールボックスの設定ファイルを開く

3 メールが保存される場所を設定する



HINT!

設定ファイルは分割されている

Dovecotの設定ファイルは、デフォルトで設定の種類別に分割されて、/etc/dovecot/conf.dディレクトリに「*.conf」というファイル名で置かれています。メインの設定ファイルである/etc/dovecot/dovecot.confもありますが、分割された設定ファイルを読み込む指定がほとんどです。ファイル名などを参考に、目的に合った設定ファイルを編集しましょう。

HINT!

メールボックスの設定をPostfixと合わせる

手順❸では、Dovecotが読みにいくメールボックスの場所とその形式を設定しています。この設定は、メールボックスに保存するPostfixの設定と合わせましょう。ここで設定している内容のうち、「maildir:」がMaildir形式を表し、「~/Maildir」がディレクトリを表します。

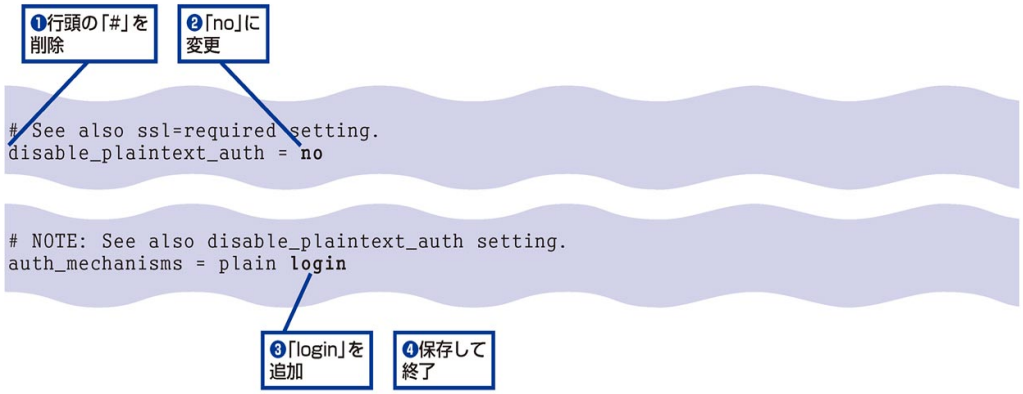
HINT!

プレーンテキストで認証する

手順❺では、まず一番簡単なユーザー認証方法として、暗号化しない方式の禁止を解除してから、プレーンテキストのパスワード認証方式である「plain」と「login」を指定しています。この2つの方式では、OSにログインするときのユーザーとパスワードで認証します。パスワードは暗号化されずに送信されるので、もし盗聴されるとパスワードを盗まれてしまう危険性があります。なお、「login」はOutlookなど一部のメールクライアントが使う方式です。

4 認証の設定ファイルを開く

5 プレーンテキストのパスワード認証を許可する



次のページに続く

6 SSLの設定ファイルを開く

変更する前の設定ファイルをコピーしておく

❶コマンドを入力

```
[root@host1 ~]# cp -p /etc/dovecot/conf.d/10-ssl.conf /etc/dovecot/conf.d/10-ssl.conf.
orig
[root@host1 ~]# vi /etc/dovecot/conf.d/10-ssl.conf
```

設定ファイルを編集する

❷コマンドを入力

7 SSLを不要にする

❶「no」に変更

```
# plain imap and pop3 are still allowed for local connections
ssl = no
```

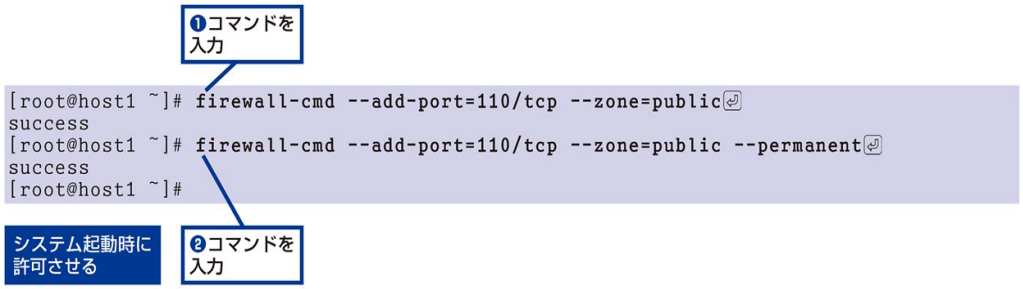
❷保存して終了

HINT!

IMAP4も使える

本章では、Dovecotを設定してメールクライアントがPOP3によりメールを取得できるようにしました。Dovecotからメールを取り出すプロトコルとしては、POP3のほかにIMAP4も使えます。POP3では基本的に、取り出したメールはサーバーから削除され、メールクライアントで管理します。一方、IMAP4では、一度読んだメールもサーバー上に残して管理するため、複数のメールクライアントでメールを読んでも一元的に管理できあす。

8 ファイアウォールで許可する



9 Dovecotを開始する

12-3 POP3の認証のセキュリティレベルを上げるには

チャレンジ&レスポンス認証

レッスン12-2では、Dovecotの基本的な設定をしてPOP3でメールを受信できるようにしました。ただし、すでに書いたように、プレーンテキストのパスワードがネットワーク上を流れるため、セキュアではありません。

このレッスンでは、ネットワークにプレーンテキストのパスワードを流さないように、POP3認証にチャレンジ&レスポンス認証（CRAM-MD5）を利用する方法を説明します。なお、CRAM-MD5認証に対応していないメールクライアントもありますので、事前にご確認ください。

1 Dovecotのパスワードファイルを作る

/etc/passwordからユーザーを抽出する

コマンドを入力　抽出するユーザー名　Dovecotのパスワードファイル

```
[root@host1 ~]# grep htaira /etc/passwd >> /etc/dovecot/users
[root@host1 ~]#
```

HINT!

/etc/passwdとDovecotの互換性

手順❶では、/etc/passwdファイルからDovecotのパスワードファイルを作っています。Dovecotのパスワードファイルは、/etc/passwdファイルと互換性があります。そのため、Dovecotのpasswdファイルを作る場合、POP3を利用するユーザー名によりgrepコマンドで抽出し、リダイレクトで追記すればミスも少なくなるでしょう。

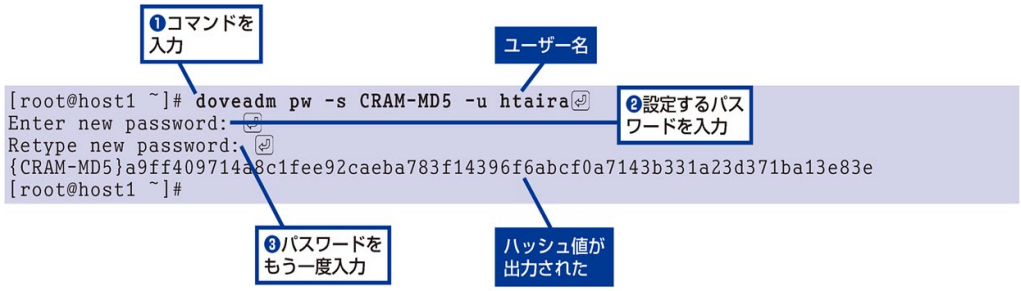
HINT!

本文は暗号化されない

このレッスンのように設定して、メールクライアントでPOP3認証の種類としてCRAM-MD5を設定すれば、認証が暗号化されます。ただし、メール本文は暗号化されませんのでご注意ください。

2 パスワードのハッシュ値を生成する

パスワードファイルに記述するハッシュ値を生成する



3 ハッシュ値をパスワードファイルに記述する

パスワードファイルを編集する

❶コマンドを入力

```
[root@host1 ~]# vi /etc/dovecot/users
```

❷「:」で区切った2つ目のフィールドの「x」を、生成したハッシュ値に変更

```
htaira:{CRAM-MD5}a9ff409714a8c1fee92caeba783f14396f6abcf0a7143b331a23d371ba13e83e:1000
:1000:Hajime Taira:/home/htaira:/bin/bash
```

❸保存して終了

4 パスワード認証の設定ファイルを開く

変更する前の設定ファイルをコピーしておく

❶コマンドを入力

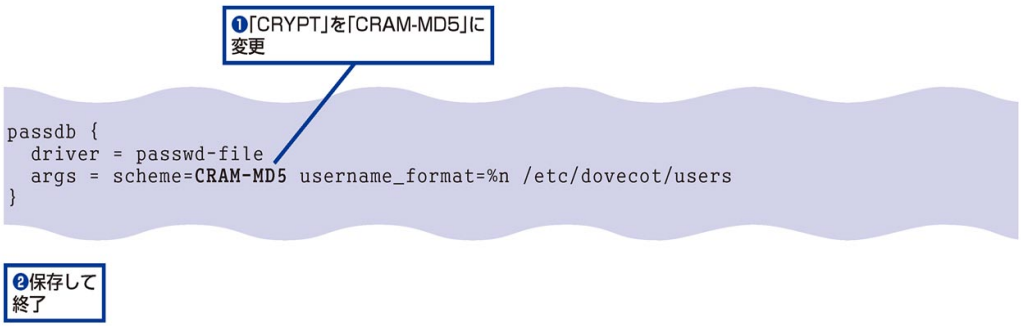
```
[root@host1 ~]# cp -p /etc/dovecot/conf.d/auth-passwdfile.conf.ext /etc/dovecot/conf.
d/auth-passwdfile.conf.ext.orig
[root@host1 ~]# vi /etc/dovecot/conf.d/auth-passwdfile.conf.ext
```

設定ファイルを編集する

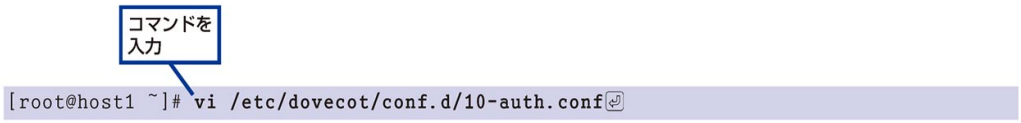
❷コマンドを入力

次のページに続く

5 チャレンジ&レスポンス認証を指定する



6 認証の設定ファイルを開く



HINT!

プレーンテキストのパスワード認証を禁止するには

レッスン12-2で設定した「disable_plaintext_auth」の値を「no」にすれば、プレーンテキスト認証が許可されなくなります。また、Linuxのユーザーでの認証を無効にするには、手順❻で設定している「auth_mechanisms」の値を「cram-md5」のみにすれば、Dovecotのパスワードファイルに登録されているユーザーしかPOP3を利用できなくなります。

HINT!

POP3の通信を暗号化するには

このレッスンで設定したチャレンジ&レスポンス認証を利用しても、POP3の通信の全体が暗号化されるわけではありません。一番安全な方法は、POP3S（POP3 over SSL/TLS）を使用することです。ただし、POP3Sの場合には、古いメールクライアントでPOP3Sが利用できなかったりするほか、SSL証明書を証明局に発行してもらう手続きと費用が必要となります（**レッスン7-3**を参照）。

7 チャレンジ&レスポンス認証を有効にする

❶「cram-md5」を追加

```
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = cram-md5 plain login
```

```
#!include auth-ldap.conf.ext
!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
```

❷行頭の「#」を削除

パスワード認証の設定ファイルを有効にする

❸保存して終了

8 Dovecotを再起動する

設定の変更を反映させる

コマンドを入力

```
[root@host1 ~]# systemctl restart dovecot.service
[root@host1 ~]#
```

12-4 メール送信に認証をかけるには

SMTP-AUTH

メールクライアントからPOP3でメールを受信する際には、認証を通る必要があります。しかし、メールクライアントからSMTPサーバーに送信するときには、デフォルトでは認証がありません。

メール送信時のSMTPサーバーに対してもユーザー認証を設けたい場合、SMTP-AUTHという仕組みを利用します。具体的な実装方法はいくつかありますが、このレッスンではPOP3サーバーのDovecotの認証を「SASL」という機構で利用してSMTP-AUTHの認証を行う方法を解説します。

1 DovecotでSASLを有効にする

変更する前の設定ファイルをコピーしておく

❶コマンドを入力

```
[root@host1 ~]# cp -p /etc/dovecot/conf.d/10-master.conf /etc/dovecot/conf.d/10-master.conf.orig
[root@host1 ~]# vi /etc/dovecot/conf.d/10-master.conf
```

設定ファイルを編集する

❷コマンドを入力

```
  # Postfix smtp-auth
 unix_listener /var/spool/postfix/private/auth {
    mode = 0666
 }
```

❸3行の行頭の「#」を削除

「#」を削除した行の間に設定を追加する

❹2行を追加

```
 unix_listener /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    group = postfix
 }
```

❺保存して終了

2 Dovecotを再起動する

設定の変更を反映させる　コマンドを入力

```
[root@host1 ~]# systemctl restart dovecot.service
[root@host1 ~]#
```

3 PostfixでSMTP-AUTHを設定する

❶コマンドを入力

```
[root@host1 ~]# vi /etc/postfix/main.cf
```

❷行頭に「#」を追加

```
#mynetworks = 192.168.0.0/24, 127.0.0.0/8
```

❸1行を追加

```
mynetworks = 127.0.0.0/8
```

```
# SMTP-AUTH
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated, reject_
unauth_destination
```

❹ファイル末尾に追加　❺保存して終了

4 Postfixを再起動する

設定の変更を反映させる　コマンドを入力

```
[root@host1 ~]# systemctl restart postfix.service
[root@host1 ~]#
```

STEP UP

Outbound Port 25 Blocking

インターネットサービスプロバイダーによっては、SMTPのポート番号であるTCPポート 25番のやりとりに制限を行っている場合があります。これは迷惑メールを送信するスパム業者が一般家庭向けのインターネット回線を使って、バルクで迷惑メールを送れないようにしたり、第三者からメールサーバーを悪用されないようにする対策であり「Outbound Port 25 Blocking（通称：OP25B)」と呼ばれています。

そのため、メールサーバーをインターネットに公開する場合には、ご契約されているインターネットサービスプロバイダーに事前に確認した方がいいでしょう。法人契約向けの固定IP付きのインターネット回線の場合、この制限がされていないことが多いでしょう。もしも制限があるインターネット回線であれば、プロバイダー側の設備でポートをブロックしてしまうため、SMTPサーバーを構築しても外部宛てにメールが送信できなかったりするトラブルが発生します。

設置したSMTPサーバーが不正なメール配信に使われないように保護されている面もあるので、メールサーバーの設置を禁止されていたり、転送制限されたりしていた場合には、割り切って他のプロバイダーや法人向けメニューを検討しましょう。

第13章 データベースサーバーを作る

定型化されたデータを保存する仕組みとしてデータベースがあります。この章では、RHEL 7に含まれるMariaDBとPostgreSQLについて説明します。インフラ担当者としてデータベースサーバーを構築する程度の範囲ですので、データベースの基礎知識はなくても結構です。インフラ担当者が一般知識としてデータベースを知っておくというのが、この章のゴールです。

●この章の内容

13-1 MariaDBを知ろう

MariaDBの概要

MariaDBは、RHEL 7から従来のMySQLの代わりに新しく採用されたRDBMS（Relational Database Management System）です。MySQLのオリジナルの開発者であり、MySQL ABの創立者でもあるMichael "Monty" Widenius氏によって、MySQLから派生したプロジェクトです。MariaDBのプログラム名、設定ファイル名および操作コマンドなどMySQLとほぼ同様です。ちなみに、RHEL 7からMySQLは含まれません。

このレッスンでは、RHEL 7に採用されたMariaDB 5.5とMySQL 5.5の違いや、MariaDBのインストール方法、起動方法について説明します。

RDBMS ってなに？

データベースでは、RDB（Relational Database、リレーショナルデータベース）という方式がよく使われています。RDBのサーバーをRDBMS（Relational Database Management System、リレーショナルデータベース管理システム）と呼びます。

RDBでは列と行という単位でテーブルという領域にデータを格納します。そして、問い合わせ言語SQLを使ってデータの変更や抽出、整形処理を行います。

単なるファイルにデータを格納する場合と違う点の一つとして、複数のクライアントから同じデータに対して同時にアクセスした場合に、RDBMSがテーブルや行の排他制御を行う点があります。これにより、万が一複数のクライアントが同時に変更要求を送った場合でもデータが破壊されないようにできます。

MariaDB ってなに？

MariaDBはオープンソースのRDBMSで、同じくRDBMSであるMySQLから派生したデータベースです。RHEL 6ではMySQLが採用されていましたが、RHEL 7からはMariaDBが採用されました。

RHEL 7に搭載されているMariaDB 5.5はMySQL 5.5をベースに、スレッドプール追加と実行計画の改良が行われています。MariaDB 5.5とMySQL 5.5の操作系は一緒であり、起動や停止などの制御コマンドも同じです。また、MariaDB 5.5が受け付けるSQLのクエリーの文法も、基本的な部分はMySQL 5.5と同じです。よって、RHEL 7にはMySQLは収録されていません。

MariaDBのインストール

1 MariaDBをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install mariadb-server
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package                    アーキテクチャー
                                    バージョン              リポジトリー              容量
================================================================================
インストール中：
 mariadb-server             x86_64  1:5.5.41-2.el7_0    rhel-7-server-rpms    11 M
依存性関連でのインストールをします：

インストール容量： 107 M
Is this ok [y/d/N]: y
Downloading packages:


完了しました！
[root@host1 ~]#
```

MariaDBと依存パッケージが表示される

❷「y」と入力してEnterキーを押す

インストールが完了した

2 MariaDBを開始する

❶コマンドを入力

```
[root@host1 ~]# systemctl start mariadb.service
[root@host1 ~]# systemctl enable mariadb.service
ln -s '/usr/lib/systemd/system/mariadb.service' '/etc/systemd/system/multi-user.
target.wants/mariadb.service'
[root@host1 ~]#
```

システム起動時に開始するようにする

❷コマンドを入力

13-2 MariaDBを使ってみる

MariaDBの操作

MariaDBを操作するには、コマンドラインインターフェイスのmysqlコマンドを使います。mysqlコマンドだけインストールして、リモートホストで動いているMariaDBに接続することもできます。Windows版やMac OS X版のmysqlもあるので、RHEL 7以外の環境から直接接続することも可能です。

このレッスンでは、クライアントから接続し、簡単なSQL文を発行して、ユーザーの追加、テーブルの作成、データの挿入、データの参照を試します。データベース操作としては基本中の基本であり、SQLに慣れる程度です。

データベースの作成

MariaDBに新規データベースを作成してみましょう。MariaDBのデータベースを作成するには、mysqladminコマンドを実行します。

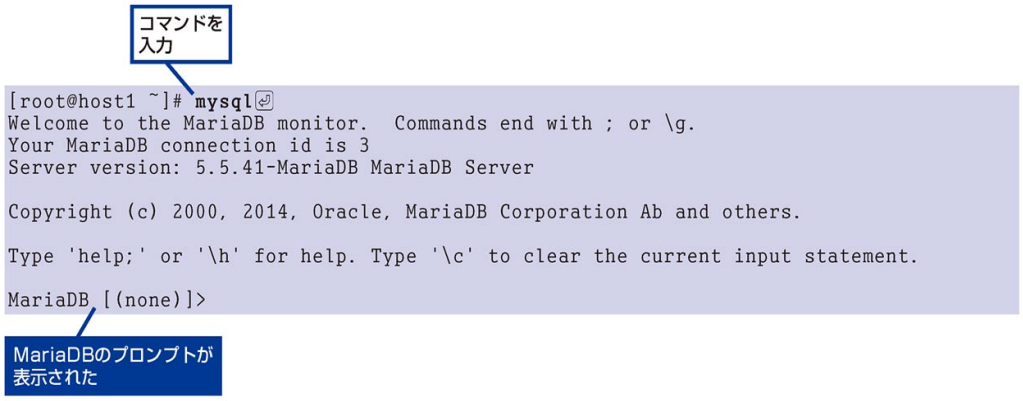
1 データベースを作る

MariaDBへの接続

MariaDBでは、Linuxのユーザーとは別にユーザーを管理しています。MariaDBのユーザーでも、rootユーザーが管理者権限を持ち、データベースの設定変更や操作が自由にできます。デフォルトではrootユーザーのみ用意されており、MariaDBがインストールされているホストからのみ接続できます。

MariaDBへ接続するには、rootユーザーのコマンドラインからmysqlコマンドを実行します。

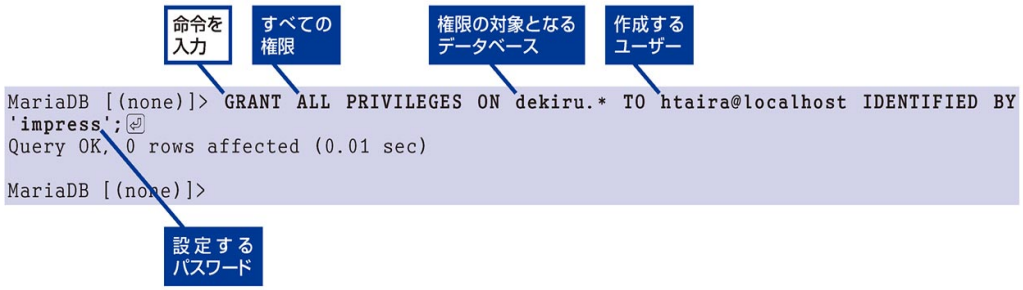
2 MariaDBに接続する



ユーザーの作成

MariaDBでは、一般ユーザーを作成するだけの操作は通常は使われません。ユーザーにアクセス権限を与えるGRANT文によって、ユーザーの新規作成も行います。このときに、パスワードも一緒に設定します。

3 ユーザーを作成する

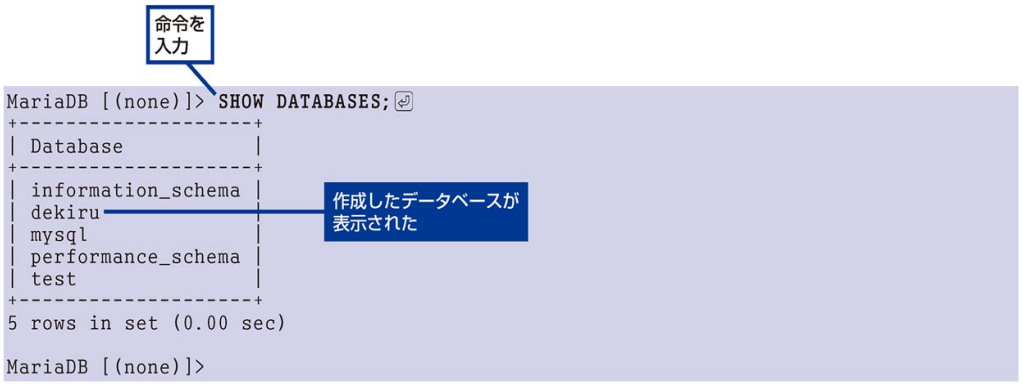


次のページに続く

データベースへの接続

SELECTやINSERTなどのSQL文を発行するには、任意のデータベースに接続する必要があります。MariaDBのプロンプトからデータベースに接続するには、USE命令を実行します。

4 データベースの一覧を表示する

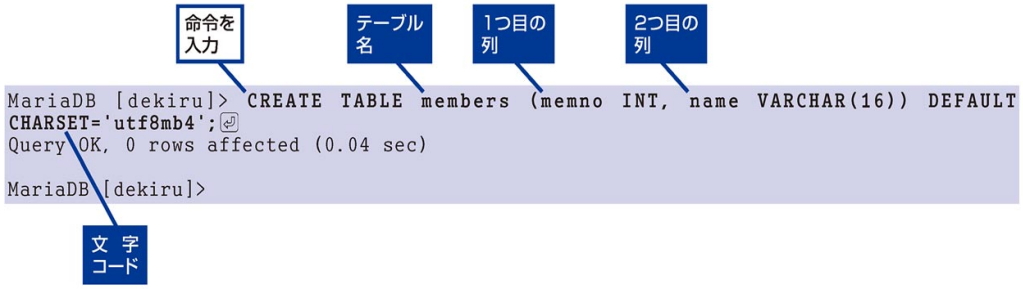


5 データベースに接続する

データの操作

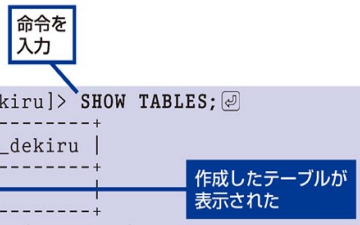
データを操作するには、SQLの命令を発行します。ここでは、テーブルを作って、INSERT文でデータを追加し、SELECT文でそのデータを取得してみます。

6 テーブルを作る




```
MariaDB [dekiru]> CREATE TABLE members (memno INT, name VARCHAR(16)) DEFAULT
CHARSET='utf8mb4';
Query OK, 0 rows affected (0.04 sec)

MariaDB [dekiru]>
```

7 テーブルの一覧を表示する




```
MariaDB [dekiru]> SHOW TABLES;
+------------------+
| Tables_in_dekiru |
+------------------+
| members          |
+------------------+
1 row in set (0.00 sec)

MariaDB [dekiru]>
```

次のページに続く

8 データを1つ追加する

命令を入力

```
MariaDB [dekiru]> INSERT INTO members VALUES (0, '平初');
Query OK, 1 row affected (0.00 sec)

MariaDB [dekiru]>
```

9 データを取得する

命令を入力

```
MariaDB [dekiru]> SELECT * FROM members;
+-------+--------+
| memno | name   |
+-------+--------+
|     0 |  平初   |
+-------+--------+
1 row in set (0.00 sec)

MariaDB [dekiru]>
```

データが表示された

HINT!

DDL文とDML文

SQLでは大きく分けて、DDL（データ定義言語）とDML（データ操作言語）の2つの種類のSQLがよく使われます。CREATE TABLEなどはDDL文と呼ばれ、テーブルの作成や、変更、削除などを行います。SELECTやINSERTなどはDML文と呼ばれ、データの取得や追加、削除、更新などデータの操作を行います。そのほかにはUPDATE、DELETEなどのコマンドもあります。RHELの管理者は、基本的なDDL文ぐらいは理解しておくと、データベース管理者と会話がかみ合います。

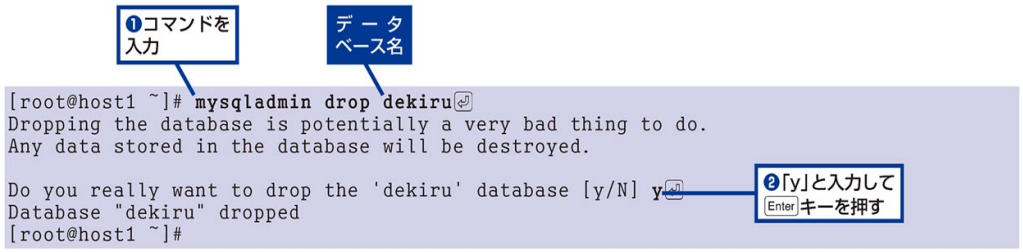
MariaDBからの切断

MariaDBのプロンプトから抜けて終了するには、\qコマンドを実行します。

10 MariaDBから切断する

データベースの削除

MariaDBのデータベースを削除したい場合が出てくると思います。データベースを削除したい場合は、rootユーザーでmysqladminコマンドを実行します。



13-3 PostgreSQLを知ろう

PostgreSQLの概要

PostgreSQLはオブジェクト型RDBMSです。Berkleyで書かれたPostgresから派生しています。オブジェクト型RDBMSなのでテーブル定義の継承も可能で、テーブルに親子関係を定義してスキーマ設計ができます。トランザクションやトリガー機能はもちろんのこと、同期／非同期レプリケーション機能も備え、多機能です。MariaDBと同じく、問い合わせ言語にはSQLを使います。

このレッスンでは、PostgreSQLのインストール方法、データベースの処理化、サービスの開始方法について説明していきます。

PostgreSQL ってなに？

PostgreSQLはMariaDBと同様に、問い合わせ言語にSQLをサポートするオープンソースのRDBMSです。PostgreSQLはオープンソースのRDBMSの中で人気が強く、主要なプログラミング言語にはライブラリが提供されており、幅広い用途で利用されています。そのためにPostgreSQLをサポートするWebアプリケーションも多く、現在も活発に開発が続いています。

HINT!

データベースの初期化が必要

PostgreSQLをインストールしたら、使う前にデータベースを初期化する必要があります。RHEL 6では「service postgresql initdb」を実行してデータベースを初期化していました。RHEL 7ではサービスの管理方法が変わった関係で、手順❷のように「postgresql-setup initdb」を実行します。

PostgreSQLのインストール

1 PostgreSQLをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install postgresql-server
```

```
================================================================================
 Package                アーキテクチャー
                                バージョン              リポジトリー              容量
================================================================================
インストール中：
 postgresql-server      x86_64    9.2.10-2.el7_1      rhel-7-server-rpms    3.8 M
依存性関連でのインストールをします：
```

PostgreSQLと依存パッケージが表示される

```
インストール容量： 33 M
Is this ok [y/d/N]: y
Downloading packages:

完了しました！
[root@host1 ~]#
```

❷「y」と入力してEnterキーを押す

インストールが完了した

2 PostgreSQLを初期化する

RHEL 7ではサービスを起動する前に初期化する必要がある

コマンドを入力

```
[root@host1 ~]# postgresql-setup initdb
Initializing database ... OK

[root@host1 ~]#
```

3 PostgreSQLを開始する

❶コマンドを入力

```
[root@host1 ~]# systemctl start postgresql.service
[root@host1 ~]# systemctl enable postgresql.service
ln -s '/usr/lib/systemd/system/postgresql.service' '/etc/systemd/system/multi-user.
target.wants/postgresql.service'
[root@host1 ~]#
```

システム起動時に開始するようにする

❷コマンドを入力

13-4 PostgreSQLを使ってみる

PostgreSQLの操作

PostgreSQLを操作するには、コマンドラインインターフェイスのpsqlコマンドを使いこなす必要があります。psqlコマンドだけインストールして、リモートホストで動いているPostgreSQLに接続することもできます。Windows版やMac OS X版のpsqlもあるので、RHEL 7以外の環境から直接接続することも可能です。

このレッスンでは、psqlコマンドの使い方を説明します。psqlコマンドでPostgreSQLに接続し、MariaDBと同じようにユーザーとデータベースの作成／削除、パスワードと権限付与、テーブルの作成など基本的な操作について解説します。

PostgreSQLへの接続

PostgreSQLでも、Linuxのユーザーとは別にユーザーを管理しています。デフォルトの設定では、Linuxと同じユーザーとして接続する必要があります。PostgreSQLの管理権限を持つpostgresユーザーで接続するため、suコマンドでpostgresユーザーへ切り替えてからpsqlコマンドを実行します。

1 ユーザーを切り替える

コマンドを入力　ユーザー名

```
[root@host1 ~]# su - postgres
-bash-4.2$
```

2 PostgresSQLに接続する

コマンドを入力

```
-bash-4.2$ psql
psql (9.2.10)
"help" でヘルプを表示します .

postgres=#
```

PostgreSQLのプロンプトが表示された

HINT!

「\」で始まるコマンド

PostgreSQLでは、ユーザーの一覧表示や、データベースの一覧表示、テーブルの一覧表示などにも「\」で始まるコマンドを使います。これらのコマンドの一覧は、「\?」コマンドで表示できます。

ユーザーの作成

PostgreSQLに一般ユーザー apacheを追加してみましょう。そのためには、SQLのCREATE USER文を発行します。

3 ユーザーを作成する

命令を入力　　ユーザー名

```
postgres=# CREATE USER apache;
CREATE ROLE
postgres=#
```

4 ユーザーの一覧を表示する

命令を入力

```
postgres=# \du
                                   ロール一覧
   ロール名 |                                   属性
 |  メンバー
----------+------------------------------------------------------------------------
-+----------
 apache    |
 | {}
 postgres  | スーパーユーザ， ロールを作成できる， DB を作成できる， レプリケーション
 | {}

postgres=#
```

作成したユーザーが表示された

次のページに続く

データベースの作成

PostgreSQLに新規データベースを作成してみましょう。そのためには、SQLでCREATE DATABASE文を発行します。

5 データベースを作る

命令を入力
データベース名
文字コード
所有者

```
postgres=# CREATE DATABASE webdb ENCODING 'UTF8' OWNER apache;
CREATE DATABASE
postgres=#
```

6 データベースの一覧を表示する

命令を入力

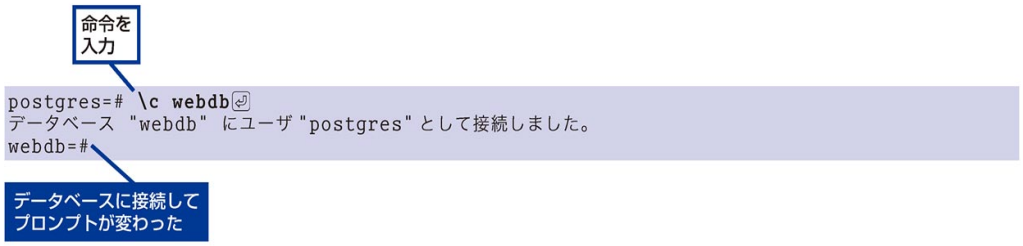
```
postgres=# \l
                                        データベース一覧
    名前     |   所有者   |   エンコーディング   |     照合順序    |  Ctype( 変換演算子 )  |
  アクセス権
-----------+----------+------------------+-------------+-------------------+----
------------------
 postgres  | postgres | UTF8             | ja_JP.UTF-8 | ja_JP.UTF-8       |
 template0 | postgres | UTF8             | ja_JP.UTF-8 | ja_JP.UTF-8       | =c/
postgres          +
           |          |                  |             |                   | pos
tgres=CTc/postgres
 template1 | postgres | UTF8             | ja_JP.UTF-8 | ja_JP.UTF-8       | =c/
postgres          +
           |          |                  |             |                   | pos
tgres=CTc/postgres
 webdb     | apache   | UTF8             | ja_JP.UTF-8 | ja_JP.UTF-8       |
(4 行 )

postgres=#
```

作成したデータベースが表示された

データベースへの接続

SELECTやINSERTなどのSQL文を発行するには、任意のデータベースに接続する必要があります。psqlのプロンプトからデータベースに接続するには、\cコマンドを実行します。

7 データベースに接続する



ユーザーのパスワードの設定

ここまで、PostgreSQLの一般ユーザー apacheにはパスワードが付与されていません。そのため、ALTER USER文を発行してパスワードを設定します。なお、2回目以降にALTER USER文を発行すると、パスワードを変更できます。

8 パスワードを設定する

次のページに続く

ユーザーの権限の付与

apacheユーザーに対して、データベースwebdbに対するすべての権限を与えるには、GRANT文を発行します。権限を付与することにより、apacheユーザーでデータベースwebdbへ接続できるようになります。

9 権限を付与する


```
webdb=# GRANT ALL PRIVILEGES ON DATABASE webdb TO apache;
GRANT
webdb=#
```

データの操作

データを操作するには、SQLの命令を発行します。ここでは、テーブルを作って、INSERT文でデータを追加し、SELECT文でそのデータを取得してみます。

10 テーブルを作る


```
webdb=# CREATE TABLE members (memno INT, name VARCHAR(16));
CREATE TABLE
webdb=#
```

HINT!

ユーザーを作る方法

このレッスンではCREATE USERでPostgreSQLのユーザーを作ってから、後でALTER USERでパスワードを設定していました。このほかに、ユーザーを作るときにパスワードを設定することもできます。なお、MariaDBでは、GRANT文でユーザー作成と権限設定を同時にしていましたが、CREATE USERでユーザー作成をすることもできます。

11 テーブルの一覧を表示する

命令を入力

```
webdb=# \d
              リレーションの一覧
  スキーマ  |   名前    |    型     |  所有者
----------+---------+----------+----------
 public   | members |  テーブル  | postgres
(1 行)

webdb=#
```

作成したテーブルが表示された

12 データを1つ追加する

命令を入力

```
webdb=# INSERT INTO members VALUES (0, '平初');
INSERT 0 1
webdb=#
```

13 データを取得する

命令を入力

```
webdb=# SELECT * from members;
 memno | name
-------+------
     0 | 平初
(1 行)

webdb=#
```

データが表示された

次のページに続く

ユーザーの削除

PostgreSQLの一般ユーザーを削除する方法もご紹介しておきます。一般ユーザーを削除するには管理者ユーザーでログインしてDROP USER文を発行します。なお、その際に対象の一般ユーザーがデータベースの所有者になっていた場合、事前にデータベースの所有者を他のユーザーへ変更する（ALTER DATABASE文）か、データベースを削除する必要があります。

14 データベースの所有者を変更する


```
webdb=# ALTER DATABASE webdb OWNER TO postgres;
ALTER DATABASE
webdb=#
```

15 ユーザーを削除する


```
webdb=# DROP USER apache;
DROP ROLE
webdb=#
```

データベースの削除

PostgreSQLのデータベースを削除したい場合が出てくると思います。データベースを削除したい場合は、データベースpostgresへ接続し直して、その上でDROP DATABASE文を発行します。

16 postgresデータベースに接続する


```
webdb=# \c postgres
データベース "postgres" にユーザ "postgres" として接続しました。
postgres=#
```

17 データベースを削除する

PostgreSQLからの切断

psqlのプロンプトから抜けて終了するには、\qコマンドを実行します。

18 PostgreSQLから切断する

19 元のユーザーに戻る

STEP UP

PostgreSQLへ接続するGUIクライアント

GUIベースのpgAdminというWindowsやMac OS Xで動くGUIベースのクライアントがあります。RHEL 7にはpgAdminは含まれていませんが、pgAdminの公式サイトからRHEL 7向けのRPMが用意されているため、簡単にインストールできます。

pgAdminの公式サイト

http://www.pgadmin.org/

psqlコマンドに慣れている人でも、pgAdminを使うことでPostgreSQL内のデータベースやテーブル構造をより直感的に把握することができて便利です。

pgAdminを利用した場合、リモートからアクセスすることが多いと思います。その際にはPostgreSQLの設定postgresql.confを変更する必要があります。

❶コマンドを入力

```
# cp -p /var/lib/pgsql/data/postgresql.conf{,.orig}
# vi /var/lib/pgsql/data/postgresql.conf
```

❷コマンドを入力　❸1行を追加

```
listen_addresses = '*'
```

ローカルネットワークからの接続の場合に、apacheユーザーからのパスワード認証を許可する場合には、次のようにpg_hba.confへ追記します。

❶コマンドを入力

```
# cp -p /var/lib/pgsql/data/pg_hba.conf{,.orig}
# echo "host  all  apache  192.168.0.0/24 md5" >> /var/lib/pgsql/data/
pg_hba.conf
```

❷コマンドを入力

なお、これらの設定を反映するには、postgresqlサービスの再起動が必要です。

また、ファイアウォールの設定も忘れずに行ってください。

❶コマンドを入力　❷コマンドを入力

```
# firewall-cmd --add-service=postgresql --zone=public
# firewall-cmd --add-service=postgresql --zone=public --permanent
```

第14章 CMSサーバーを作る

この章では、今までのWebサーバーとデータベースサーバーの章で習得した知識を活かして、Webサーバー上にCMS（Content Management System）の1つであるWordPressを構築してみましょう。WordPressの構築を通じて、Webとデータベースが連携するシステムを構築する上での勘所を掴むのが、この章のゴールです。

●この章の内容

14-1 CMSを作る前準備をするには

WordPressの準備

WordPressは、2003年に最初のバージョンがリリースされてから世界中で多く使われているブログシステムです。PHPというプログラミング言語で書かれており、生成できるページの柔軟性が高いため、CMSとしても使われます。

WordPressを動作させるには、WebサーバーとしてApacheとPHP、データベースサーバーとしてMariaDBを準備する必要があります。このレッスンでは、**第7章**のApacheと**第13章**のMariaDBがインストールされていることを前提に、その上にWordPressをインストールする手順を説明していきます。

PHPのインストール

1 PHPをインストールする

注意 この章の内容を実行するには、第7章のApacheと第14章のMariaDBを動くようにしておいてください。

❶コマンドを入力

```
[root@host1 ~]# yum install php php-mysql
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package         アーキテクチャー
                                バージョン                  リポジトリー                容量
================================================================================
インストール中：
 php             x86_64      5.4.16-23.el7_0.3       rhel-7-server-rpms      1.3 M
 php-mysql       x86_64      5.4.16-23.el7_0.3       rhel-7-server-rpms       97 k
依存性関連でのインストールをします：

インストール容量： 18 M
Is this ok [y/d/N]: y
Downloading packages:


完了しました！
[root@host1 ~]#
```

PHPと依存パッケージが表示される

❷「y」と入力してEnterキーを押す

インストールが完了した

HINT!

PHPでインストールされるもの

手順❶で「php」パッケージと、MariaDBに接続するための「php-mysql」をインストールしています。PHPはApacheのモジュールとして動的に組み込まれて動きます。「php」パッケージにはそのApacheのモジュールとApache設定が含まれますが、インストールするときに依存パッケージとして実際にPHPを使うためのファイルやライブラリなどがインストールされます。

データベースの用意

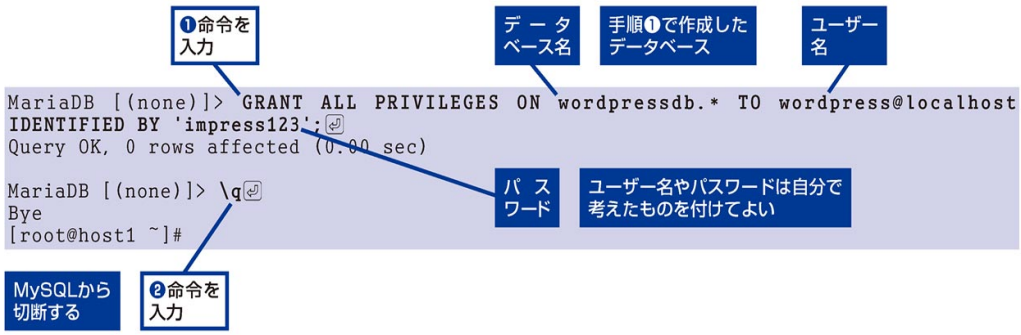
1 MariaDBのデータベースを作る


```
[root@host1 ~]# mysqladmin create wordpressdb
[root@host1 ~]#
```

2 MariaDBに接続する


```
[root@host1 ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.


MariaDB [(none)]>
```

3 データベースのユーザーを追加する




```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpressdb.* TO wordpress@localhost
IDENTIFIED BY 'impress123';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> \q
Bye
[root@host1 ~]#
```

14-2 CMSを作るには

WordPressのインストール

WordPressは、RHEL 7には含まれていません。そこで、WordPress.orgからtar.gz形式アーカイブのソースコードのダウンロードしてきます。そして、そのアーカイブを展開して、コピーするとインストール作業がほぼ完了します。また、MariaDBと連携して動くため、データベース名やユーザー名、パスワードを間違えて設定してしまうとデータベースに接続できず、WordPressが動きません。このレッスンでは、WordPressの入手から展開方法、ファイルのオーナーの変更、WordPressの設定方法、それにともなうSELinuxの設定まで説明していきます。

1 WordPressをダウンロードする

コマンドを入力

WordPress日本語版の最新版

```
[root@host1 ~]# wget http://ja.wordpress.org/latest-ja.tar.gz
--2015-05-25 22:22:16--  http://ja.wordpress.org/latest-ja.tar.gz
ja.wordpress.org (ja.wordpress.org) をDNSに問いあわせています... 66.155.40.249, 66.155.40.250
ja.wordpress.org (ja.wordpress.org)|66.155.40.249|:80 に接続しています... 接続しました。
HTTP による接続要求を送信しました、応答を待っています... 200 OK
長さ: 6773205 (6.5M) [application/octet-stream]
`latest-ja.tar.gz' に保存中

100%[======================================>] 6,773,205   1.97MB/s 時間 3.7s

2015-05-25 22:22:20 (1.74 MB/s) - `latest-ja.tar.gz' へ保存完了 [6773205/6773205]

[root@host1 ~]#
```

ダウンロードが完了した

HINT!

WordPress日本語版の最新版を入手

手順❶では、「WordPress日本語ローカルサイト」から、最初から日本語化されている「WordPress日本語版」を入手しています。WordPress日本語版では、日本語リソースの同梱や、日本語関連の不具合を修正するプラグインの同梱、デフォルトで日本語に設定、ドキュメントの翻訳などがなされています。また、最新のリリース版がlatest-ja.tar.gzという名前で提供されるため、同じURLで最新版がリリースできます。

HINT!

wgetコマンドでダウンロード

手順❶では、WordPressのアーカイブをダウンロードするのにwgetコマンドを使っています。wgetコマンドはコマンドラインからURLを指定してファイルをダウンロードするツールです。wgetでは、URLで指定されたファイルだけでなく、HTMLファイルからリンクしている先もダウンロードする、といった指定もできます。

2 アーカイブを展開する

コマンドを入力

```
[root@host1 ~]# tar xvf latest-ja.tar.gz
wordpress/
wordpress/wp-mail.php

wordpress/wp-admin/press-this.php
wordpress/readme.html
[root@host1 ~]#
```

アーカイブの内容が展開された

3 WordPressをコピーする

ApacheのDocument Rootの下に置く

コマンドを入力

```
[root@host1 ~]# cp -r wordpress /var/www/html
[root@host1 ~]#
```

次のページに続く

4 WordPressのディレクトリに移動する

コマンドを入力

```
[root@host1 ~]# cd /var/www/html/wordpress
[root@host1 ~]#
```

5 ファイルのオーナーを変更する

WordPressのすべてのファイルとディレクトリのオーナーをapacheにする

コマンドを入力

```
[root@host1 wordpress]# chown -R apache:apache .
[root@host1 wordpress]#
```

6 設定ファイルをコピーする

設定ファイルの雛形からコピーする

コマンドを入力

```
[root@host1 wordpress]# cp wp-config-sample.php wp-config.php
[root@host1 wordpress]#
```

HINT!

ファイルのオーナーの変更を忘れない

手順❺では、WordPressのファイルすべてのオーナーを、apacheグループのapacheユーザーに変更しています。WordPressのファイルは、Apacheに組み込まれたPHPが、読むだけでなく書き込む権限を持っている必要があります。そのため、オーナーをapacheに変更しています。オーナーの変更を忘れてしまうと、正常に動作せずにハマりますのでご注意ください。

HINT!

設定している内容

手順❼では、WordPressの設定を変更して、**レッスン14-1**で作ったデータベースの名前と、同じく**レッスン14-1**で作ったデータベースのユーザー名とパスワードを設定しています。

HINT!

外部のデータベースサーバーを使うには

手順❽では、WordPressを動かすための最低限のSELinuxの設定として、Apacheからのファイル書き込みとメール送信を許可しています。このほか、Webサーバーとは異なるサーバー上でデータベースサーバーが稼働している場合には、ApacheからMariaDBにネットワーク経由でアクセスするための設定も必要です。その場合は、SELinuxのブール値「httpd_can_network_connect」を「on」にします。

```
# setsebool -P httpd_can_network_connect on
```

7 設定ファイルを編集する

データベースに接続する情報を設定する

❶コマンドを入力

```
[root@host1 wordpress]# vi wp-config.php
```

❷データベース名を入力

```
** WordPress のためのデータベース名 */
define('DB_NAME', 'wordpressdb');

/** MySQL データベースのユーザー名 */
define('DB_USER', 'wordpress');

/** MySQL データベースのパスワード */
define('DB_PASSWORD', 'impress123');
```

❸データベースのユーザー名を入力

❹データベースのパスワードを入力

❺保存して終了

8 SELinuxの設定を変更する

❶コマンドを入力

Apacheからすべてのwebリソースへの書き込みを許可する

```
[root@host1 wordpress]# setsebool -P httpd_unified on
[root@host1 wordpress]# setsebool -P httpd_can_sendmail on
[root@host1 wordpress]#
```

❷コマンドを入力

Apacheからのメール送信を許可する

14-3 CMSを使い始めるには

WordPressの初期設定

WordPressを使い始める前に、Webブラウザーで WordPressの管理画面にログインして、そこでMariaDBのデータベースのテーブル定義や初期値の反映を行います。もしもWebサーバーにグローバルIPが設定されていた場合、その時点から、インターネットへ情報発信ができます。このレッスンでは細かく説明しませんが、WordPressのテーマを変更したり、背景画像を変えるだけでも、自分だけのWordPressを作り上げることができます。もしも何かで壊れてしまっても自分だけのサーバーですので恐れることなく、いろいろと試してみてください。

1 ブログと管理者の情報を設定する

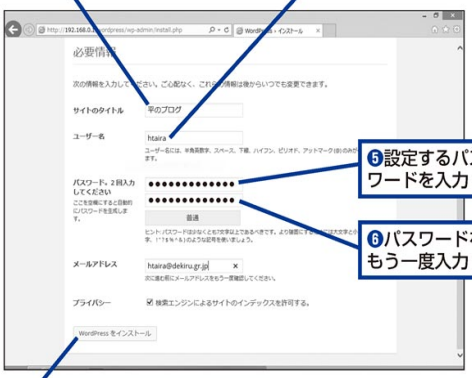
❶PCでWebブラウザを起動

❷以下のURLを入力

http://192.168.0.1/wordpress/

❸ブログ名を入力

❹管理者のユーザー名を入力



❺設定するパスワードを入力

❻パスワードをもう一度入力

❼[WordPressをインストール]をクリック

2 設定が成功した

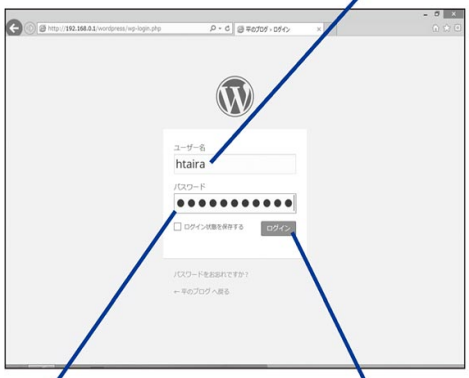
完了メッセージが表示された

[ログイン]をクリック

3 ログインする

手順1で設定した管理ユーザーとしてログインする

❶管理者のユーザー名を入力



❷パスワードを入力

❸[ログイン]をクリック

HINT!

パスワードを忘れた場合どうなるのか？

「パスワードをお忘れですか？」をクリックするとユーザー名かメールアドレスを質問されます。その後にメールで新しいパスワードの設定フォームのURLが通知されます。URLにアクセスして新しいパスワードを設定してください。この際にメールサーバーがきちんと機能していない場合には、メールが送れませんのでお気を付けください。

4 管理画面が表示された

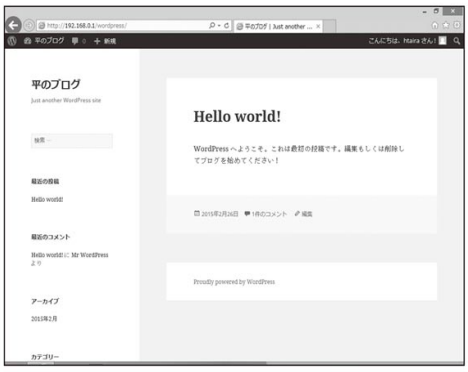
この画面からブログを設定する

5 ブログを表示する

再び以下のURLを入力

http://192.168.0.1/wordpress/

ブログが表示された



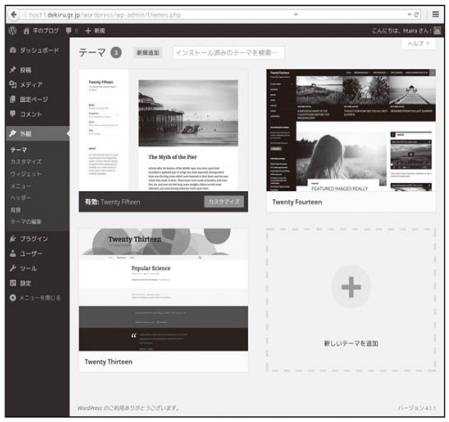
STEP UP

WordPressのテーマを変更するには

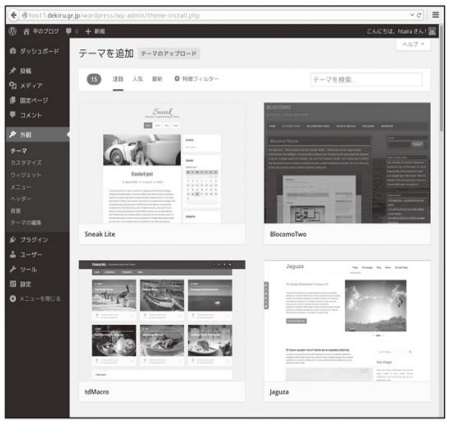
WordPressのデザインを変更する仕組みをテーマと呼びます。

WordPressの管理者にログインし、［メニュー］－［外観］－［テーマ］を選ぶと、テーマの一覧画面が表示されます。WordPressに同梱された3つのテーマが一覧で出ますが、「新規追加」をクリックするとテーマディレクトリから最新の人気テーマが取得できます。世界中の有志によって作られた洗練されたテーマから選んでデザインを変更することができます。運用中に変更してもWordPressのデータが初期化されることはありませんので、お気軽にお試しください。

テーマの一覧



テーマの追加



第15章 仮想マシンを動かす

この章で紹介する仮想化技術Linux KVMを活用することで、RHEL 7上で複数の仮想マシンを動かすことができます。仮想化技術やLinux KVMを完全に把握しようと思うと非常に奥深いものがありますが、この章では、Linux KVMを使い始めるために必要な知識を中心に説明します。

●この章の内容

15-1 Linux KVMを知ろう

Linux KVMの概要

Linux KVMは、Linuxカーネルに組み込まれた仮想化機能です。Linuxカーネル2.6.20からLinuxカーネルにマージされました。実体としては1MBにも満たないカーネルモジュールです。

スケジューラー、NUMA対応、メモリー集約、帯域幅制御など、Linuxがすでに持つさまざまな仕組みを流用できるため、ものすごい勢いで開発が行われてきました。RHELでは、RHEL 5.4に搭載されて以降、パブリッククラウドを中心に、さまざまな環境で利用されてきました。

このレッスンでは、Linux KVMの概要について説明します。

Linux KVMとは

RHEL 7には仮想化技術として、Linux KVM（Kernel-based Virtual Machine）が搭載されています。Linux KVMでは、Linuxカーネルが仮想化ソフトウェアとなり、複数の異なる種類のOSを仮想マシン上で動かすことを可能にします。つまり、RHEL 7がインストールされた1台のシステムにおいて、Linux KVMの仮想マシン上で、RHEL 5やRHEL6、Windows Server 2012といったゲストOSを動作させることができます。　なお、Linux KVMを利用するために、仮想化支援機能としてIntel VTもしくはAMD-Vを搭載したCPUが必要となります。そして仮想化支援機能がBIOSもしくはUEFIで有効化されている必要があります。

KVMの実体

Linux KVMは元々イスラエルのQumranet社（Red Hatが2008年に買収）が開発したオープンソースの仮想化ソフトウェアで、2006年10月に公開された後、同年12月にLinuxカーネルにマージされ、Linuxカーネル2.6.20から標準機能として利用できるようになりました。

Linux KVMはカーネルモジュールで実装されており、利用の有無によらずシステム上にインストールされています。

以下のパスに存在する、kvm.koという1MBにも満たないカーネルモジュールがLinux KVMの正体です。

```
/usr/lib/modules/3.10.0-229.el7.x86_64/kernel/arch/x86/kvm/kvm.ko
```

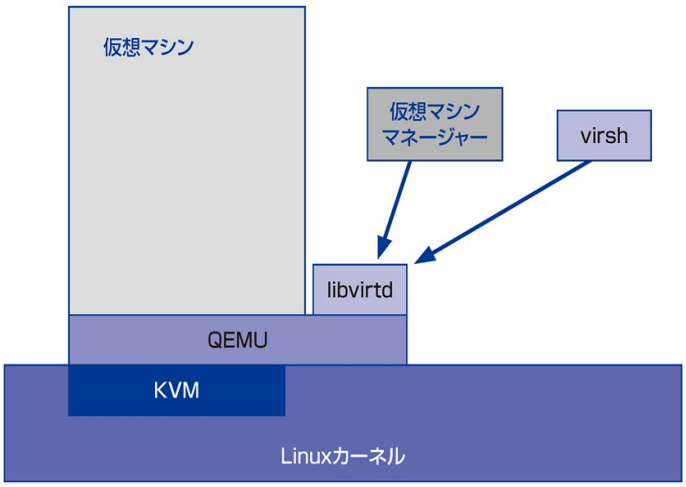
KVMのユーザーランド

KVMのカーネルモジュールと連携するユーザーランドのアプリケーションとして、QEMUというエミュレーターの存在があります。これはqemu-kvmパッケージとして提供されており、このパッケージを追加でインストールすることで、初めて仮想マシンをエミュレーションできます。

QEMUはKVMよりも昔からあるマシンエミュレーターで、x86、MIPS、PowerPC、ARMなど多くのCPU環境とI/Oデバイスをエミュレーションできます。QEMUがKVMと一緒に動くときは、CPUエミュレーション部分をKVM側に任せて、QEMUは主にメモリー管理とI/Oデバイスのエミュレーションに徹します。

仮想マシンの中のゲストOSで認識されている仮想ハードウェアは、QEMUがほぼすべてエミュレーションしています。なお、KVMの場合には物理マシン上のCPUと同じアーキテクチャーのCPUしか、仮想マシン上に提供することはできません。よって、x86_64アーキテクチャーにおいては、仮想マシンの中にもx86_64アーキテクチャーの仮想CPUが提供され、ppc64アーキテクチャーの場合には仮想マシンの中にもppc64アーキテクチャーの仮想CPUが提供されます。

また、RHEL 7において、直接的にユーザーランドとして見えるアプリケーションは、**レッスン15-3**で解説する仮想マシンマネージャーです。



15-2 仮想マシンを操作できるようにするには

管理ツールのインストール

RHEL 7に含まれる仮想マシンの管理ツールとしては、仮想マシンマネージャー（virt-manager）とvirshがあります。これらは両方ともlibvirtdというサービスに対してlibvirt APIにて指示を出しているクライアントであり、厳密にはlibvirtdが本当の意味で仮想マシンを管理していると言えます。どちらの管理ツールで指示を出したとしても、libvirtdが仲介してくれるので、不整合が起きることはありません。ただし、libvirtdではサーバー1台しか管理できません。このレッスンでは、仮想マシンマネージャーをインストールします。

1 libvirtdとvirt-managerをインストールする

❶コマンドを入力

```
[root@host1 ~]# yum install libvirt virt-manager
読み込んだプラグイン：langpacks, product-id, subscription-manager


================================================================================
 Package                          アーキテクチャー
                                        バージョン          リポジトリー          容量
================================================================================
インストール中：
 libvirt                          x86_64 1.2.8-16.el7_1.1 rhel-7-server-rpms  96 k
 virt-manager                     noarch 1.1.0-12.el7     rhel-7-server-rpms 631 k
依存性関連でのインストールをします：

インストール容量： 12 M
Is this ok [y/d/N]: y
Downloading packages:


完了しました！
[root@host1 ~]#
```

libvirtとvirt-managerのパッケージが表示される

❷「y」と入力してEnterキーを押す

インストールが完了した

HINT!

KVMを管理するそのほかのツール

別製品で提供しているRed Hat Enterprise Virtualization（RHEV）や、RHEL OpenStack PlatformなどのKVMの管理ツールもあります。これらは複数台の物理サーバーを管理できます。

2 libvirtdを開始する

KVMを管理するlibvirtdサービスを起動する

❶コマンドを入力

```
[root@host1 ~]# systemctl start libvirtd.service
[root@host1 ~]# systemctl enable libvirtd.service
[root@host1 ~]#
```

システム起動時に開始するようにする

❷コマンドを入力

3 libvirtdの動作を確認する

virshコマンドから確認する

コマンドを入力

```
[root@host1 ~]# virsh version
コンパイル時に使用したライブラリ： libvirt 1.2.8
使用中のライブラリ： libvirt 1.2.8
使用中の API: QEMU 1.2.8
実行中のハイパーバイザー： QEMU 1.5.3

[root@host1 ~]#
```

libvirtやQEMUのバージョンが表示された

正常にlibvirtdサービスが稼働している

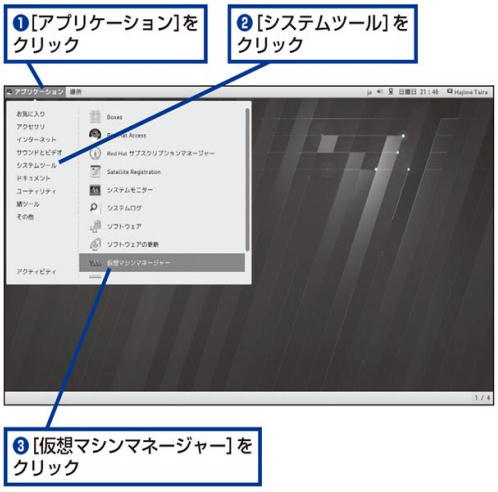
15-3 仮想マシンを作るには

仮想マシンマネージャー

仮想マシンマネージャー（virt-manager）は、GUIベースで直感的に利用できる仮想化管理インターフェイスを提供します。元々はXen Hypervisorの管理ツールとして開発されていましたが、現在ではKVMの管理ツールとして使われることが多くなってきました。ローカルだけではなく、リモートの仮想化ホストにSSH経由で接続し管理することもできます。

このレッスンでは、仮想マシンマネージャーによる仮想マシンの作り方や、OSのインストール、そして仮想マシンの起動／停止の方法について説明していきます。

1 仮想マシンマネージャーを起動する



2 パスワードを入力する

3 仮想マシンマネージャーが起動した

仮想マシンマネージャーのウィンドウが開いた

HINT!

仮想マシンの動作中に仮想マシンマネージャーを終了した場合

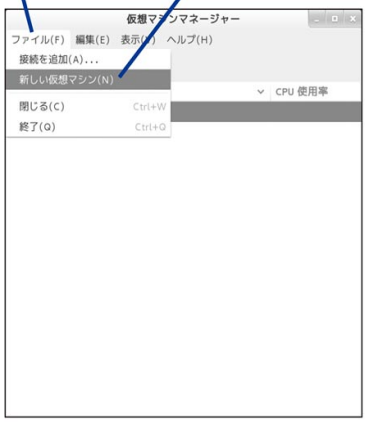
仮想マシンマネージャーで仮想マシンのインストール中に何か他の作業をしたい場合や、間違ってウィンドウを閉じて終了してしまった場合でも、心配無用です。仮想マシンマネージャーで一度指示を出した仮想マシンは、libvirtdサービスが管理しており、バックグラウンドで動いています。もう一度接続すれば続きから作業ができます。よって、安定稼働後はログアウトしても大丈夫です。

4 仮想マシンの作成を開始する

ここではローカルホストのlibvirtdへ接続する

❶[ファイル]をクリック

❷[新しい仮想マシン]をクリック



5 OSのインストール方法を選ぶ

新しい仮想マシンの作成ウィザードが表示された

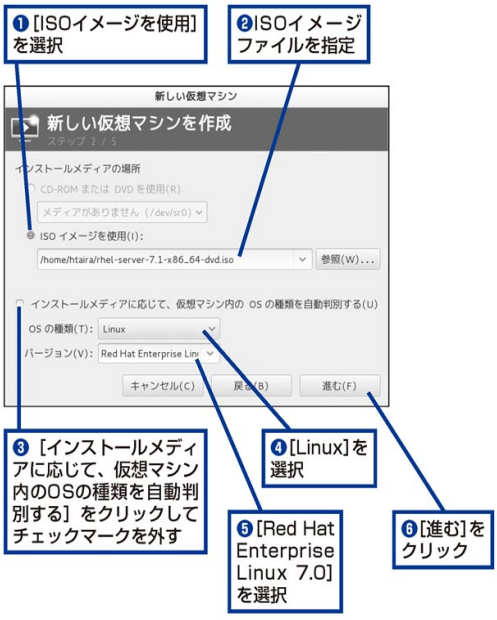
ここではRHEL 7.1をISOイメージファイルからインストールする

❶[ローカルのインストールメディア]を選択

❷[進む]をクリック

次のページに続く

6 OSを指定する

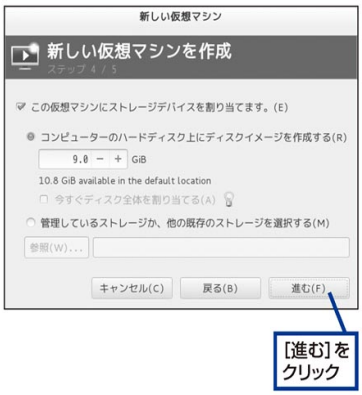


7 メモリー量とCPU数を指定する

特に必要がなければデフォルトのままでよい

8 ディスクを指定する

特に必要がなければデフォルトのままでよい

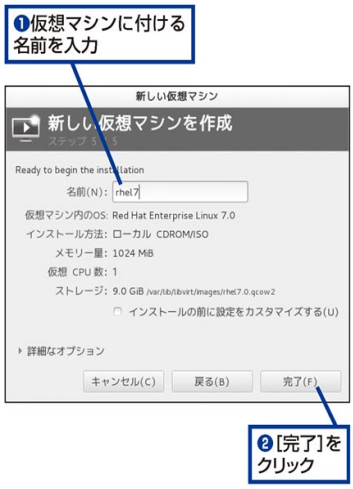


HINT!

OSの種類で何が変わるか

仮想マシンマネージャーでは仮想マシンを作成する際にOSの種類を選択する項目があります。最近ではインストールDVDやISOイメージの内容から判断してOSを自動判別することもできます。この項目を適切に選択することにより、仮想マシン内でエミュレーションするデバイスのデフォルトのタイプが変化します。たとえば、[Windows]を選んだ場合には、IDEディスクのエミュレーションがデフォルトになります。

9 仮想マシン名を指定する



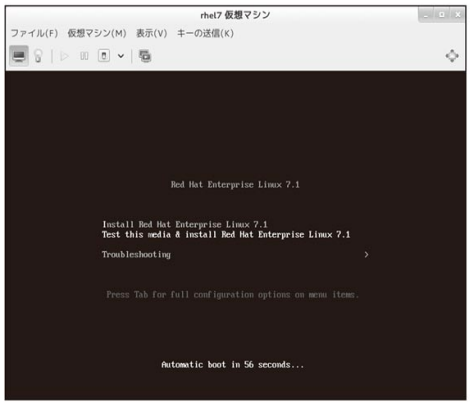
HINT! 仮想マシンの画面から抜ける 左Ctrl＋左Alt

仮想マシンマネージャーでゲストOSの画面操作をしているときに、マウスカーソルが画面から出られなくなることがあります。仮想マシンマネージャーでは、グラブキーという特殊なショートカットキーが定義されています。デフォルトは左Ctrl＋左Altとなっており、メニューの［編集］-［設定］-［コンソール］から変更できます。

10 仮想マシンが起動した

RHEL 7.1のインストーラーが起動した

以降、レッスン2-4を参考にインストールする



11 仮想マシンを終了した

仮想マシンをシャットダウンすると仮想マシンマネージャーに戻る

15-4 コマンドラインから仮想マシンを操作するには

virshの使い方

レッスン15-2でlibvirtdの動作確認コマンドとして利用したvirshコマンドは、libvirtdを操作するコマンドラインインターフェイスです。リモートのKVM環境にSSHでログインして仮想マシン制御を行うことができます。virshの操作に慣れると日常的な運用管理のほとんどはできてしまいます。仮想マシンマネージャーで設定項目がない細かいパラメーターもvirshを使うことで指定できます。

このレッスンでは、知っておくと便利なvirshコマンドについて、よく使うものをご紹介しましょう。

ヘルプを表示する

virsh helpを実行すると、virshコマンドの使い方を表示してくれます。

コマンドを入力

```
[root@host1 ~]# virsh help
グループ別コマンド：

 Domain Management （ヘルプのキーワード 'domain'）:
    attach-device                  XML ファイルによるデバイスの接続
    attach-disk                    ディスクデバイスの接続
    attach-interface               ネットワークインターフェースの接続
    autostart                      ドメインの自動起動

    echo                           引数のエコー
    exit                           対話式ターミナルの終了
    help                           ヘルプの表示
    pwd                            カレントディレクトリーの表示
    quit                           対話式ターミナルの終了

[root@host1 ~]#
```

HINT!

virshで仮想マシンを作成

virshにXML形式の仮想マシンの定義ファイルを読み込ませて仮想マシンを作成することもできます。

サブコマンドのヘルプを表示する

virsh helpの後にvirshのサブコマンドを指定すると、その詳細なオプションを表示します。

「virsh list」のヘルプを表示する

コマンドを入力

```
[root@host1 ~]# virsh help list
  名前
    list - ドメインの一覧表示

    --table            表形式の一覧表示 （初期値）
    --managed-save     管理済み保存状態を持つ停止状態のドメインに印をつけます
    --title            show domain title


[root@host1 ~]#
```

仮想マシンの一覧を表示する

「virsh list」を実行すると、現在稼働中の仮想マシンの一覧を表示します。また、「virsh list --all」と「--all」を付けると、稼働していないものも含めたすべての仮想マシンの一覧を表示します。

起動していないとき

コマンドを入力

```
[root@host1 ~]# virsh list --all
 Id    名前                           状態
----------------------------------------------------
 -     rhel7                          シャットオフ

[root@host1 ~]#
```

起動しているとき

コマンドを入力

```
[root@host1 ~]# virsh list
 Id    名前                           状態
----------------------------------------------------
 5     rhel7                          実行中

[root@host1 ~]#
```

次のページに続く

仮想マシンの情報を表示する

「virsh dominfo <仮想マシン名>」を実行すると、対象の仮想マシンの情報を表示します。

コマンドを入力

```
[root@host1 ~]# virsh dominfo rhel7
Id:             -
名前:           rhel7
UUID:           b79fc2bc-90d4-4f47-964a-848d2233975f
OS タイプ:    hvm

管理済み保存： いいえ (no)
セキュリティモデル： selinux
セキュリティ DOI: 0

[root@host1 ~]#
```

仮想マシンを起動する

「virsh start <仮想マシン名>」を実行すると、登録されている仮想マシンを起動します。また、「virsh start <仮想マシン名> --console」と「--console」を付けると、起動時に仮想マシンのテキストコンソールに接続します。

コマンドを入力

```
[root@host1 ~]# virsh start rhel7
ドメイン rhel7 が起動されました

[root@host1 ~]#
```

仮想マシンを再起動する

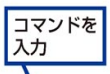
「virsh reboot <仮想マシン名>」を実行すると、稼働中の仮想マシンを再起動します。

コマンドを入力

```
[root@host1 ~]# virsh reboot rhel7
ドメイン rhel7 を再起動しています

[root@host1 ~]#
```

自動起動を設定する

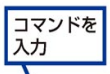
「virsh autostart <仮想マシン名>」を実行すると、仮想マシンをシステム起動時に自動起動させるよう設定します。また、「virsh autostart <仮想マシン名> --disable」と「--disable」を指定すると、自動起動しないよう設定します。




```
[root@host1 ~]# virsh autostart rhel7
ドメイン rhel7 が自動起動に設定されました

[root@host1 ~]#
```

仮想マシンを停止する

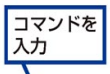
「virsh shutdown <仮想マシン名>」を実行すると、稼働中の仮想マシンを穏やかに停止します。




```
[root@host1 ~]# virsh shutdown rhel7
ドメイン rhel7 はシャットダウン中です

[root@host1 ~]#
```

仮想マシンを強制停止する

「virsh destroy <仮想マシン名>」を実行すると、稼働中の仮想マシンを強制停止します。




```
[root@host1 ~]# virsh destroy rhel7
ドメイン rhel7 は強制停止されました

[root@host1 ~]#
```

STEP UP

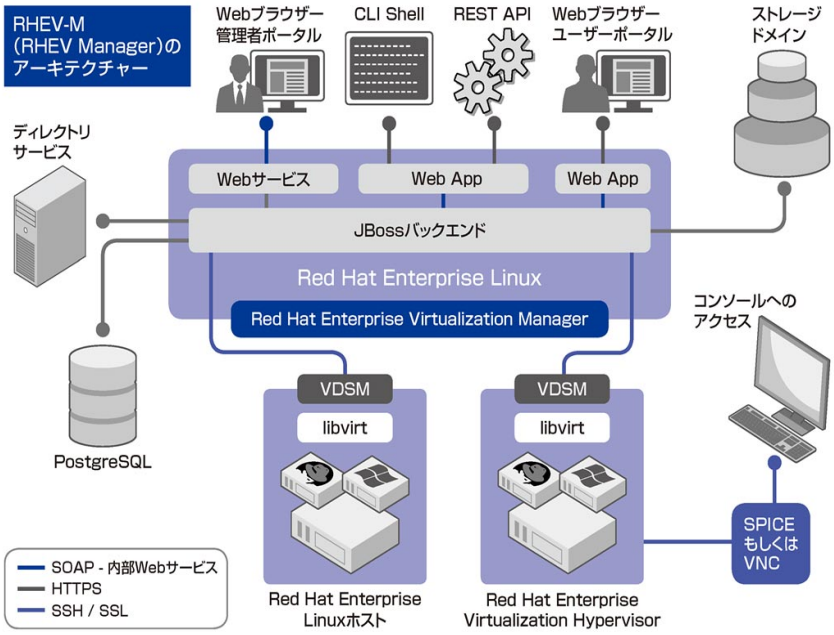
複数ホストの管理を行うためには

仮想マシンマネージャーでも、接続先に複数登録すれば、まったく管理できないわけでもありません。しかし、仮想化環境が大きくなってくると、共有ストレージリソースの割り当てルールや、仮想マシンのテンプレート、MACアドレスの管理などが大変になります。

そこで統合仮想化管理ソフトウェアの「Red Hat Enterprise Virtualization（RHEV）」が別製品で用意されています。この管理ソフトウェアを使うことで、数百台レベルの仮想化ホストと数千台レベルの仮想マシンまで効率よく管理することができます。

また、仮想マシンマネージャーを利用するためには、ローカルの環境にもRHEL 7が必要ですが、RHEVではWebブラウザーから管理することができます。管理者が指定したユーザーに権限分与することもできるため、複数人の管理者ユーザーを付与することもできます。

まず仮想マシンマネージャーでLinux KVMに慣れ、仮想マシンの管理が大変になったらRHEVに切り替えることで、管理者の運用負担を軽減できます。



第16章 コンテナーを使う

この章で紹介するコンテナー技術を活用することで、RHEL 7上で複数のアプリケーションを効率的に動かすことができます。第15章で紹介した仮想化のKVMと同じような目的で利用できますが、Dockerというコンテナー管理の仕組みを利用して、仮想化を行うことなく相互に隔離された空間で動かすことができます。この章では、Linuxにおけるコンテナーの仕組みとDockerによるコンテナーの管理方法を中心に説明します。

●この章の内容

16-1 Dockerを知ろう

Dockerの概要

Dockerは、Docker社が開発を行っているコンテナー管理の仕組みです。Dockerはコンテナーイメージを定義し、その管理を中心とし、コンテナーの実行自体はLinuxのNamespaceの仕組みやcgroupsを使って実現しています。また、Dockerは開発者が動かしていた環境をそのままDockerイメージで本番環境に展開することができます。よって、Dockerを利用することで開発者と運用担当者がDevOpsのアプローチで共同作業できるようになります。このレッスンでは、まず、Dockerについて理解を深めるために、その特徴と利用形態についてご紹介します。

Dockerとは

RHEL 7から、コンテナー管理ツールとしてDockerが搭載されました。

Dockerはコンテナー管理のための管理フレームワークです。Linuxが標準で提供しているコンテナー技術であるLinuxコンテナー（lxc）や、RHEL 7で採用されているlibcontainerと呼ばれるコンテナー管理の仕組みを使い、アプリケーション実行環境を軽量かつ効率的に整えることを目的に開発されています。

コンテナー技術を利用することにより、1つのサーバー上で複数のアプリケーションを稼働させることができます。それぞれのアプリケーションはコンテナーの中で稼働します。その点では、本書で紹介している仮想化技術Linux KVMに似た側面もあります。

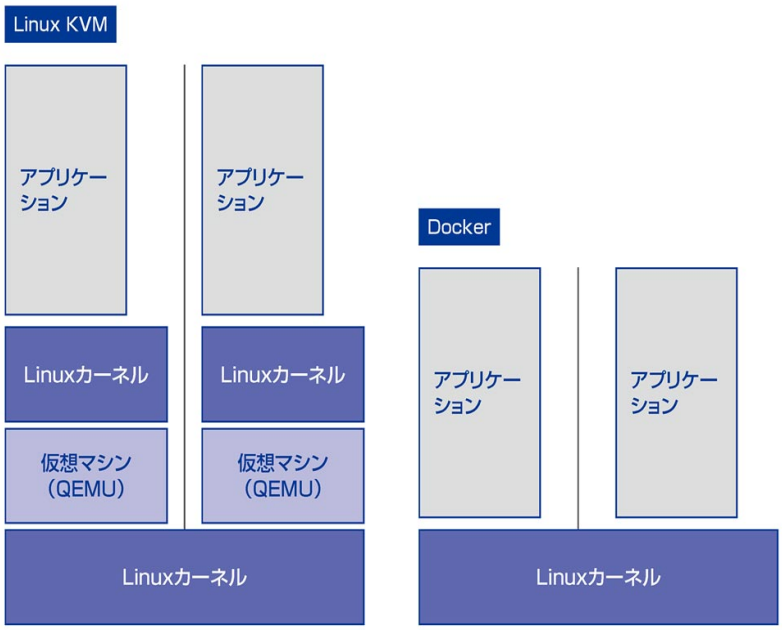
コンテナー技術は仮想化技術よりもリソースの共有度合いが大きいため、仮想化技術よりもオーバーヘッドも少なく、また、各種リソースの集約度を高めることができます。

Dockerの特徴

Dockerはコンテナーイメージの管理を中心として、コンテナー内のアプリケーションの稼働状況の管理や、稼働中のコンテナーが利用できるリソースの制限を行います。

仮想化技術と似ていますが、対象を隔離する際に仮想マシンを作るか否かが異なる点です。Dockerでは、すべてのコンテナーは1つのLinuxカーネルを共用しており、そのカーネルに実装されているプロセス、ディスク、ネットワークなどのNamespaceの仕組みをうまく使うことによって、各コンテナー間で相互にリソースを干渉しないように振る舞います。仮想化技術の利用目的の1つであるアプリケーションのカプセル化という観点では共通します。

また、Dockerは既存のコンテナーイメージから派生してコンテナーイメージを作ることができます。派生したコンテナーイメージはDockerが差分イメージとして管理するため、この仕組みによりディスク容量の節約にもつながります。



Dockerの利用形態

コンテナー技術は仮想化技術と似ていますが、仮想化技術の代替ではありません。アプリケーションやライブラリをコンテナーが持つディスクイメージの中に格納して、アプリケーションにポータビリティをもたらします。コンテナーを稼働する環境が、物理サーバーであろうと、仮想サーバーであろうと、パブリッククラウドであろうと、Dockerが管理する環境であれば、コンテナー作成者が期待するとおりにデプロイされ格納されているアプリケーションが期待どおりに動きます。

つまり、コンテナー技術は仮想化技術よりも上位層でアプリケーション開発者が期待するアプリケーション実行環境を提供し、Dockerは、コンテナー上で動かすアプリケーションの配信基盤を提供します。

また、Dockerで生成したコンテナーは、1コンテナー＝1プロセスで構成することを原則としています。そのため、コンテナー内で起動するプロセスの指定は、シンプルに起動コマンドを指定しましょう。複数プロセスを起動したいのであれば、運用管理の観点から見ると、OS一式が動作する仮想化技術の方が適切な選択肢と言えます。

16-2 Dockerを使えるようにするには

Dockerのインストール

Dockerのインストール先には、コンテナーの中のアプリケーションが消費するメモリーと10GB程度のディスクの空き領域があれば、ひとまず、Dockerを動かすことができます。マシンは物理マシンであっても仮想マシンであってもかまいません。

Dockerをインストールするには、Extrasチャンネルを有効化させて、そのチャンネルからdockerパッケージをインストールします。そして、インストール後にdockerサービスを起動するだけで、セットアップが完了します。サービス起動時にディスク領域が確保されます。

1 Dockerをインストールする

RHEL 7のExtrasチャンネルを有効にする

❶コマンドを入力

```
[root@host1 ~]# subscription-manager repos --enable=rhel-7-server-extras-rpms
リポジトリー 'rhel-7-server-extras-rpms' はこのシステムに対して有効になっています。
[root@host1 ~]#
```

❷コマンドを入力

```
[root@host1 ~]# yum install docker
読み込んだプラグイン：langpacks, product-id, subscription-manager

インストール中：
 docker      x86_64      1.4.1-37.el7      rhel-7-server-extras-rpms      7.1 M

インストール容量： 31 M
Is this ok [y/d/N]: y
Downloading packages:

完了しました！
[root@host1 ~]#
```

Dockerのパッケージが表示される

❸「y」と入力してEnterキーを押す

インストールが完了した

2 Dockerサービスを開始する

❶コマンドを入力

```
[root@host1 ~]# systemctl start docker.service
[root@host1 ~]# systemctl enable docker.service
ln -s '/usr/lib/systemd/system/docker.service' '/etc/systemd/system/multi-user.target.
wants/docker.service'
[root@host1 ~]#
```

❷コマンドを入力

システム起動時に開始するようにする

メモ 初回サービス開始時には、コンテナーのデータ領域を確保するため、起動に少し時間がかかります。

3 Dockerサービスの状態を確認する

コマンドを入力

```
[root@host1 ~]# docker info
Containers: 0
Images: 0
Storage Driver: devicemapper
 Pool Name: docker-253:1-68947407-pool
 Pool Blocksize: 65.54 kB
 Backing Filesystem: <unknown>
 Data file: /dev/loop0
 Metadata file: /dev/loop1
 Data Space Used: 307.2 MB
 Data Space Total: 107.4 GB
 Metadata Space Used: 729.1 kB
 Metadata Space Total: 2.147 GB
 Udev Sync Supported: true
 Data loop file: /var/lib/docker/devicemapper/devicemapper/data
 Metadata loop file: /var/lib/docker/devicemapper/devicemapper/metadata
 Library Version: 1.02.93-RHEL7 (2015-01-28)
Execution Driver: native-0.2
Kernel Version: 3.10.0-229.el7.x86_64
Operating System: Red Hat Enterprise Linux
CPUs: 1
Total Memory: 1.791 GiB
Name: host1.dekiru.gr.jp
ID: C3KE:UIDY:2JNW:FUGG:EIBT:TPS3:RREG:YM3L:MPXQ:AXVO:YFL6:NYQH
[root@host1 ~]#
```

Dockerサービスの状態が表示された

16-3 コンテナーを動かすには

Dockerの使い方

Dockerを使うには、前のレッスンでセットアップしたdockerサービスに対して、そのフロントエンドであるdockerコマンドを使い、コンテナーを操作します。それにより、コンテナーイメージの検索、コンテナーの起動／停止／削除などの操作を行うことができます。Dockerコンテナーを生成する前には、元となるコンテナーイメージが必要となります。

このレッスンでは、コンテナーイメージを入手し、Dockerコンテナーを生成した後にコンテナーの仕組みを理解するためにインタラクティブモードでコンテナーを操作します。

1 イメージを検索する

名前が「centos」にマッチするイメージを検索する　コマンドを入力　イメージの検索

```
[root@host1 ~]# docker search centos
NAME                              DESCRIPTION                                     STARS   OFFICIAL  AUTOMATED
centos                            The official build of CentOS.                   884     [OK]
ansible/centos7-ansible           Ansible on Centos7                              35                [OK]

solict/provisionous-puppet-centos CentOS provisions with Puppet included          0                 [OK]
insaneworks/centos                CentOS 6.5 x86_64 + @update                     0                 [OK]
[root@host1 ~]#
```

HINT!

Dockerイメージの入手元

Dockerイメージを格納する場所をDockerレジストリと言います。Docker公式レジストリ（registry.hub.docker.com）から、公式イメージをはじめ、他のユーザーが作成して登録したDockerイメージをダウンロードできます。また、RHELのDockerでは、Red Hat社が提供するDockerリポジトリ（registry.access.redhat.com）も最初から利用できるようになっています。

2 イメージを入手する

イメージをダウンロードしてみる

コマンドを入力

イメージのダウンロード

```
[root@host1 ~]# docker pull centos
Pulling repository registry.access.redhat.com/centos
511136ea3c5a: Pull complete
5b12ef8fd570: Pull complete
88f9454e60dd: Pull complete
centos:latest: The image you are pulling has been verified. Important: image
verification is a tech preview feature and should not be relied on to provide
security.
Status: Downloaded newer image for centos:latest
[root@host1 ~]#
```

イメージがダウンロードされた

3 イメージの一覧を表示する

Dockerリポジトリから入手したイメージの一覧を表示する

コマンドを入力

イメージの一覧の表示

```
[root@host1 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
centos              latest              88f9454e60dd        2 weeks ago         210 MB
[root@host1 ~]#
```

一覧が表示された

4 Dockerコンテナーを生成する

入手したイメージからコンテナーを生成する

コマンドを入力

コンテナーを生成して実行

イメージ

コンテナーで実行するコマンド

```
[root@host1 ~]# docker run -it centos:latest /bin/bash
[root@ab47bcafe63b /]#
```

bashの新しいプロンプトが表示された

次のページに続く

5 コンテナーの中にいることを確認する

/etc/centos-releaseを表示してみる

コマンドを入力

```
[root@ab47bcafe63b /]# cat /etc/centos-release
CentOS Linux release 7.0.1406 (Core)
[root@ab47bcafe63b /]#
```

CentOSの環境が確認された

6 カーネルを確認する

カーネルのバージョンを表示してみる

コマンドを入力

```
[root@ab47bcafe63b /]# uname -r
3.10.0-229.el7.x86_64
[root@ab47bcafe63b /]#
```

RHELのカーネルが確認された

HINT!

インタラクティブモードでの実行

手順❹では、/bin/bashを起動するコンテナーを実行し、コンテナー内部のコマンドラインに入出力を接続しています。-iオプションでインタラクティブモードを指定し、-tオプションで仮想端末を割り当てます。

HINT!

コンテナー環境とカーネル

手順❺で/etc/centos-releaseを表示すると、CentOSが動いている気分になります。しかし、コンテナーではコンテナー実行環境とカーネルを共有しているので、手順❻でunameコマンドを実行すると、RHEL7のカーネルバージョンが表示されます。つまりOSの実体としては1つなのです。

HINT!

コンテナーから見えるプロセス

手順❼のようにコンテナーの中でpsコマンドを実行すると、面白い結果が返ってきます。/bin/bashがプロセスID:1で動いているのです。ベースのRHEL 7上でプロセスID:1にて動いているはずのsystemd（/usr/lib/systemd/systemd）のプロセスも表示されません。コンテナー実行環境で動いているプロセスや、他のコンテナー上で動いているプロセスは隠蔽されます。

7 プロセス一覧を表示する

コマンドを入力

```
[root@ab47bcafe63b /]# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:00 /bin/bash
   20 ?        R+     0:00 ps ax
[root@ab47bcafe63b /]#
```

2つのプロセスだけが表示された

8 コンテナーから抜ける

コマンドを入力

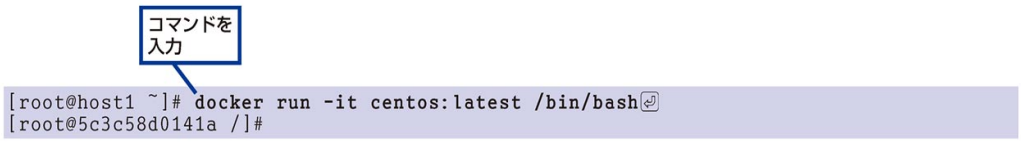
```
[root@ab47bcafe63b /]# exit
exit
[root@host1 ~]#
```

元のプロンプトに戻った

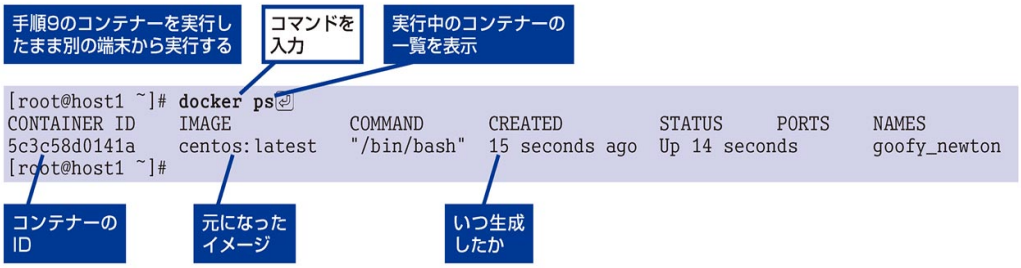
コンテナーの実行は終了した

次のページに続く

9 新しいコンテナーを生成する



10 コンテナーの稼働状況を確認する

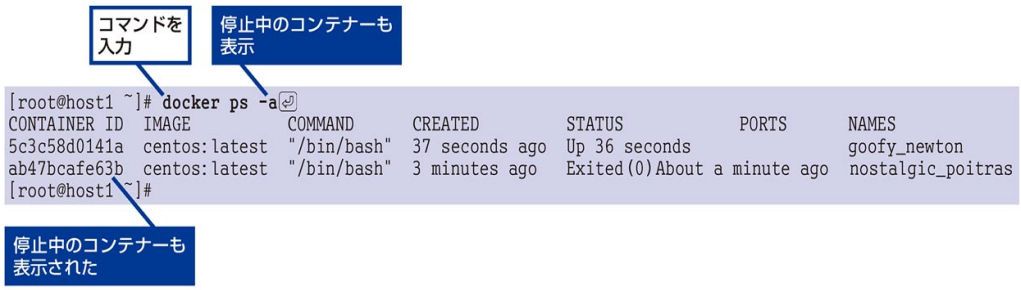


HINT!

終了したコンテナー

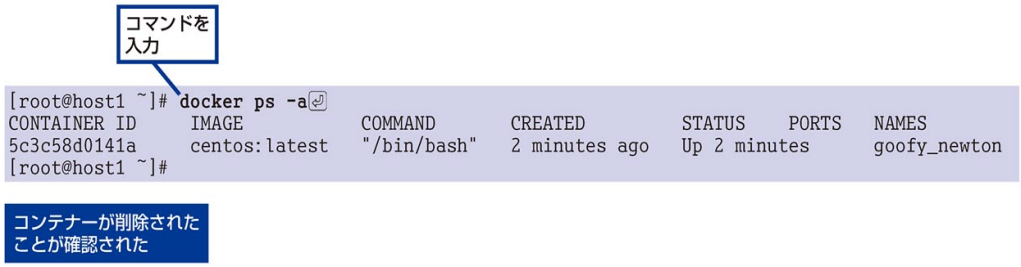
プロセスが終了したDockerコンテナーや、明示的にstopさせたDockerコンテナーは、停止状態になっていますが、ディスクのデータは残っています。そこで手順⓬のように、コンテナーIDを指定してdocker rmコマンドを実行することで、削除できます。

11 終了したコンテナーも表示する



12 コンテナーを削除する

13 再度コンテナーを表示する



16-4 コンテナーでサーバーソフトを実行するには

バックグラウンドでの実行

前のレッスンでは、Dockerコンテナーの仕組みを把握するために対話型インターフェイスを持つプログラムをフォアグラウンドで起動しました。しかし、これではrootユーザーがコンソールでログインしている間しかDocker内でプログラムを動かせません。

サーバーアプリケーションの大半はバックグラウンドプロセスとして動きます。このレッスンでは、RHEL 6の公式コンテナーイメージを利用し、Dockerコンテナー内で任意のプログラムをバックグラウンドプロセスとして動かす方法を説明していきます。

1 RHEL 6のイメージを入手する

イメージをダウンロードする

コマンドを入力

```
[root@host1 ~]# docker pull rhel6
Pulling repository registry.access.redhat.com/rhel6
f5f0b338bbd6: Pulling image (latest) from registry.access.redhat.com/rhel6, endpf5f0b338bbd6:
Download complete
Status: Downloaded newer image for registry.access.redhat.com/rhel6:latest
[root@host1 ~]#
```

イメージがダウンロードされた

2 イメージの一覧を表示する

コマンドを入力

```
[root@host1 ~]# docker images
REPOSITORY                          TAG           IMAGE ID        CREATED         VIRTUAL SIZE
centos                              latest        88f9454e60dd    3 weeks ago     210 MB
registry.access.redhat.com/rhel6    6.6-6         f5f0b338bbd6    7 weeks ago     155.6 MB
registry.access.redhat.com/rhel6    latest        f5f0b338bbd6    7 weeks ago     155.6 MB
[root@host1 ~]#
```

RHEL 6のイメージが表示された

HINT!

RHELの公式イメージについて

現在のところ、Red HatからRHEL 6とRHEL 7の公式イメージが提供されています。これらはカスタマーポータルのダウンロードページ、またはRed Hatが提供するDockerリポジトリからコンテナーイメージ形式で入手することができます。コンテナー内でのホスト登録作業は不要です。そのままyumコマンドを実行するとコンテナーホストのサブスクリプションを利用して同じチャンネルをすぐに利用することができます。

3 コンテナーを生成する

コンテナーを生成してApacheをインストールする

コマンドを入力

コンテナーに名前を付ける

イメージ

```
[root@host1 ~]# docker run --name=httpd_temp registry.access.redhat.com/rhel6:latest /bin/
bash -c "yum install httpd -y;"
Loaded plugins: product-id, subscription-manager
```

Apacheをインストール

```
Complete!
[root@host1 ~]#
```

コンテナーの中でyumコマンドが実行されたあと、元のコンテナー実行環境に戻った

4 イメージとして保存する

Apacheをインストールしたコンテナーをイメージとして保存する

コマンドを入力

イメージを生成

元のコンテナー

作成するイメージ

```
[root@host1 ~]# docker commit httpd_temp rhel6_httpd_base
555cd9946accb88c12222098ea366042bacb499401e73e66ec82aeb248446502
[root@host1 ~]#
```

次のページに続く

5 Apacheをコンテナーで実行する

rhel6_httpd_baseのイメージから、さらにコンテナーを生成する

コマンドを入力 / バックグラウンドで実行 / コンテナー内の80番ポートを、外に8080番ポートとして見せる / コンテナー名 / イメージ

```
[root@host1 ~]# docker run -d -p 8080:80 --name=webcontainer01 rhel6_httpd_base /usr/
sbin/httpd -D FOREGROUND
41dea674440158cf0b8f039fb30a510bbe105d752984c1ab77121f7c854e18e6
[root@host1 ~]#
```

Apacheをコンテナー内でフォアグラウンドで起動

6 ポートを確認する

8080番ポートをリッスンしていることを確認する / コマンドを入力

```
[root@host1 ~]# ss -lnt
State       Recv-Q Send-Q        Local Address:Port          Peer Address:Port
LISTEN      0      128                       *:22                       *:*
LISTEN      0      128           127.0.0.1:631                      *:*
LISTEN      0      100           127.0.0.1:25                       *:*
LISTEN      0      128                  :::8080                    :::*
LISTEN      0      128                  :::22                      :::*
LISTEN      0      128                 ::1:631                     :::*
LISTEN      0      100                 ::1:25                      :::*
[root@host1 ~]#
```

8080番ポートをリッスンしている

HINT!

イメージやコンテナーのID

手順❹でdocker commitによりイメージを作ったときに、長い16進数の値が表示されました。これがイメージの固有のIDで、ほかのイメージと重複しないようになっています。IDを直接指定するのは大変なので、イメージにはリポジトリ名（「registry.access.redhat.com」など）、イメージ名（「rhel6」など）、タグ（「latest」など）が付いています。コンテナーも同じく、手順❺で表示されたように、固有のIDを持っています。

7 Apacheにアクセスしてみる

8080番ポートのApacheにアクセスしてみる

コマンドを入力

```
[root@host1 ~]# curl http://localhost:8080/
<!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">

        </body>
</html>
[root@host1 ~]#
```

Apacheのテストページが表示された

8 コンテナーを停止する

コマンドを入力　コンテナーの停止　コンテナー名

```
[root@host1 ~]# docker stop webcontainer01
webcontainer01
[root@host1 ~]#
```

9 コンテナーを削除する

コマンドを入力

```
[root@host1 ~]# docker rm webcontainer01
webcontainer01
[root@host1 ~]#
```

STEP UP

Dockerfile

Dockerfileと呼ばれる定義ファイルを記述することで、目的のサービスを起動するコンテナーイメージを自動的に生成させることができます。

Dockerfileの中にはベースのコンテナーイメージ、収録したいパッケージ、イメージ生成時に実行するコマンド、コンテナーの起動時に実行するコマンドを列挙します。

Dockerfile

```
FROM rhel7
MAINTAINER Hajime Taira
RUN yum update -y
RUN yum install -y httpd
ADD ./index.html /var/www/html/
EXPOSE 80
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
```

index.html

```
<html>
<head>
<title>from container</title>
</head>
<body>Hello Docker!</body>
</html>
```

Dockerfileが存在するディレクトリで、docker buildコマンドを実行することで、Dockerイメージがビルドされます。


```
# docker build -t rhel7_httpd:dev .
```

ビルドされたDockerイメージからコンテナーを生成するには次のように実行します。


```
# docker run -d -p 8080:80 --name webcontainer02 rhel7_httpd:dev
```

第17章 RHEL 7をメンテナンスする

実際にRHEL 7を動かしていくと、メンテナンスが必要となります。この章では、RHEL 7のメンテナンスとして、まずハードディスクの増設方法について、パーティションにファイルシステムを作成する場合と、LVMで拡張する場合を解説します。続いて、新しいブートローダーであるGRUB2の操作や設定方法を説明し、最後にバックアップとリストアについて説明します。

●この章の内容

17-1 ハードディスクを増設するには

parted、mkfs.xfs、LVM

ディスクの容量を増やすために、ハードディスクを追加します。新しく用意した追加のハードディスクを利用するには、2つの方法があります。ひとつは、ハードディスクをコンピューターに接続した後に、パーティションを定義し、パーティションにファイルシステムを作成した上で、フォーマットする方法です。もうひとつは、LVMを使って、既存のファイルシステムを拡張して容量を増やす方法です。LVMを使ったファイルシステムの拡張は、オンラインで行えるため、運用中のサーバーでもサービスを止めずにメンテナンスが行えます。

パーティションの形式

ハードディスク上にはファイルシステムを作る区画を割り当てる前に、事前にパーティション定義を行う必要があります。パーティション定義はハードディスク上のパーティションテーブルに格納されます。

現在のシステムで利用されているパーティションテーブルの形式には、従来からのMBR（Master Boot Record）形式と、GPT（GUID Partition Table）形式の2種類があります。

MBR形式には基本パーティションと拡張パーティションという概念があり、基本パーティションは4つまでしか定義できません。基本パーティションの1つを消費して拡張パーティション領域を作ることができます。拡張パーティション領域を作ることで、基本パーティションと拡張パーティションを合わせて上限63個までパーティション定義することができます。

GPT形式には基本パーティションと拡張パーティションという概念はなく、上限128個までパーティション定義することができます。

2TBを超えるハードディスクでは、GPT形式のパーティションテーブルが必須となっています。最近のシステムではGPT形式のパーティションからでも起動できるようになっていますので、特段理由がない限り、GPT形式を選択した方がよいでしょう。

ディスクのデバイスファイル

Linux上で認識されているハードディスクには、それぞれ対応するブロックデバイスファイルが存在します。SATAやSAS、Fibre Channelのディスクの場合、/dev/sdaから始まるブロックデバイス名が振られます。また、KVM上の仮想マシンで動いているマシンの場合でVirtIO Blockのハードディスクの場合、/dev/vdaから始まるブロックデバイスファイルが振られます。これらのブロックデバイスファイルはハードディスク全体を操作したい場合に指定するものです。

一方、ブロックデバイスファイルの名前の後ろに数字がついているデバイスファイルもあります。これらは、そのハードディスクのパーティション番号を指します。よって、ファイルシステムを作成するときには、後ろに数字が付いているブロックデバイスファイルを指定します。

この数字はMBR形式の場合、基本パーティションは1から4、拡張パーティションは5から始まる数字が振られます。

パーティションにファイルシステムを作成する場合

パーティションの定義

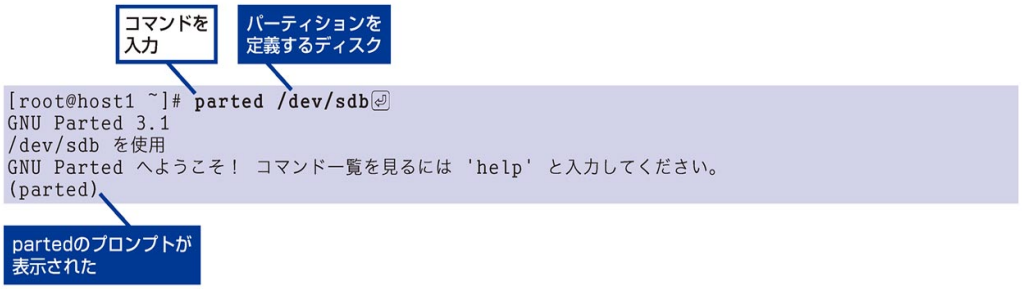
パーティション定義は、RHEL 7をインストールする際に一緒に行うこともできます。新しく用意した追加のハードディスクの場合は、RHELに用意されているストレージ管理ツールを使って定義する必要があります。

パーティション操作を行うツールとしては、fdiskとpartedコマンドがあります。fdiskは従来から利用されている管理コマンドで、主にMBR形式のパーティションを操作するために利用します。partedは管理コマンドで、MBR形式とGPT形式の両方のパーティション操作を行うことができます。なお、partedは、RHEL 6のときにも存在していました。

ここでは、partedを使ってパーティションを定義する方法を紹介します。

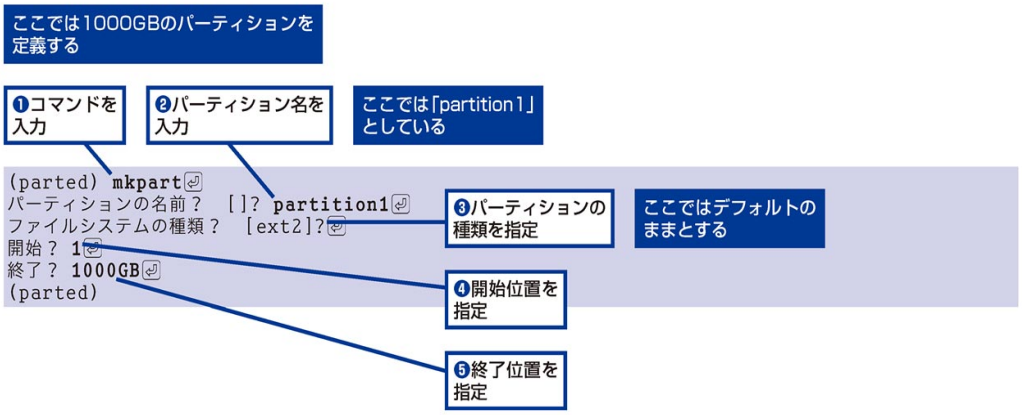
次のページに続く

1 partedを起動する



2 GPT形式を選ぶ

3 パーティションを作る



HINT!

RHEL 7のfdiskとparted

RHEL 7に収録されているfdiskコマンドではGPT形式のパーティションも操作できます。ただし、fdisk起動時に右のような警告メッセージが出ます。そのため、GPT形式の場合はpartedコマンドかgdiskコマンドをお使いください。

```
WARNING: fdisk GPT support is currently
new, and therefore in an experimental
phase. Use at your own discretion.
```

4 パーティションテーブルを表示する

コマンドを入力

```
(parted) print
モデル：VMware, VMware Virtual S (scsi)
ディスク /dev/sdb: 1074GB
セクタサイズ（論理/物理）: 512B/512B
パーティションテーブル：gpt
ディスクフラグ：

番号    開始      終了      サイズ    ファイルシステム      名前          フラグ
 1      1049kB    1000GB    1000GB                          partition1

(parted)
```

パーティションが作られている

5 partedを終了する

コマンドを入力

```
(parted) quit
通知：必要であれば /etc/fstab を更新するのを忘れないようにしてください。

[root@host1 ~]#
```

Linuxのプロンプトに戻った

次のページに続く

6 ファイルシステムを作成する

作成した/dev/sdb1にファイルシステムを作成する

RHEL 7のデフォルトであるXFSのファイルシステムを作る

コマンドを入力

```
[root@host1 ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1              isize=256    agcount=4, agsize=61035072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=244140288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=119209, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@host1 ~]#
```

7 マウントポイントを作る

新しいマウントポイント/mnt/newhomeを作る

コマンドを入力

```
[root@host1 ~]# mkdir /mnt/newhome
[root@host1 ~]#
```

8 増設したディスクをマウントする

コマンドを入力

```
[root@host1 ~]# mount /dev/sdb1 /mnt/newhome
[root@host1 ~]#
```

HINT!

起動時にマウントさせるには

次の起動時に、最初からマウントした状態で利用するには、/etc/fstabに次のような記述を追記してください。

```
/dev/sdb1    /mnt/newhome    xfs     defaults        1 2
```

LVMでファイルシステムを拡張する場合

LVMによるディスク管理

RHEL 7では、従来型のパーティションよりも賢いストレージ管理の仕組みとして、LVM（Logical Volume Manager）を利用できます。インストール時にインストール先ディスクのデフォルトの設定を選んでいると、ルートファイルシステムもLVMを利用して作られます。

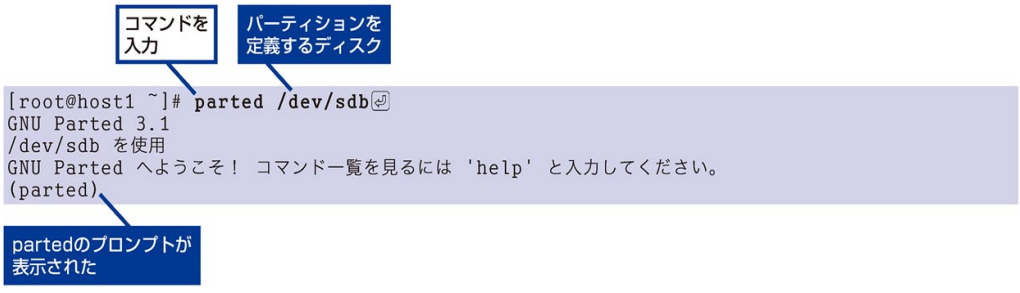
LVMでは、ひとつのハードディスクを複数のボリュームとして割り当てたり、複数のハードディスクを各ハードディスクよりも大きな容量のボリュームとしてまとめたりすることができます。

LVMでは物理的なハードディスクのことを物理ボリュームと呼び、LVMによって構成されたボリュームのことを論理ボリュームと呼びます。従来のパーティション定義では、パーティション番号で管理することしかできませんでしたが、LVMでは論理ボリュームに分かりやすい名前を付けて管理することができますし、後に説明するオンラインでのファイルシステムの拡張も可能になります。

LVMでは、物理ボリュームを束ねてボリュームグループを構成し、そのボリュームグループから必要な容量を切り出して論理ボリュームと定義する手続きを踏みます。そして、論理ボリュームの中にファイルシステムを作ることが可能です。

ここでは、増設したディスクを物理ボリュームとしてLVMに追加し、論理ボリュームのサイズを拡張する方法を紹介します。

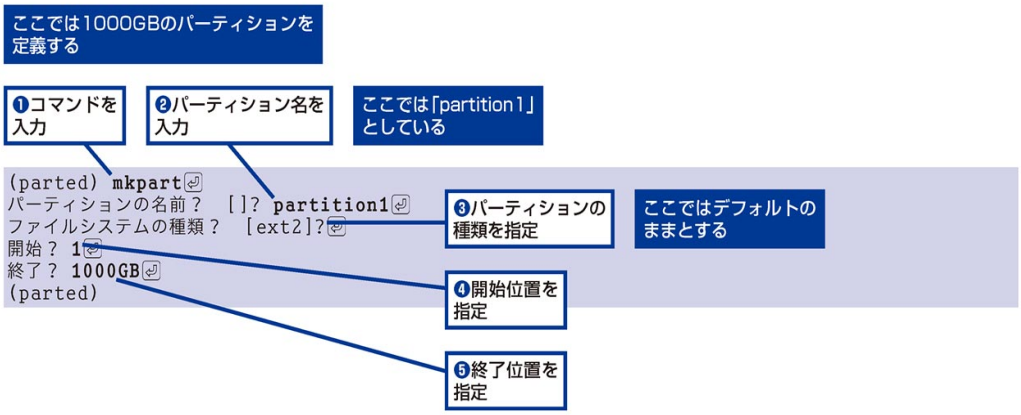
1 partedを起動する



次のページに続く

2 GPT形式を選ぶ

3 パーティションを作る



4 lvmフラグを有効にする

HINT!

すでにパーティションがある場合

すでにパーティションが作られていた場合、手順❷のようにパーティションテーブルを作成すると、以前のパーティションは失われます。そのため、partedがその旨を表示して、続行するかどうかを確認されます。またその後、手順❼で物理ボリュームを初期化したときにも以前の内容を消去するかどうか確認されることがあります。

5 パーティションテーブルを表示する

コマンドを入力

```
(parted) print
モデル： VMware, VMware Virtual S (scsi)
ディスク /dev/sdb: 1074GB
セクタサイズ (論理 / 物理): 512B/512B
パーティションテーブル： gpt
ディスクフラグ：

番号    開始      終了      サイズ    ファイルシステム      名前          フラグ
 1      1049kB    1000GB    1000GB    xfs                   partition1    lvm

(parted)
```

パーティションが作られている

6 partedを終了する

コマンドを入力

```
(parted) quit
通知： 必要であれば /etc/fstab を更新するのを忘れないようにしてください。

[root@host1 ~]#
```

Linuxのプロンプトに戻った

7 物理ボリュームを初期化する

コマンドを入力

```
[root@host1 ~]# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
[root@host1 ~]#
```

次のページに続く

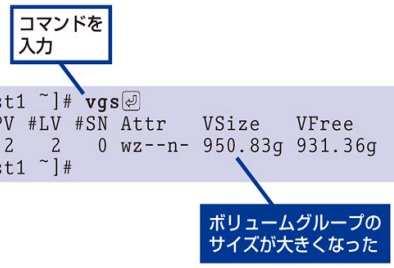
8 ボリュームグループを表示する


```
[root@host1 ~]# vgs
  VG   #PV #LV #SN Attr   VSize  VFree
  rhel   1   2   0 wz--n- 19.51g 40.00m
[root@host1 ~]#
```

9 ボリュームグループに物理ボリュームを追加する


```
[root@host1 ~]# vgextend rhel /dev/sdb1
  Volume group "rhel" successfully extended
[root@host1 ~]#
```

10 再びボリュームグループを表示する




```
[root@host1 ~]# vgs
  VG   #PV #LV #SN Attr   VSize   VFree
  rhel   2   2   0 wz--n- 950.83g 931.36g
[root@host1 ~]#
```

HINT!

拡大と縮小

手順⓫と手順⓬では、先に論理ボリュームを大きくしてから、その中のファイルシステムのサイズを拡大しています。縮小する場合には逆にファイルシステムを縮小してから論理ボリュームを縮小します。なお、ファイルシステムの拡大縮小の機能はファイルシステムの種類ごとに異なり、XFSでは縮小には対応していません。

HINT!

LVMのスナップショット機能

LVMの便利な機能にスナップショットがあります。スナップショットは、その時点の内容を記録するもので、オンラインバックアップなどに利用できます。なお、スナップショットをとると、全内容分のディスク領域が消費されるわけではなく、差分の領域のみ消費されます。

11 論理ボリュームのサイズを拡張する

ここではボリュームグループから100GBを割り当てる


```
[root@host1 ~]# lvresize -L +100G /dev/rhel/root
  Size of logical volume rhel/root changed from 17.47 GiB (4472 extents) to 117.47 GiB
(30072 extents).
  Logical volume root successfully resized
[root@host1 ~]#
```

12 ファイルシステムのサイズを拡張する

論理ボリュームのサイズにファイルシステムを合わせる


```
[root@host1 ~]# xfs_growfs /
meta-data=/dev/mapper/rhel-root  isize=256    agcount=4, agsize=1144832 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=4579328, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 4579328 to 30793728
[root@host1 ~]#
```

17-2 ブートローダーを設定するには

GRUB2

RHEL 7ではIntel 64bitとAMD 64bit（x86_64）、IBM Power System（ppc64）向けのデフォルトのブートローダーとして、GRUB2（GRand Unified Bootloader version 2）が採用されました。ブートローダーはOSが起動する前の部分を担当するプログラムです。

GRUB2はRHEL 6まで採用されていたGRUBの上位バージョンですが、その仕組みや設定方法がガラリと変わっています。そのため、設定はまったく新しい方法で行う必要があります。

このレッスンでは、GRUBとGRUB2を比較しながら説明していきます。

GRUB2の特徴

GRUB2はBIOSブートとUEFIブートの両方をサポートする高機能なブートローダーです。さまざまなファイルシステムからの起動や、LVMやRAIDのサポートが強化されています。その他、カーネルの提供者の署名で確認してから起動するUEFI Secure Bootにも対応しています。GRUBとGRUB2との設定ファイルおよびコマンドの違いは、以下の表のようになります。

GRUBとGRUB2の違い

	GRUB（RHEL 6）	GRUB2（RHEL 7）
設定ファイル	/boot/grub/grub.conf※1 /boot/efi/EFI/redhat/grub.conf※1	/boot/grub2/grub.cfg※1 ※2 /boot/efi/EFI/redhat/grub.cfg※1 ※2 /etc/default/grub /etc/grub.d/*
GRUBモジュール	/boot/grub/*_stage1_5	/boot/grub2/i386-pc/*.mod※1 /boot/grub2/x86_64-efi/*.mod※1
ブートローダーのインストール	grub-install	grub2-install
設定ファイル生成	ー	grub2-mkconfig
パスワード生成	grub-crypt	grub2-mkpasswd-pbkdf2
デフォルトエントリー指定	ー	grub2-set-default

※1 BIOSブートとUEFIブートとでファイルが異なります
※2 GRUB2のgrub.cfgは直接編集不可

GRUB2を設定するには

GRUB2になってから、設定ファイルが細分化されて見通しがよくなりました。RHEL 6までは、/boot/grub/grub.confや/boot/efi/EFI/redhat/grub.confを直接編集して設定を行うのが通常の手順でした。一方RHEL 7では、/boot/grub2/grub.cfgや/boot/efi/EFI/redhat/grub.cfgの直接編集は禁じられています。

GRUB2の設定ファイル

- /boot/grub2/grub.cfg……※BIOSブートのシステムの場合
- /boot/efi/EFI/redhat/grub.cfg……※UEFIブートのシステム
- /etc/default/grub
- /etc/grub.d
  - 00_header
  - 10_linux
  - 20_linux_xen
  - 20_ppc_terminfo
  - 30_os-prober
  - 40_custom
  - 41_custom

そこでGRUB2では、/etc/default/grubや/etc/grub.d/ディレクトリの中の設定ファイルを編集してから、以下のコマンドを実行する必要があります。

BIOSブートのシステム環境の場合には、grub2-mkconfigコマンドで/boot/grub2/grub.cfgへ書き出します。


```
[root@host1 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
```

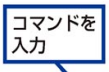
UEFIブートのシステム環境の場合には、grub2-mkconfigコマンドで/boot/efi/EFI/redhat/grub.cfgへ書き出します。


```
[root@host1 ~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

次のページに続く

GRUB2のエントリーを確認するには

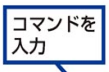
現在のGRUB2のエントリーを確認するには、grub2-editenvコマンドを実行します。




```
[root@host1 ~]# grub2-editenv list
saved_entry=Red Hat Enterprise Linux Linux, with Linux 3.10.0-121.el7.
x86_64
```

デフォルトエントリーを変更するには

システム起動時に自動的に選ばれるデフォルトエントリーを切り替えたい場合には、grub2-set-defaultコマンドを実行します。指定は番号となっています。




```
[root@host1 ~]# grub2-set-default 2
```

HINT!

UEFI Secure Bootとは

UEFI Secure Bootとは、デジタル署名によってOSのブートをチェックし、認証を通らないOSのブートを禁止する仕組みです。LinuxでUEFI Secure Bootに対応するには、第1ステージのブートローダーや、カーネル、カーネルモジュールが秘密鍵で署名され、対応する公開鍵で認証される必要があります。RHEL 7ではこれらのファイルが署名されるほか、ブートローダーとカーネルに埋め込み公開鍵が含まれています。UEFI Secure Bootを有効にしたシステムで外部のドライバーを読み込みたい場合は、そのドライバーも署名されていることを確認してください。

GRUB2のパラメーターやカーネルオプションを変更するには

GRUB2のパラメーターやカーネルオプションを変更したい場合、GRUB2では/etc/default/grubの設定ファイルを書き換える必要があります。

たとえば、GRUBのタイムアウト時間を延ばすには、GRUB2の設定ファイルの中のGRUB_TIMEOUTの値（単位は秒）を大きくします。また、カーネルのオプションを変更したい場合には、GRUB2の設定ファイルの中のGRUB_CMDLINE_LINUXを変更します。

/etc/default/grub

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=vg_root/lv_root crashkernel=auto
vconsole.font=latarcyrheb-sun16 rd.lvm.lv=vg_root/lv_swap vconsole.
keymap=us rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

RHEL 7以外のエントリーを記述したい場合には、カスタムエントリー用の設定ファイル/boot/grub.d/40_customを編集します。

他のLinuxディストリビューションを起動する場合の例

```
menuentry "Other Linux" --class gnu-linux --class os {
    insmod gzio
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos5'
    linux16 /boot/vmlinuz-custom ro
    initrd16 /boot/initramfs-custom.img
}
```

編集したら、「GRUB2を設定する」で説明したようにgrub2-mkconfigコマンドを実行します。

次のページに続く

レスキューモードと緊急モード

RHEL 7では、systemdの採用にともない、システムをレスキューモードで起動する方法も変わっています。また、レスキューモード（rescue.target）以外に似たモードとして緊急モード（emergency.target）も追加で提供されるようになりました。

●**レスキューモード（rescue.target）**：従来のレスキューモード（シングルユーザーモード、runlevel 0）に相当するモードです。主にハードウェアのメンテナンスや、バックアップ/リストア作業の場合などに利用します。

●**緊急モード（emergency.target）**：従来のレスキューモードよりも、さらに深刻な場合に利用する起動モードです。ルート（/）ファイルシステムすらマウントができない場合などに利用します。

レスキューモードで起動するには

起動時にGRUB2の画面でカーネルオプションを追加して、一時的にレスキューモードで起動することができます。

カーネルオプションのエントリーまでカーソルキーで移動して、Ctrl+Eでエントリーの最後に移動し、「systemd.unit=rescue.target」と追記します。

```
linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=a5d808c5-2b83-427c-
b831-5285b2afbed2 ro rd.lvm.lv=rhel/root vconsole.keymap=jp106 rd.lvm.
lv=rhel/swap vconsole.font=latarcyrheb-sun16 crashkernel=auto  rhgb
quiet LANG=ja_JP.UTF-8 systemd.unit=rescue.target
```

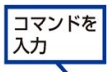
オプションを追加

その後に、Ctrl+Xキーを押すと、GRUB2はカーネルのロードを開始します。

```
Welcome to rescue mode! Type "systemctl default" or ^D to enter
default mode.
Type "journalctl -xb" to view system logs. Type "systemctl reboot" to
reboot.
Give root password for maintenance
(or type Control-D to continue):
```

rootのパスワードを入力

レスキューモードから抜けて通常起動時のtargetへ戻るには、次のようにコマンドを実行します。




```
[root@host1 ~]# systemctl default
```

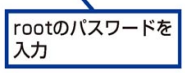
緊急モードで起動するには

同じく起動時にGRUB2の画面でカーネルオプションのエントリーまでカーソルキーで移動して、Ctrl＋Eキーでエントリーの最後に移動し、「systemd.unit=emergency.target」と追記します。

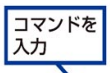
```
linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=a5d808c5-2b83-427c-
b831-5285b2afbed2 ro rd.lvm.lv=rhel/root vconsole.keymap=jp106 rd.lvm.
lv=rhel/swap vconsole.font=latarcyrheb-sun16 crashkernel=auto  rhgb
quiet LANG=ja_JP.UTF-8 systemd.unit=emergency.target
```

その後に、Ctrl＋Xキーを押すと、GRUB2はカーネルのロードを開始します。

```
Welcome to emergency mode! After logging in, Type "journalctl -xb" to
view
system logs, "systemctl reboot" to reboot, "systemctl default" to try
again
to boot into default mode.
Give root password for maintenance
(or type Control-D to continue):
```



緊急モードから再起動する場合には、次のようにコマンドを実行します。




```
[root@host1 ~]# systemctl reboot
```

17-3 バックアップするには

バックアップとリストア

システムが稼働し始めると、予期せぬシステムダウンやそれにともなうデータ損失などが発生します。そのようなときに必要となる対策がバックアップと、そのバックアップデータからデータ修復を行うリストアと呼ばれる作業です。

いくらデータをバックアップしていても、リストアの手順が分からないと修復ができません。したがって、バックアップとリストアの手順は、対で考えるべきです。

このレッスンでは、RHEL 7におけるバックアップ／リストアの考え方と、その手法について説明していきます。

RHEL7のバックアップ／リストア

Linuxのバックアップ／リストアの方法は、いろいろ存在します。RHELにおけるバックアップ／リストアのセオリーとしては、まず、OS部分はRPMのパッケージリストでリストアします。そして、設定データが含まれる/etcや、スプールやデータベースなどが含まれる/var、ユーザーデータが含まれる/home以下の内容はバックアップデータから個別にリストアします。OSが含まれるディスクをまるごとバックアップするのはストレージリソースの無駄なので避けた方がいいでしょう。リストアできないバックアップデータは取得しない方がマシです。

◆バックアップ時　任意の形式でパッケージ名を出力する

```
[root@host1 ~]# rpm -qa > rpm-qa.txt
[root@host1 ~]# rpm -qa --queryformat='%{NAME}.%{ARCH}\n' > rpm-qa-qf.txt
```

◆リストア時　まずRHEL 7のインストールメディアからOSを再インストールする　インストール後に、次のコマンドを実行してリストアする

```
[root@host1 ~]# subscription-manager register --auto-attach
[root@host1 ~]# while read RPM; do yum install -y $RPM; done < rpm-qa-qf.txt
```

注意 リストア時には、Subcription ManagerでRed Hatカスタマーポータルにシステムが登録されている必要があります。

注意 デフォルトで有効にならないレポジトリからRPMをインストールする場合には、事前にsubscription-managerコマンドやyum-config-managerコマンドなどでレポジトリを追加してください。

設定ファイルのバックアップ／リストア

設定ファイルのバックアップには、tarコマンドを使うとよいでしょう。tarコマンドが優れている点は、以下のとおりです。

・RHEL 7に標準で提供されているツールである
・レスキューモードでも提供される（どんな環境でもリストア可能）
・一般ユーザーでもファイルの自分の所有するファイルをバックアップ／リストアできる
・無圧縮／圧縮（gzip ／ bzip2 ／ xz）が選べ、圧縮率も非常に高い
・任意のファイルをピンポイントでリストアできる
・SELinuxのコンテキストをバックアップ／リストアできる
・パイプを使ってリモートからバックアップ／リストアできる

バックアップを行う場合に、何をバックアップすべきか見極めることが必要です。最近はハードディスク容量の増加にともない、メンテナンスの時間内にバックアップがとりきれないシステムも多く見受けられます。そのようなシステムに共通するのが、データの重要度に関係なく、何から何でも全部バックアップを取得している点です。大量に取得したバックアップデータは、リストア対象のデータも探しにくく、何かあったときのリストア作業にも時間がかかってしまいます。

RHEL 7に標準で含まれるアプリケーションは、バックアップすべき重要なファイルは/etcと/varにしか作ることはありません。また、/homeには一般ユーザーが作ったファイルが格納されていますが、これは別途説明します。それ以外の場所のファイルは、RPMのパッケージリストでリストア可能なものだとお考えください。

OSに起動に必要な設定ファイルが格納されている/etcディレクトリのバックアップを行う場合には、次のように実行します。

```
[root@host1 ~]# tar -C / --selinux --acls --xattrs -zcvf /opt/etc.tar.gz /etc
```

「-C /」：「/」をカレントディレクトリとする
「--selinux --acls --xattrs」：SELinuxのコンテキスト、POSIX ACL、拡張属性(xattrs)をtarファイル内に含める

HINT!

以前登録したシステムとして再登録するには

Red Hatカスタマーポータルで Consumer IDが分かっていれば、subscription-managerコマンドにて以前登録したシステムとして登録できます。

```
[root@host1 ~]# subscription-manager register --consumerid=xxxx-xxxx-xxxx-xxxx
```

Consumer IDが分からなくなった場合、カスタマーポータルから調べることも可能です。
https://access.redhat.com/management/consumers

リストアを行う場合には、次のように実行します。

```
[root@host1 ~]# tar -C / -zxvf /opt/etc.tar.gz
```

各種サービスのバックアップ／リストア

本書の各章で紹介している各種サーバーのバックアップ／リストアの方法をご紹介します。個別のサービスごとにバックアップを取得しておいた方が、部分的にファイルが損傷した場合にリストアの手順が簡素化できます。

Apache

以下のファイルをtarでバックアップ／リストアします。

・/etc/httpd/
・/etc/pki/
・/var/www/html/ ……※SSLの設定を行っている場合

PHP

以下のファイルをtarでバックアップ／リストアします。

・/etc/php.ini
・/etc/php.d/
・/etc/httpd/conf.d/php.conf

Dovecot

以下のファイルをtarでバックアップ／リストアします。

・/etc/dovecot/

Postfix

以下のファイルをtarでバックアップ／リストアします。

・/etc/postfix/
・/home/<username>/Maildir/……※Maildir形式の場合
・/var/spool/mail/ ……※mailbox形式の場合

Squid

以下のファイルをtarでバックアップ／リストアします。

・/etc/squid/

vsFTPd

以下のファイルをtarでバックアップ／リストアします。

・/etc/vsftpd/
・/var/ftp/ ……※必要があれば

Samba

以下のファイルをtarでバックアップ／リストアします。
・/etc/samba/

MariaDB

mysqldumpでバックアップ／リストアするか、サービスを停止して/etc/my.cnf、/var/lib/mysql/、/var/log/mysqld.logをtarでバックアップ／リストアします。

```
[root@host1 ~]# mysqldump -uroot -p wordpressdb > /opt/mariadb-wordpressdb.sql
```

PostgreSQL

pg_dumpでバックアップ／リストアするか、サービスを停止して/var/lib/pgsql/data/*.confをtarでバックアップ／リストアします。

```
[root@host1 ~]# pg_dump -U root -c webdb > /opt/postgresql-webdb.sql
```

BIND

以下のファイルをtarでバックアップ／リストアします。
・/etc/named.conf
・/var/named/

dnsmasq

以下のファイルをtarでバックアップ／リストアします。
・/etc/dnsmasq.conf
・/etc/hosts
・/etc/resolv.conf

DHCP

以下のファイルをtarでバックアップ／リストアします。
・/etc/dhcp/dhcpd.conf

NTP

以下のファイルをtarでバックアップ／リストアします。
・/etc/chrony.conf

STEP UP

KVMゲストやDockerコンテナーのバックアップ／リストア

KVMゲスト

KVMでバックアップ／リストアの対象となるファイルは、以下のとおりです。

- /etc/libvirt/
- /etc/libvirt/qemu.conf
- /etc/libvirt/qemu/
- /var/lib/libvirt/images/

仮想マシンのデータは、デフォルトではディスクイメージの形式のストレージプール「default」が定義されており、各仮想マシンのディスクイメージは/var/lib/libvirt/imagesに格納されています。こちらをコピーしておくことでバックアップが可能です。

ディスクイメージは、すべての仮想マシン分を1つのtar形式で固めるとサイズが大きくなりすぎるため、1仮想マシンずつ固めるか、どこかの場所にコピーしておくのがよいでしょう。

仮想マシンのリストア時は、先にディスクイメージを元の場所へコピーし、その後にvirshコマンドを実行して、ストレージプールのリフレッシュや仮想マシンの登録作業を行います。

```
[root@host1 ~]# cp rhel7.img /var/lib/libvirtd/images/rhel7.img
[root@host1 ~]# virsh pool-refresh default
[root@host1 ~]# virsh define rhel7.xml
```

Dockerコンテナー

Dockerでは、コンテナーイメージをtar形式にダンプすることができます。そして他の環境で読み込んで利用することもできます。

```
[root@host1 ~]# docker save rhel7_httpd > rhel7_httpd.tar
```

似たコマンドにdocker exportコマンドもあります。こちらはコンテナーイメージの親子関係を保持せずダンプするので、リストア後に差分イメージが結合された状態でリストアされてしまいます。

```
[root@host1 ~]# docker load -i rhel7_httpd.tar
```

また、Dockerfileを使っていた場合、Dockerfileから再度コンテナーイメージを再生成することもできます。

索引

数字・記号

ABC

DEF

GHI

JKL

MNO

PQR

STU

VWX

YZ

ア

カ

サ

タ

ナ

ハ

■著者

平 初（たいら はじめ）

レッドハット株式会社

北海道滝川市出身。商社系システムインテグレーター、外資系ハードウェアベンダーを経て、現在、レッドハット株式会社にてクラウドエバンジェリストとして活躍。仮想化技術の黎明期から、IA サーバーにおける仮想化技術の啓蒙活動に力を注ぎ、日本国内における仮想化技術の普及拡大に貢献した。

16 才の頃に、雑誌の付録についてきた FTP 版の Red Hat Linux と出会い、それからずっと使い続けている。上京してから The Fedora Project に参加し、今では Fedora L10N の日本語翻訳チームをリードしている。職場でも仕事環境は Fedora を愛用。

現在、２人の子供がおり、休日は息子を連れて山手線沿線にて新幹線をウォッチングしている。田端駅、日暮里駅がお気に入りのスポット。最近の趣味は旅行と一眼レフカメラでの風景撮影。

STAFF

本文オリジナルデザイン	川戸明子
オリジナルタイトルデザイン	山岡デザイン事務所 <yamaoka@mail.yama.co.jp>
カバーデザイン	ティーワイ ファクトリー株式会社　佐藤ともゆき
本文機器イラスト	有限会社リンクスヘンダー
DTP 制作	有限会社リンクスヘンダー
進行	TSUC
編集協力	高橋正和
編集	石橋克隆 <ishib-ka@impress.co.jp>
制作	今津幸弘 <imazu@impress.co.jp>
	鈴木　薫 <suzu-kao@impress.co.jp>

本書のご感想をぜひお寄せください

http://book.impress.co.jp/books/1114101057

読者登録サービス CLUB Impress

アンケート回答者の中から、抽選で商品券（1万円分）や図書カード（1,000円分）などを毎月プレゼント。当選は賞品の発送をもって代えさせていただきます。

●本書の内容に関するご質問は、書名・ISBN・お名前・電話番号と、該当するページや具体的な質問内容、お使いの動作環境などを明記のうえ、インプレスカスタマーセンターまでメールまたは封書にてお問い合わせください。電話やFAX等でのご質問には対応しておりません。なお、本書の範囲を超える質問に関しましてはお答えできませんのでご了承ください。

●落丁・乱丁本はお手数ですがインプレスカスタマーセンターまでお送りください。送料弊社負担にてお取り替えさせていただきます。但し、古書店で購入されたものについてはお取り替えできません。

■読者の窓口
インプレスカスタマーセンター
〒101-0051 東京都千代田区神田神保町一丁目105番地
TEL 03-6837-5016 ／ FAX 03-6837-5023
info@impress.co.jp

■書店／販売店のご注文窓口
株式会社インプレス 受注センター
TEL 048-449-8040
FAX 048-449-8041

できるPRO Red Hat Enterprise Linux 7（レッド ハット エンタープライズ リナックス セブン）

2015年7月1日 初版発行

著　者　平 初（たいら はじめ）＆できるシリーズ編集部（へんしゅうぶ）
発行人　土田米一
発行所　株式会社インプレス
〒101-0051　東京都千代田区神田神保町一丁目105番地
TEL 03-6837-4635（出版営業統括部）
ホームページ　http://book.impress.co.jp/

本書は著作権法上の保護を受けています。本書の一部あるいは全部について（ソフトウェア及びプログラムを含む）、株式会社インプレスから文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

Copyright © 2015 Hajime Taira and Impress Corporation. All rights reserved.

印刷所　株式会社廣済堂

ISBN978-4-8443-3839-0 C3055

Printed in Japan