

Linuxサーバー実務ガイド

〈2026年版〉 中小企業のIT担当者のための安全な構築とトラブル対応

— 「何を・どの順で・何に注意して」がわかる、兼任担当者のための実務書 —

- ✓ 構築前に決めること：ディストリビューションと設置場所の選び方
- ✓ 最初の1時間でやる初期設定と、締め出されないSSH・ファイアウォール設定
- ✓ バックアップ・ログ・監視を「続けられる仕組み」にする運用設計
- ✓ 症状別トラブル早見表、チェックリスト20、コマンド早見表つき

発行：Harmonic Society株式会社

（千葉市 | ホームページ制作・AI導入支援・IT顧問）

はじめに — 「サーバーを任された」その日から使える一冊

「何から手をつければいいのか分からない」「セキュリティ設定に不安がある」——初めてLinuxサーバーを任されたIT担当者の方から、最もよく聞く声です。中小企業では、サーバー管理は専任ではなく、総務や営業と兼任の担当者が担うことも珍しくありません。

一方で、「うちは小さい会社だから攻撃の標的にはならない」という考えは危険です。ランサムウェアやボットネットによる攻撃は、企業規模を問わず無差別に行われます。インターネットに公開したサーバーには、公開した直後から自動化されたログイン試行が届きます。

Linuxは、世界のWebサーバーやクラウドサーバーの大半で使われている、無料で商用利用できるOSです。ライセンス費用がかからず、安定性も高い——ただしそれは「正しく設定し、運用を続けていれば」の話です。

本書は、当社ブログのLinux関連記事30本を、**サーバーを安全に立ち上げ、止めずに運用するために必要な順番**で一冊に再構成したものです。コマンドを網羅的に覚える必要はありません。「何を・どの順で・何に注意して」行うかを押さえ、細かなコマンドは巻末の早見表と各記事で確認する——そんな使い方を想定しています。

- 第1～2章：構築前の判断と、最初の1時間でやる初期設定
- 第3～5章：セキュリティ設定、Webサービスの公開、クラウドVM特有の注意点
- 第6～7章：バックアップ・ログ・監視の運用と、トラブル時の切り分け

本書の情報について：ディストリビューションのバージョン・サポート期限、クラウドサービスの料金体系などは2026年9月時点の情報です。コマンドは Ubuntu Server (LTS版) を基本とし、必要に応じて RHEL系 (Rocky Linux/AlmaLinux) を併記しています。環境によって設定ファイルの場所やサービス名が異なる場合があります。本番環境で実行する前に、必ず検証環境で確認し、設定ファイルのバックアップを取ってから作業してください。

第1章 構築前に決めること

1-1. 「Linux」と「ディストリビューション」の違い

厳密には、Linuxとはハードウェアとソフトウェアを仲介するOSの中核部分（カーネル）を指します。実際に使うのは、カーネルに各種ソフトウェアやパッケージ管理ツールをまとめた**ディストリビューション**です。カーネルがエンジンなら、ディストリビューションは完成車にあたります。サーバー用途ではGUI（画面操作）を使わず、コマンドで操作するのが一般的です。

図表1：サーバー用途の主なディストリビューション（2026年9月時点）

ディストリビューション	系統・パッケージ管理	サポートの目安	向いている使い方
Ubuntu Server（LTS版）	Debian系・apt	LTS版は5年間のセキュリティ更新（有償のUbuntu Proで延長可）	初めての1台、クラウドVM、情報量を重視する場合
Rocky Linux／AlmaLinux	RHEL系・dnf	メジャーバージョンごとに約10年	旧CentOSからの移行、長期の安定運用
Debian	Debian系・apt	安定版を長期運用できる	安定性重視のサーバー

1-2. 選び方の結論

- **迷ったらUbuntu Server のLTS版**。情報が見つけやすく、主要クラウドで公式イメージが提供されています。2026年9月時点では24.04 LTSと26.04 LTSが選択肢です。LTSでない通常版はサポートが9か月しかないため、本番サーバーには使いません。
- **CentOSが残っているなら移行が最優先**。CentOS 8は2021年末、CentOS 7は2024年6月にサポートが終了しており、脆弱性が見つかってても修正が提供されません。移行先はRHEL互換のRocky Linux（RHELとの厳密な互換性を重視）かAlmaLinux（CentOS 7からの移行ツールELevateを提供）が現実的です。CentOS Streamは位置づけが異なるため、本番利用は慎重に検討してください。
- RHEL系の現行メジャーバージョンは9系と10系です。いずれを選ぶ場合も、**サポート終了日を公式サイトで確認し、社内の管理台帳に書いておく**ことが、将来の「気づいたら期限切れ」を防ぎます。

1-3. どこに置くか — 社内設置・クラウドVM・小型サーバー

同じLinuxサーバーでも、どこに置くかで費用と責任の範囲が変わります。多くの中小企業にとっては、公開Webサイトや検証環境はクラウド、社内限定の用途は社内設置、という**使い分け（ハイブリッド）**が現実的です。

図表2：設置場所ごとの特徴

観点	社内設置（オンプレミス）	クラウドVM（AWS EC2等）	小型サーバー（Raspberry Pi等）
初期費用	ハードウェア購入が必要	ほぼ不要（従量課金）	低い（周辺機器込みで2万円前後から）
ハードの管理	故障・電源・設置場所も自社	クラウド事業者が担当	自社。microSDは耐久性に難があり、SSD推奨
向く用途	社内ファイル、社外に置きたくないデータ	公開Webサイト、検証環境、リモートワーク	社内の小規模用途、検証・学習
注意点	停電・故障時の復旧手段を用意	常時稼働では割高になる場合がある。課金の管理が必須	性能に限界。業務の中核には向かない

1-4. 構築前に決めておく5つのこと

サーバーの事故の多くは、設定ミスよりも「決めていなかったこと」から起きます。構築を始める前に、次の5点を1枚のメモにしておきましょう。

- 用途と公開範囲**：インターネットに公開するのか、社内LAN限定か。これで開けるポートが決まります（第3章）。
- 管理者は誰か**：担当者ごとに個人アカウントを作ります。共有アカウントは、誰が何をしたか追跡できなくなるため作りません。
- 必要なスペック**：Ubuntu Server 24.04 LTSの最低要件はメモリ1GB・ストレージ2.5GBですが、実運用ではWebサーバーならメモリ4GB以上、データベースなら8GB以上が目安です。
- バックアップの保存先**：サーバーとは別の場所をもう1か所確保します（第6章）。
- 障害時の連絡と判断**：止める・戻すを誰が判断するか。社外の協力先がいれば連絡先も。

あわせて、**作業記録（いつ・誰が・何を・なぜ変更したか）を残す場所**を用意しておく、と、トラブル時の切り分けが格段に楽になります。

第2章 初期構築 — 最初の1時間でやること

2-1. インストール時の3つのポイント

まずVirtualBox等の仮想環境で手順を試してから本番機に入れるのが安全です。インストーラーでは次の3点に注意します。

- **言語は英語を推奨**（キーボードは日本語配列）。エラー文で検索したとき情報が見つけやすいためです。
- **LVMを有効にする**。後からディスク容量を拡張しやすくなります。
- **OpenSSHサーバーをインストール**にチェック。パスワードは記号を含む12文字以上に。

インストール後は、本番利用の前に次の順で初期設定を行います。**順番に意味がある**点がポイントです（例：ファイアウォールはSSHを許可してから有効化）。バックアップとログの設定（第6章）も公開前に済ませます。

図表3：インストール直後の初期設定（推奨順）

順	作業と注意点	代表的なコマンド（Ubuntu）
1	パッケージの最新化	<code>sudo apt update && sudo apt upgrade -y</code>
2	タイムゾーン（初期値はUTC）	<code>sudo timedatectl set-timezone Asia/Tokyo</code>
3	ホスト名（役割＋番号で）	<code>sudo hostnamectl set-hostname web01</code>
4	固定IP（社内設置の場合）	<code>/etc/netplan/</code> を編集し <code>sudo netplan apply</code>
5	自動セキュリティ更新	<code>sudo apt install unattended-upgrades</code>
6	管理ユーザー作成	<code>sudo adduser 名前、sudo usermod -aG sudo 名前</code>
7	SSH鍵認証へ切替（第3章）	手元のPCで <code>ssh-keygen -t ed25519</code>
8	ファイアウォール（SSH許可が先）	<code>sudo ufw allow ssh、sudo ufw enable</code>
9	不正ログイン対策（第3章）	<code>sudo apt install fail2ban</code>

RHEL系では、1は `sudo dnf update`、5は `dnf-automatic`、6のグループは `wheel` に読み替えます。

2-2. rootを直接使わない — 個人アカウント + sudo

日常作業は一般ユーザーで行い、管理者権限が必要なときだけ sudo を使うのが基本です。

- **rootの直接ログインを禁止する**（SSHの設定は第3章）。
- **sudoの設定は必ず visudo で編集する**。直接編集すると、構文ミスでsudo自体が使えなくなる恐れがあります。設定は /etc/sudoers.d/ にユーザーごとに分けると管理しやすくなります。
- **パスワードなしのsudo (NOPASSWD) は避け**、必要ならコマンドを限定します。
- **sudo usermod -aG の「-a」を忘れない**。付け忘れると既存のサブグループから外れます。
- 退職・異動時は sudo passwd -l ユーザー名 で**速やかにロック**し、月1回は棚卸しします。

2-3. パーミッションは「必要最小限」

ファイルの権限（パーミッション）は「所有者・グループ・その他」の3者に対し、読み（4）・書き（2）・実行（1）の合計で表します。よく使う値は限られています。

図表4：よく使うパーミッションの目安

対象	推奨値	意味
~/ssh ディレクトリ	700	本人のみアクセス可
秘密鍵・authorized_keys	600	本人のみ読み書き可（緩いとSSHが接続を拒否）
公開鍵	644	本人は読み書き、他は読み取りのみ
Web公開ディレクトリ	755	Webサーバーが中を読める
HTML・CSS・画像	644	実行権限は不要

やってはいけない：「動かないから chmod 777」。全ユーザーに書き込みを許し、改ざんの入口になります。まずファイルの所有者とサービスの実行ユーザー（Nginxなら通常 www-data）を確認しましょう。

第3章 セキュリティ設定 — 締め出されずに守りを固める

3-1. SSHは「鍵認証のみ」が基本

SSHはサーバーを遠隔操作する入口であり、最も攻撃を受けやすい場所です。パスワード認証は総当たり攻撃（ブルートフォース攻撃）に弱いため、**秘密鍵を持っている人しか入れない公開鍵認証に切り替える**のが第一歩です。鍵の種類はEd25519が現在の推奨で、秘密鍵にはパスフレーズを設定します。

切り替えで最も多い事故は、設定ミスで自分自身が締め出されることです。次の順番を守ってください。

図表5：SSH鍵認証への安全な切り替え手順

1. 手元のPCで `ssh-keygen -t ed25519` を実行し、鍵ペアを作る（秘密鍵は誰にも渡さない）
2. `ssh-copy-id ユーザー名@サーバー` で公開鍵をサーバーに登録する
3. 鍵でログインできることを確認する
4. `/etc/ssh/sshd_config` で `PasswordAuthentication no` と `PermitRootLogin no` を設定する
5. `sudo sshd -t` で設定ファイルの文法をチェックする
6. **今のセッションを閉じずに** `sudo systemctl restart ssh` を実行する（RHEL系は `sshd`）
7. 別のターミナルから新しく接続できることを確認してから、元のセッションを閉じる

追加で検討したい設定は、ログイン試行回数の制限（`MaxAuthTries 3`）、ログイン猶予時間の短縮（`LoginGraceTime 30`）、接続できるユーザーの限定（`AllowUsers`）、不要な機能の無効化（`X11Forwarding no`）です。

注意：クラウドのイメージなどでは、`/etc/ssh/sshd_config.d/` 配下のファイルが本体の設定を上書きしていることがあります。`sudo sshd -T` で「実際に有効になっている値」を確認しましょう。

SSHのポート番号変更について：22番から変更すると自動スキャンによるログイン試行は大きく減りますが、防御の本体はあくまで鍵認証と接続元の制限です。変更する場合は、①先にファイアウォールで新ポートを許可、②新ポートで接続確認、③旧ポートを閉じる、の順で。Ubuntu 22.10以降はSSHがソケット起動のため、`sudo systemctl daemon-reload` と `sudo systemctl restart ssh.socket` も必要です。RHEL系ではSELinuxにも新ポートを登録します（`semanage port` コマンド）。

3-2. ファイアウォールは「既定で拒否、必要な分だけ許可」

原則はひとつ、**受信はすべて拒否し、必要な通信だけを許可**することです。Ubuntu系ではUFW、RHEL系ではfirewalldを使います。

図表6：用途別に開けるポートの目安

用途	ポート	誰に開けるか
SSH（管理用）	22/tcp（変更時はそのポート）	管理者の接続元IPのみが理想
Webサイト	80/tcp・443/tcp	全体に公開
データベース（MySQL等）	3306/tcp など	公開しない。同一サーバー内か社内LANのみ
管理画面・社内ツール	8080/tcp など	社内LANまたはVPN経由のみ

図表7：UFWとfirewalldの基本操作

操作	UFW（Ubuntu系）	firewalld（RHEL系）
SSHを許可	<code>sudo ufw allow ssh</code>	<code>sudo firewall-cmd --permanent --add-service=ssh</code>
Webを許可	<code>sudo ufw allow 80/tcp</code> （443も同様）	<code>sudo firewall-cmd --permanent --add-service=http</code> （httpsも同様）
特定IPからのみSSH	<code>sudo ufw allow from 203.0.113.10 to any port 22</code>	リッチルールで送信元アドレスを指定
有効化・反映	<code>sudo ufw enable</code>	<code>sudo firewall-cmd --reload</code>
設定の確認	<code>sudo ufw status verbose</code>	<code>sudo firewall-cmd --list-all</code>

締め出し事故を防ぐ3つの鉄則

- **SSHの許可を最初に入れる**。許可せずに有効化すると、その瞬間に接続が切れます。
- **「裏口」を確認してから作業する**。クラウドのWebコンソールや物理コンソールから入れることを事前に確かめます。
- **保険をかける**。`sudo at now + 15 minutes` で「ufw disable」を予約し、接続を確認できたら `atrm` で取り消す方法があります。

3-3. 更新を止めない

最も基本的で、最も効果の大きい対策が**OSとパッケージを最新に保つ**ことです。既知の脆弱性を放置するのは、玄関の鍵を開けたままにするのと同じです。

- Ubuntu系は unattended-upgrades、RHEL系は dnf-automatic（設定で upgrade_type = security）で、セキュリティ更新を自動適用します。
- メジャーアップデートや大きな変更は「検証環境で確認 → バックアップ → 本番適用」の順で。
- カーネルの更新は再起動するまで反映されません。Ubuntuでは /var/run/reboot-required というファイルがあれば再起動待ちの状態です。**再起動の時間枠を月1回など決めておく**と、先延ばしを防げます。
- 信頼できないリポジトリは追加しません。何をインストール・更新したかは記録に残します。

3-4. 不正ログイン対策と「攻撃される面」を減らす

- **fail2ban**：ログを監視し、ログイン失敗を繰り返すIPアドレスを自動でブロックします。設定は /etc/fail2ban/jail.local に書きます（jail.confは更新で上書きされるため直接編集しない）。「10分間に3回失敗で1時間ブロック」などが一例です。自社の固定IPは除外（ignoreip）しておきます。
- **不要なサービスを止める**：sudo ss -tlnp で待ち受けているポートを、systemctl list-units --type=service --state=running で稼働中のサービスを確認し、不要なものは sudo systemctl disable --now サービス名 で停止・無効化します。
- **SELinux（RHEL系）は安易に無効化しない**：getenforce で状態を確認し、動かないときは拒否ログ（sudo ausearch -m avc -ts recent）を見て原因に対処します。

3-5. セキュリティは「点検の予定」で続ける

図表8：定期点検カレンダー

頻度	点検内容
週次	認証ログ（auth.log／secure）の不審なログイン失敗、fail2banのブロック状況
月次	アカウントの棚卸し、不要サービスの確認、更新の適用状況
四半期	監査ツールLynisによる点検、バックアップの復元テスト、登録済みSSH鍵の確認
年次	セキュリティ方針の見直し、証明書・ドメインの期限確認、障害復旧訓練

第4章 Webサービスを公開する — Nginx・HTTPS・Docker

4-1. Nginxで公開する流れ

Nginxは少ないメモリで大量の同時接続を処理でき、静的サイトの配信やアプリへの仲介（リバースプロキシ）を得意とするWebサーバーです。公開までの流れは次のとおりです。

1. インストール：`sudo apt install nginx`（RHEL系は `sudo dnf install nginx`）
2. ファイアウォールで80番・443番を許可（UFWなら `sudo ufw allow 'Nginx Full'`）
3. サイトごとの設定を `/etc/nginx/sites-available/` に作成し、`sites-enabled/` にリンクして有効化（RHEL系は `/etc/nginx/conf.d/` に配置）
4. **`sudo nginx -t` で文法チェック**してから `sudo systemctl reload nginx` で反映
5. ドメインのDNSをサーバーに向け、HTTPS化する

反映は接続を切らない **reload を優先**します。Node.jsやPythonのアプリは 127.0.0.1（サーバー内部）でだけ待ち受け、外部にはNginxだけを見せる構成が安全です。

4-2. HTTPS化は「自動更新の確認」までが作業

Let's Encryptなら無料で証明書を取得できます。`sudo certbot --nginx -d example.com` を実行すると、取得とNginx設定の修正が自動で行われます。有効期間は90日と短く（2026年9月時点。さらに短縮される予定が公表されています）、**自動更新が前提**です。`sudo certbot renew --dry-run` で更新テストを行い、`systemctl list-timers` で更新タイマーの稼働まで確認しましょう。

図表9：Webサーバーでよく出るエラーと最初に見る場所

症状	よくある原因	最初に見る場所
403 Forbidden	ファイルの所有者・パーミッション	公開ディレクトリの権限と、Nginxの実行ユーザー
502 Bad Gateway	裏側のアプリが停止している	アプリのサービス状態とログ
Nginxが起動しない	設定ファイルの文法ミス、ポート競合	<code>sudo nginx -t</code> 、 <code>sudo ss -tlnp</code>
証明書の期限切れ警告	自動更新の失敗	<code>sudo certbot renew --dry-run</code> の結果

4-3. Dockerを使うなら知っておくべきこと

Dockerは、アプリケーションと動作環境をまとめて「コンテナ」として動かす技術です。仮想マシンと違いLinuxカーネルを共有するため軽量で、1台のサーバーに複数のサービスを同居させたり、開発環境と本番環境を同じ状態で再現したりできます。Gitea（Gitサーバー）、Redmine（プロジェクト管理）、Nextcloud（ファイル共有）など、公式イメージが用意された社内ツールも多くあります。

便利な反面、次の点は必ず押さえてください。

- **公開ポートはファイアウォールをすり抜ける**：docker run -pなどで公開したポートは、UFWのルールを経由せず外部から到達できる状態になります。社内専用のツールは127.0.0.1:8080:80のように公開先をサーバー内部に限定し、外部公開はNginx経由にします。
- **dockerグループへの追加は、実質的にroot権限の付与**です。所属させるユーザーは最小限にします。
- **公式イメージを使い、定期的に更新する**。コンテナ内は非rootユーザーで動かします。
- **パスワードやAPIキーをイメージや設定ファイルに直書きしない**。環境変数ファイル等で分離し、その管理にも注意します。
- **データはボリュームに置き、バックアップ対象に含める**。コンテナを削除するとコンテナ内のデータは消えます。
- **お掃除コマンドは慎重に**。docker system df で使用量を確認できますが、docker system prune に --volumes を付けるとデータのボリュームまで消えます。

4-4. 自社アプリはsystemdで管理する

自社のアプリケーションやスクリプトを常駐させるときは、systemdのサービスとして登録すると、自動起動・ログの一元管理・異常終了時の自動再起動が使えます。

- /etc/systemd/system/ にユニットファイルを作成し、専用の実行ユーザー（User=）と、**フルパスでの起動コマンド**（ExecStart=）を指定します。systemdはPATHを参照しないためです。
- Restart=on-failure と RestartSec=5 で、異常終了時に5秒後に自動再起動します。
- ユニットファイルを変更したら sudo systemctl daemon-reload、有効化と起動は sudo systemctl enable --now サービス名。
- パッケージで入れたサービスの設定を変えるときは、元ファイルを直接編集せず sudo systemctl edit サービス名 で上書き用の設定を作ります。
- 定期実行はcronのほか、systemdタイマーも使えます。Persistent=true を付けると、停止中で実行できなかった分を起動後に実行します。

第5章 クラウドVM（AWS EC2）で運用する際の注意点

5-1. OSの中は同じ、違うのは「外側」

クラウドVMでも、OSの中の設定（第2～4章）は同じです。違うのはサーバーの外側に加わる仕組みです。AWS EC2を例に整理します。

図表10：クラウドVM特有の設定と注意点（AWS EC2の例）

項目	役割	注意点
セキュリティグループ	サーバー外側のファイアウォール	SSHを全世界（0.0.0.0/0）に開けない。自社IPのみ許可し、OSのUFWと二重で守る
キーペア	SSH接続用の鍵	秘密鍵は作成時にしかダウンロードできない。chmod 400 で保管し、紛失に備える
パブリックIP	外部からの接続先	停止・起動で変わる。固定するならElastic IP。パブリックIPv4アドレスは使用中でも課金対象（2024年2月以降）
EBS・スナップショット	仮想ディスクとそのバックアップ	使っていないボリュームやスナップショットも課金され続ける
AMI	サーバー丸ごとのイメージ	定期的に作成しておく、同じ構成で素早く復旧できる

初回ログイン後は、既定のユーザー（UbuntuのAMIなら ubuntu）とは別に担当者ごとのユーザーを作り、第2～3章の初期設定を行います。AWSアカウント自体も、ルートユーザーに多要素認証を設定し、日常操作は権限を絞ったユーザーで行います。管理者の接続元IPを固定できない場合は、AWS Systems Managerのセッションマネージャーなど、SSHポートを公開せずに操作する方法も検討できます。

5-2. 課金事故を防ぐ

- 最初にAWS Budgetsで月額予算を設定し、80%と100%到達時にメール通知を受け取ります。
- 検証環境は使わない時間帯に停止します。
- 小さく始め、CloudWatchで実際の使用率を見てからサイズを調整します。常時稼働の本番はリザーブドインスタンス等の割引も検討できます。
- 無料利用枠の条件は変更されることがあるため、公式情報を確認してください。

第6章 運用 — バックアップ・ログ・監視を「仕組み」にする

6-1. バックアップは3-2-1ルールから

故障、誤操作、ランサムウェア——データ消失のリスクは常にあります。基本方針として広く推奨されているのが**3-2-1ルール**です。**3つ**のコピーを持ち（元データ+バックアップ2つ）、**2種類**の媒体に保存し、**1つ**は遠隔地（別拠点やクラウドストレージ）に置く。完全な実施が難しくても、**最低限「サーバー外」を含む2か所**に保管しましょう。

図表11：バックアップ対象と頻度の目安

優先度	対象	具体例	推奨頻度
最高	業務データ	データベース、顧客情報	毎日（または更新の都度）
高	設定ファイル	/etc/ 以下、アプリの設定	変更時+週次
中	アプリ・Webコンテンツ	/var/www/ 以下	デプロイ時+週次
低	ログ	/var/log/ 以下	必要に応じて

6-2. 道具の使い分けと注意点

- **tar**でアーカイブ、**rsync**で変更分だけをコピー。rsyncはコピー元末尾の「/」の有無で動作が変わるため、-n（ドライラン）で確認してから実行します。
- **データベースはファイルコピーではなくダンプ機能（MySQLなら mysqldump）で取得**します。
- **「同期」はバックアップではありません**。--delete で同期すると、誤削除や暗号化されたファイルもそのまま複製されます。日付ごとのフォルダなどで**世代を残**します。
- 機密データは暗号化し、cronやsystemdタイマーで自動化して、**失敗時は通知が届く**ようにします。

6-3. 「戻せる」ことを確かめる

月1回はアーカイブが壊れていないか（tar -tzf で一覧表示）、四半期に1回はテスト環境へ実際に復元、年1回は全体を戻す訓練を行い、**復元手順書**を最新に保ちましょう。

6-4. ログは「どこに何があるか」を知っておく

障害調査も不正アクセスの発見も、出発点はログです。

図表12：主なログの場所

内容	Ubuntu/Debian系	RHEL系
システム全般	/var/log/syslog	/var/log/messages
認証（SSH・sudo）	/var/log/auth.log	/var/log/secure
Webサーバー	/var/log/nginx/	同左
サービス単位	journalctl -u サービス名	同左

journalctlは、時間（--since "1 hour ago"）、重要度（-p err でエラー以上）、起動単位（-b で今回の起動以降）で絞り込めるのが強みです。

- **肥大化対策**：logrotate（/etc/logrotate.d/）で定期的に圧縮・削除。journaldは journalctl --disk-usage で使用量を確認し、journald.conf の SystemMaxUse で上限を設定します。
- **保持期間の目安**：セキュリティ関連は最低90日（可能なら1年）、アプリケーションは30～90日。業界の規制があればそちらに従います。

6-5. 監視は「閾値」と「通知先」を決める

図表13：最低限の監視項目と目安

項目	確認コマンド	目安
CPU負荷	uptime (load average)	CPUコア数（nproc で確認）を継続的に超えたら要調査
メモリ	free -h	free ではなく available を見る。スワップの多用は不足の兆候
ディスク	df -h、df -i	使用率80%で警戒、90%で即対処。inodeの枯渇にも注意
サービス	systemctl status サービス名	failed になっていないか
証明書	sudo certbot certificates	期限の手前で更新されているか

sysstatの sar で履歴を残すと「いつから遅いのか」をさかのぼれます。1台ならスクリプト+cronのメール通知から始め、台数が増えたらZabbix、Prometheus+Grafana、Mackerel、CloudWatchなどを検討します。

第7章 トラブル対応 — 慌てずに切り分ける

7-1. 基本の6ステップ

トラブル時にいちばん避けたいのは、闇雲に設定を変えたり再起動を繰り返したりして、状況を悪化させることです。次の流れを守るだけで、対応の質は大きく変わります。

1. **症状を正確に把握する**：いつから、何が、どう動かないのか
2. **ログを確認する**：直近のエラーメッセージを探す
3. **直前の変更を確認する**：設定変更、更新、デプロイはなかったか
4. **仮説を立てる**：ログとエラーメッセージから原因の候補を挙げる
5. **1つずつ検証する**：一度に複数の変更を加えない
6. **記録する**：行った対処と結果を残す（次回と再発防止のため）

7-2. 最初に打つ5つのコマンド

図表14：トラブル時の初動コマンド

コマンド	分かること
uptime	稼働時間と負荷（load average）。直近で再起動していないか
df -h	ディスクの空き容量。満杯はあらゆる不具合の原因になる
free -h	メモリの残り（available）とスワップの使用状況
sudo journalctl -xe	直近のシステムログとエラーの説明
top	CPUやメモリを大量に使っているプロセス。wa（I/O待ち）の高さ

7-3. 「再起動すれば直る」の前に

再起動で症状が消えることは多いものの、原因の手がかりも一緒に消え、多くの場合は再発します。本番サーバーでは、再起動の前に上記コマンドの結果とログを保存し、影響範囲（誰が困っているか）と連絡先を確認してから判断しましょう。

ネットワークの不具合は、**下の層から順に**確かめるのが鉄則です。自分のIPアドレス（ip addr）→ 経路（ip route）→ ゲートウェイへの疎通（ping）→ 外部IPへの疎通（ping -c 3 8.8.8.8）→ 名前解決（dig や resolvectl status）→ 目的のポート（nc -zv ホスト ポート番号、curl -I URL）。どこで途切れるかが分かれば、原因はほぼ絞り込めます。

7-4. 症状別・トラブル対応早見表

中小企業のサーバーで実際に起きやすいトラブルを、症状から引けるようにまとめました。

図表15：症状別・トラブル対応早見表

症状	まず疑う原因	確認と対処
「No space left on device」	ログの肥大化、パッケージのキャッシュ、Dockerの不要データ	df -h で満杯の場所を特定し、du -h --max-depth=1 で掘り下げる。journalctl --vacuum-size=500M、sudo apt clean、logrotate設定の見直し
ファイルを消しても容量が空かない	削除済みファイルをプロセスが開いたまま	sudo lsof +L1 で特定し、該当サービスを再起動
容量はあるのにファイルが作れない	inodeの枯渇（小さなファイルの大量発生）	df -i で確認し、大量のファイルがあるディレクトリを整理
プロセスが突然消える	メモリ不足によるOOM Killer	sudo journalctl -k で「Out of memory」を確認。原因プロセスの特定、スワップ追加、メモリ増設
応答が極端に遅い	CPUの高負荷、ディスクI/O待ち	top の wa 値、ps aux --sort=-%cpu、vmstat 2 5
SSH：Connection refused	SSHサービスの停止、ポート違い	コンソールから systemctl status ssh、sudo ss -tlnp
SSH：Permission denied	鍵の登録漏れ、権限が緩すぎる	~/ssh は700、authorized_keys は600。サーバー側の auth.log を確認
SSH：タイムアウト	ファイアウォール、セキュリティグループ	sudo ufw status、クラウドのセキュリティグループ、nc -zv で到達確認
サービスが起動しない	設定ミス、ポート競合、権限、SELinux	systemctl status → journalctl -u サービス名 → 設定の文法チェック（nginx -t 等）→ ss -tlnp
外部に出られない・名前解決できない	ゲートウェイ、DNS設定	ping -c 3 8.8.8.8 が通るか。通るならDNS（resolvectl status）

予防がいちばんの対策です。ディスク・メモリの監視と通知（第6章）、設定変更前のバックアップ（例：sudo cp 設定ファイル 設定ファイル.bak）、変更内容の記録、検証環境での事前確認、戻し方（ロールバック手順）の準備——この5つを習慣にしておけば、トラブルの多くは「起きてても慌てない」ものになります。

巻末付録 Linuxサーバー セキュリティ・運用チェックリスト20

【構築・初期設定】

- 1. 用途と公開範囲（開けるポート）をメモに残している
- 2. サポート期間内の長期サポート版を使っている（サポート終了のOSが残っていない）
- 3. 担当者ごとの個人アカウントを使い、共有アカウントを作っていない
- 4. パッケージを最新化し、自動セキュリティ更新を有効にしている
- 5. タイムゾーン（Asia/Tokyo）とホスト名を設定している

【アクセス制御】

- 6. SSHは鍵認証のみ（PasswordAuthentication no）にしている
- 7. rootの直接ログインを禁止している（PermitRootLogin no）
- 8. SSHの接続元をIPアドレスで制限している（またはSSHを外部公開していない）
- 9. ファイアウォールは既定で拒否し、必要なポートだけを開けている
- 10. fail2ban等でログイン失敗を繰り返すIPを自動ブロックしている
- 11. sudo権限は必要な人だけに付与し、visudoで管理している
- 12. パーミッション777のファイル・ディレクトリを放置していない

【サービス公開】

- 13. HTTPS化し、証明書の自動更新をテストした（certbot renew --dry-run）
- 14. データベースや管理画面をインターネットに公開していない（Dockerの公開ポートを含む）
- 15. 不要なサービスを停止・無効化した（ss -tlnp で待ち受けを確認）

【運用・障害対応】

- 16. バックアップを自動で取得し、サーバー外にも保管している
- 17. 直近3か月以内に、実際に復元できることを確認した
- 18. ディスク・メモリ・サービス停止を監視し、通知が届く
- 19. 設定変更の記録（いつ・誰が・何を・なぜ）を残している
- 20. 障害時の連絡先、コンソールへの入り方、復旧手順を文書にしている

チェックが10個以下の場合は、まず6～9（SSHとファイアウォール）と16～17（バックアップと復元）から着手してください。この6項目だけで、不正侵入とデータ消失という二大リスクを大きく下げられます。

巻末付録 よく使うコマンド早見表

Ubuntu系を基本に、RHEL系で異なるものは括弧内に記載しています。迷ったときは、①変更前に .bak を作る、②文法チェックしてから反映、③反映後は別の方法で動作確認、の3原則を守ってください。

図表16：用途別コマンド早見表

目的	コマンド
パッケージを最新化	sudo apt update && sudo apt upgrade (sudo dnf update)
サービスの状態を見る	systemctl status サービス名
起動・停止・再起動・無停止反映	sudo systemctl start/stop/restart/reload サービス名
自動起動を有効にして起動	sudo systemctl enable --now サービス名
サービスのログ（直近50行）	journalctl -u サービス名 -n 50
直近1時間のエラー	journalctl -p err --since "1 hour ago"
ログをリアルタイム表示	journalctl -f、tail -f ファイル名
負荷・メモリ	uptime、free -h
重いプロセスを探す	top、ps aux --sort=-%mem
ディスクの使用率	df -h (inodeは df -i)
容量を使っている場所	sudo du -h --max-depth=1 /var
待ち受けポートの確認	sudo ss -tlnp
IP・経路・疎通・名前解決	ip addr、ip route、ping -c 3 ホスト、dig ドメイン名
ポートへの到達確認	nc -zv ホスト ポート番号、curl -I URL
ファイアウォールの確認	sudo ufw status verbose (sudo firewall-cmd --list-all)
SSH設定の検証・実効値	sudo sshd -t、sudo sshd -T
ユーザーに管理者権限	sudo usermod -aG sudo 名前 (wheelグループ)
アカウントのロック	sudo passwd -l 名前
Nginx設定の検証・証明書更新テスト	sudo nginx -t、sudo certbot renew --dry-run

おわりに — 1台を安全に回せれば、次の1台は怖くない

Linuxサーバーの運用で大切なのは、難しいコマンドを覚えることではなく、**正しい順番で設定し、点検とバックアップを習慣にすること**です。本書の内容は、1台目のサーバーで一度やり切れば、2台目以降はほぼ同じ手順で再現できます。まずは巻末のチェックリストで現状を確認し、できていない項目から1つずつ埋めていってください。

発行元：Harmonic Society株式会社

千葉市を拠点に、中小企業のホームページ制作・AI導入支援・IT顧問を提供しています。本書は、当社ブログで発信してきたLinux解説記事群をもとにまとめたものです。「社内にIT専任者がいない」会社の相談相手として、技術の分かる相手に直接ご相談いただけます。オンライン対応・全国OK。

サービス	内容・価格
IT顧問ライトプラン	AI・ITの継続相談を月1.5万円から
ホームページ制作	月額制 月5,500円～（初期0円）／LP制作 1本2.5万円～（いずれも税込）
ホームページ保守運用	GA4×AIの週次レポート付き 月額5,500円

まずは30分の無料相談から

「このサーバー、今のままで大丈夫？」「クラウドに移すべきか」——本書を読んで浮かんだ疑問を、そのままお持ちください。売り込みはしません。

- Web：<https://harmonic-society.co.jp/contact/>
- お役立ち資料一覧：<https://harmonic-society.co.jp/whitepaper/>

本書の内容は2026年9月時点の情報に基づきます。ディストリビューションのサポート期限、クラウドサービスの料金や仕様は変更される場合があります。コマンド・設定例は一般的な環境を想定したものであり、実行の際は必ず検証環境で確認のうえ、自己責任で行ってください。無断転載を禁じます。