## ✅ ****CPU**使用率の理想的な目安**

| 状態 | 使用率の目安 | 備考 |
| --- | --- | --- |
| **通常時** | 10〜50% | 常に100%に張り付いているのは問題あり。 |
| **一時的なピーク** | 70〜90% | 処理が集中するバッチやアクセス集中など。 |
| **常時90%以上** | 問題あり | サーバー増設やアプリ最適化を検討。 |

### ✔ 補足

* CPU使用率が**常時80%以上**なら、**ボトルネック**を疑うべきです。
* 「CPUアイドルが0%」になると、負荷に耐えられずパフォーマンスが劣化します。
* CPUコア数が多い場合は**スレッド単位での分布**も見るべきです。

## ✅ **メモリ使用率の理想的な目安**

| 状態 | 使用率の目安 | 備考 |
| --- | --- | --- |
| **通常時** | 50〜80% | キャッシュやバッファとして使われる分も考慮。 |
| **一時的に90%超** | 許容範囲 | キャッシュなら問題なし。swapの使用がなければOK。 |
| **常時90%以上** | 要注意 | swapが発生していないか確認。スラッシングに注意。 |

### ✔ 補足

* **Linuxでは「空きメモリが少ない＝悪」ではありません。**
* free や top で「バッファ・キャッシュ」を除いた**本当の空きメモリ**を確認すべきです。
  + free -m コマンド例：available を見る
* **swapの使用量が多いと危険信号**（ディスクI/Oが激増します）

## ✅ 監視ツールの設定目安（アラート閾値例）

| 指標 | 通常アラート閾値 |
| --- | --- |
| CPU使用率 | 85%以上（5分継続） |
| メモリ使用率 | 90%以上（swap発生時） |
| swap使用量 | 1GB以上で要警戒 |

## ✅ 種類別の使用率の考え方（実務経験から）

| サーバー種別 | CPU使用率 | メモリ使用率 | 備考 |
| --- | --- | --- | --- |
| Webサーバー（Nginxなど） | 10〜30% | 30〜70% | 軽量な処理が多いため余裕あり |
| アプリケーションサーバ | 30〜70% | 50〜90% | Java/Pythonなどで上がりやすい |
| データベースサーバー | 20〜80% | 70〜90% | メモリキャッシュが重要。swap厳禁 |
| ファイルサーバー | 10〜40% | 30〜70% | CPUよりI/Oがボトルネックになる傾向 |

## ✅ まとめ

| 項目 | 理想値（目安） |
| --- | --- |
| CPU | 10〜70%以内で推移 |
| メモリ | 50〜80%以内が理想 |
| swap | 常時0、使うならごくわずか |
| 常時高負荷 | 要調査（アプリ or リソース不足） |

必要であれば top, vmstat, sar, htop, glances などのツールを使って、状況に応じて可視化や監視強化をするのもおすすめです。

**あまりにもCPUやメモリ使用率が低すぎる状態が常態化している場合、それは「オーバースペック（過剰リソース）」のサイン**です。

## ✅ オーバースペックの具体的な例と影響

| 状態 | 説明 |
| --- | --- |
| CPU使用率が常時10%未満 | 処理能力に対して負荷が小さすぎる可能性。 |
| メモリ使用率が常時30%未満 | メモリを持て余しており、コスト効率が悪い。 |
| スケールアウトしてるのに各ノードが余裕過ぎる | ノード台数や構成を見直すべきかも。 |

### ❗ 問題点

* **コスト面の無駄**（電力・冷却・ライセンス・クラウド課金など）
* **物理スペースの無駄**（オンプレならラックに余分な機材）
* **管理リソースの分散**（無駄にホストが多いと障害対応も増える）

## ✅ オーバースペックかどうかを判断する目安

| 項目 | 見るべきポイント |
| --- | --- |
| 平常時CPU | 5〜15%以下が続いていないか？ |
| 平常時メモリ | 30%以下しか使っていないか？ |
| スケール状況 | 分散数と負荷のバランスは取れているか？ |
| アプリ特性 | キャッシュ戦略やピークアクセス想定済みか？ |

## ✅ 実務的な対策

| 状況 | 対策例 |
| --- | --- |
| オンプレ（物理）サーバがオーバースペック | 小型化 or VM統合 |
| 仮想環境でリソースが余っている | vCPUやメモリを縮小して別用途に再分配 |
| クラウドでリソース余りがち | インスタンスタイプを下げてコスト削減 |
| 日中と夜間で使用率が極端に違う | オートスケーリングやリソース予約見直し |

## ✅ 理想のバランス感覚

サーバーは\*\*「リソースを使い切らない程度に使い倒す」のが最も経済的かつ安定的\*\*です。

* CPU：**30〜60%台を普段使い**
* メモリ：**60〜80%を有効活用**
* スパイク時だけ80〜90%を許容範囲とし、常時は避ける

**VM（仮想マシン）上のサーバー群のバックアップ**には、いくつかの**定石**や**ベストプラクティス**があります。VMホスト自体が潰れたときの対策を含めて、全体像を整理してお伝えします。

## ✅ 【前提】VM上のサーバー群とは？

あなたの言う「VMにいっぱいサーバーがある」というのは、おそらく以下のような構成です：

コピーする編集する

物理マシン（ホストOS or ハイパーバイザ）

└─ VM1（Webサーバ）

└─ VM2（DBサーバ）

└─ VM3（DNSサーバ）など

つまり、**一台の物理ホスト上で複数の仮想マシンが動いている構成**ですね。

## ✅ 物理ホストが壊れたらどうなる？

* **ハードウェア障害＝全VMが一気に停止**
* バックアップがなければ、**VMごと消滅の可能性あり**
* → **VM単体だけでなく、ホストごと対策すべき**

## ✅ バックアップの定石（層別で考える）

### 🔹 レイヤー1：****VM**レベルのバックアップ**

| 方法 | 特徴 |
| --- | --- |
| **スナップショット** | 一時的なバックアップ向け。運用中は向かない（破損の可能性）。 |
| **VMDK/VDIなどの仮想ディスクコピー** | 電源OFFの状態で取得するのが確実。 |
| **ホスト単位のバックアップ** | 物理マシン上のVMファイル群ごと取得（例：/var/lib/libvirt/images/）。 |

### 🔹 レイヤー2：**ゲスト**OS**内でのバックアップ**

| 方法 | 特徴 |
| --- | --- |
| OS上でrsync/tar/backupソフトなど | 柔軟だが復旧にはOSが必要。 |
| DBやサービスは**アプリケーション整合性**に注意（例：MySQLならmysqldumpやxtrabackup）。 |  |
| cronやsystemd-timerで**定期実行**するのが実務的。 |  |

### 🔹 レイヤー3：**リモートバックアップ **/ DR**対策**

| 方法 | 特徴 |
| --- | --- |
| NFS、rsync、rclone、S3連携など | オフサイトに保存できるのが強み。 |
| 物理ホストが死んでもバックアップが別サーバにあれば復旧可 |  |
| \*\*DR（ディザスタリカバリ）\*\*環境も用意できれば理想 |  |

## ✅ ベストプラクティス（実務の定石）

| 項目 | 実務でのおすすめ |
| --- | --- |
| VMのスナップショット | 一時退避には便利だが、本番バックアップとしては危険（破損のリスク）。長期保管には不向き。 |
| 仮想ディスクの定期コピー | 週1回は電源を止めて「コールドバックアップ」を取るのが安心。 |
| OS内のバックアップ | 毎日夜間にアプリ・データのバックアップ。rsync + cronが定番。 |
| ホストバックアップ | /etc/libvirt/ や VM構成ファイル・ディスクファイル も保管対象に。 |
| 自動化 | ansible や バックアップスクリプト + cron で手間を削減。 |
| リストア手順 | **「復旧手順書」** を作成しておくと、いざというときに安心。 |

## ✅ 実例：VM群を動かす物理ホストが壊れた場合の復旧戦略

1. **別のホストに仮想環境（libvirt / VirtualBox / ESXiなど）を再構築**
2. バックアップしていたVMイメージや設定ファイルをリストア
3. ゲストVMを起動
4. サービスが正常稼働するか確認

## ✅ まとめ：VM群のバックアップの定石

| レイヤー | 推奨手法 |
| --- | --- |
| VMスナップショット | 短期保護（実験用途向け） |
| 仮想ディスクコピー | 長期保存（週次など） |
| OS内バックアップ | 毎日取得（rsync等） |
| リモート保存 | NAS/外部ストレージ/S3等 |
| 復旧手順書 | 文書化必須 |

**ハイパーバイザー（＝物理ホスト上の仮想化基盤）ごと丸ごとバックアップする方法**も確立されています。これは「ホスト丸ごと障害」に備える強力な戦略です。

## ✅ 「ハイパーバイザー丸ごとバックアップ」とは？

### 意味：

仮想マシンだけでなく、**ホストOS + ハイパーバイザー設定 + VM構成ファイル + VMディスクイメージ**など、**仮想環境をまるごと1台バックアップ**すること。

## ✅ どうやってやるのか？（代表例別）

### 🔹 KVM/QEMU（libvirt環境）

| 方法 | 備考 |
| --- | --- |
| /etc/libvirt/ の設定保存 | VM定義ファイル（XML形式）が格納されている。 |
| /var/lib/libvirt/images/ を保存 | 仮想ディスク（qcow2など）本体。 |
| virsh dumpxml <vmname> | 個別VMの設定をバックアップできる。 |
| ホストOSごとrsyncやtarで別ストレージに退避 | フルバックアップに最適。 |
| クローン作成（例：Clonezilla） | OSイメージごとディスク単位で複製可能。 |

### 🔹 VMware ESXi

| 方法 | 備考 |
| --- | --- |
| vCenter + ESXiの設定バックアップ機能 | WebUIやCLIで取得可能。 |
| .vmxや.vmdkファイルを取得 | 仮想マシンごとに構成・ディスクが分かれている。 |
| Veeamなど商用ソリューション | 丸ごとスナップショット + リモート保存が可能。 |

### 🔹 Proxmox VE

| 方法 | 備考 |
| --- | --- |
| vzdumpコマンドで仮想マシン丸ごとバックアップ | LXC/VM両対応。スケジューリングも可能。 |
| /etc/pve/ ディレクトリの保存 | クラスタ構成やVM設定が入っている。 |

## ✅ 実務的なやり方（KVM系の場合）

たとえばKVMホストを丸ごとバックアップするならこんな構成です：

bash

コピーする編集する

#!/bin/bash

BACKUP\_DIR="/backup/kvmhost-$(date +%F)"

mkdir -p "$BACKUP\_DIR"

# VM設定ファイルの保存

cp -a /etc/libvirt "$BACKUP\_DIR/libvirt-config"

# VMディスクイメージの保存

rsync -a --sparse /var/lib/libvirt/images "$BACKUP\_DIR/images"

# ネットワーク設定など

cp -a /etc/sysconfig/network-scripts "$BACKUP\_DIR/network"

# fstabやfirewalldなどの設定もお好みで

## ✅ 丸ごとバックアップの利点

| 項目 | 内容 |
| --- | --- |
| 🔁 復旧が早い | ホストOS＋仮想マシンを一気に復元できる |
| 🧱 仮想化環境をそっくり再現 | 環境差異が少なくなる（特にテスト環境と揃えやすい） |
| 🧯 DR対策 | 別ホストで即時展開がしやすい（例：PXE + rsyncで再構築） |

## ✅ 注意点

| 注意 | 内容 |
| --- | --- |
| 大容量になりがち | 仮想ディスクが多いとTB単位になる |
| ライブ中のコピーは不整合のリスク | 可能ならVMを止めてコールドバックアップ推奨 |
| 定期的に検証する | 復旧テストを忘れると「取れてるだけのバックアップ」に |

## ✅ 実務ベースのアプローチまとめ

| 対象 | 方法 |
| --- | --- |
| VM単体 | rsync or virsh dumpでディスク・設定取得 |
| ハイパーバイザー全体 | /etc/libvirt, /var/lib/libvirt, ネットワーク設定、OS全体をバックアップ |
| DR目的 | 外部ストレージ or 別ホストへ定期同期（rsync + cron） |
| 監視 | 成功/失敗をログに記録 or 通知（mail/slack） |

「**KVMホストの丸ごとバックアップ＆復元スクリプト**」

## ✅ バックアップスクリプト（kvm\_backup.sh）

bash

コピーする編集する

#!/bin/bash

# バックアップ先（変更してください）

BACKUP\_BASE="/backup/kvm"

DATE=$(date +%F)

BACKUP\_DIR="$BACKUP\_BASE/$DATE"

# ログファイル

LOGFILE="$BACKUP\_DIR/backup.log"

mkdir -p "$BACKUP\_DIR"

log() {

echo "$(date '+%F %T') - $1" | tee -a "$LOGFILE"

}

log "=== KVM ホストバックアップ開始 ==="

# 1. libvirtの設定

log "[1/5] libvirt設定を保存中..."

mkdir -p "$BACKUP\_DIR/libvirt"

cp -a /etc/libvirt "$BACKUP\_DIR/libvirt"

# 2. VM定義情報（XML形式）を保存

log "[2/5] 各VMの定義（XML）を保存中..."

mkdir -p "$BACKUP\_DIR/vm\_xml"

for vm in $(virsh list --all --name); do

virsh dumpxml "$vm" > "$BACKUP\_DIR/vm\_xml/${vm}.xml"

done

# 3. VMディスクイメージをバックアップ

log "[3/5] 仮想ディスクイメージをバックアップ中..."

mkdir -p "$BACKUP\_DIR/images"

rsync -a --sparse /var/lib/libvirt/images/ "$BACKUP\_DIR/images/"

# 4. ネットワーク設定など

log "[4/5] ネットワーク設定などを保存中..."

mkdir -p "$BACKUP\_DIR/etc"

cp -a /etc/sysconfig/network-scripts "$BACKUP\_DIR/etc/"

cp -a /etc/hosts "$BACKUP\_DIR/etc/"

cp -a /etc/fstab "$BACKUP\_DIR/etc/"

cp -a /etc/firewalld "$BACKUP\_DIR/etc/" 2>/dev/null

# 5. 完了

log "[5/5] バックアップ完了しました"

## ✅ 復元スクリプト（kvm\_restore.sh）

bash

コピーする編集する

#!/bin/bash

# バックアップ元（変更してください）

BACKUP\_DIR="/backup/kvm/2025-05-28" # 日付はバックアップ日時に合わせて変更

LOGFILE="/var/log/kvm\_restore.log"

log() {

echo "$(date '+%F %T') - $1" | tee -a "$LOGFILE"

}

log "=== KVM ホスト復元開始 ==="

# 1. libvirt設定復元

log "[1/4] libvirt設定を復元中..."

cp -a "$BACKUP\_DIR/libvirt"/\* /etc/libvirt/

# 2. VM定義の再インポート

log "[2/4] VM定義を再インポート中..."

for xml in "$BACKUP\_DIR/vm\_xml"/\*.xml; do

virsh define "$xml"

done

# 3. 仮想ディスクの復元

log "[3/4] 仮想ディスクイメージを復元中..."

rsync -a --sparse "$BACKUP\_DIR/images/" /var/lib/libvirt/images/

# 4. ネットワーク設定復元（再起動必要）

log "[4/4] ネットワーク設定など復元中..."

cp -a "$BACKUP\_DIR/etc/network-scripts" /etc/sysconfig/

cp -a "$BACKUP\_DIR/etc/hosts" /etc/

cp -a "$BACKUP\_DIR/etc/fstab" /etc/

cp -a "$BACKUP\_DIR/etc/firewalld" /etc/ 2>/dev/null

log "復元完了。必要であれば再起動してください。"

## ✅ 運用上のポイント

| 項目 | 内容 |
| --- | --- |
| 🎯 cron登録 | kvm\_backup.sh を週1回程度でスケジューリングすると良いです。 |
| 🔐 バックアップ先 | 別ディスク/別ホスト/クラウド（rsyncやNFS）に逃がすのがベスト。 |
| 🧪 復元テスト | テスト用VM環境で復元手順を定期的に確認してください。 |
| 🛑 VM稼働中の注意 | 一部ディスクはスナップショットが必要な場合があります。止められるなら一時停止 or シャットダウンしてからがベストです。 |

これからの知識キャッチアップ

* 特定VMのバックアップだけしたいとき
* スナップショットとrsyncの併用方法
* バックアップ監視やメール通知の実装