

# リナックスマスター実務ラボ

無料サンプル 課題#001

模範解説つき完全版

Linuxmaster.JP

## 本サンプルについて

課題#001「止まっていたのは cron ではありません」

分野：障害調査 / 難易度：中級 / 想定所要時間：60～90分

## 収録内容

会員には毎月届くものと同じ完成品を、1冊に収録しています。

- ・第1部 課題（シナリオ・調査素材・設問3問）
- ・第2部 ヒント（3段階）
- ・第3部 模範解説（調査の型・原因の3層整理・復旧手順・改善版スクリプト・現場の教訓）

会員には毎月、課題（1日）→ヒント（15日頃）→模範解説（翌月1日）の順に、3通に分けてメールが届きます。本サンプルは、その1ヶ月分を1冊に収録したものです。第2部・第3部を開く前に、まず課題だけで解いてみることをおすすめします。

出題・解説：宮崎 智広（リナックスマスター.JP 代表・現役サーバー管理者）

発行：リナックスマスター.JP（株式会社イーネットマーキュリー）

入会のご案内：<https://lab.linuxmaster.jp/>

## 第 1 部 課題

※会員には毎月 1 日に届きます

### 【実務ラボ 課題#001】止まっていたのは cron ではありません

- ・分野: 障害調査 / 難易度: 中級 / 想定所要時間: 60~90 分
- ・提出締切: 今月 20 日 (課題メールへの返信で提出)
- ・ヒント配信: 15 日頃 / 模範解説: 翌月 1 日

### 1. シナリオ

あなたは、従業員 50 名の製造業の会社で、Web サーバー (社外向けサイト + 社内ポータル) の管理を前任者から引き継いだ担当者です。

7 月 9 日の朝、開発担当からこう頼まれました。

「7 月 8 日のバックアップから、間違えて消したファイルを 1 つ取り出してほしい」

ところが、バックアップファイルを展開しようするとエラーになります。

調べてみると、7 月 7 日以降の 3 本がすべて展開できません。

バックアップは毎日深夜、cron で自動取得しているはずでした。

引き継ぎ書には「バックアップは自動化済み。/var/backup に 30 日分保存」とだけ書かれています。

**このサーバーで何が起きているのかを、以下の調査素材から特定してください。**

## 2. 前提環境

- OS: AlmaLinux 9 (RHEL 9 系)
- 対象サーバー: web01 (Web サーバー)
- バックアップ対象: /var/www (サイトのコンテンツ一式)
- バックアップ先: /var/backup (専用の 20GB ディスク /dev/sdb1 をマウント)
- コマンド出力は英語ロケール環境での表示です
- 手元での環境構築は不要です。以下の調査素材 (実際のコマンド出力) を読み解いて解答してください。

### 3. 調査素材

#### 素材 1: バックアップ先の状態

```
# ls -lht /var/backup/ | head -7
total 19G
-rw-r--r--. 1 root root 663M Jul  9 02:33 www-20260709.tar.gz
-rw-r--r--. 1 root root 648M Jul  8 02:33 www-20260708.tar.gz
-rw-r--r--. 1 root root 655M Jul  7 02:33 www-20260707.tar.gz
-rw-r--r--. 1 root root 672M Jul  6 02:36 www-20260706.tar.gz
-rw-r--r--. 1 root root 668M Jul  5 02:35 www-20260705.tar.gz
-rw-r--r--. 1 root root 664M Jul  4 02:35 www-20260704.tar.gz
(以下、約 30 日分のファイルが続く)
```

#### 素材 2: 展開できないという 7 月 8 日のファイルを検証した結果

```
# tar tzf /var/backup/www-20260708.tar.gz > /dev/null
gzip: stdin: unexpected end of file
tar: Unexpected EOF in archive
tar: Error is not recoverable: exiting now
```

7 月 7 日・7 月 9 日のファイルも同じエラーになります。7 月 6 日以前のファイルは正常に展開できます。

#### 素材 3: cron の設定と実行ログ

```
# crontab -l
30 2 * * * /usr/local/bin/backup-www.sh

# grep backup-www /var/log/cron | tail -3
Jul  7 02:30:01 web01 CROND[91542]: (root) CMD (/usr/local/bin/backup-www.sh)
Jul  8 02:30:01 web01 CROND[95217]: (root) CMD (/usr/local/bin/backup-www.sh)
Jul  9 02:30:01 web01 CROND[98844]: (root) CMD (/usr/local/bin/backup-www.sh)
```

#### 素材 4: バックアップスクリプト本体

```
# cat /usr/local/bin/backup-www.sh
#!/bin/bash
# Web コンテンツ日次バックアップ (30 世代保持) 前任者作成
BACKUP_DIR=/var/backup
DATE=$(date +%Y%m%d)

tar czf ${BACKUP_DIR}/www-${DATE}.tar.gz /var/www 2>/dev/null
find ${BACKUP_DIR} -name "www-*.tar.gz" -mtime +30 -delete
```

## 素材 5: ディスクの状態

```
# df -h /var/backup
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1       20G   20G  631M  97% /var/backup

# du -sh /var/www
2.1G    /var/www
```

補足: 半年前の引き継ぎ時のメモには「/var/www は約 1.4GB」と記録されています。

## 4. 設問 (3 問)

### 設問 1: 原因の特定

7月7日以降のバックアップが壊れている直接の原因と、それが3日間誰にも気づかれなかった理由を、素材のどの行を根拠にしたかを挙げて説明してください。

### 設問 2: 復旧手順

今夜のバックアップから正常に取得できる状態に戻すための手順を、実行するコマンドを含めて順番に書いてください。

### 設問 3: 再発防止策

同じ事故を繰り返さないために、このスクリプトと運用をどう変えるべきか提案してください（スクリプトの改善版を書ける方は書いてみてください。箇条書きの提案だけでも構いません）。

## 5. 提出方法 (会員の場合)

- ・課題メールに返信する形で、設問 1~3 の解答をお送りください (A4 相当・書式自由)
- ・「実行したつもりのコマンド／確認した箇所／そこから何を判断したか」の調査記録の形がおすすめです
- ・完璧な答えでなくても大丈夫です。途中まででも、調査の道筋そのものが財産になります
- ・提出いただいた方の良い着眼点は、模範解説の中で匿名でご紹介します

## 6. ヒント配信の予告

15日頃に3段階のヒントをお送りします

(本サンプルでは次のページに収録しています。自力で解きたい方は、解答をまとめてから先へ進んでください)。

## 第 2 部 ヒント (3 段階)

※会員には 15 日頃に届きます

### ヒント 1 (見る場所)

「cron が止まっている」と「バックアップが止まっている」は別の話です。

素材 3 を見ると、cron はこの 3 日間もジョブを起動しています。

では、起動したジョブは「成功」していたのでしょうか。

ファイルが毎日作られていること (素材 1) は、成功の証拠になりますか。

### ヒント 2 (確かめ方)

素材 1 のファイルサイズをよく見てください。

壊れている 3 本 (655M・648M・663M) は、正常だった頃 (672M・668M) と見分けがつかない大きさです。

「ファイルがある」「サイズもそれらしい」でも中身は壊れている。

これが素材 2 の検証コマンド `tar tzf` が毎回の運用に必要な理由です。

では、書き込みが途中で止まる原因は何か。素材 5 と突き合わせてみてください。

残り 631M に、今夜の 1 本 (約 660M) は入りきるでしょうか。

### ヒント 3 (核心の一步手前)

素材 4 のスクリプトの「`2>/dev/null`」に注目してください。

`tar` が致命的なエラーを出しても、それは通知されるでしょうか。

さらに、`tar` が失敗してもスクリプトは次の `find` を実行して正常終了します。

`cron` は「出力があればメールする」仕組みですが、このスクリプトは失敗しても何も出力しません。

エラーは 2 段階で握り潰されています。

## 第3部 模範解説

※会員には翌月 1 日に届きます

### 解説 1: 調査の型（切り分けの順序）

この課題の調査は、次の順で絞り込むのが定石です。

- |             |   |
|-------------|---|
| (1) 事象の確認   | 壊れているのは「何」か → ファイルは存在するが展開不能（素材 1・2）          |
| (2) 起動の確認   | cron はジョブを起動しているか → している（素材 3）                |
| (3) 成果物の検証  | 起動と成功は別。tar tzf で中身を検証 → 7/7 以降の 3 本が破損（素材 2） |
| (4) 実行環境の確認 | 書き込み先は健全か → 残り 631M で 1 本分（約 660M）に足りない（素材 5） |
| (5) コードの精読  | なぜ気づけなかったか → 2>/dev/null と終了コードのマスク（素材 4）     |

現場で最初にやりがちなのは「cron の設定を疑う」ことですが、素材 3 で起動が確認できた時点で、疑うべきは cron ではなくジョブの中身に移ります。課題タイトルの「止まっていたのは cron ではありません」はこの切り分けを指しています。

### 解説 2: 根本原因の 3 層整理

層	原因
直接原因	バックアップ先 /var/backup の満杯。/var/www が半年で 1.4GB→2.1GB に肥大し、圧縮後サイズ×30 世代がディスク容量 20GB に到達した。深夜の tar は書き込み途中で「No space left on device」で異常終了し、途中まで書かれた壊れたアーカイブが残った
構造原因	エラーの二重握り潰し。(a) 2>/dev/null が tar のエラー出力を捨てているため、cron のメール通知機構に何も渡らない。(b) tar が失敗してもスクリプトは最後の find を実行して終了するため、スクリプト全体の終了コードは 0（正常）になる
運用原因	容量の監視がなく、復元テスト（展開検証）の習慣もないため、「ファイルが毎日増えている」ことをバックアップ成功と同一視していた

補足: 朝に確認した df が 100%ではなく 97%（素材 5）だったのは、tar が満杯で異常終了した「後」に、スクリプト後半の find が 30 日前の 1 本（約 630M）を削除して空きが戻ったためです。

それでも今夜の 1 本分 (約 660M) にはわずかに足りず、今夜も途中で満杯になります。df が出力するのは「実行した今この瞬間」だけで、深夜のピーク時に満杯だった証拠は残りません。ここに気づけると、切り分けが一段速くなります。なお、この `2>/dev/null` は前任者の嫌がらせではありません。tar に絶対パスを渡すと毎回

```
tar: Removing leading `/' from member names
```

という警告 (メンバ名から先頭の「/」を取り除きます、の意) が stderr に出て、cron から毎日メールが届きます。

それを黙らせるために付けたと推測できます。警告を消すつもりで、致命的な警告まで消してしまった。

現場で本当によくあるパターンです。

実際にディスクが満杯の状態での tar を実行すると、次のように失敗します (終了コードは 2) 。

```
# tar czf /var/backup/test.tar.gz /var/www
tar: Removing leading `/' from member names
tar (child): /var/backup/test.tar.gz: Cannot write: No space left on device
tar (child): Error is not recoverable: exiting now
tar: /var/backup/test.tar.gz: Wrote only 2048 of 10240 bytes
tar: Child returned status 2
tar: Error is not recoverable: exiting now
# echo $?
2
```

`2>/dev/null` はこのエラーメッセージを非表示にします。

### 解説 3: 復旧手順

```
# 1. 壊れた 3 本を削除して空きを確保 (削除前に tar tzf で破損を再確認)
tar tzf /var/backup/www-20260707.tar.gz > /dev/null # エラーになることを確認
rm -f /var/backup/www-20260707.tar.gz /var/backup/www-20260708.tar.gz /var/backup/www-20260709.tar.gz

# 2. 空き容量を確認
df -h /var/backup

# 3. 手動でバックアップを取得 (エラーを画面で見えるため 2>/dev/null は付けない)
tar czf /var/backup/www-$(date +%Y%m%d)-manual.tar.gz -C / var/www

# 4. 取得したアーカイブを展開検証
tar tzf /var/backup/www-$(date +%Y%m%d)-manual.tar.gz > /dev/null && echo OK

# 5. 空き容量を再確認し、翌朝、cron 実行分が正常に作られたことを確認
```

ポイントは手順 3 の `-C / var/www`。ルートに移動してから相対パスでアーカイブ化することで、「Removing leading `/'」警告そのものがなくなります。警告を握り潰すのではなく、警告が出ない書き方に直す。これが正しい方法です。

なお、そもそもの発端だった開発担当の依頼（7月8日時点のファイルの取り出し）は、7月7日～9日のバックアップが破損しているため叶えられません。最後に正常だった7月6日のファイルに目的のものが入っているかを確認し、「戻せるのはここまで」という事実を依頼者に伝えるところまでが、この障害対応になります。

あわせて判断が必要なのは世代数です。2.1GB（圧縮後約660MB）×30世代は約20GBで、20GBディスクでは再発は時間の問題です。「30日分」が業務要件（どこまで遡って復元したいか）なら**ディスク増設**、要件でないなら**世代数の削減**。これは技術ではなく業務の判断なので、上長・関係者と「何日前まで戻れば良いか」を確認して決めます。

## 解説 4: 再発防止（改善版スクリプト）

```
#!/bin/bash
# backup-www.sh 改善版: 失敗が「見える」バックアップ
set -u

BACKUP_DIR=/var/backup
DATE=$(date +%Y%m%d)
ARCHIVE=${BACKUP_DIR}/www-${DATE}.tar.gz
LOG=/var/log/backup-www.log

log() { echo "[$(date '+%F %T')] $1" >> "${LOG}"; }

log "backup start"

# (1) 事前チェック: 使用率 80%以上なら開始前に異常終了
USAGE=$(df --output=pcent "${BACKUP_DIR}" | tail -1 | tr -dc '0-9')
if [ "${USAGE}" -ge 80 ]; then
    log "ERROR: ${BACKUP_DIR} usage ${USAGE}% (>=80%)"
    exit 1
fi

# (2) 世代整理を先に実行（空きを作ってから書く）
find "${BACKUP_DIR}" -name "www-*.tar.gz" -mtime +30 -delete

# (3) バックアップ本体: エラーは捨てずにログへ。失敗したら中途半端なファイルを残さない
if ! tar czf "${ARCHIVE}" -C / var/www >> "${LOG}" 2>&1; then
    log "ERROR: tar failed"
    rm -f "${ARCHIVE}"
    exit 1
fi

# (4) 展開検証: 「作れた」ではなく「戻せる」ことを確認
if ! tar tzf "${ARCHIVE}" > /dev/null 2>> "${LOG}"; then
    log "ERROR: verify failed"
    exit 1
fi

log "backup OK: $(du -h "${ARCHIVE}" | cut -f1)"
```

そして cron 側で、終了コードで通知を実装します。

```
MAILTO=admin@example.com
30 2 * * * /usr/local/bin/backup-www.sh || echo "backup-www.sh FAILED on $(hostname)" |
mail -s "[ERROR] backup failed" admin@example.com
```

改善点の対応表:

元の問題	改善
エラーが /dev/null に消える	すべてログファイルに記録 (2>&1 でログへ)
tar 失敗でも終了コード 0	if 文で判定し exit 1。壊れた中途半端なファイルは残さない
満杯になるまで気づけない	実行前に使用率 80%で早期警報 (満杯になる前に止めて知らせる)
「ファイルがある」=成功の思い込み	tar tzf の展開検証を毎回の処理に組み込む
削除が tar の後	世代整理を先に実行し、書き込み前に空きを確保
失敗しても通知が飛ばない	cron 側で終了コードを判定してメール通知

さらに一段上を目指すなら、月 1 回の「リストア訓練」(実際に 1 ファイル取り出してみる)を定例化すると、今回のような破損は最長でも 1 ヶ月で発見できます。バックアップの価値は「取れていること」ではなく「戻せること」で決まります。

## 解説 5: 現場の教訓

・「起動した」と「成功した」は別物。

cron のログはジョブの起動しか教えてくれません。成功の証拠は成果物の検証だけです。

・エラーを握り潰す 1 行は、将来の自分への時限爆弾。

2>/dev/null を書きたくなったら、「この出力が致命的エラーだった場合、自分はどこで気づけるか」を先に自問してください。

・バックアップは「戻せて」初めてバックアップ。

ファイルの存在・サイズ・日付は、いずれも成功の証拠になりません。

・提出いただいた解答の中から、良い着眼点を匿名でご紹介します。

(例:「最初に tar tzf を 7/6 のファイルにも実行して『どこまでが正常か』の境界を確定させた」という調査手順。境界の確定は切り分けの基本で、とても良い進め方だと思います。)

## この品質の課題が、毎月 1 本届きます

リナックスマスター実務ラボは、毎月 1 本の実務課題（障害調査・構築・自動化・セキュリティ点検）と 3 段階ヒント・模範解説、メール質問権（月 2 件）、課題アーカイブを提供する月額制の会員サービスです。会員の月次サイクルは課題#002 から始まります（本サンプル#001 と重複しません）。

会費: 月額 11,000 円（税込）／入会金なし・最低契約期間なし

入会のご案内: <https://lab.linuxmaster.jp/>

お問い合わせ: [pakira@linuxmaster.jp](mailto:pakira@linuxmaster.jp)（リナックスマスター.JP 宮崎）